Bisbot
Recipes – Behavior Graphs

Recipes are hierarchical data-structures that group and sequence actions and responses that are expected of the simulated entity. QuadRuple = (B,S,V,b₀) :

- B set of task achieving behaviors
- S, V sets of directed edges between behaviors
- S sequential edges - specify temporal order of execution behaviors
- V set of vertical decomposition edges - allow a single higher level behavior to be broken down into execution chains containing multiple lower level behaviors.
- b₀ in B – Behavior in which execution begins
Recipes collect together Soar operator templates.

Each operator template has preconditions that can be tested to see if the operator is selectable.

It also has termination conditions that can be tested to see if the operator should terminate execution (if it is currently selected).

The termination conditions have indication of results: successful termination, failed termination, preemptive termination.
Adding Operators

`programmer->ops`

Add a new behavior. Each behavior has preconditions, endconditions and application code.

Be sure to condition your preconditions and endconditions only by the use of events.

DO NOT use O Support.
Former Move Operator

Move forward if not blocked

## Move Operator Proposal

# If the task is tanksoar and the tank is not blocked in the forward direction, propose the move operator.
Former Move Operator - Propose

sp {tanksoar*propose*move

(state <s> ^name tanksoar
^io.input-link.blocked.<dir> no)

-->

(<s> ^operator <o> +)

(<o> ^name move
^actions.move.direction <dir>}}
Former Move Operator - Apply

Same as eaters example

# Apply*move

# If the move operator for a direction is selected,

# generate an output command to move in that direction.
Former Move Operator - Apply

sp {apply*move

  (state <s> ^io.output-link <ol>

       ^operator <o>)

  (<o> ^name move

       ^direction <dir>)}

-->

  (<ol> ^move.direction <dir>>)
Former Move Operator - Termination

# Apply*move*remove-move:

# If the move operator is selected,

# and there is a completed move command on the output link,

# then remove that command.
Former Move Operator - Termination

sp {apply*move*remove-move

  (state <s> ^io.output-link <ol>
     ^operator.name move)

  (<ol> ^move <direction>)

  (<direction> ^status complete)

  -->

  (<ol> ^move <direction> -))

Receipes

rsc - receipe shortcuts, constructed automatically

sc - mulitple allowed, pointers into receipe under “root”

root - points at first, top node

child - multiplie allowed - has next (mult. allowed), prev (mult allowed) parent.
Recipes

recipe

name (string)

rsc (recipe short cuts constructed automatically)

sc - multiply allowed, pointers into recipe

root - points at first, top node

child - multiply allowed

next (mult), prev (mult), parent, active (if currently in control), terminated <yes> in case of endcondition, first (true, false - is it a first child)
I found the explanation in : SingleEntity24-01-06.doc (in the downloads section in github ).
New Move Operator

in pogrammer->ops->move.soar  (very-simple-wanderer->ops)

sp {ops*move*param
   ^params <p>}
   (<o>
   (state <ss> ^state <s>)
   (<p> ^timeout <t>)
   (<s> ^recipe <r>)
   (<t> ^time 10000
     (<r> ^rsc <rsc>)
     ^type failure)}
   (<rsc> ^sc <o>)
New Move Operator

sp \{ops*move*precondition*not-blocked\}

\[
\begin{align*}
\text{(state } \langle s_s \rangle & \wedge \text{state } \langle s \rangle) & \rightarrow & \text{(o} \wedge \text{precondition} \\
\text{(<s> } \wedge \text{recipe } \langle r \rangle \wedge \text{inputs } \langle i \rangle) & & & \text{(con} \wedge \text{name move} \\
\text{(<r> } \wedge \text{rsc } \langle rsc \rangle) & & & \text{^value true} \\
\text{(<rsc> } \wedge \text{sc } \langle o \rangle) \wedge \text{dparams } \langle dp \rangle & & & \text{\wedge type} \\
\text{(o} \wedge \text{name move)} & & & \text{nil} \\
\text{(<i> } \wedge \text{blocked } \langle b \rangle) & & & \text{(dp} \wedge \text{direction}}
\end{align*}
\]
New Move Operator

```
sp {ops*move*precondition*blocked

  (state <ss> ^state <s>) →
  (<o> ^precondition
  (<con> ^name move

  (<s> ^recipe <r> ^inputs <i>)

  (<r> ^rsc <rsc>)

  (<rsc> ^sc <o>)

  ^dparams <dp>

  (<o> ^name move)

  nil)

  (<i> ^blocked <b>)

  (<dp> ^direction

  none))
```
New Move Operator

sp {ops=move*commands=apply-and-terminate:o-support

(state <ss> ^state <s>) →

(<s> ^outputs <out> ^recipe <r>)

(<r> ^rsc <rsc>
^direction <dir> )

(<rsc> ^sc <o>)
#terminate

(<o> ^name move ^dparams <dp> ^active yes)  (<o> ^endcondition <end>)

(<dp> ^direction <dir>
^value true})

}
Preferences

In programmer->preferences->simple-defaults.soar

sp {preferences*motion*move-better-than-rotate

  (state <s> ^state <state>) → (<state> ^operator <o1> > <o2>) }

  (<state> ^operator <o1> + ^operator <o2> +)

  (<o1> ^name move)

  (<o2> ^name rotate)