The Programming Language AGENT0

Koen Hindriks
Utrecht University
koenh@cs.uu.nl
www.cs.uu.nl/people/koenh

AGENT0 agents

This presentation of AGENT0 is based on Shoham's paper of 1993:

Agent-oriented Programming,

in: Artificial Intelligence **60**.

Ingredients of AGENT0:

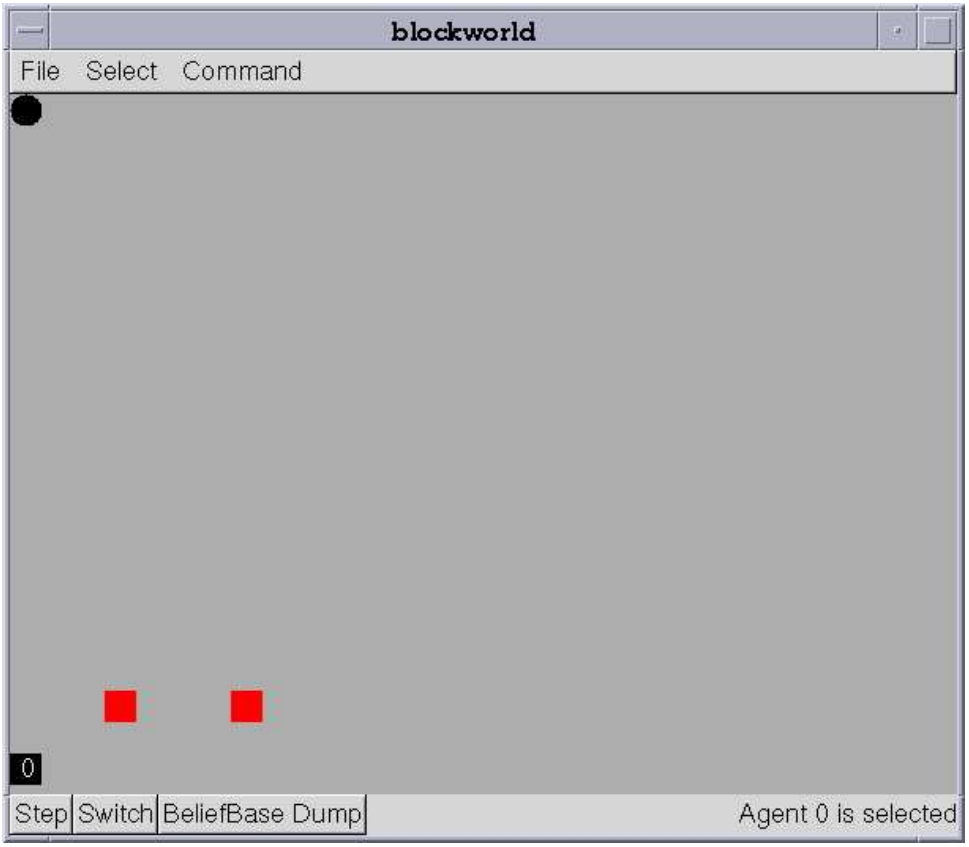
- beliefs,
- commitments (=actions),
- commitment rules
  (making decisions=committing to actions),
- communication,
- time.

First, we focus on writing programs for
- single agents
- without communication.

The Stack Block Example

The Block World

There is one robot wandering in a Block World.
In this Block World there are no obstacles.
There are two blocks in the Block World.
The Block World has no boundaries.



A Logical Language for Talking about the Block World

A Logical Language can be used to express facts about the Block World.

names for objects:
block a and block b.

predicates to express properties:
$robot(X, Y)$: the robot is at position X, Y
$nextto(X, Y)$: the robot is next to X, Y
$block(B, X, Y)$: block B is at X, Y
$clear(B)$: there is no block on top of B
$hold(B)$: the robot holds block B
$on(B1, B2)$: block $B1$ is on block $B2$

Facts are time stamped.

example of a fact: $(0\ (block(a, 4, 3)))$:
block a is at position $4, 3$ at Time 0.

Initial Beliefs

Initial beliefs consist of database of initial facts.

$$\begin{array}{l}\mathbf{BELIEFS} := \\ \quad (0\ block(a,4,3)), \\ \quad (0\ block(b,8,3)), \\ \quad (0\ clear(a)), \\ \quad (0\ clear(b)), \\ \quad (0\ robot(1,1))\end{array}$$

What does an agent believe?

BEL$(T\ fact)$:
The agent believes that $fact$ at Time T.
Example: **BEL**$(3\ (0\ block(a,4,3)))$.

$\neg$**BEL**$(T\ fact)$:
The agent does not believe $fact$ at Time T.

Specifying Agent Capabilities in AGENT0

A specification of an action A should specify
- when it is possible to execute the action, and
- what the effects of executing the action are.

Components of Action Specifications:

- name of primitive action,
- preconditions of action,
- postconditions of action*.

(*) Here we deviate from Shoham'93.

The $PickUp$ action

The $PickUp(B, X, Y)$ action picks up block B if
(i) the robot is next to position X, Y,
(ii) block B is clear, and
(iii) the robot does not hold any blocks.

$$
\begin{array}{l}
(\textbf{DO}\ T\ PickUp(B, X, Y)) \\
\quad \textbf{PRE}: \\
\qquad \textbf{BEL}(T\ nextto(X, Y)) \wedge \\
\qquad \textbf{BEL}(T\ block(B, X, Y)) \wedge \\
\qquad \textbf{BEL}(T\ clear(B)) \wedge \\
\qquad \neg\textbf{BEL}(T\ hold(a)) \wedge \\
\qquad \neg\textbf{BEL}(T\ hold(b)) \\
\quad \textbf{POST}: \\
\qquad \textbf{BEL}(T+1\ nextto(X, Y)) \wedge \\
\qquad \neg\textbf{BEL}(T+1\ block(B, X, Y)) \wedge \\
\qquad \textbf{BEL}(T+1\ clear(B)) \wedge \\
\qquad \textbf{BEL}(T+1\ hold(B)) \wedge \\
\qquad B \neq a \rightarrow \neg\textbf{BEL}(T+1\ hold(a)) \wedge \\
\qquad B \neq b \rightarrow \neg\textbf{BEL}(T+1\ hold(b))
\end{array}
$$

Belief Persistence in AGENT0

What does an agent belief at time $T + 1$?

Belief Persistence Assumption:
Beliefs persist over time.
If a belief $\mathbf{BEL}(T \ \ fact)$ is not changed by performing some action at time T, then the agent $\mathbf{BEL}(T + 1 \ fact)$.

Note:
The issue of belief persistence arises because beliefs are time stamped.

Remark:
The issue is a special case of the frame problem.

The $StackOn$ action

The $StackOn(B1, B2, X, Y)$ action stacks $B1$ on $B2$ if
(i) the robot is next to X, Y
(ii) holds block $B1$
(iii) block $B2$ is at X, Y
(iv) block $B2$ is clear.

$$
\begin{array}{l}
(\mathbf{DO}\ T\ StackOn(B1, B2, X, Y)) \\
\quad \mathbf{PRE}: \\
\quad\quad \mathbf{BEL}(T\ nextto(X, Y)) \wedge \\
\quad\quad \mathbf{BEL}(T\ hold(B1)) \wedge \\
\quad\quad \mathbf{BEL}(T\ block(B2, X, Y)) \wedge \\
\quad\quad \mathbf{BEL}(T\ clear(B2)) \\
\quad \mathbf{POST}: \\
\quad\quad \neg\mathbf{BEL}(T+1\ hold(B1)) \\
\quad\quad \mathbf{BEL}(T+1\ block(B1, X, Y)) \wedge \\
\quad\quad \mathbf{BEL}(T+1\ on(B1, B2)) \wedge \\
\quad\quad \neg\mathbf{BEL}(T+1\ clear(B2))
\end{array}
$$

Logical Relations between Predicates

If block $B1$ is on top of block $B2$, then $B2$ is not clear:

$$on(B1, B2) \rightarrow \neg clear(B2)$$

Robot is at X, Y iff the robot is next to $X-1, Y$, etc.:

$$[nextto(X-1, Y) \wedge nextto(X+1, Y) \wedge nextto(X, Y-1) \wedge nextto(X, Y+1)] \leftrightarrow robot(X, Y)$$

Remark:
Unfortunately AGENT0 does not have facilities to adequately deal with such logical relations:

- no facilities for reasoning with beliefs,
- updating problem.

Compare the programming languages ConGolog and 3APL.

Robot Movements in the Block World

$$
\begin{array}{l}
(\mathbf{DO}\ T\ North) \\
\quad \mathbf{PRE}: \\
\qquad \mathbf{BEL}(T\ robot(X, Y)) \wedge \\
\qquad \neg\mathbf{BEL}(T\ block(a, X, Y+1)) \wedge \\
\qquad \neg\mathbf{BEL}(T\ block(b, X, Y+1)) \wedge \\
\quad \mathbf{POST}: \\
\qquad \mathbf{BEL}(T+1\ robot(X, Y+1))
\end{array}
$$

Remark:
We assume a robot can be at most at one position at the same time.

$$
\begin{array}{l}
[robot(X, Y) \wedge (X \neq X' \vee Y \neq Y')] \\
\qquad \rightarrow \neg robot(X', Y')
\end{array}
$$

We can give similar specifications for $East$, $South$, and $West$.

The Frame Problem

Facts an action does change and does not change.

Frame Problem:

- An Action Specification states what effects that action has.
- It does not state what the action does not change.

General Rule for Action Specifications:

If an action specification does not say anything about a predicate, the predicate is not changed.

Example:
The $clear$ predicate and the $North$ action.

Capability Database

A capability database is a list of primitive actions + their action specifications.

$$
\begin{array}{l}
\textbf{CAPABILITIES} := \\
\quad (\textbf{DO}\ T\ PickUp(B, X, Y)) \\
\quad (\textbf{DO}\ T\ StackOn(B1, B2, X, Y)) \\
\quad (\textbf{DO}\ T\ North) \\
\quad (\textbf{DO}\ T\ East) \\
\quad (\textbf{DO}\ T\ South) \\
\quad (\textbf{DO}\ T\ West)
\end{array}
$$

Writing an AGENT0 Program

An AGENT0 program consists of:

- a database of initial beliefs, ✓
- a database of capabilities, ✓
- a database of commitment rules.

Commitment Rules

A commitment rule selects the actions an agent should perform at each time point.

A commitment rule is used by the agent to make decisions as to which action to perform.

Components of a Commitment Rule:

- condition on the mental state,
- action to perform.

What do we want the Robot to do?

Specification of the Task of the Agent (Robot):

> Build a stack of all the blocks (in our example two) lying around in the Block World.

Remark:

- we don't care where the stack is build,
- we don't care how the stack is build.

Commitment Rules for Moving

IF:

- block B is clear,
- block B is not on top of another block, and
- it is not nextto block B,

THEN it should move towards block B.

Commitment Rules for Moving (cont'ed)

$$
\begin{array}{l}
(\mathbf{COMMIT} \\
\quad \mathbf{BEL}(now\ block(B, X, Y)) \wedge \\
\quad \mathbf{BEL}(now\ clear(B)) \wedge \\
\quad \neg\mathbf{BEL}(now\ on(B, a)) \wedge \\
\quad \neg\mathbf{BEL}(now\ on(B, b)) \wedge \\
\quad \mathbf{BEL}(now\ robot(X', Y')) \wedge \\
\quad \mathbf{BEL}(now\ Y' + 1 < Y) \wedge \\
\qquad (DO\ now\ North) \\
)
\end{array}
$$

Similar rules for $East$, $South$, and $West$.

Note special constant now.

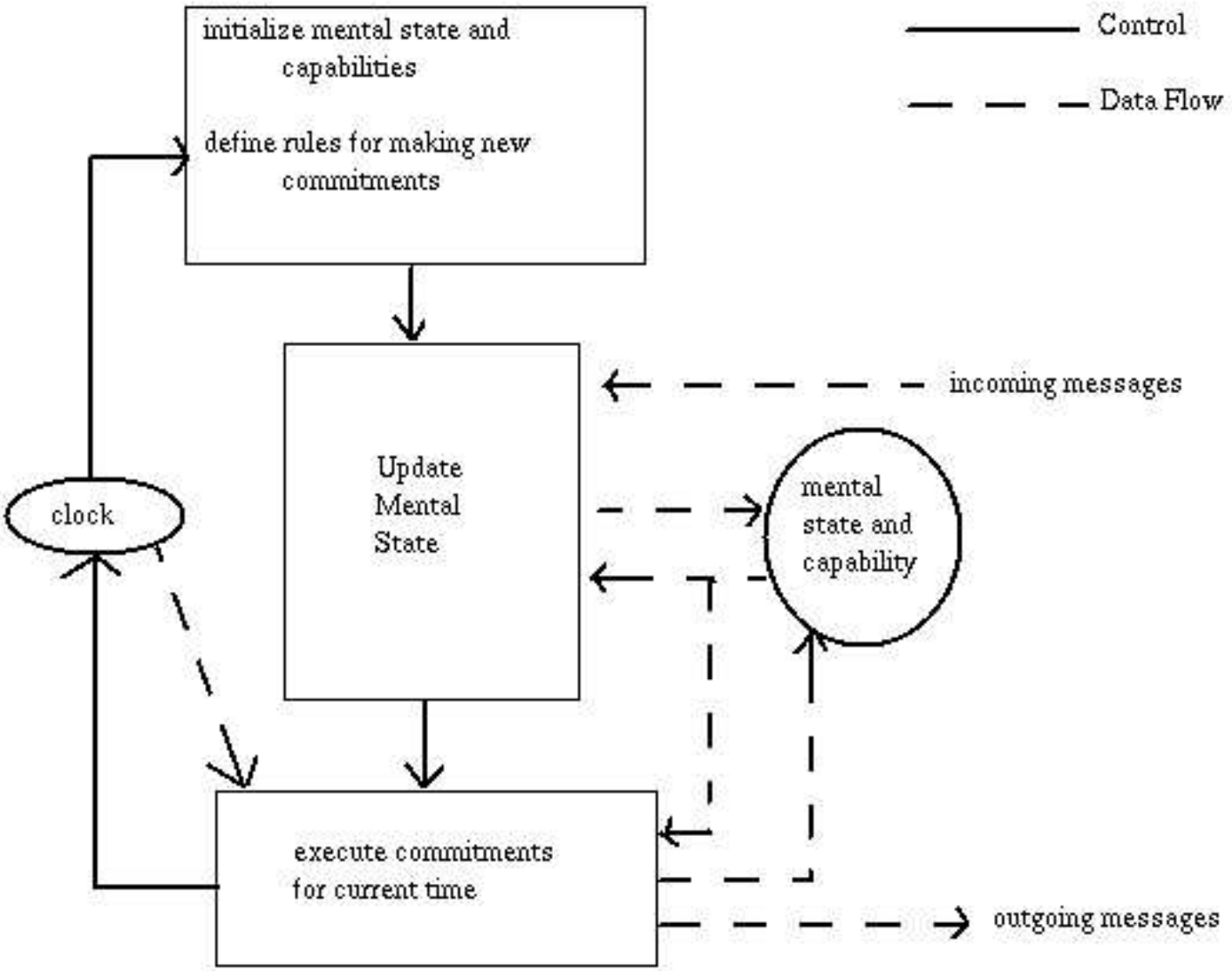
How is the Robot going to Move?

How is the program interpreted?

The AGENT0 interpreter cycle consists of 3 Steps:

1. Fire all applicable commitment rules,
2. Remove any infeasible actions,
3. Execute all actions scheduled for the current time (now).

The AGENT0 Interpreter



re 1 : Control/Data Flow Of The Agent–0 Language Interpreter (Shoham, 93)

The Role of the Clock

Time is assumed to be discrete and has a beginning: Time 0, Time 1, etc.

In contrast with Shoham'93, we do not assume a synchronised clock between agents. The reason is that this leads to problems with updating the beliefs of an agent.

Instead, we assume that with the execution of an action and only with the execution of an action the current time is increased with one time unit.

Firing a Commitment Rule

- Firing a Commitment Rule results in commitments.
- A commitment is an action.

Fire a Rule:

1. match the mental state condition with the mental state,

2. add instantiated actions to current commitments.

Firing a Commitment Rule (Example)

Current belief base:

$$\mathbf{BELIEFS} :=$$
$$(0\ block(a, 4, 3)),$$
$$(0\ block(b, 8, 3)),$$
$$(0\ clear(a)),$$
$$(0\ clear(b)),$$
$$(0\ robot(1, 1))$$

Current Time: 0.

Commitment Rule:

$$(\mathbf{COMMIT}$$
$$\mathbf{BEL}(now\ block(B/a, X/5, Y/4)) \wedge$$
$$\mathbf{BEL}(now\ clear(B/a)) \wedge$$
$$\neg\mathbf{BEL}(now\ on(B/a, a)) \wedge$$
$$\neg\mathbf{BEL}(now\ on(B/a, b)) \wedge$$
$$\mathbf{BEL}(now\ robot(X'/1, Y'/1)) \wedge$$
$$\mathbf{BEL}(now\ Y'/1 + 1 < Y/4 \wedge$$
$$(DO\ now\ North))$$

The Commitment Rule matches with substitutions as indicated in blue, and will result in adding action $(DO\ now\ North)$ to the commitments.

Firing All Commitment Rules at Time 0

Firing all Commitment Rules at Time 0 yields:

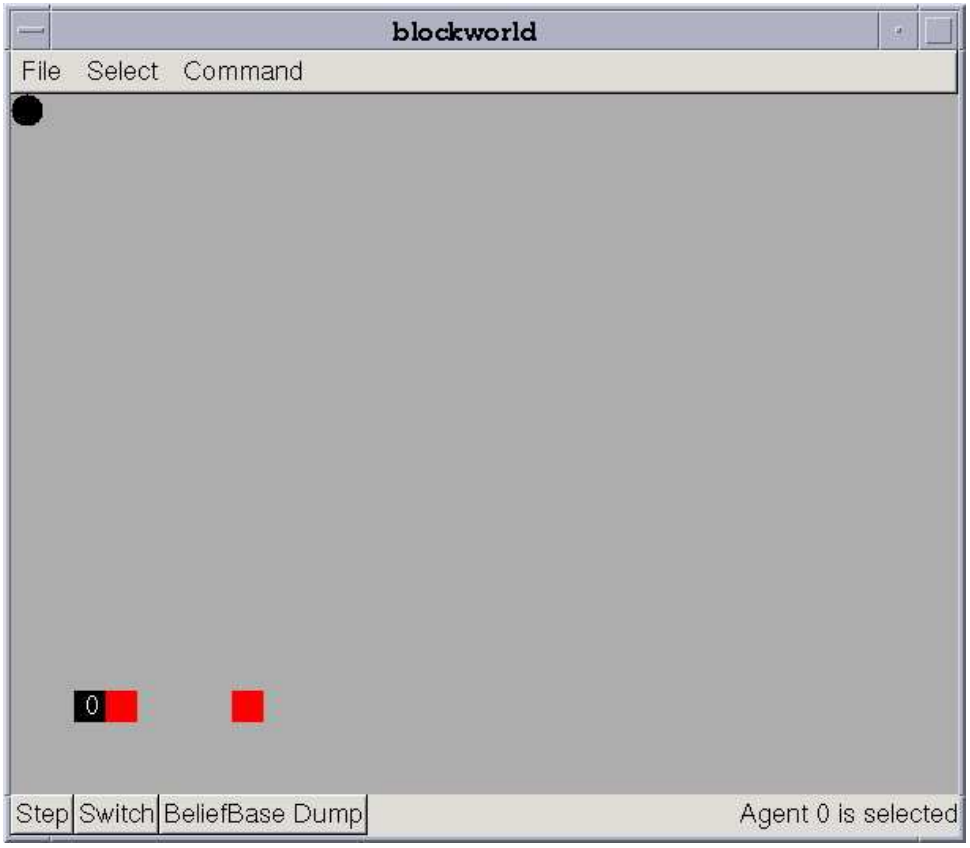
$(DO\ now\ North)$,
$(DO\ now\ North)$,
$(DO\ now\ East)$,
$(DO\ now\ East)$

All of these actions are feasible at Time 0 $\Rightarrow$
Step 2 does not remove any commitments.

After executing all of these actions:

- The robot is at position $3, 3$,
- The current time is Time 4, since 4 actions have been executed,
- Only the position of the robot has changed.

Block World At Time 4



Commitment Rule for Picking up a block

IF:

- block B is clear,
- block B is not on top of another block,
- the robot does not hold a block, and
- the robot is next to block B,

THEN the robot should pick up block B.

Commitment Rule for Picking up Block (cont'ed)

$$
\begin{array}{l}
(\mathbf{COMMIT} \\
\quad \mathbf{BEL}(now\ block(B, X, Y)) \wedge \\
\quad \mathbf{BEL}(now\ clear(B)) \wedge \\
\quad \neg\mathbf{BEL}(now\ on(B, a)) \wedge \\
\quad \neg\mathbf{BEL}(now\ on(B, b)) \wedge \\
\quad \neg\mathbf{BEL}(now\ holds(a)) \wedge \\
\quad \neg\mathbf{BEL}(now\ holds(b)) \wedge \\
\quad \mathbf{BEL}(now\ nextto(X, Y)) \\
\qquad (DO\ now\ PickUp(B, X, Y)) \\
)
\end{array}
$$

Remark:
At Time 0 this rule could not be fired.

Firing Commitment Rules at Time 4

Firing all Commitment Rules at Time 4 yields:

Current Commitments:

$(DO\ now\ East)$,
$(DO\ now\ PickUp(a, 4, 3))$

Removing Infeasible Actions At Time 4

At Time 4:
$\Rightarrow$ Step 2 in the AGENT0 Interpreter comes into play, because action $East$ is not feasible:
block a is in the way.

Remark:
Because of Step 2 in the AGENT0 Interpreter, we do not have to check if a movement action is feasible in a commitment rule!

After Step 2 at Time 4, which removes action $East$, the commitment database looks like:

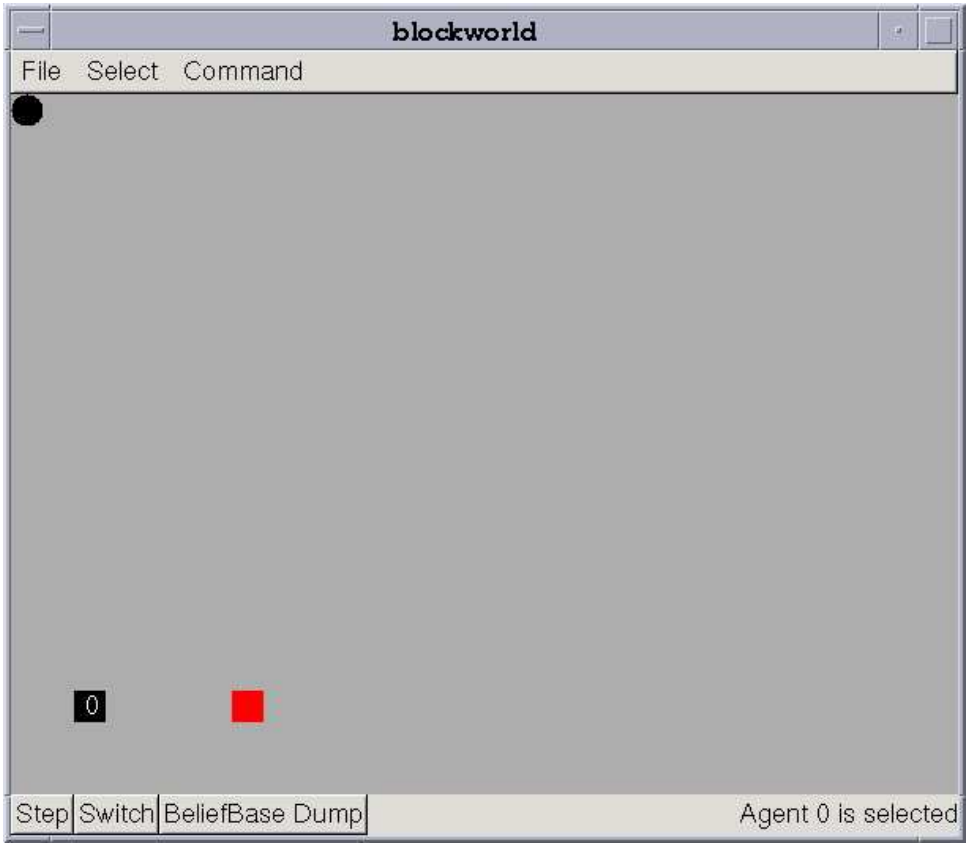
Current Commitments:

$(DO\ now\ PickUp(a, 4, 3))$

Belief Base at Time 5

After executing the action at Time 4, the facts time stamped with Time 5 in the belief base are:

$(5\ (hold(a))),$
$(5\ (block(b, 8, 3)),$
$(5\ (clear(a)),$
$(5\ (clear(b)),$
$(5\ (robot(3, 3)))$



Firing Commitment Rules at Time 5

Step 1 at Time 5:
Computing the new commitments for Time 5 by firing commitment rules yields:

Current Commitments:

$$(DO\ now\ East)$$

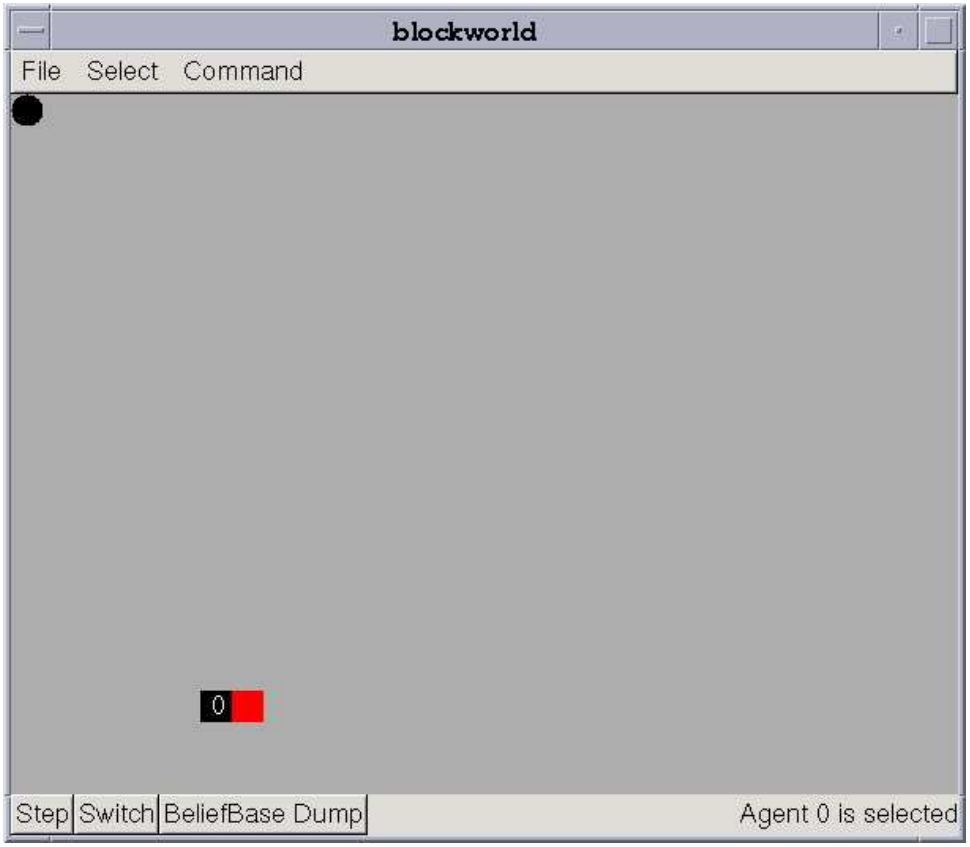
The new position of the robot at Time 6 is: $4, 3$.

This is repeated 3 more times;
therefore, at Time 9 the position of the robot is: $7, 3$.

Belief Base at Time 9

The facts time stamped with Time 9 in the belief base are:

$(9\ (hold(a))),$
$(9\ (block(b, 8, 3)),$
$(9\ (clear(a)),$
$(9\ (clear(b)),$
$(9\ (robot(7, 3)))$



Commitment Rule for Building Stack

IF:

- block B is clear,
- the robot holds block B', and
- the robot is nextto block B,

THEN the robot should stack B' on B.

Commitment Rule for Building Stack (cont'ed)

$$
\begin{array}{l}
(\textbf{COMMIT} \\
\quad \textbf{BEL}(now\ block(B, X, Y)) \wedge \\
\quad \textbf{BEL}(now\ clear(B)) \wedge \\
\quad \textbf{BEL}(now\ holds(B')) \wedge \\
\quad \textbf{BEL}(now\ nextto(X, Y)) \\
\qquad (DO\ now\ StackOn(B', B, X, Y)) \\
)
\end{array}
$$

Problem with Stack Program

- Would the robot build a stack if the initial position is $6, 3$? Why not?
- How would you fix this?