UNICAMP
Universidade Estadual de Campinas

FEEC
Faculdade de Engenharia Elétrica e de Computação

Aluno: Mateus Neves Barreto
R.A.: 142358
Disciplina: IA006
Professor: Ricardo R. Gudwin

Relatório – Aula 4

1 Atividade 1

A execução e ambientalização com o aplicativo TankSoar foi bem simples. No ambiente de 14 por 14 quadros, podemos encontrar objetos como:

- Tanques;
- Obstáculos;
- Recarregador de vida;
- Recarregador de energia;
- Pacotes de mísseis.

Cada pacote de míssil contém 7 unidades. Cada tanque possui energia (de 0 a 1000 pontos), saúde (de 0 a 1000 pontos), mísseis, sensores e pontos para classificar sua eficiência. Um tanque possui seis diferentes sensores:

- **Blocked:** verifica a vizinhança de um quadro do tanque e informa se ao norte, sul, leste e oeste possuem ou não obstáculos;
- **Incoming:** verifica a aproximação a qualquer distância de um míssil inimigo. Informando existem o mísseis se aproximando por norte, sul, leste e oeste, não identifica mísseis atrás de obstáculos e/ou tanques e não identifica quem lançou o míssil;
- **Radar:** o radar possui 5 qtributos de identificação:
  - energy: se existe um carregador de energia é informado a distância em quadros e a direção: direita, esquerda ou a frente;
  - health: se existe um carregador de vida é informado a distância em quadros e a direção: direita, esquerda ou a frente;
  - obstacle: se existem obstáculos são informados a distância e a direção: direita, esquerda ou a frente de cada obstáculo;
  - open: se existem quadros vazios são informados a distância e a direção: direita, esquerda ou a frente de cada quadro vazio;
  - tank: se existem tanques são informados a distância, a direção: direita, esquerda ou a frente e a cor de cada tanque;
- **Rwaves:** verrifica pelas quatro direções se o radar de algum tanque o está detectando;
- **Smell:** detecta a cor e a distância do tanque mais próximo; Se não existir tanque mais próximo é retornado none/nada.
- **Sound:** detecta a direção do tanque mais próximo que se moveu na última jogada.

Além destes sensores os tanques possuem outros atributos como status do radar, cor, escudos e outros. A Figura 1 e 2 apresenta a criação dos tanques e a execução da aplicação respectivamente.

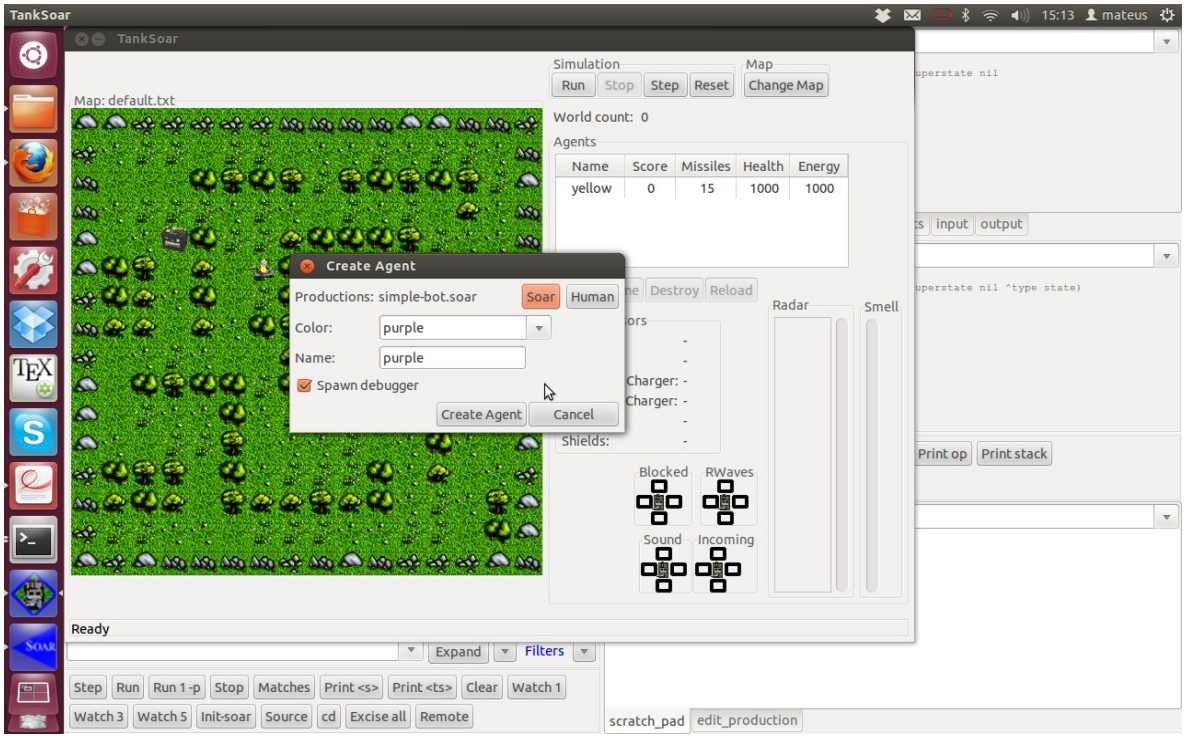

Figura 1 – Criando tanques.

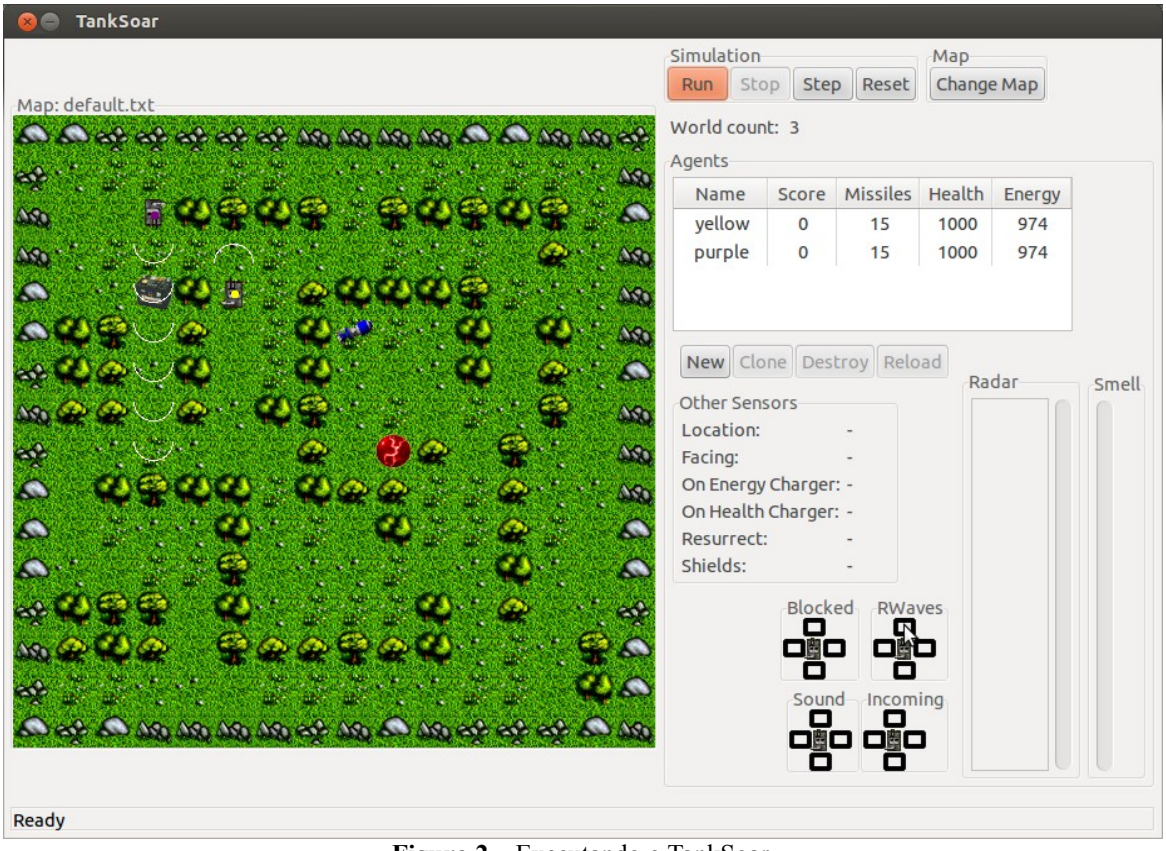

Figura 2 – Executando o TankSoar.

2 Atividade 2

A Figura 3 apresenta a criação do projeto template tanksoar no VisualSoar.

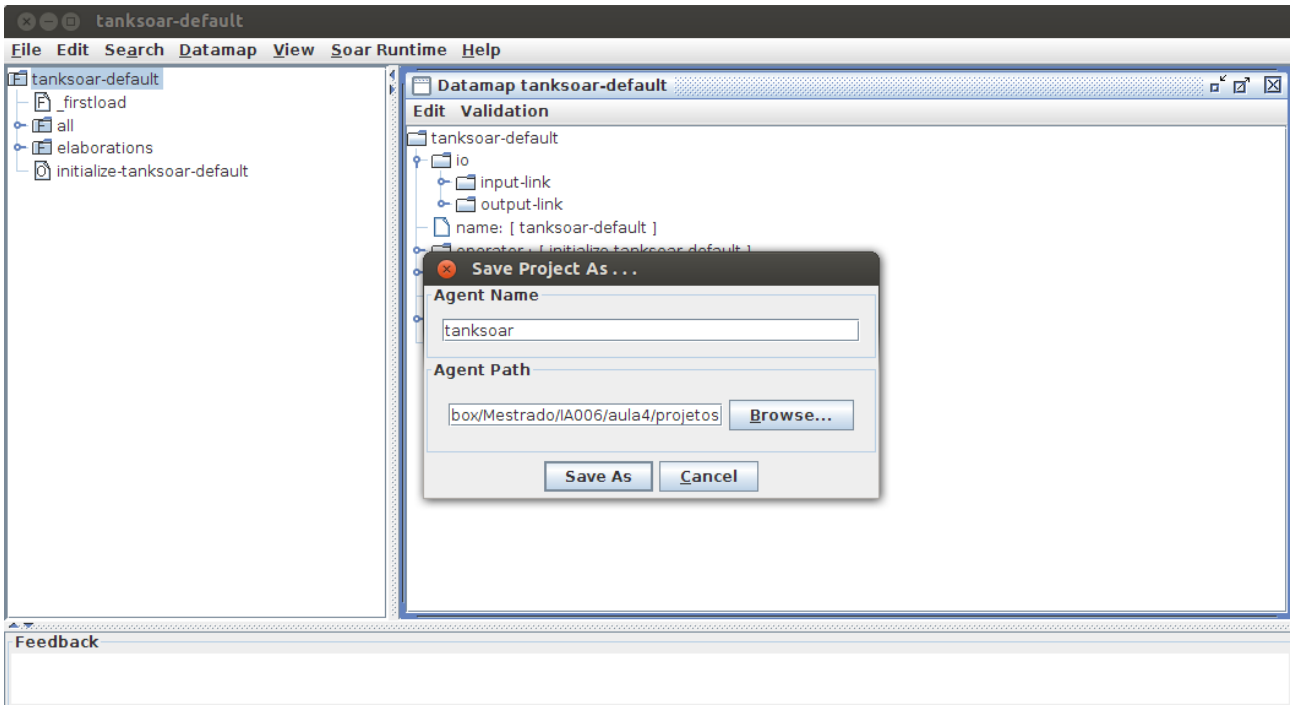

Figura 3 – Salvando template no VisualSoar.

2.1 Move e Turn

Para o tanque poder se locomover, é preciso propor operadores que possibilite tal ação. Neste subtópico do tutorial, são apresentadas 3 regras: a regra move, turn e a turn*backward. Dentre elas, é possível andar para frente (*move*), virar para à direita ou para esquerda (*turn*) e voltar (*turn*backward*). A Figura 4 apresenta a implementação destas regras.

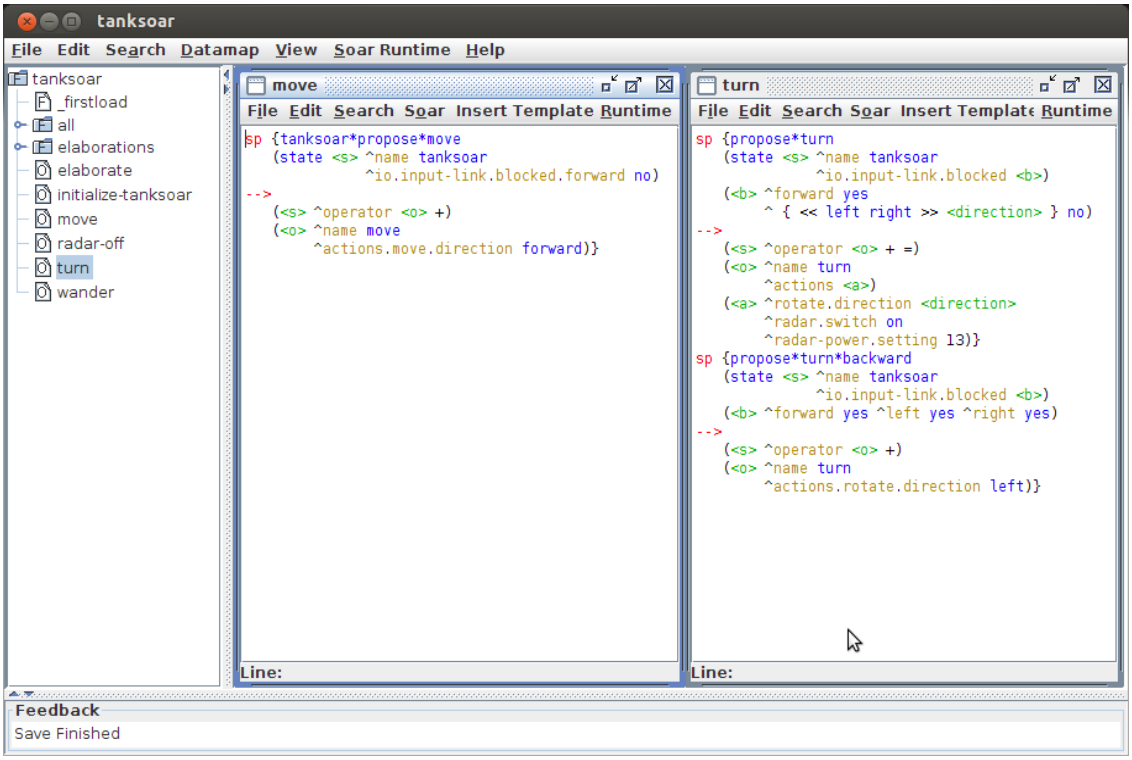

Figura 4 – Operadores *move* e *turn*.

2.2 Economizando energia

Para economizar ernergia, o tem que precisa evitar deixar o radar ligado. Para isso o tutorial propõe regras para otimizar a utilização do mesmo, evitando o desperdício de energia. As Figuras 5 e 6 apresentam respectivamente a implementação e análise das regras.

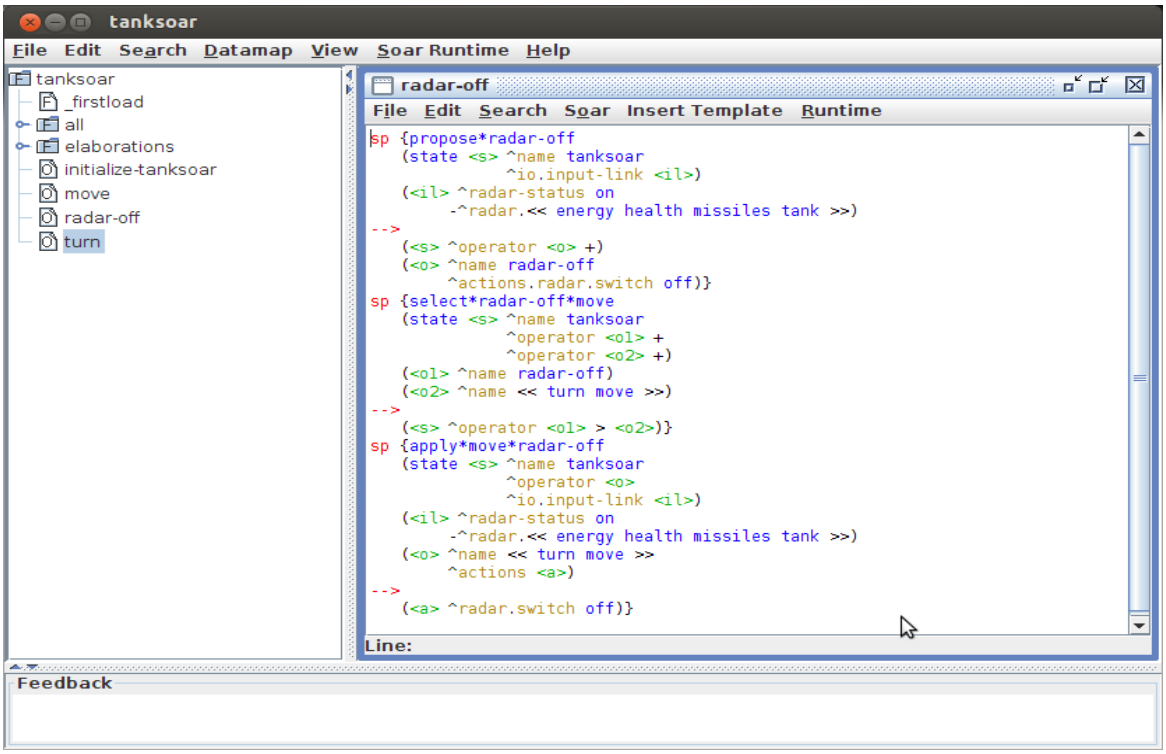

Figura 5 – Regras radar-off.

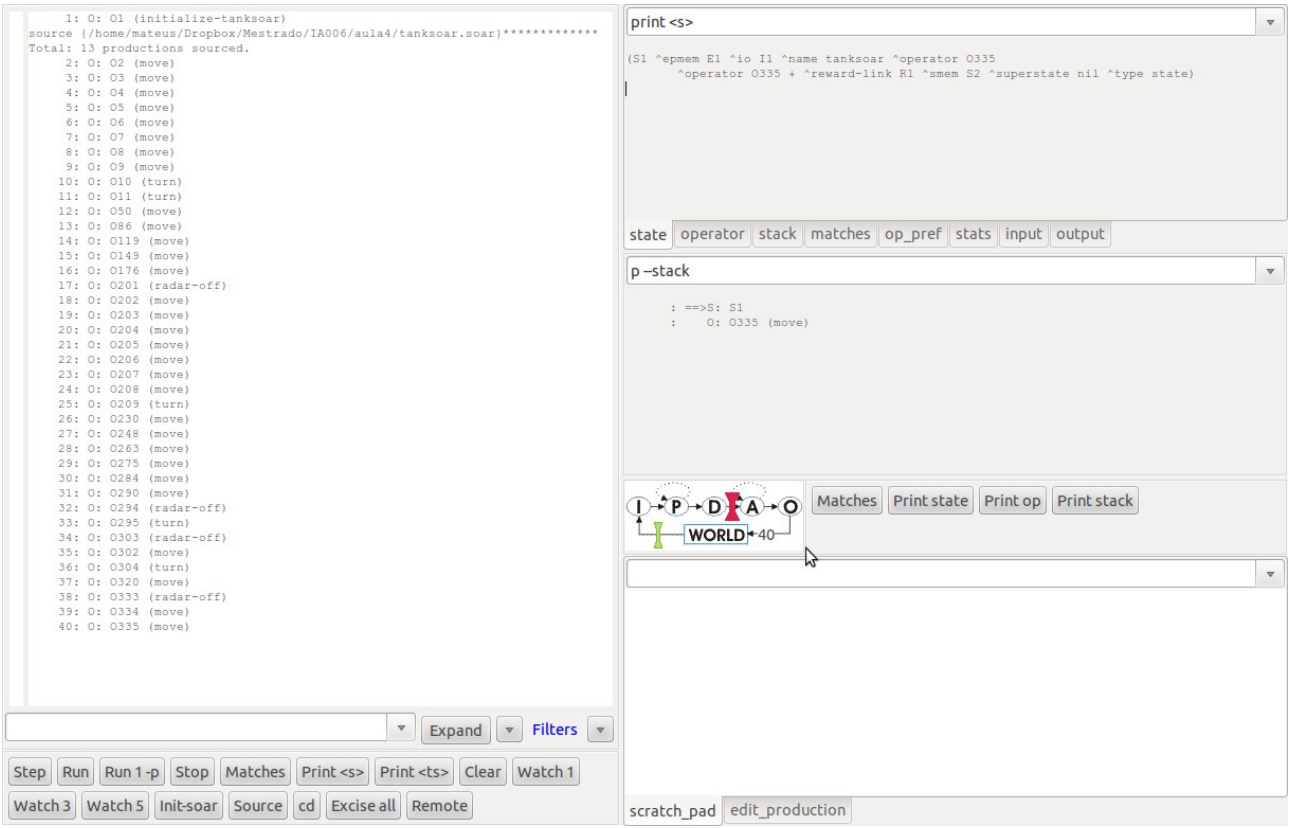

Figura 6 – Análise de execução do tanque contendo a regra radar-off.

2.3 Objetivos do tanque

O tanque que é apresentado no tutorial possui 4 objetivos básicos. O esquema presente na Figura 7, apresenta a dispocição e possíveis estados dos 4 objetivos ou operações que o tanque deve seguir.

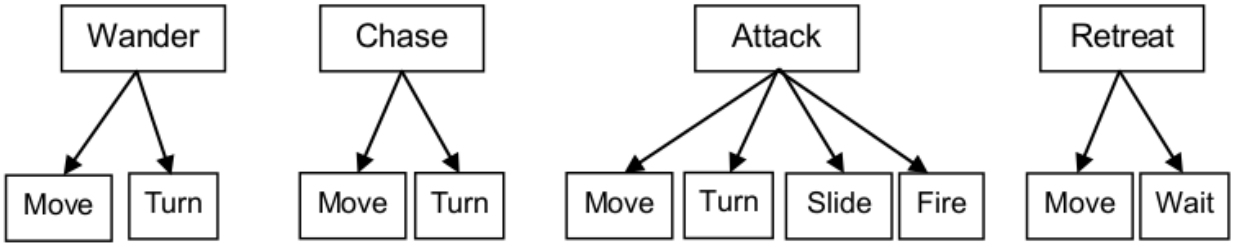

Figura 7 – Operações que o tanque deve seguir.

2.3.1 Wander

Se o tanque não identificar nada nos sensores, ele deve realizar o operador "*wander*", que possui um significado de "andar sem destino", com o intuito de identificar algo nos sensores. A Figura 8 presenta a implementação básica desta regra.

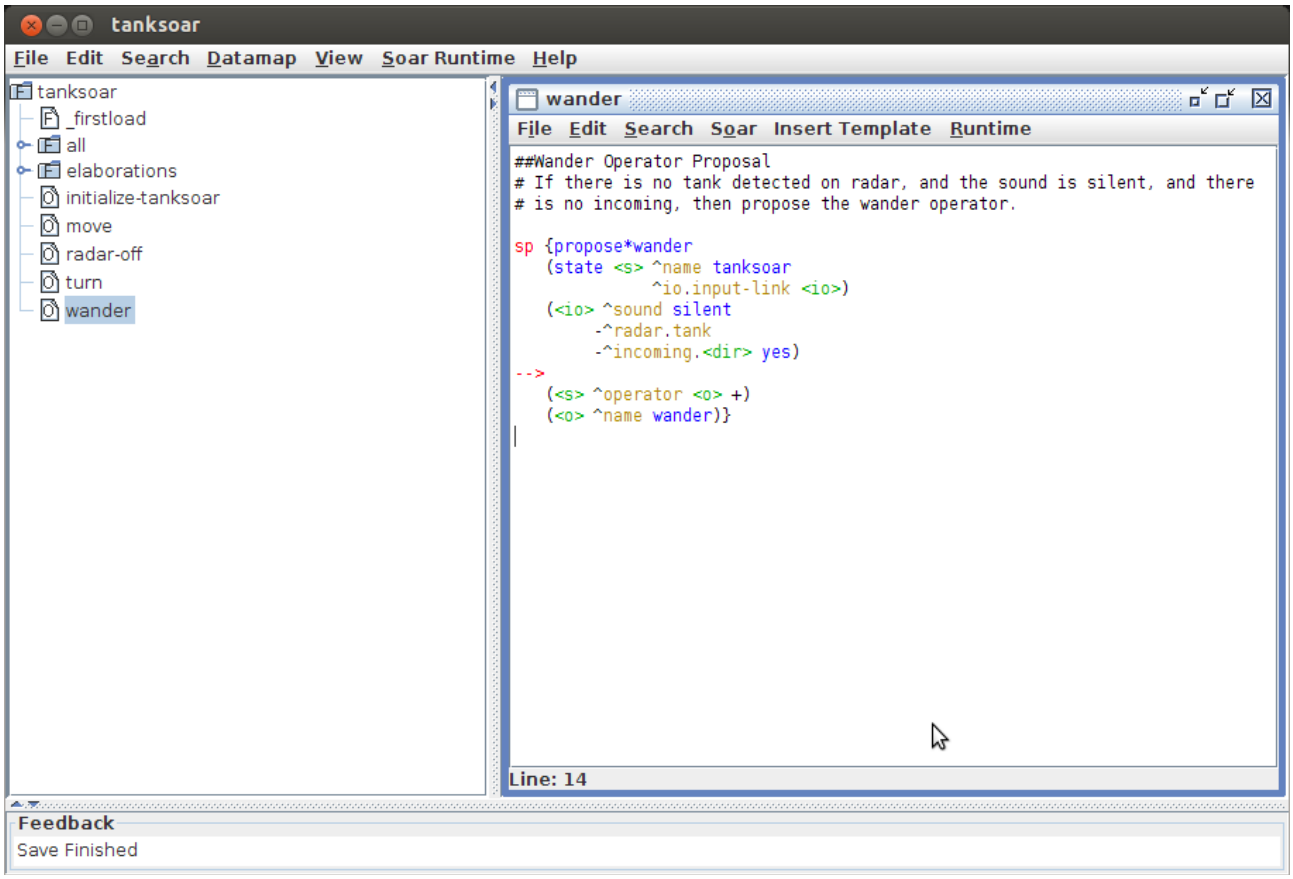

Figura 8 – Regra do operador *wander*.

Porém quando se utiliza apenas esta regra é gerado um estado de *"state no-change"*, significando que as regras propostas não estão mudando, ou conseguindo ser combinadas com outras regras. Então sempre é proposta a regra inicial.

Para contornar este problema, o tutorial apresenta e detalha e implementação das regras de modo correto. A Figura 9 apresenta o código das regras para o suboperador "*wander*" e as regras gerais necessárias na combinação presentes no arquivo *"directory"* presente na diretório "*elaborations*".

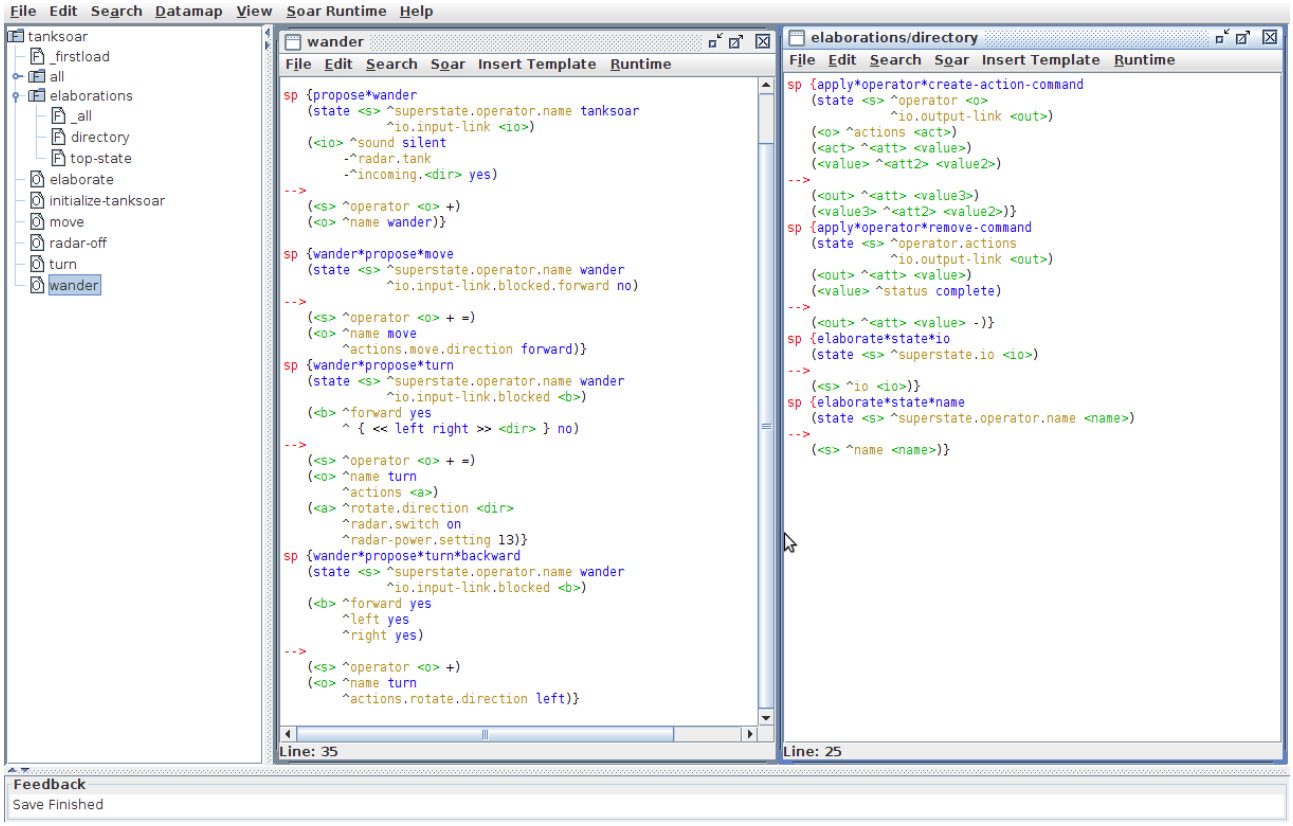

Figura 9 – Regras do operador *wander* e regras gerais.

2.3.2 *Chase, Attack and Retreat*

O operador *"chase"* é o operador de perseguição. Ele deve ser ativado somente quando o sensor de som detecta um tanque, porém não sabe exatamente onde o mesmo se encontra. Também deve ser evidado de ser ativado quando o tanque está com baixa energia ou baixa vida, pois ele possivelmente irá encontrar o objeto caçado, o tanque adversário. Se o caçador estiver com baixas energias não poderá ativar os escudos; outro caso é se o caçador não possuir mísseis, não fara sentido caçar o adversário.

Após a criação das propostas do operador *"chase"* foram implementadas as propostas de regras dos operadores *"attack"* e *"retreat"* ataque e recuar respectivamente. Essas duas funções completam os 4 objetivos que o tanque deve realizar e buscar cumpri-los. A Figura 10 apresenta a realização desta tarefa.

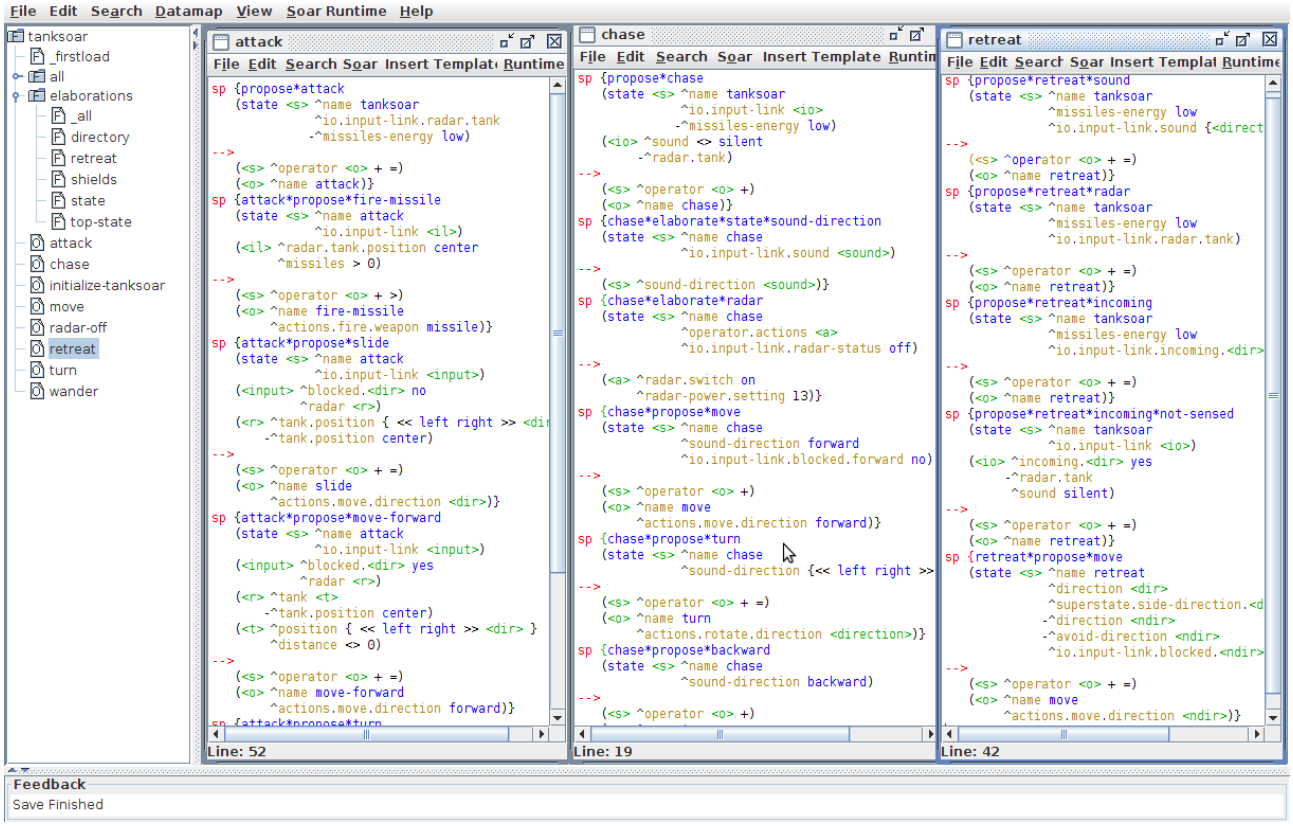

Figura 10 – Regras dos operadores *chase, attack* e *retreat*.

Além dessas regras, foi implementada a regra de escudos (*shields*). O escudo auxilia o tanque para poder se aproximar do adversário. A implementação desta regra pode ser observada na Figura 11.

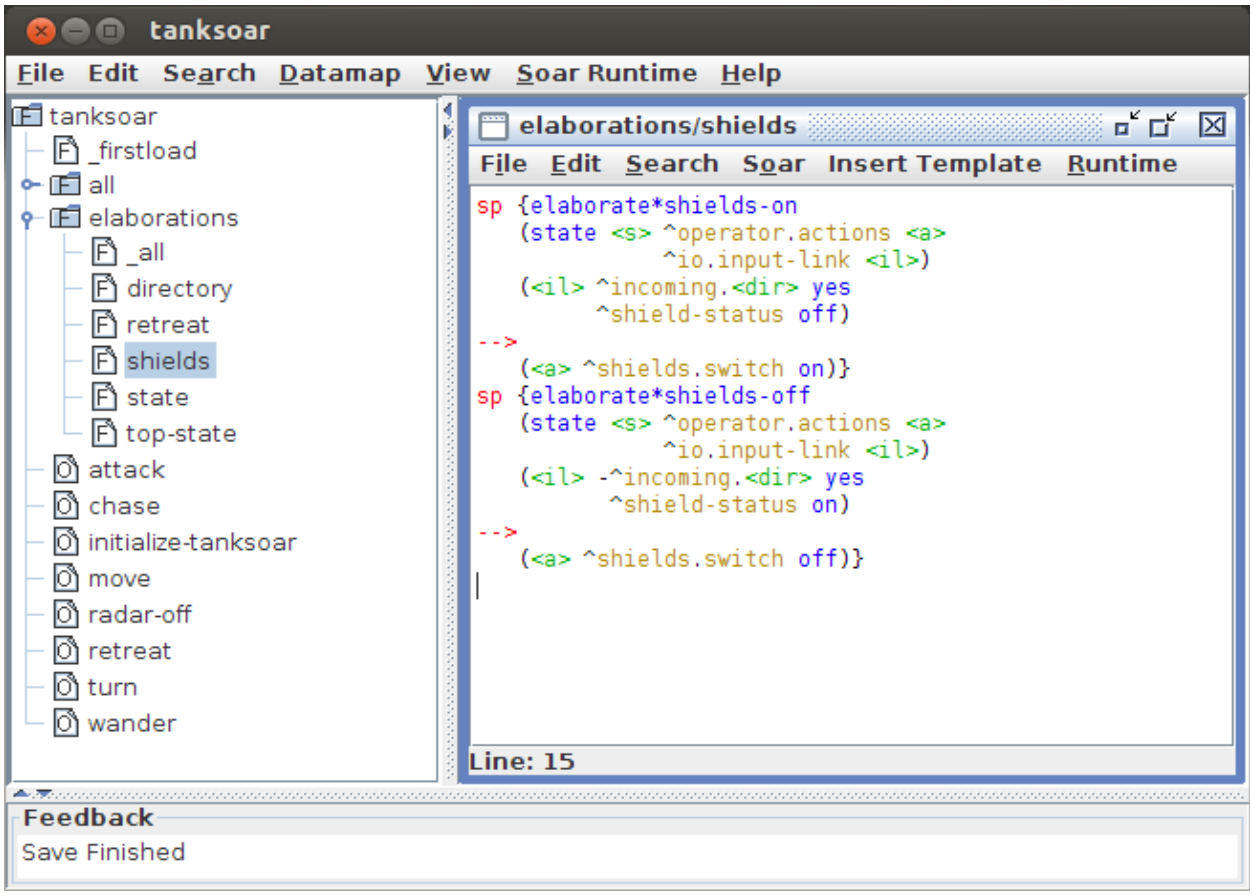

Figura 11 – Regrado escudo *(shield)*.

Quando nenhum operador é proposto, por algum caso, ou de bug do sistema ou de erro lógico, o estado "*no-change*" é atingido e o programa trava. Para evitar este estado, e trata-lo, o tutorial apresenta a regra *"wait"* como solução, Figura 12.

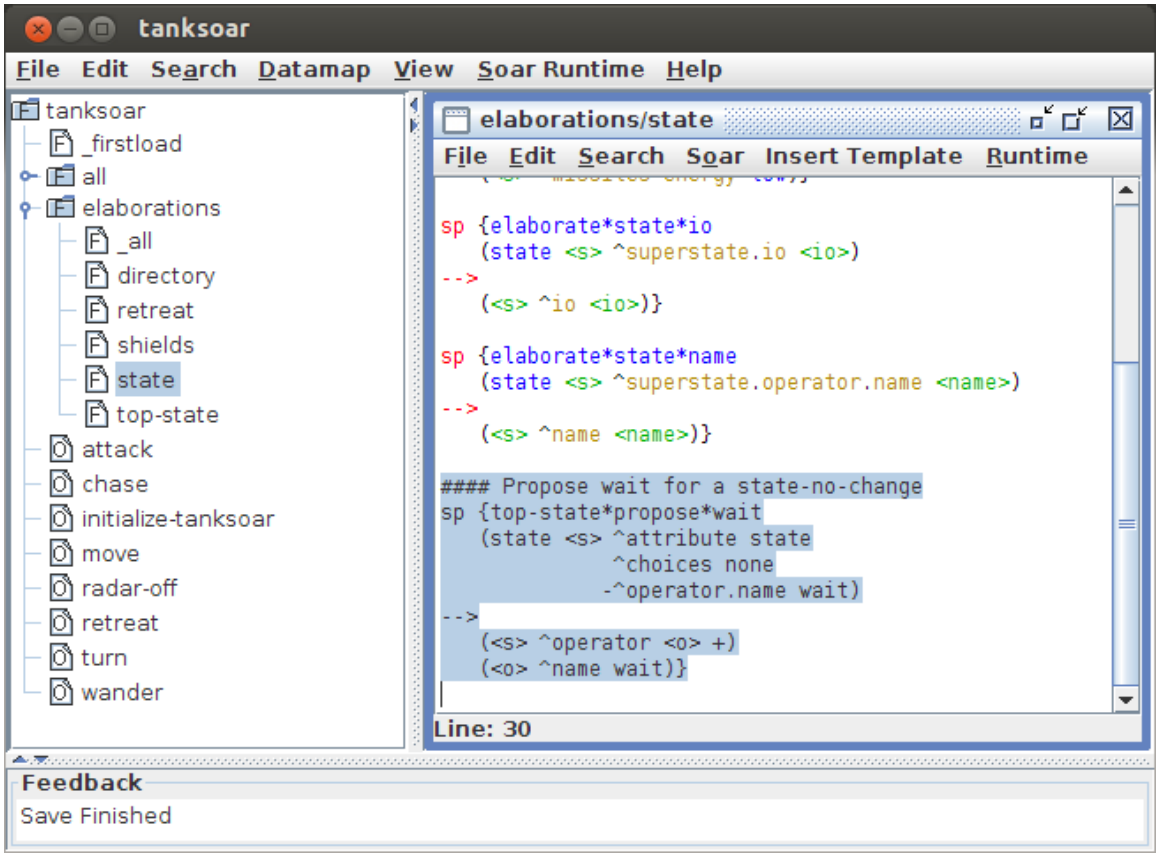

Figura 12 – Estado *no-change.*

2.4 Melhorando a detecção de som

Essa técnica proposta no tutorial, visa a melhora do dispositivo de sensor sonoro. Pois se um tanque detecta um inimigo e o adversário para de se mover, não é possível mais rastreá-lo pelo som. Deste modo, a melhoria proposta, visa persistir a direção em que o som foi observado para que o tanque caçador possar continuar a sua busca pelo seu adversário. Esse som persistido perdura por 5 ciclos, pois se o tanque adversário não foi encontrado nos 5 próximos ciclos possivelmente o mesmo teve êxito na fuga. A Figura 13 apresenta a implementação da melhoria da utilização do dispositivo de som.

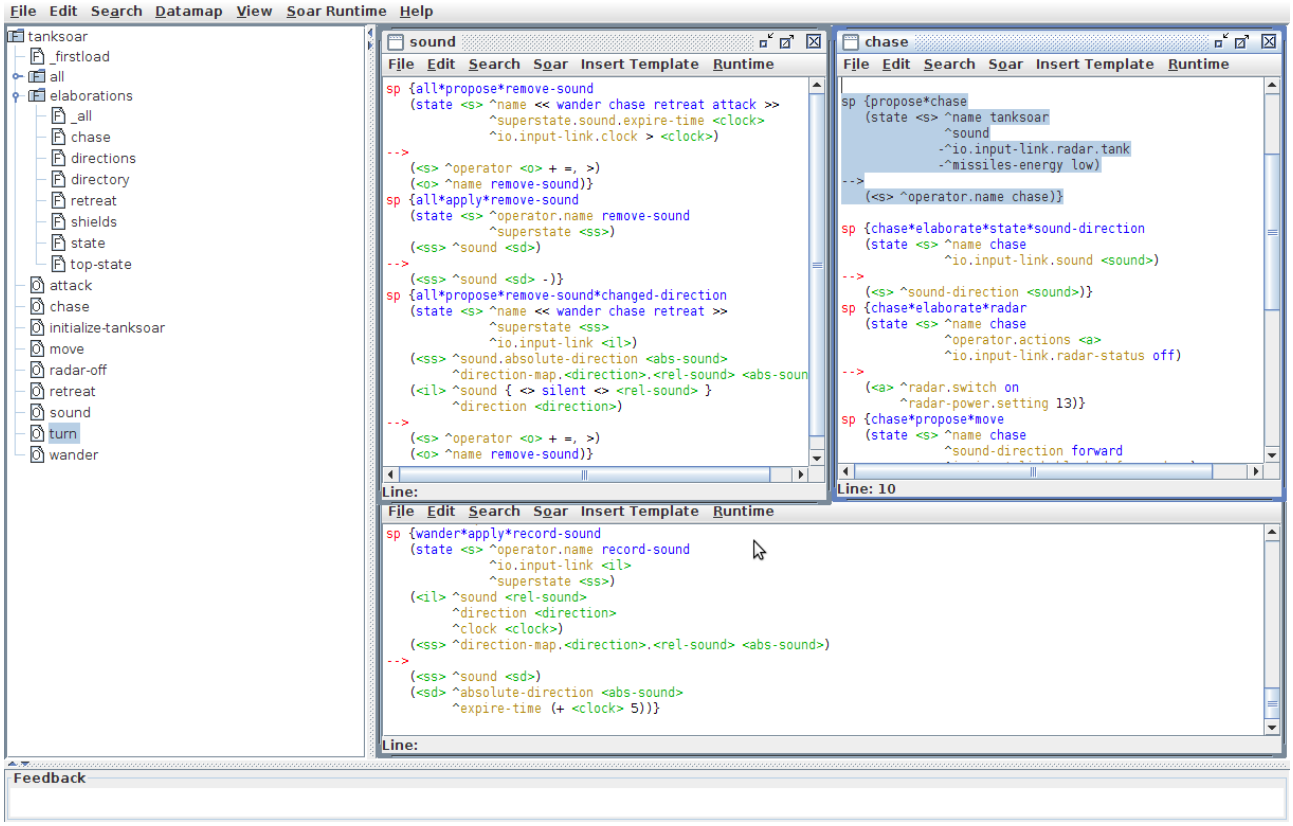

Figura 13 – Regras para melhorias dispositivo som.

3 Atividade 3

A atividade 3 propõe a criação de um tanque com preferência de busca de saúde e energia com base no map de estrutura e inicialização *"init-map"*.

O mapa de estrutura deve ser armazenado e atualizado enquanto está participando do jogo; o mapa do tanque é armazenado na memória de trabalho durante o jogo. A implementação das regras do mapa estão presentes na Figura 14.

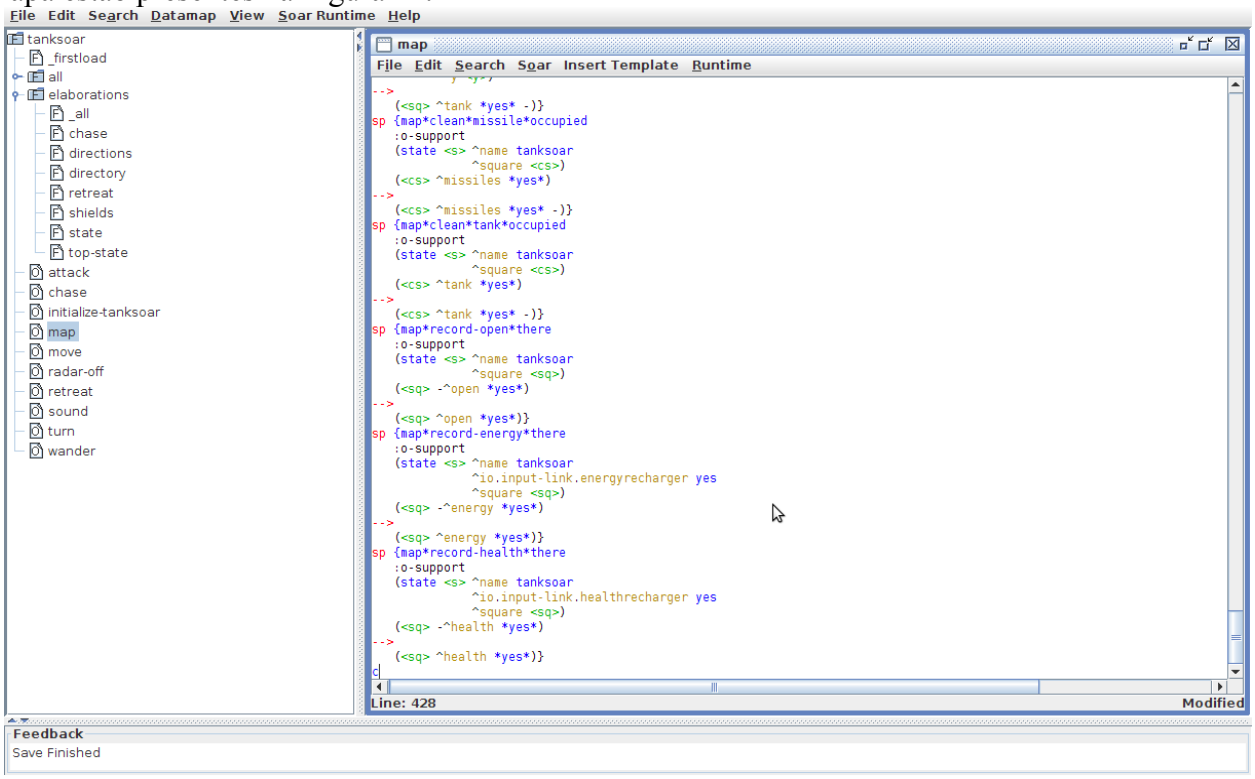

Figura 14 – Implementação das regras *map*.