Team Automata for Spatial Access Control

Maurice H. ter Beek[1]

Tuesday 18 September 2001
ECSCW 2001

⇒ joint work with Clarence A. (Skip) Ellis[2],
Jetty Kleijn[1], and Grzegorz Rozenberg[1,2]

[1]Leiden Institute of Advanced Computer Science, Univ. Leiden

[2]Dept. of Computer Science, Univ. of Colorado

Access Control

- information access control vital component of any (computer) system

- CSCW / groupware systems require:

  - more refined access rights
    e.g. right to scroll file synchronously edited by group in real time

  - more control of meta access rights
    e.g. grant/revoke a team member the right to grant/revoke file access to a non-team member (i.e. delegation)

Spatial Access Control

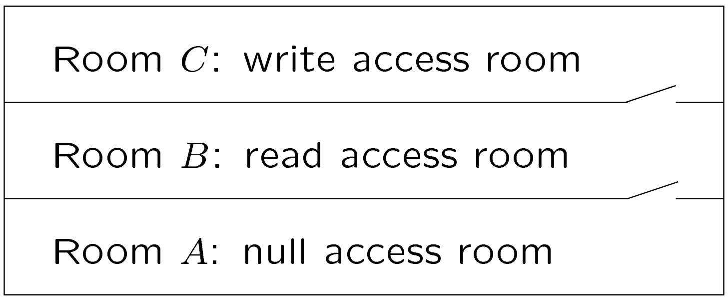
SPACE metaphor (Bullock et al., 1998)

$\Downarrow$

access control governed by spaces (rooms)
in which subjects and objects reside

and

the ability of a subject to traverse space
in order to get closer to an object



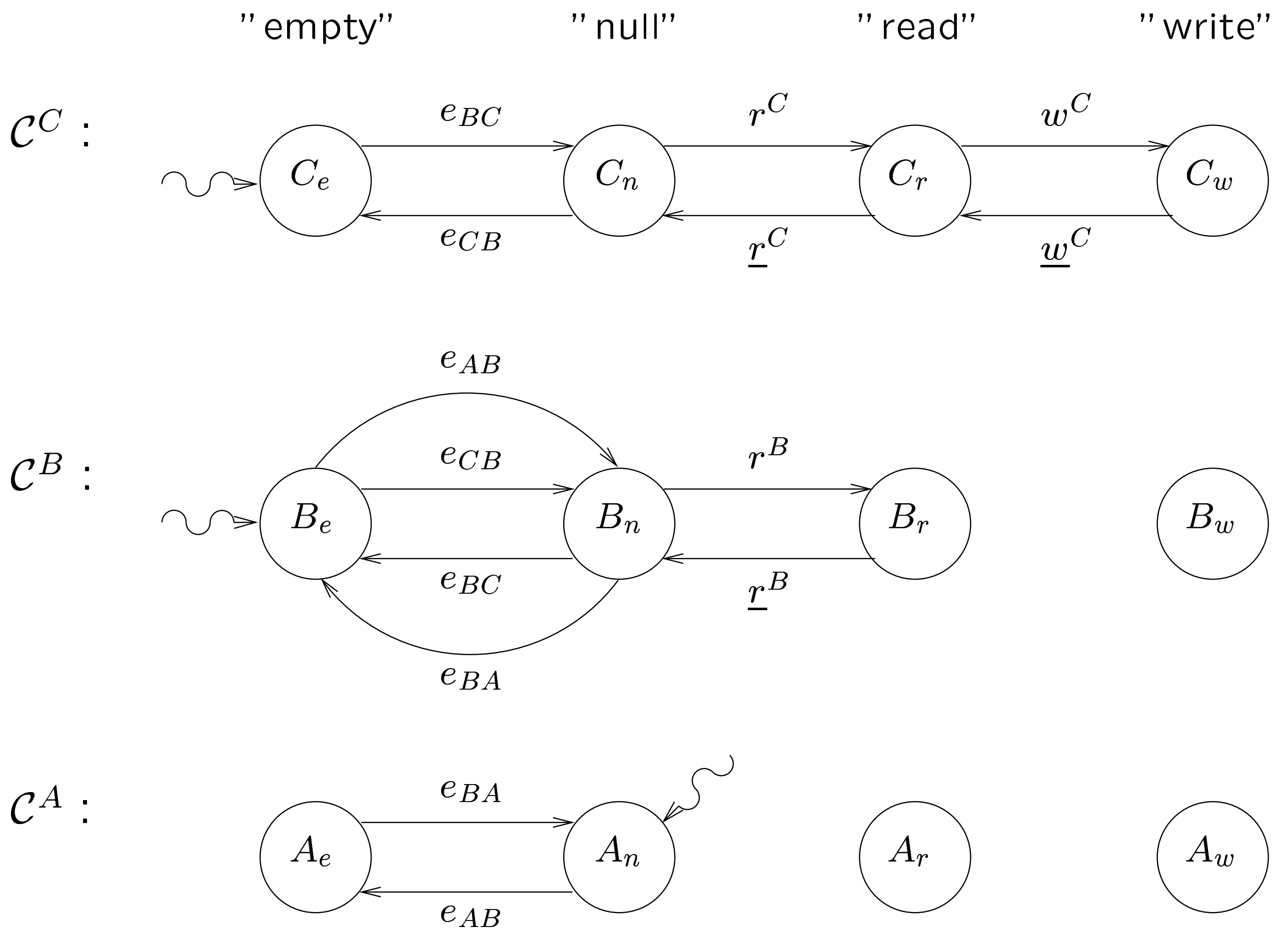
Team Automata

•

A formal framework for modelling
communication, coordination,
cooperation, and collaboration
among system components
(ter Beek, 2001)

- model logical architecture of a design
- abstract from concrete data and actions
- describe behaviour in terms of
  - state-action diagram (structure)
  - role of actions (input, output, internal)
  - shared actions (synchronizations)

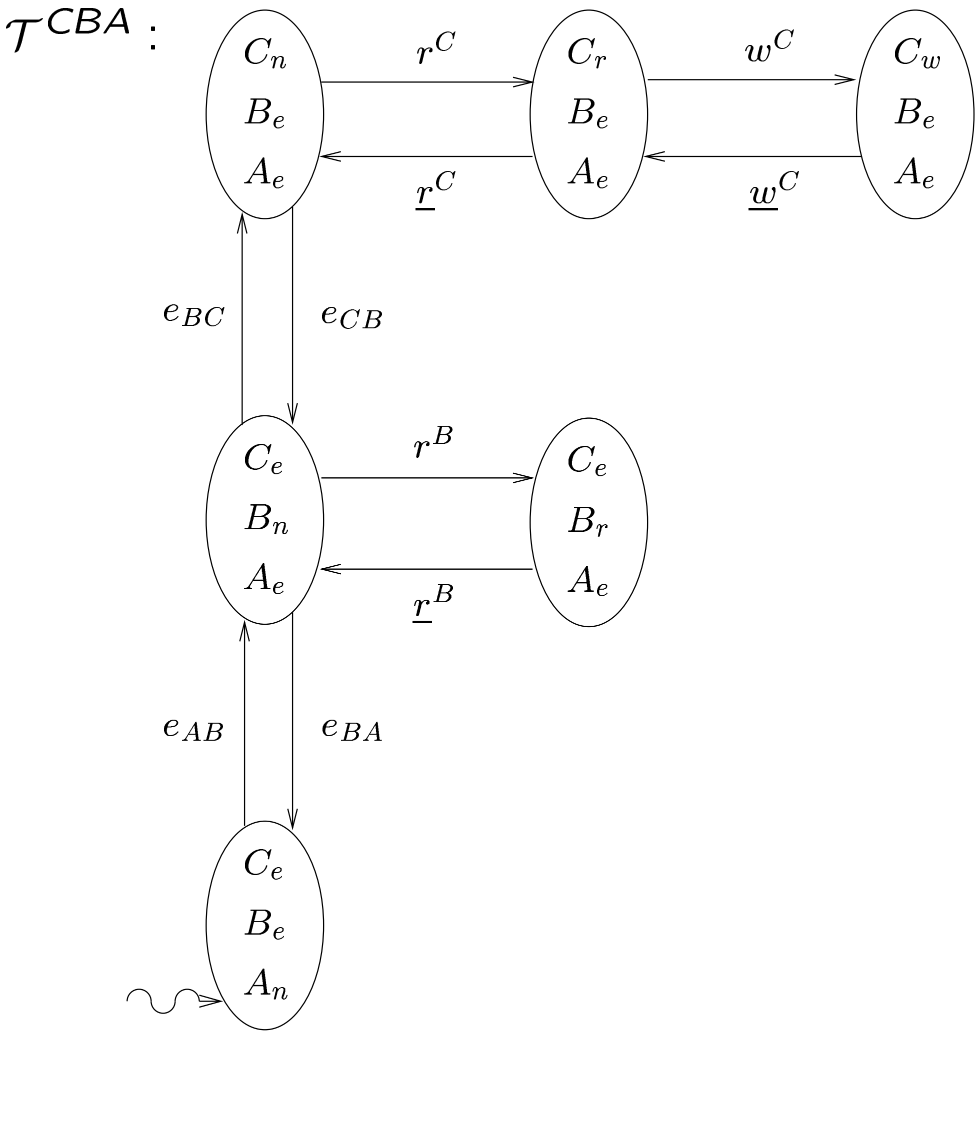
Component Automata $\mathcal{C}^C, \mathcal{C}^B, \mathcal{C}^A$: Rooms C, B and A



r^C, $\underline{r}^C$, w^C, $\underline{w}^C$, r^B, and $\underline{r}^B$ **internal** actions
e_{BC}, e_{CB}, e_{AB}, and e_{BA} **external** actions

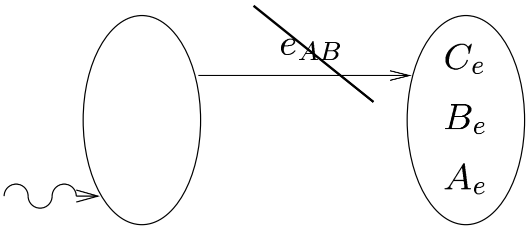
? how to form a team automaton over $\{\mathcal{C}^C, \mathcal{C}^B, \mathcal{C}^A\}$

Example Team Automaton $\mathcal{T}^{CBA}$ over $\{\mathcal{C}^C, \mathcal{C}^B, \mathcal{C}^A\}$



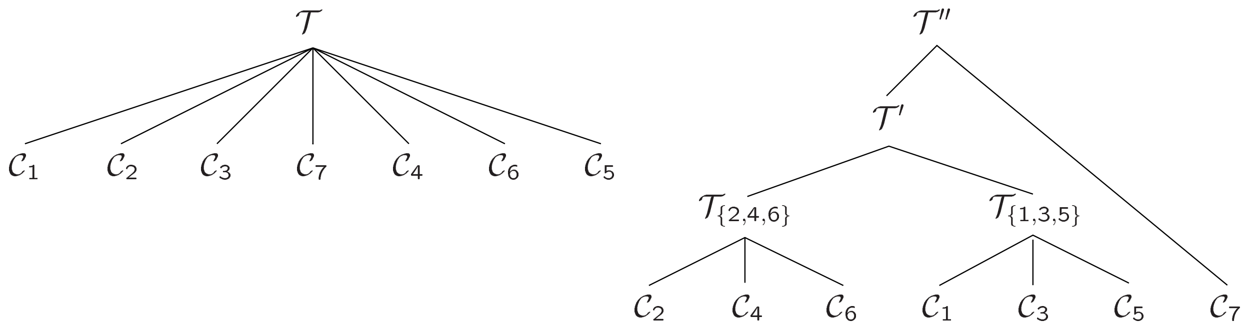
$\Rightarrow$ team automaton defined by <u>choice</u> of synchronizing transitions!

$\mathcal{T}^{CBA}$:



Teams over Teams

Note: every team automaton can be used
as a component



$\mathcal{T}$ team automaton over $\{\mathcal{C}_1, \ldots, \mathcal{C}_7\}$
$\mathcal{T}''$ *iterated* team automaton over $\{\mathcal{C}_1, \ldots, \mathcal{C}_7\}$

Fact: after *reordering* elements of $\mathcal{T}''$:
$\mathcal{T}''$ team automaton over $\{\mathcal{C}_1, \ldots, \mathcal{C}_7\}$

Meta Access Control

$\Rightarrow$ the rights to grant and revoke access can themselves be granted and revoked

e.g. read access rights during refereeing:

(m^3) authors of paper grant the right to grant read access to editors-in-chief

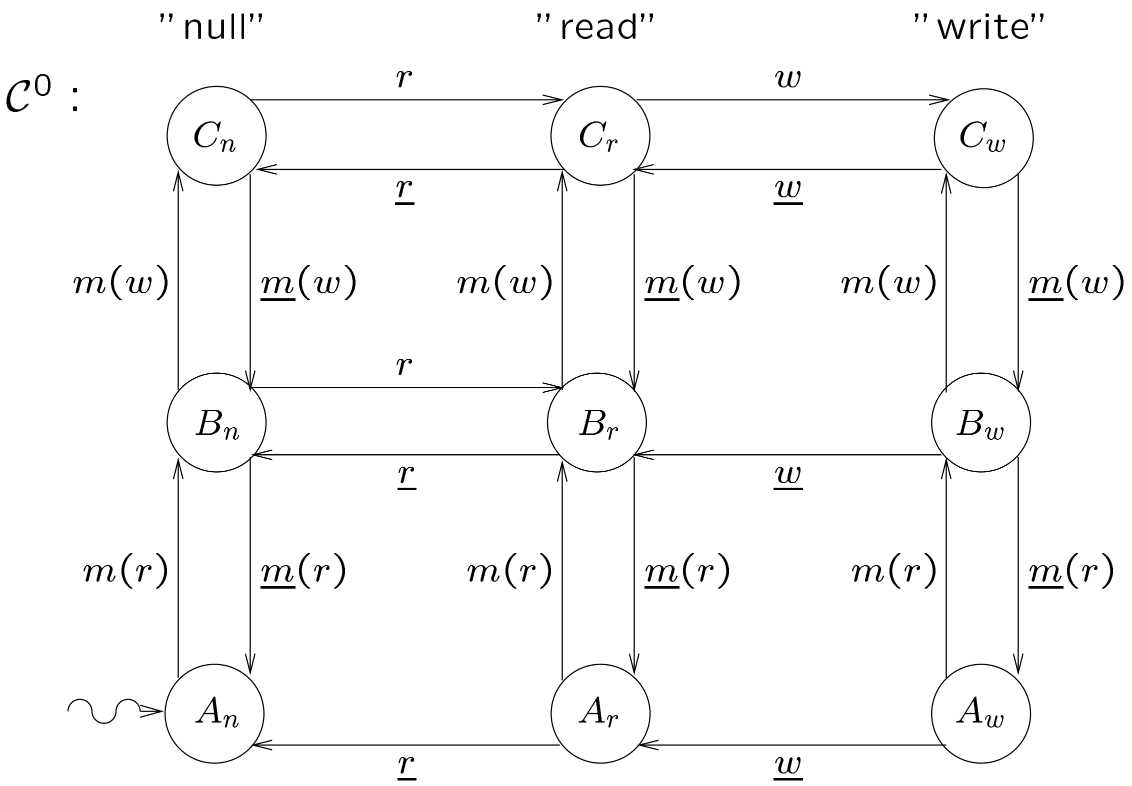
(m^2) editors-in-chief grant the right to grant read access to assistent editors

(m^1) assistent editors grant the right to read access to reviewers

(r) reviewers have read access

$$\underbrace{\text{grant right to}}_{m}\ \underbrace{\text{grant right to}}_{m}\ldots\underbrace{\text{read access}}_{r}$$

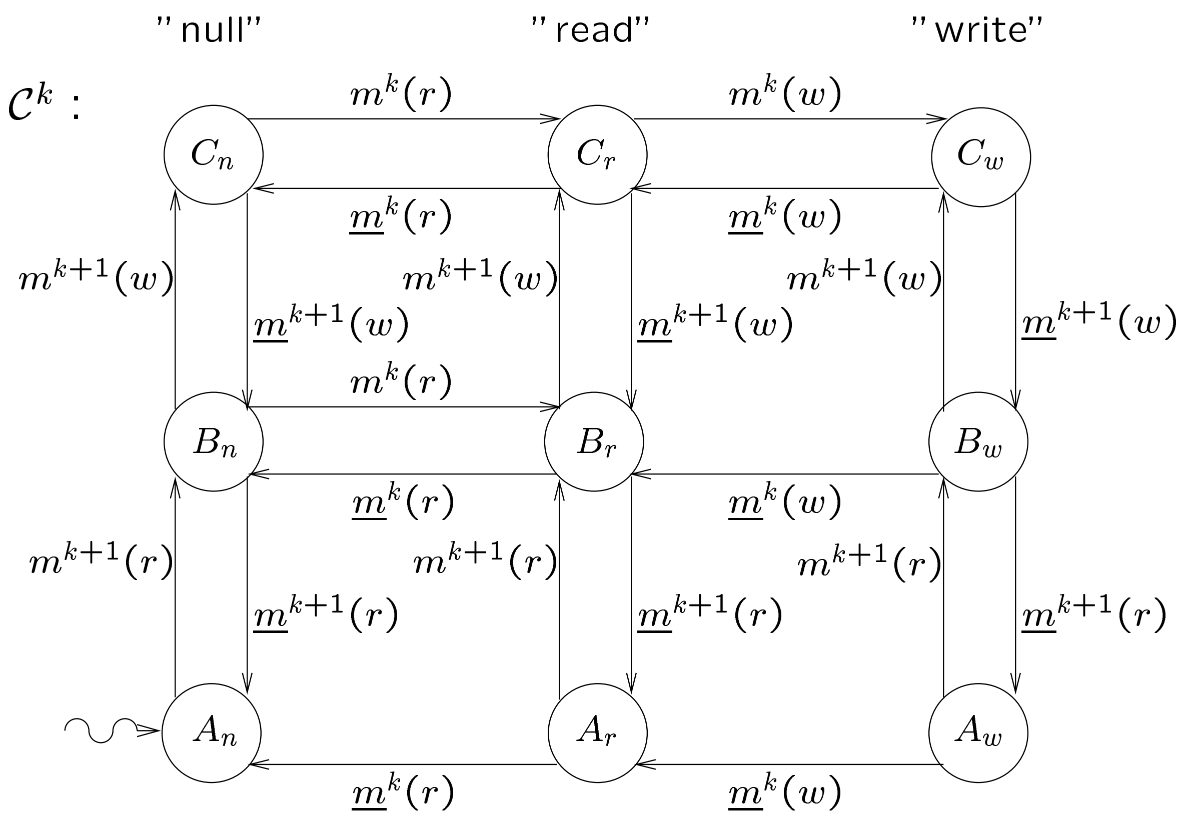
Component Automaton $\mathcal{C}^0$



$$\left.\begin{array}{lr} m^0(r) = r & \text{read access} \\ \underline{m}^0(r) = \underline{r} & \text{no read access} \\ m^0(w) = w & \text{write access} \\ \underline{m}^0(w) = \underline{w} & \text{no write access} \end{array}\right\} \begin{array}{l} \textbf{internal} \\ \text{actions} \end{array}$$

$$\underbrace{\begin{array}{lr} m^1(r) = m(r) & \text{grant right to read access} \\ \underline{m}^1(r) = \underline{m}(r) & \text{revoke right to read access} \\ m^1(w) = m(w) & \text{grant right to write access} \\ \underline{m}^1(w) = \underline{m}(w) & \text{revoke right to write access} \end{array}}_{\textbf{input} \text{ actions}}$$

Component Automaton $\mathcal{C}^k$: Meta Access at Layer k

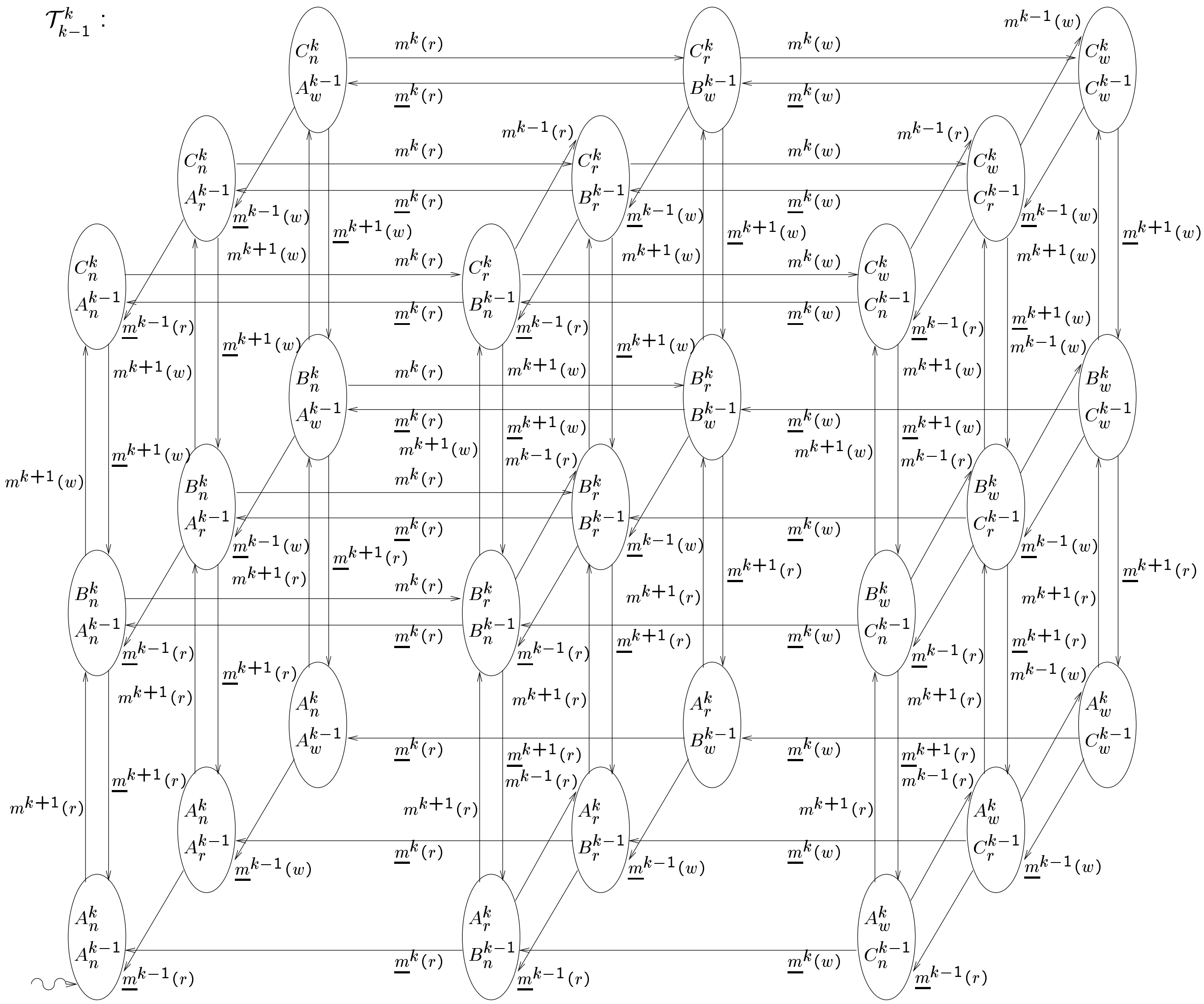


$m^k(r)$, $\underline{m}^k(r)$, $m^k(w)$, and $\underline{m}^k(w)$ **output** actions

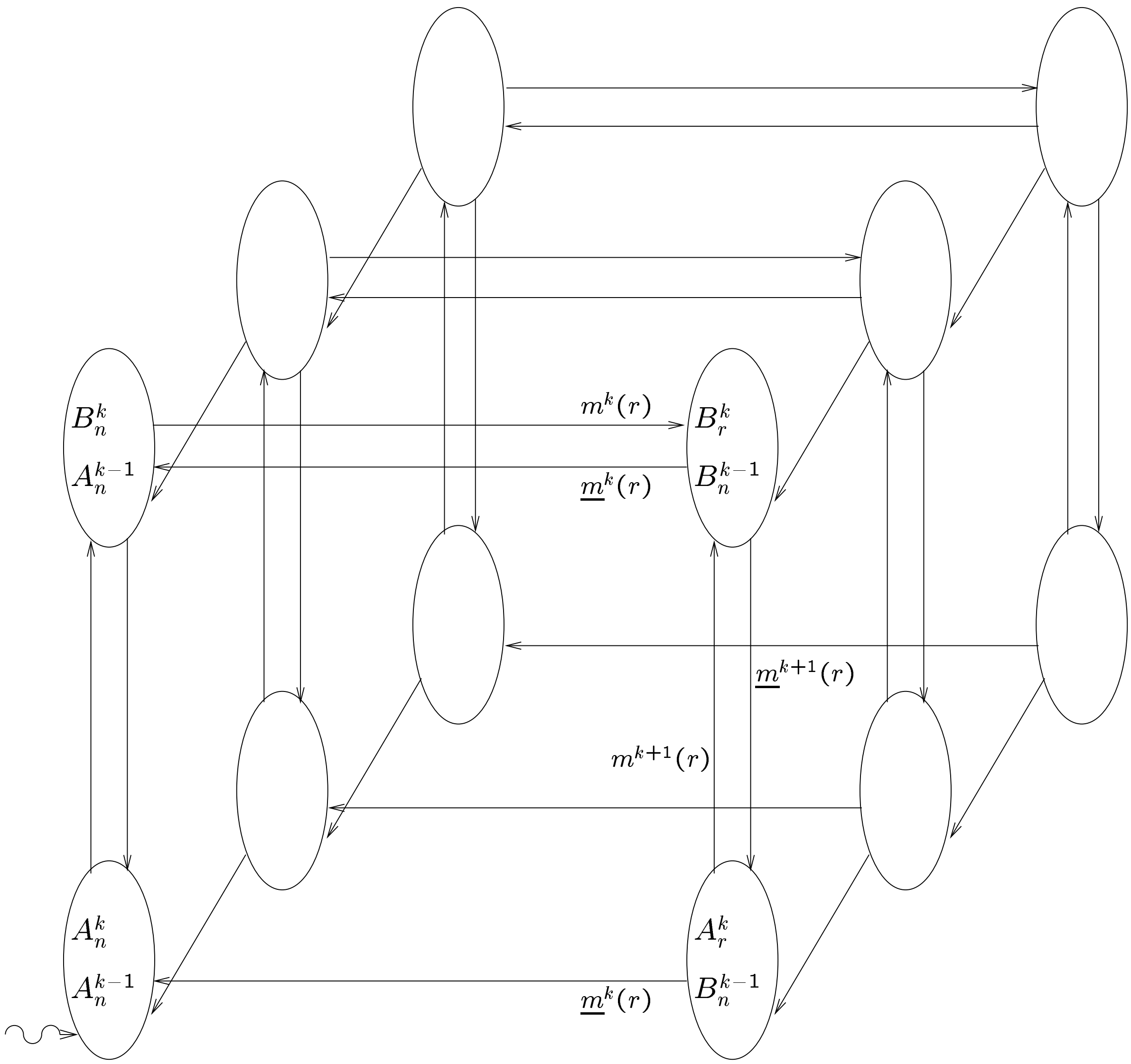
$m^{k+1}(r)$, $\underline{m}^{k+1}(r)$, $m^{k+1}(w)$, and $\underline{m}^{k+1}(w)$ **input**

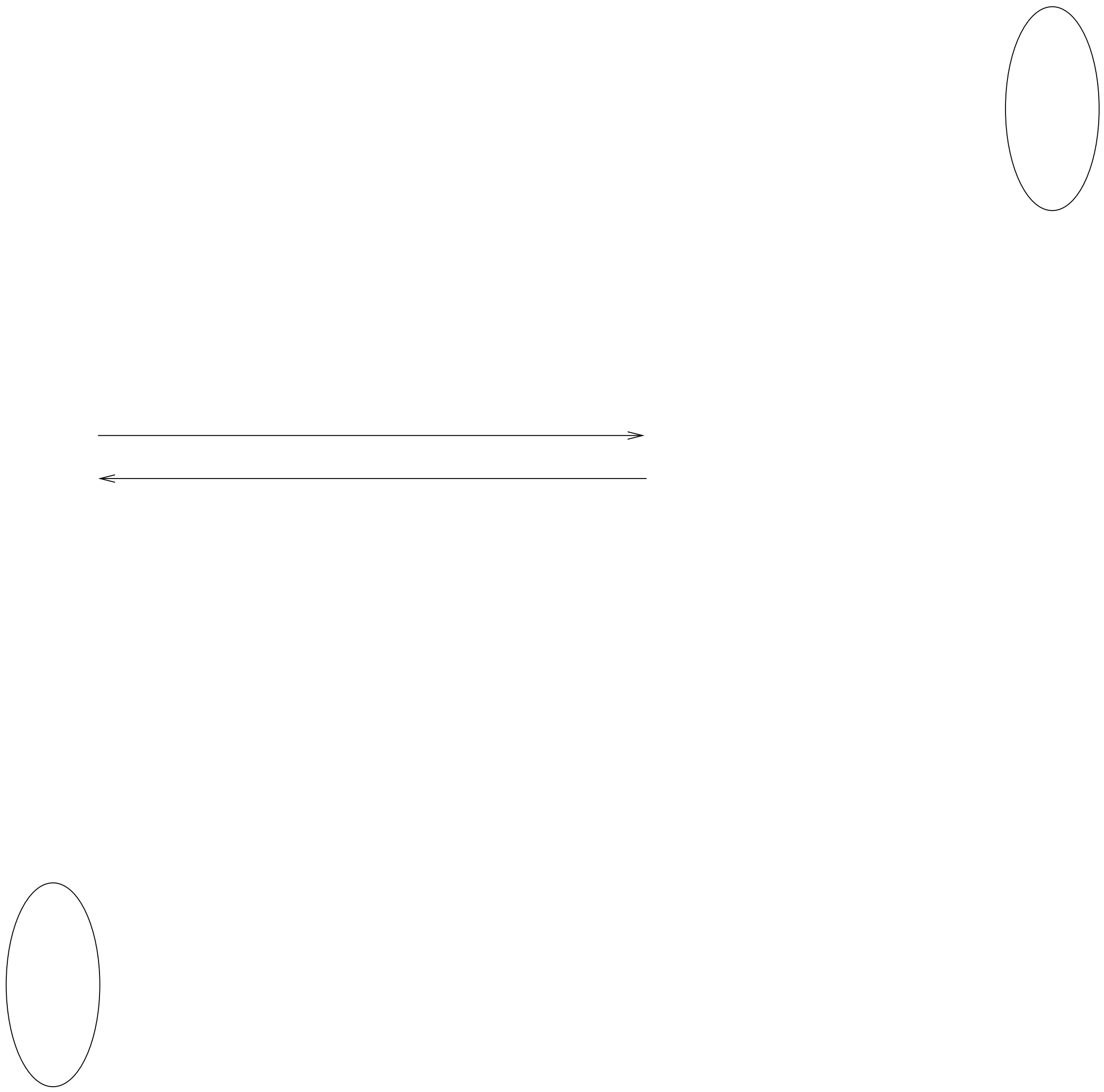
$\Rightarrow$ compose *iterated* team over $\{\mathcal{C}^0, \ldots, \mathcal{C}^n\}$ by choosing its synchronizing transitions!

Example Team Automaton $\mathcal{T}^k_{k-1}$ over $\{\mathcal{C}^{k-1}, \mathcal{C}^k\}$

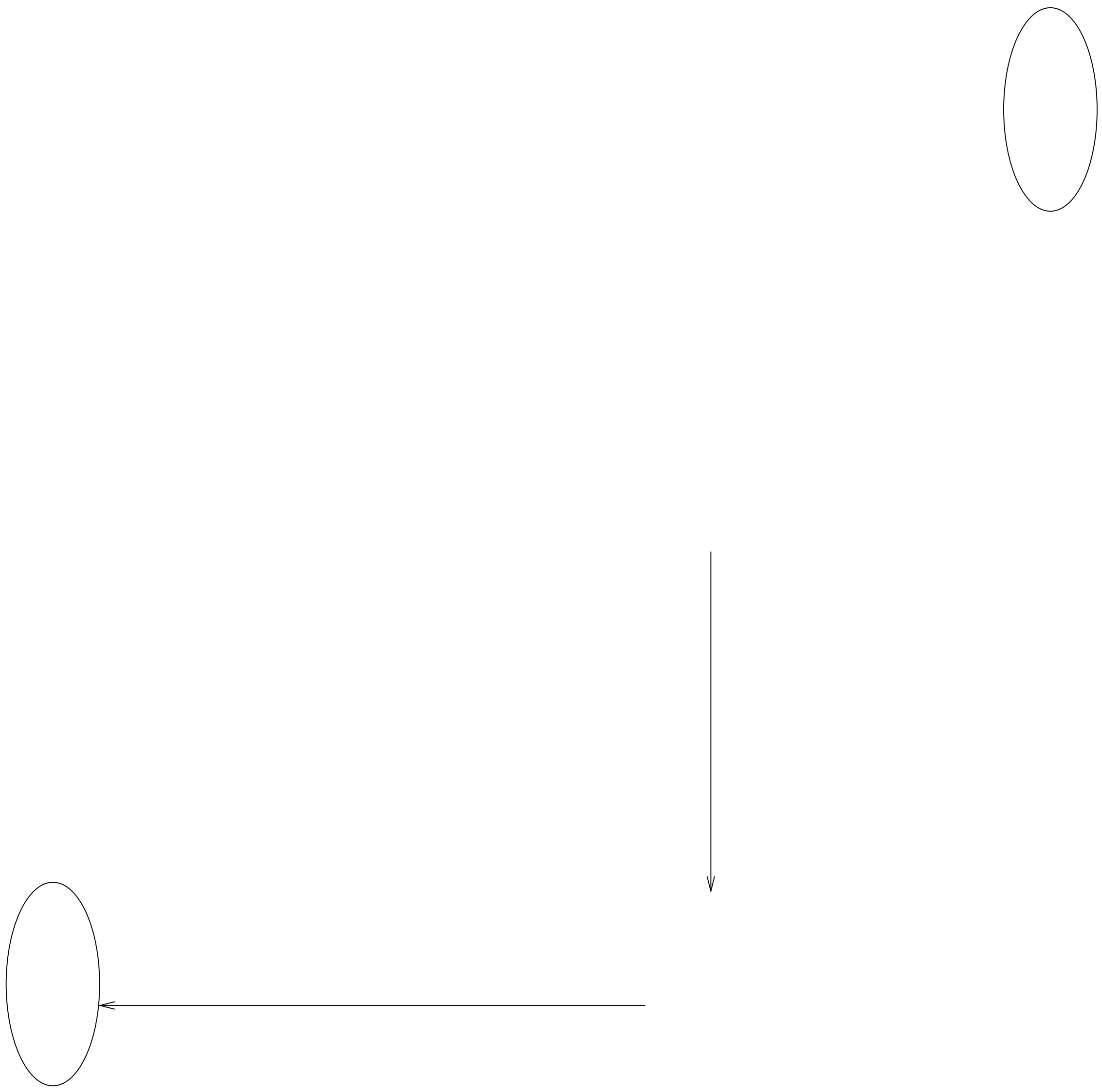


Detail from Team Automaton $\mathcal{T}_{k-1}^{k}$ over $\{\mathcal{C}^{k-1}, \mathcal{C}^{k}\}$





$m^k(r)$, $\underline{m}^k(r)$, $m^k(w)$, and $\underline{m}^k(w)$ **master-slave** synchronizations with master $\mathcal{C}^k$ and slave $\mathcal{C}^{k-1}$



shallow and *deep* revocation: here **deep**, which is known as hard to model (Dewan et al., 1998)

Conclusion

- spatial access control metaphor end user friendly:

  + demystifies concepts of access control
  + understandable by non-computer people

- team automata model precise and rigorous:

  + flexible (role of actions, choice of transitions)
  + scalable (modular construction, iterated teams)
  + extendible (time, priorities)
  + verifiable (automata theoretic tools)

  – no tool (yet)

$\Rightarrow$ metaphor and model together form
a powerful team!