

MODELAGEM DE FORÇAS DE DEFESA BASEADA EM AGENTES

Gustavo Henrique Soares de Oliveira Lyrio

Instituto Nacional de Matemática Pura e Aplicada – IMPA
Estrada Dona Castorina, 110 – Rio de Janeiro, RJ – 22460-320
glyrio@impa.br

Grupo de Tecnologia em Computação Gráfica – Tecgraf/PUC-Rio
Rua Marquês de São Vicente, 225 – Rio de Janeiro, RJ – 22453-900
glyrio@tecgraf.puc-rio.br

Roberto de Beauclair Seixas

Instituto Nacional de Matemática Pura e Aplicada – IMPA
Estrada Dona Castorina, 110 – Rio de Janeiro, RJ – 22460-320
rbs@impa.br

Grupo de Tecnologia em Computação Gráfica – Tecgraf/PUC-Rio
Rua Marquês de São Vicente, 225 – Rio de Janeiro, RJ – 22453-900
rbs@tecgraf.puc-rio.br

Resumo

Este trabalho apresenta uma abordagem para modelar agentes com comportamento pré-definido usando máquinas de estado de modo a figurarem como força de defesa, controlados por um centro de comando e controle.

Palavras-Chaves: Modelagem por Agentes; Comportamento Pré-Definido; Máquinas de Estado; Defesa; Comando e Controle.

Abstract

This paper presents an approach for modeling pre-defined behavior agents through state machines to figure out as a defense force, controlled by a command and control center.

Keywords: Agent Modeling; Pre-Defined Behavior; State Machines; Defense; Command and Control;

1. INTRODUÇÃO

O objetivo é desenvolver um sistema capaz de simular estratégias e linhas de ações elaboradas pelos oficiais alunos do Curso de Aperfeiçoamento de Oficiais do Corpo de Fuzileiros Navais (CAO-CFN) permitindo ou não a validação destas estratégias.

O sistema atual é de dupla ação, significando que tanto as forças de ataque quanto de defesa são controladas por oficiais alunos do CAO-CFN. Como a defesa não é uma tarefa normalmente atribuída aos fuzileiros navais, o grupo de alunos que recebe esta tarefa tem seu treinamento prejudicado.

Para resolver este problema introduzimos inteligência artificial utilizando o conceito de agentes para modelar unidades de defesa, onde uma unidade pode ser qualquer fração de uma força militar (esquadras de tiro, pelotão, companhias, etc.).

Para criar um modelo usando inteligência artificial deve-se analisar a sua forma de pensar (percepção, ação, interação,...) e identificar suas principais características. As primeiras características que se destacam quando analisamos a forma de pensar de unidades de defesa é que elas não agem de acordo com sua própria vontade de forma independente. Unidades militares, tanto de defesa quanto de ataque, obedecem ordens provenientes de um comando hierarquicamente superior, assim como seguem estratégias de combate e doutrinas.

Seu comportamento é bem conhecido e amplamente descrito nos manuais de treinamento. Essa característica levou-nos a construir nossos modelos de agentes usando comportamento pré-definido, baseando o comportamento das unidades nas técnicas descritas nos manuais de treinamento e no conhecimento de um instrutor de defesa do CAO-CFN.

Este trabalho apresenta uma abordagem para modelar rapidamente agentes com comportamento pré-definido usando máquinas de estado de modo a figurarem como força de defesa, controlados por um centro de comando e controle.

Embora a descentralização seja a filosofia mais adotada de comando e controle, algumas decisões ainda tem de ser tomadas por um comando (hierarquicamente) superior que conhece a situação como um todo, bem como as vantagens e fraquezas de cada uma das suas unidades de combate, sendo capaz de coordenar seus esforços provendo uma melhor defesa.

1.1. MODELAGEM DE COMANDO E CONTROLE

Conforme dito anteriormente, usar agentes para modelar unidades militares de defesa traz a necessidade de algum tipo de comando centralizado. As unidades não podem figurar no terreno e atuar por conta própria ou então o modelo passaria a ser um modelo de foco de resistência descentralizado. É necessário alguma centralização para coordenar essas unidades. Em um cenário de combate real esse papel seria realizado por um posto de comando (PC) que receberia informações de suas unidades, analisaria o cenário e tomaria decisões gerais como por exemplo, decidir quais missões de apoio de fogo deveriam ser cumpridas.

No modelo proposto os agentes se mantém em suas posições de defesa até que algum tipo de ameaça seja verificada. Então o agente que verificou a ameaça coloca a posição que a possível ameaça ocupa em uma lista de alvos (simulando a mensagem “plotado inimigo nas seguintes coordenadas” enviada ao posto de comando). Após o término do turno de cada agente, o posto de comando analisa a lista de alvos verificando quais alvos podem ser atingidos por fogos de artilharia, quais alvos demandam apoio aéreo aproximado e quais alvos não são uma ameaça imediata. Então o posto de comando solicita as missões de tiro as unidades melhores posicionadas no terreno (Figura 1).



Figura 1: Esquema mostrando a atividade do posto de comando.

2. TÉCNICAS DE MODELAGEM DE AGENTES

“Um agente é uma entidade que pode ser perceber seu ambiente através de *sensors* e atuar sobre este ambiente através de *effectors* (Russell e Norving)”.

Partindo desta definição o objetivo é identificar o que é relevante para o agente perceber e que ações devem ser tomadas pelo agente. Dois modelos são necessários aqui: *sensors* e *effectors*.

Nossos *sensors* são modelados através de questões que um agente precisa responder quando imerso no ambiente. As respostas destas questões serão sempre verdadeiro ou falso. Agrupando-se algumas destas questões e suas respostas podem-se representar as situações relevantes em que o agente pode encontrar-se. Nós chamaremos estas situações de estados. Quanto mais complexo for o modelo, maior será o número de *sensors* e estados.

Agora que encontramos um meio de modelar os *sensors*, precisamos modelar os *effectors*. Faremos isso definindo ações que o agente deverá realizar quando encontrar-se em um determinado estado. Devemos também identificar quais eventos podem ocorrer no ambiente que podem vir a alterar o estado de nossos agentes. Chamaremos estes eventos de transições.

Vamos modelar um sentinela em uma torre por exemplo. Esta torre terá uma chave que dispara um alarme quando o sentinela verificar alguma ameaça se aproximando. Nosso agente sentinela deverá ter então dois *sensors*. Um *sensor* define se ele pode ver algum inimigo se aproximando; o outro *sensor* define se o alarme está ligado. Então, teremos 2 estados: vigiando e disparando alarme. Enquanto vigiando, o sentinela toma a ação de olhar para o horizonte verificando se algum inimigo se aproxima. Se nenhum inimigo se aproxima, o sentinela permanece no estado de vigia, mas se algum inimigo se aproxima ocorre uma transição para o estado de disparando alarme. Então o sentinela toma a ação de correr até a chave que aciona o alarme e acioná-lo e retorna para o estado de vigia.

É verdade que um sentinela modelado desta forma está bem longe da realidade. Dentre muitas outras coisas, um sentinela precisa descansar. Ele não pode permanecer vigiando para sempre. Bem, você sempre pode melhorar o modelo adicionando mais estados e transições. Vamos supor que o nosso sentinela tem um atributo chamado energia que varia de 100% (completamente ativo) até 0% (extremamente cansado). Quando no estado de vigia a energia do sentinela decresce de 1% a cada 7 minutos. Atingindo 20% o sentinela executa uma transição para um novo estado chamado: dormindo. Quando neste estado, o sentinela irá recuperar 3% de sua energia a cada 7 minutos e mudar para o estado de vigia quando a energia voltar a 100% novamente.

Poderíamos adicionar muitos outros estados para tornar nosso sentinela mais próximo da realidade, porém o objetivo aqui é apenas exemplificar como a idéia funciona.

Tipo de agente	Sensors	Effectors	Transições
Sentinela	- Existe alguma ameaça? - Alarme ligado?	- Buscar inimigo no campo visual. - Soar alarme.	- Inimigo avistado. - Alarme ligado.
Sentinela melhorado	- Existe alguma ameaça? - Alarme ligado? - Energia < 20%?	- Buscar inimigo no campo visual. - Soar alarme. - Dormir	- Inimigo avistado. - Alarme ligado. - Energia completa (100%)

Tabela 1. Tabela mostrando *sensors*, *effectors* e transições para ambos os exemplos de modelo de agentes.

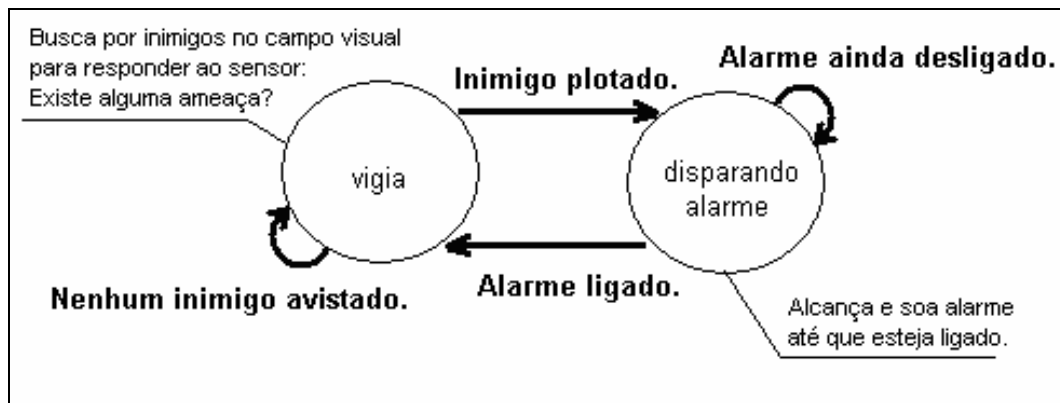


Figura 2: Diagrama de autômatos para modelo de agente sentinela.

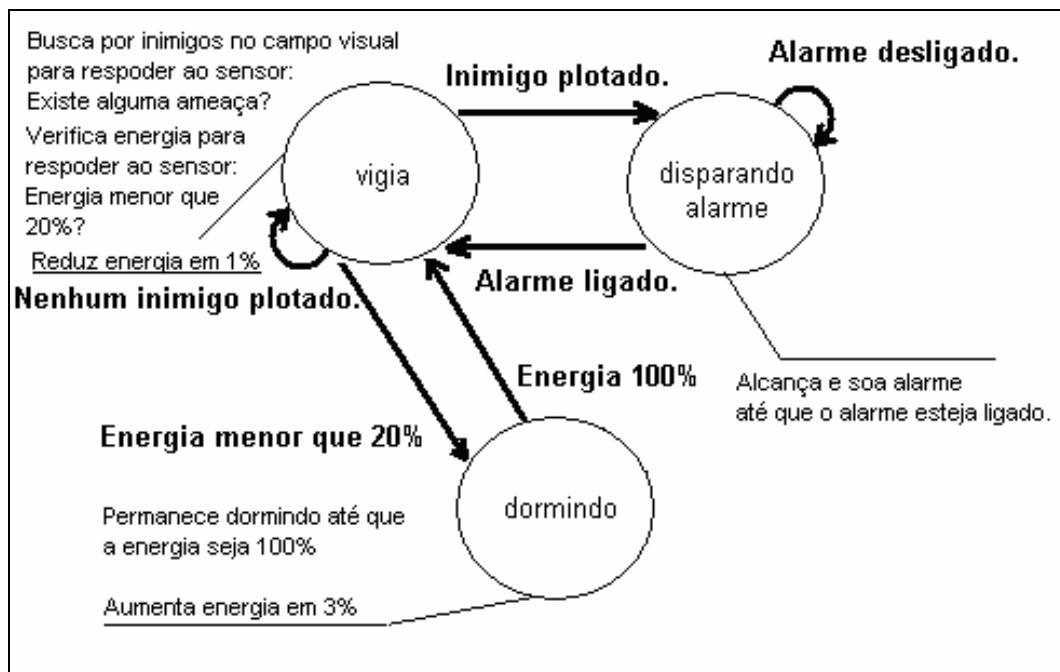


Figura 3: Diagrama de autômatos para modelo de agente sentinela melhorado.

Como pode-se verificar nas Figuras 2 e 3, essa técnica se adapta perfeitamente ao conceito de máquinas de estados finitos.

“Uma máquina de estados finitos (MEF) consiste em um conjunto de estados (incluindo um estado inicial), um conjunto de entradas, um conjunto de saídas e uma função de transição de estados. A função de transição de estados recebe a entrada e o estado corrente e retorna um único novo estado e um conjunto de saídas. Como existe um único novo estado possível, MSF são usadas para codificar comportamentos determinísticos (Funge)”.

O estado inicial da nossa MEF será o estado inicial do agente. No nosso exemplo do sentinela, o estado de vigia. As entradas serão os *sensors* e as saídas serão esses *sensors* atualizados. Finalmente, as transições são condições que devem ser satisfeitas para permitir a função de transição modificar o estado corrente, como algum inimigo se aproximando ou alarme ligado.

Em resumo, estamos usando máquinas de estado finito para modelar o comportamento de agentes. Para isso agrupamos um ou mais *sensors* para criar estados. Quando em cada estado o agente irá executar uma ou mais ações sobre si mesmo ou sobre o ambiente. Essas ações são nossos *effectors*. Alguns eventos farão com que o agente mude de um estado para outro. Esses eventos são as transições da máquina de estados.

3. DOCTRINA E COMPORTAMENTO DAS UNIDADES DE DEFESA

Nosso modelo de agente de defesa é baseado na doutrina de guerra de atrito. Nós escolhemos a guerra de atrito por ser a doutrina de defesa mais utilizada. A doutrina de guerra de atrito pode ser resumida em:

- Destruição cumulativa do inimigo através de aplicação de poder de fogo superior.
- Minimizar a força do inimigo
- Controle centralizado
- Terreno capturado (defendido em nosso caso) é a métrica primária de batalha.

Usando esse resumo como objetivo e com a ajuda de um oficial instrutor de defesa dividimos o comportamento dos agentes em dois tipos distintos de comportamento de defesa: comportamento de defesa de posição e comportamento de retardo do inimigo.

No entanto, ambos os comportamentos são de unidades de defesa, então existem muitos *sensors* comuns entre eles.

Uma unidade de defesa precisa ser capaz de perceber quando o inimigo se aproxima e executar ações em ambos os casos. No caso de nenhum inimigo estar visível no momento, a unidade precisa continuar procurando. Isso determina nossos primeiros *sensor* e *effector*:

Sensor: Posso ver algum inimigo?

Effector: Buscar por ameaça.

Quando um inimigo é avistado (Resposta afirmativa para o *sensor* anterior) a unidade necessita saber se é um inimigo novo ou se esse inimigo já havia sido avistado anteriormente, dessa forma evita-se que a unidade fique solicitando missões de tiro a cada ciclo sobre o mesmo alvo. Nossos próximos *sensor* e *effector* serão:

Sensor: É um novo alvo?

Effector : Solicitar apoio de fogo.

Devido ao longo alcance de alguns armamentos de artilharia a unidade provavelmente receberá fogos do inimigo antes que possa detectá-lo, ela terá que calcular a possível posição do inimigo (novo *effector*) para solicitar apoio de fogo sobre esta posição gerando assim um novo *sensor*:

Sensor: - Estou sobre fogos de artilharia?

Effector: Calcular posição do inimigo.

A unidade também necessitará saber quando está sobre ataque de fogos diretos (tiro tenso) e em caso afirmativo se o contra-ataque é possível. Visto que o contra-ataque é possível a unidade engaja o inimigo. Outros dois *sensors* comuns a ambos os comportamentos são:

Sensor: Estou sobre fogo direto?

Sensor: Posso engajar o inimigo?

Effector: Engajar o inimigo.

O último *sensor* comum está relacionado com o efetivo da unidade. Será necessário determinar quando a unidade perdeu metade de seus homens. Este é um valor crítico e um novo e bastante particular (uma vez que o *effector* que ele demanda depende do comportamento adotado) *sensor* será necessário aqui:

Sensor: Perdi metade do meu efetivo?

Effector: O *effector* aqui depende do comportamento e de outros *sensors* e será explicado mais adiante.

Definidos os *sensors* e *effectors* comuns, vamos recordar que temos dois comportamentos distintos a serem modelados: defesa de posição e retardo do inimigo.

Quando defendendo uma posição a unidade não pode abandoná-la em quaisquer circunstâncias e quando as baixas atingirem 50% do seu efetivo essa unidade irá solicitar reforços e aguardar a sua chegada. Esses serão nossos novos *sensor* e *effector* particulares ao comportamento de defesa de posição:

Sensor: Chegaram os reforços?

Effector: Solicitar reforços.

Mas se o objetivo é retardar o inimigo, quando as baixas atingirem 50%, a unidade de defesa recua para uma nova posição no terreno. A unidade vai precisar saber quando a nova posição for alcançada. Isso proporciona nossos novos *sensor* e *effector* particulares a esse comportamento:

Sensor: Alcancei a minha nova posição?

Effector: Recuar.

Tipo de agente	Sensors	Effectors
Unidade de defesa de posição	1)Posso ver algum inimigo? 2)É um novo alvo? 3)Estou sobre fogos de artilharia? 4)Estou sobre fogo direto? 5)Posso engajar o inimigo? 6)Perdi metade do meu efetivo? 7)Chegaram os reforços?	Busca por ameaça, Solicita apoio de fogo, Calcula posição do inimigo, engaja inimigo, solicita reforços
Unidade de retardo inimigo	1)Posso ver algum inimigo? 2)É um novo alvo? 3)Estou sobre fogos de artilharia? 4)Estou sobre fogo direto? 5)Posso engajar o inimigo? 6)Perdi metade do meu efetivo? 7)Alcancei minha nova posição?	Busca por ameaça, Solicita apoio de fogo, Calcula posição do inimigo, engaja inimigo, Recua

Tabela 2: Tabela mostrando *sensors* e *effectors* para ambos os tipos de modelo.

Agora que definimos nossos *sensors* e *effectors* vamos agrupá-los em estados, ações e transições para construir nossas máquinas de estado finito.

3.1. MÁQUINA DE ESTADO DE DEFESA DE POSIÇÃO

3.1.1. Estado *idle*:

Esse é o estado *default* da unidade. Ele agrupa todos os *sensors* com respostas negativas. Nenhuma ameaça a vista, nenhum alvo novo, não foram recebidos fogos de artilharia, a unidade não está sendo atacada, as baixas são menores que 50%, não são aguardados reforços. Quando nesse estado a unidade mantém sua posição e possui apenas um *effector*: Buscar por alguma ameaça no campo visual (Posso ver algum inimigo?). As transições aqui são executadas quando algum dos *sensors* muda de resposta.

3.1.2. Ameaça plotada:

Este estado agrupa o *sensor* do campo visual (Posso ver algum inimigo?) e o *sensor* de novo alvo (É um novo alvo?), ambos com respostas positivas. O *effector* aqui é solicitar apoio

de fogo sobre o alvo. Quando um inimigo está dentro do alcance visual da unidade a unidade verifica se a posição do inimigo está ou não presente na lista de alvos como alvo já atacado. Se não estiver presente a unidade insere a posição como alvo não atacado. Se já estiver presente como alvo já atacado, a unidade muda seu status para não atacado, solicitando uma nova missão de tiro sobre o alvo. Porém se estiver listado como alvo não atacado a unidade ignora o inimigo (isso significa que o posto de comando conhece o inimigo mas ainda não foi capaz de solicitar uma missão de tiro sobre ele).

Quando o posto de comando solicita uma missão de tiro sobre um algo ele altera seu status na lista de alvos para alvo atacado. Isso faz-se necessário para evitar múltiplos pedidos de apoio de fogo sobre o mesmo alvo. Depois de inserir o inimigo na lista ou de ignorá-lo, as respostas de ambos os *sensors* mudam para negativas e uma transição é feita para o estado *idle*. Isso mantém a unidade checando quando as missões de tiro foram cumpridas.

3.1.3. Sob fogos de artilharia inimigos:

O único *sensor* neste estado é o *sensor* de recebimento de fogos de artilharia (Estou sobre fogos de artilharia?). Quando sob fogo de artilharia inimigo, a unidade executa seus *effectors*: calcular a direção e posição de onde os fogos vieram e solicitar fogo sobre o inimigo.

3.1.4. Atacado:

O estado atacado possui três *sensors* com respostas afirmativas: *sensor* que identifica que a unidade está sendo atacada (Estou sobre fogo direto?), *sensor* de possibilidade de engajamento (Posso engajar o inimigo?) e *sensor* de baixas (Perdi metade do meu efetivo?) com resposta negativa. Um agente entra nesse estado quando recebe ataque direto (tiro tenso) dos inimigos. O agente executa o *effector* engajar inimigo para contra atacar e permanece neste estado até que o engajamento não seja mais possível ou que as baixas cheguem a 50% do efetivo. No primeiro caso, uma transição é feita para o estado *idle*, no Segundo caso o *effector* de solicitação de reforços e executado e uma transição é feita para o estado aguardando reforços.

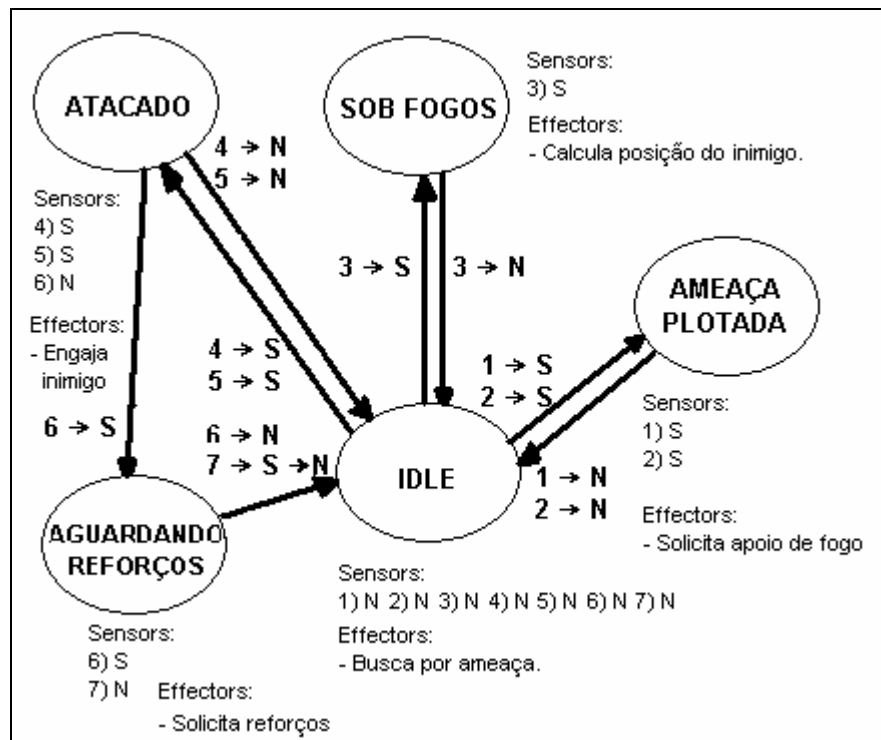


Figura 4: Máquina de estados com *sensors* e *effectors* para comportamento de defesa de posição.

3.1.5. Aguardando reforços:

Esse estado agrupa dois *sensors*: o de baixas (Perdi metade do meu efetivo?) com resposta afirmativa e o *sensor* de espera de reforços (Chegaram os reforços?) com resposta negativa. O agente permanece nesse estado até que seu efetivo seja *recompletado*. Então ele altera apenas a primeira resposta e retorna para o estado *idle*. Mesmo os reforços tendo chegado a segunda resposta permanece negativa porque uma vez que os reforços chegaram a unidade não está mais aguardando por reforços.

3.2. MÁQUINA DE ESTADOS DE RETARDO INIMIGO

Retardar o inimigo é uma estratégia usada quando a força de defesa atual é pequena e uma grande força de defesa está sendo preparada para combater o inimigo em breve. O objetivo então é retardar o inimigo proporcionando tempo para a chegada da força de defesa maior. Neste comportamento as unidades podem ocasionalmente fazer uma retirada para a próxima posição de defesa planejada

Todos os estados neste comportamento são os mesmos do comportamento anterior exceto pelo estado aguardando reforços (Chegaram os reforços?) que deve ser substituído pelo estado retraindo (Alcancei minha nova posição?).

3.2.1. Retraindo:

Agrupa dois *sensors*: *sensor* de baixas (Chegaram os reforços?) e o *sensor* de nova posição alcançada (Alcancei minha nova posição?) o primeiro com resposta positiva e o segundo com resposta negativa. Quando as baixas atingem 50% do efetivo da unidade esta recua para a próxima posição no terreno. Chegando a essa nova posição a unidade troca apenas a primeira resposta dos *sensors* para negativo e faz uma transição para o estado *idle*. Novamente a posição foi atingida porém, se a unidade atingiu a nova posição não precisa mais recuar, então a resposta para o Segundo *sensor* permanece negativa.

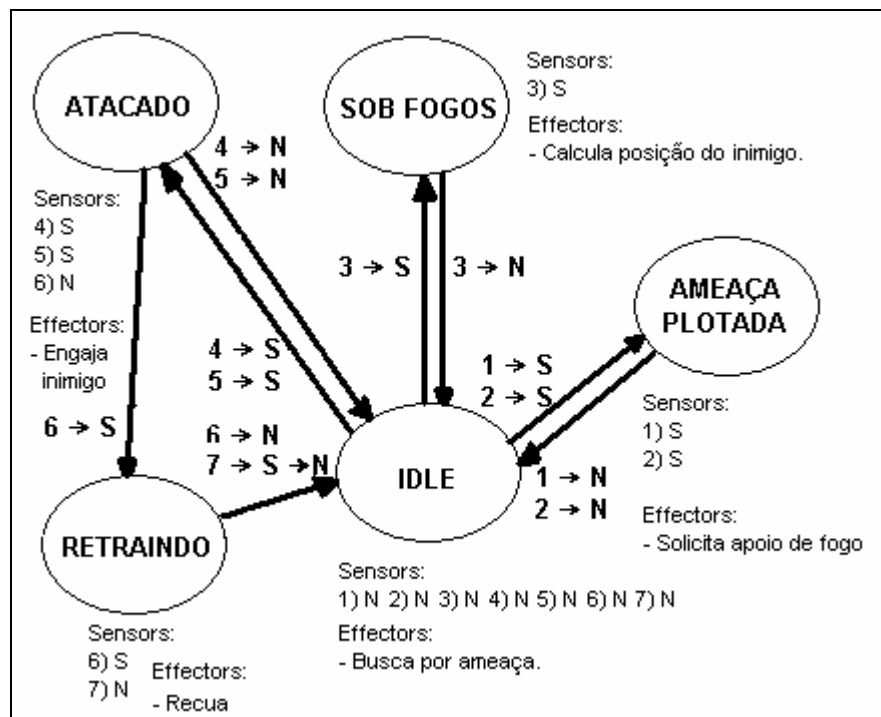


Figura 5 Máquina de estados com sensors e effectors para o comportamento de retardo inimigo.

4. OUTROS RECURSOS DO SISTEMA DE DEFESA AUTOMÁTICA

4.1. LOGÍSTICA

Quando defendendo uma área com o objetivo de retardar o inimigo o terreno é dividido em 3 linhas paralelas a costa. A força de defesa esconde provisões ao longo das duas primeiras linhas e toma posição na última linha (mais próxima da praia). Quando a força não pode mais manter sua linha de defesa ela recua para a linha anterior re-completando suas unidades com as provisões escondidas.

Neste trabalho, assumimos que o terreno é bem conhecido pela força de defesa e que provisões como munição, armamento, comida e água estão escondidas nas linhas de defesa. Quando faz-se necessário recuar as unidades são re-completadas de seus valores iniciais de provisões ao alcançarem a nova posição.

4.2. RECONHECIMENTO AÉREO PERIÓDICO

Se a força de defesa possui alguma aeronave sem missão de vôo solicitada, o posto de comando pode ocasionalmente solicitar uma missão de reconhecimento aéreo que consiste em um vôo padronizado sobre uma área de modo a plotar qualquer atividade inimiga. Se um inimigo é avistado durante o vôo a aeronave inclui sua posição na lista de alvos.

5. TRABALHOS CORRELATOS

Não há muita informação disponível sobre as técnicas usadas para modelar inteligência artificial para unidades militares. Porém, descobrimos que quase todas as forças armadas dos Estados Unidos da América utilizam uma arquitetura chamada SOAR. Nesta sessão nós vamos apresentar brevemente a arquitetura SOAR e descrever as principais diferenças entre o SOAR e as técnicas descritas neste trabalho.

5.1. SOAR

O Soar é uma arquitetura para desenvolvimento de sistemas que exibem comportamento inteligente. Os Agentes construídos no SOAR são capazes de aprender e tomar decisões a partir de combinações de conhecimento adquirido em tempo de execução. No SOAR, cada decisão é baseada na interpretação de dados sensoriais, e nos conhecimentos adquiridos durante a resolução de problemas anteriores.

A principal diferença entre o SOAR e as técnicas descritas neste trabalho é o tipo de agente que está sendo construído. O SOAR constrói agentes inteligentes, capazes de adquirir conhecimento e aplicar esse conhecimento para tomar decisões. Os agentes construídos com as técnicas descritas neste trabalho não são agentes inteligentes. São agentes baseados em comportamento pré-definido. Isso significa que os agentes não aprendem e todas as suas decisões seguem um comportamento bem conhecido.

Por que não usar agentes inteligentes então? Bem, tomamos esta decisão devido ao fato de estarmos usando agentes para substituir humanos que seguem um plano de defesa pré-definido. Caso um agente realize uma ação que não está descrita no plano, todo o exercício pode ser comprometido. Toda ação deve ser facilmente identificável no plano de defesa.

Caso você não necessite de comportamento pré-definido ou este comportamento não esteja descrito em algum tipo de manual ou ainda não é um comportamento bem conhecido, então nós aconselhamos a você tentar usar agentes inteligentes e nesse caso o Soar seria uma boa alternativa.

6. CONCLUSÃO

Agentes proporcionam um boa maneira de modelar unidades militares autônomas e o comportamento pré-definido é, neste caso o tipo de comportamento mais fácil de modelar uma vez que todo o esforço intelectual necessário que faz com que os programadores evitem este tipo de comportamento já está pronto e documentado nos manuais de estratégia militar.

Pequenas regras pré-definidas podem levar a comportamentos complexos e são mais fáceis de serem compreendidas.

Quando trabalhando com forças não convencionais o comportamento pré-definido talvez não seja a melhor opção uma vez que tais forças podem agir de forma inesperada. Neste caso o uso de agentes inteligentes, com habilidade de adquirir conhecimento deve produzir bons resultados.

7. TRABALHOS FUTUROS

É bastante difícil posicionar estrategicamente unidades em um terreno sem conhecer suas particularidades. Até agora nossas unidades são posicionadas manualmente no terreno pelos instrutores de defesa do CAO-CFN. O primeiro passo para desenvolver uma ferramenta para posicionar unidades sobre qualquer tipo de terreno é desenvolver um método de identificação de suas características. Pretende-se desenvolver um novo módulo que permita as unidades de defesa posicionarem-se estrategicamente em qualquer terreno de forma automática.

Além disso, estamos estudando a possibilidade de inserir comportamento não determinístico para modelar unidades de defesa que atuam de forma diferente das forças convencionais, como guerrilha por exemplo.

8. REFERÊNCIAS BIBLIOGRÁFICAS

- [1] FUNGE, John David. AI for Games and Animation: A Cognitive Modeling Approach, 1999, A K Peters.
- [2] Edited by Mark DeLoura, Game Programming Gems, 2000, Charles River Media.
- [3] Ioerger, T. R.; Volz, R. A.; Yen, J.; 2000, "Modeling Cooperative, Reactive Behaviors on the Battlefield with Intelligent Agents". In: Proceedings of the ninth Conference on Computer Generated Forces (9th CGF), pp. 13-23, Orlando, Florida, USA.
- [4] Smith, R. D.; 1998 "Essential Techniques for Military Modeling & Simulation". In: Proceedings of the 1998 Winter Simulation Conference, pp. 805-812, Orlando, Florida, USA.
- [5] Roth, J. T. Ph.D., CPE, 1997. Advances in Using Simulation to Enhance Military Training. In: IEEE Sixth Annual Human Factors Meeting, Orlando, Florida, USA.
- [6] Woodaman, R.F.A., 2000, Agent-Based Simulation of Military Other Than War Small Unit Combat. Tese de M.Sc., Naval PostGraduate School, Monterey, California, USA.
- [7] Zuccaro, CF (FN) P. M.; "O Batalhão de Comando e Controle: o que é e para que serve". Revista O Anfíbio, 2001.
- [8] Silva, C. A.; "[Modelagem Comportamental para Agentes em Ambientes Reais](#)"; Master Thesis, University of the State of Rio de Janeiro; 2003.
- [9] Savelli, R.M.; Lyrio, G.H.S.O; Seixas, R.B.; [The Maneuver and Attrition Warfare Simulation System](#), *Simpósio de Pesquisa Operacional e Logística da Marinha - SPOLM 2004*.

- [10] Silva, C.A.; Seixas, R.B.; [Arquitetura de um Modelo de Confronto Baseado em Agentes Autônomos e Dados Geográficos](#), *Simpósio de Pesquisa Operacional e Logística da Marinha - SPOLM 2004*.
- [11] Silva, C.A.; Seixas, R.B.; [Integração de Agentes Autônomos e SIG em uma Arquitetura para Simulação de Confrontos](#), *pre-print submitted to Simpósio de Desenvolvimento e Manutenção de Software da Marinha - SDMS 2003*.
- [12] SOAR project at <http://sitemaker.umich.edu/soar>. (Acessado em 30/06/2006)