

GOAL Programming Tips

Debug Rules

1. If it runs slow logic is not sound
2. If agent does not log anymore (and times out from then on with server) too many calculations/logic **very** wrong/empty updates
- 3.

Prolog arithmetic symbols

Good:	Bad:
=< kleinergleich	<=
\= ungleich, besser ist \== (2diff funktioniert auch bei uninitialisierten Variablen)	/=
>= größergleich	
@< (kann auch unmathematische Reihenfolgen vergleichen, @ generalisiert)	
Vergleich ==	Nicht mit Zuweisung = verwechseln

Implizite Zuweisung, Namen egal zwischen Dateien, Namen innerhalb Datei logisch aber beliebig

```
% Managing server successful action messages
if percept(lastActionResult(success)) then {
    % update XY position of agent
    if percept(lastAction(move), lastActionParameters([D])) then {
        % update agent XY position with confirmed move nswe
        Event Datei: if bel(agentAt(X, Y), translate(D, X, Y, X2, Y2)) then delete(agentAt(X, Y)) + insert(agentAt(X2, Y2)).
```

Variablenamen D, X, Y müssen auftauchen bevor sie benutzt werden (Innerhalb der if Regel, nicht hundert Zeilen vorher). Ergebnisnamen X2, Y2 tauchen danach auf, da ja erst die Berechnung in translate() gemacht werden muss.

Prolog Datei:

```
4
5 % Translate position concerning direction nswe
6 translate(n, X1, Y1, X2, Y2) :- X2 = X1, Y2 is Y1 - 1.
7 translate(s, X1, Y1, X2, Y2) :- X2 = X1, Y2 is Y1 + 1.
8 translate(e, X1, Y1, X2, Y2) :- Y2 = Y1, X2 is X1 + 1.
9 translate(w, X1, Y1, X2, Y2) :- Y2 = Y1, X2 is X1 - 1.
10
```

Die Formel translate() ist in Prolog beschrieben.

", " ist and (in Prolog und GOAL)

"is" in Prolog ist "="

":- " ist Regeloperator gleich umgekehrten Implikationspfeil, rechte Seite impliziert linke Seite

Schaubild 1: Implizite Definition von Variablen

<pre> 1 :- dynamic stepAwaitingAction/1, haveMove/1, step/1. 2 :- dynamic agentAt/2, thing/4, randomAffinity/1. 3 4 5 % Translate/update position XY concerning direction nsw_e 6 translate(n, X1, Y1, X2, Y2) :- X2 = X1, Y2 is Y1 - 1. 7 translate(s, X1, Y1, X2, Y2) :- X2 = X1, Y2 is Y1 + 1. 8 translate(e, X1, Y1, X2, Y2) :- Y2 = Y1, X2 is X1 + 1. 9 translate(w, X1, Y1, X2, Y2) :- Y2 = Y1, X2 is X1 - 1. 10 11 % Update position XY in relation to agent position 12 % localize(X1, Y1, X2, Y2, X3, Y3) :- X3 is X1 + X2, Y3 is Y1 + Y2. 13 % delocalize(X1, Y1, X2, Y2, X3, Y3) :- X3 is X1 - X2, Y3 is Y1 - Y2. 14 15 % get random nsw_e direction 16 randomDirection(Dir) :- random_between(0, 3, D), 17 integerToDirection(D, Dir). 18 19 % get random 90 degree direction to initial affinity direction 20 random90Direction(Affini, AltDir) :- random_between(0, 1, RandD), 21 flankingDirection(RandD, Affini, AltDir). 22 23 % helper function random to direction 24 integerToDirection(0, n). 25 integerToDirection(1, s). 26 integerToDirection(2, w). 27 integerToDirection(3, e). 28 29 % helper function random to 90 degree direction 30 flankingDirection(0, n, w). 31 flankingDirection(1, n, e). 32 flankingDirection(0, s, w). 33 flankingDirection(1, s, e). 34 flankingDirection(0, e, s). 35 flankingDirection(1, e, n). 36 flankingDirection(0, w, s). 37 flankingDirection(1, w, n). </pre>	<p>kleine Buchstaben sind hier voll Okay</p> <p>Anfangsbuchstabe</p> <p>Ab hier gehen nur Großbuchstaben für rechnende Variablen, sonst geht nichts, wird dann auch schön blau markiert die Var wenn korrekt, wird sich aber auch nicht beschwert da die Strings n s w e ja auch hier stehen. Hat mich 2-3 Stunden gekostet. :D</p>
--	---

Schaubild 2: Groß- und Kleinschreibung Variablen Unterschiede

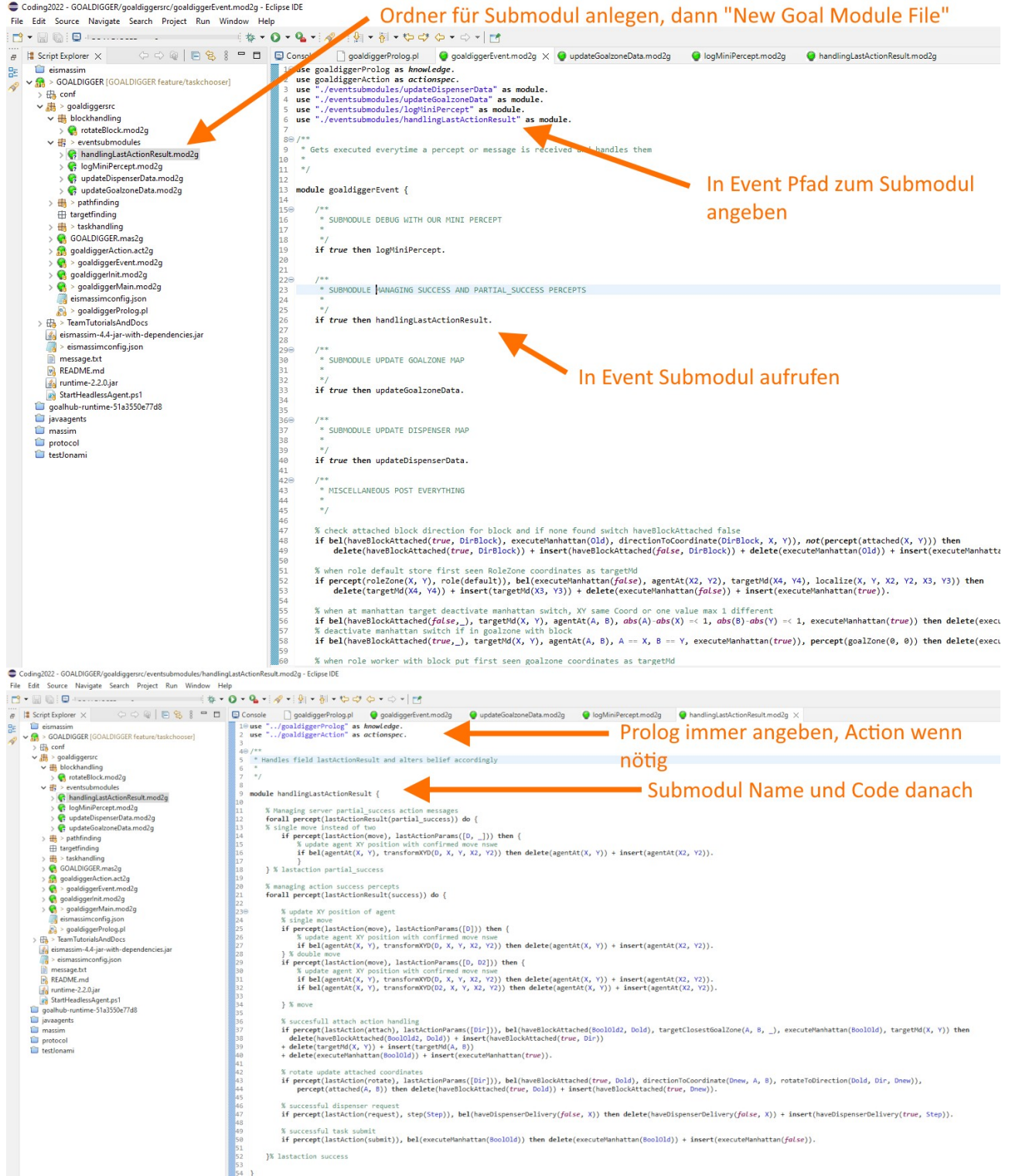


Schaubild 4: Submodul anlegen

connectionA6.", Beispiel 1 Submodule Debugging

```

43341 18:34:00.755,[connectionA6] performed 'updateDispenserData'., aus Event Module aufgerufen
43342 18:34:00.758,[connectionA6] empty update on beliefbase connectionA6.,
43343 18:34:00.758,[connectionA6] performed 'clear(0, -1)'., 2. ausgewählte Aktion beendet randomExecuteMove Modul
43344 18:34:00.758,[connectionA6] deleted 'haveMove(false)' from beliefbase connectionA6.,
43345 18:34:00.758,[connectionA6] inserted 'haveMove(true)' into beliefbase connectionA6.,
43346 18:34:00.758,[connectionA6] performed 'randomExecuteMove'., 1. Modul
43347 18:34:00.817,"[connectionA6] non-state actions: 19, send actions 0, state queries: 1068, total[beliefs: 62,
goals: 0, messages: 0, percepts: 284] Main Zyklus zu ende
43348 ++++++ Cycle 691 ++++++",

```

```

18:32:25.254,[connectionA3] inserted 'wMd(3)' into beliefbase connectionA3., Beispiel 2
18:32:25.254,[connectionA3] deleted 'eMd(6)' from beliefbase connectionA3.,
18:32:25.254,[connectionA3] inserted 'eMd(5)' into beliefbase connectionA3.,
18:32:25.254,[connectionA3] performed 2!manhattanCalcModule'., Wenn ein Modul fertig ausgeführt wurde erscheint es als performed
18:32:25.257,[connectionA3] empty update on beliefbase connectionA3.,
18:32:25.257,[connectionA3] performed 'move(n)'., 4. ausgewählte Aktion
18:32:25.258,[connectionA3] deleted 'haveMove(false)' from beliefbase connectionA3.,
18:32:25.258,[connectionA3] inserted 'haveMove(true)' into beliefbase connectionA3.,
18:32:25.258,[connectionA3] performed 3!manhattanChoiceModule'.,
18:32:25.258,[connectionA3] performed 1!manhattanExecuteMove'.,
18:32:25.334,"[connectionA3] non state actions: 19, send actions 0, state queries: 177, total[beliefs: 22,
goals: 0, messages: 0, percepts: 81]
+++++ Cycle 8 ++++++",

```

Schaubild 4: Ablauf der Modulaufrufe und Aktionsauswahl