

## 4 Cenários de Uso

Neste capítulo será apresentada a aplicabilidade do *framework* JSAN. Esta será feita através de dois cenários de uso. Primeiro será aplicado no contexto de prevenções de crimes (BOSSE e GERRITSEN, 2010) e o segundo está relacionado a missões de resgate de civis que estão em áreas de risco.

### 4.1. Suporte na Prevenção de Crimes

Criminologia Ambiental é o estudo do crime, da criminalidade e da relação entre vítimas e infratores, e como eles se relacionam (BRANTINGHAM, 1991). Um importante tópico dentro de Criminologia Ambiental é a análise do deslocamento de crimes. Alguns dos muitos desafios nesta área são a predição e prevenção com relação ao surgimento de pontos com alto índice de criminalidade, tornando o estudo do deslocamento de crimes uma das mais promissoras áreas de pesquisa (COHEN e FELSON, 1979), (HERBERT, 1982), (SHERMAN, GARTIN e BUERGER, 1989), (BOSSE e GERRITSEN, 2010). Certos tipos de crimes acontecem tipicamente em torno de locais específicos numa cidade, isto é, ruas próximas a shoppings, estações de trens e alto estradas. Tais localidades são conhecidas como pontos com alta criminalidade. Entretanto, esses pontos não persistem por um longo período de tempo. Um número de fatores conhecidos pode causar o deslocamento desses locais de criminalidades para outra região. Por exemplo, a implantação de câmeras em estações de trem pode fazer com que o índice de criminosos diminua naquela localidade (COHEN e FELSON, 1979), (HERBERT, 1982), (SHERMAN, GARTIN e BUERGER, 1989), (BOSSE e GERRITSEN, 2010).

A dinâmica natural desses pontos com alta taxa de criminalidade faz deles um tópico popular de pesquisas científicas. Por exemplo, questões típicas que são estudadas em um Ambiente Criminológico: aonde e quando esses pontos com alto índice de criminosos irão surgir? Quanto tempo vai durar? E o que nós podemos

fazer para prevenir? A abordagem clássica para investigar esses tipos de questões é uma extensa coleta de dados empíricos durante vários anos, e o uso de análises técnicas para identificar tendências nestes dados. Entretanto, uma desvantagem desta abordagem é que ela foca nos padrões de deslocamentos de crimes antigos, que não garantem que os padrões futuros serão similares.

Como alternativa, ao uso de padrões de deslocamentos antigos, criminologistas têm unido forças com pesquisadores da Ciência da Computação e Inteligência Artificial para explorar os benefícios de simulações sociais (BOSSE e GERRITSEN, 2010), com agentes, para investigar o deslocamento de crimes. Assim, a perspectiva desta abordagem é usar os ambientes de simulação para prever a dinâmica do deslocamento de crimes no futuro, melhor que analisar as dinâmicas do passado.

Diante do apresentado, Bosse e Gerritsen, (BOSSE e GERRITSEN, 2010) notaram a necessidade de lidar com tais fatores. Os autores apresentam uma simulação para dar suporte a prevenção de crimes analisando o deslocamento de crimes numa determinada região. Nesta simulação eles exploraram diferentes estratégias para o movimento dos guardas, os quais têm como objetivo proteger os civis e evitar que assaltos sejam cometidos pelos criminosos.

A inserção de normas no Suporte a Prevenção de Crimes tem como objetivo regular as ações dos agentes guardas. Por exemplo, tais normas podem influenciar os agentes guardas a agirem de forma a evitar o maior número possível de assaltos, mas, prezando por sua segurança. Ou seja, se uma norma que regula o agente guarda caso ele deseje ir para uma área com muitos bandidos e poucos guardas. Esta norma pode descrever que para o agente efetuar a prisão é necessário ele chamar reforço. Caso o agente cumpra com a norma, outros agentes são enviados a localidade e a prisão é efetuada com sucesso e recebe a recompensa de obter maior velocidade no seu deslocamento. Já se o agente guarda violar com a norma e colocar sua vida em risco, sua velocidade de deslocamento é reduzida e com isso perde seu poder de efetuar prisões.

Neste contexto, utilizando como cenário de uso, a simulação de Bosse e Gerritsen (BOSSE e GERRITSEN, 2010) de Suporte a Prevenção de Crimes, foi implementada uma extensão desta simulação com a inserção de normas, com o objetivo de regular a autonomia dos agentes guardas. Para isto, utilizou-se uma implementação do *framework* JSAN que fornece os mecanismos necessários para

criar as normas no ambiente de simulação e provê a possibilidade de estender diversas estratégias para os agentes guardas lidarem com normas, além de já oferecer algumas implementadas.

#### 4.1.1. Ideia Principal

O sistema multiagente estendido da simulação de Suporte a Prevenção de Crimes recebe informações sobre o número de agentes *criminosos*, número de agentes *guardas* e o número de agentes *civis*. A seguir, é assumido que o mundo (ou cidade) escolhido possa ser representado em termos de um número diferente de *localidades*. Isto é importante para saber a *densidade* de agentes que está presente em cada uma das localidades. Por exemplo, a *densidade* de criminosos numa determinada localidade é dada através do número de criminosos naquela localidade pelo total de criminosos na cidade. Além disso, para descrever o movimento dos diferentes tipos de agentes de uma local para outro, informações sobre a *reputação* (ou *atratividade*) de um local é necessária. Esta atratividade é diferente para cada tipo de agente. Por exemplo, civis gostam de locais onde estão a salvo, isto é, locais onde alguns guardas estão presentes e sem criminosos. Por outro lado, guardas são atraídos para lugares com um grande número de criminosos presentes, e criminosos gostam de locais onde existam muitos civis e nenhum policial. Finalmente, para ser capaz de representar a ideia de áreas quentes, o *número de assaltos* por localidade é modelado. A ideia é que mais assaltos aconteçam em localidades as quais se tenham muitos criminosos e civis e poucos guardas.

A interação entre os conceitos apresentados acima são mostrados na Figura 4.1. Esta figura retrata a influência entre os diferentes grupos em uma localidade. Aonde os círculos denotam os conceitos que são mencionados anteriormente em itálico e as setas indicam a influência entre os conceitos (influências na atratividade foram utilizadas setas pontilhadas para uma melhor visualização).

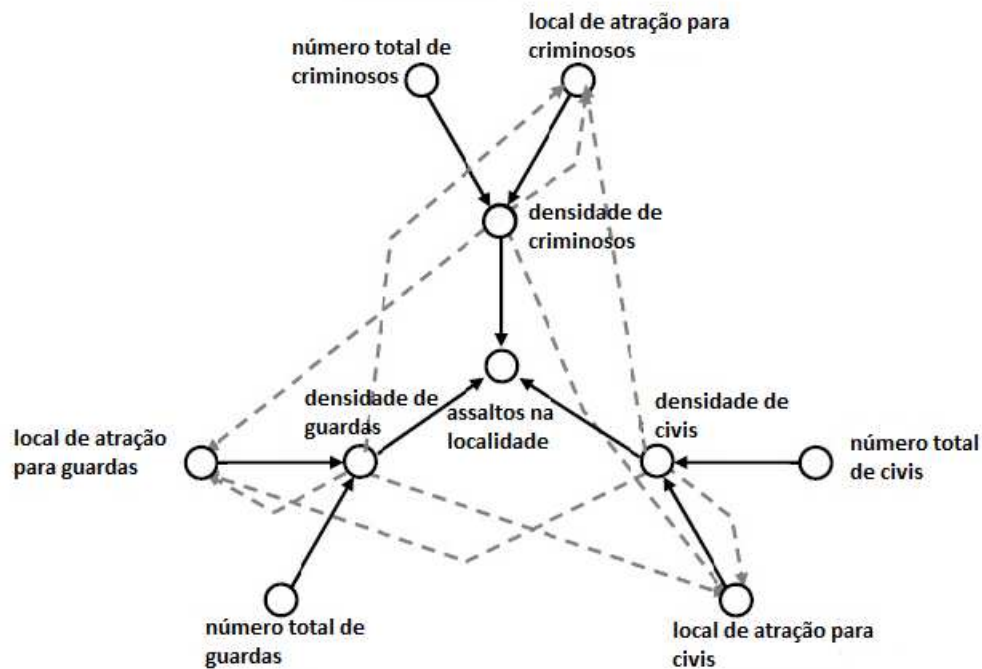


Figura 4.1 - Interação entre criminosos, guardas e civis (BOSSE e GERRITSEN, 2010)

Quando as normas são inseridas na simulação, um agente Gerenciador é responsável por recebê-las e enviá-las para o grupo de agentes para o qual a norma é endereçada. Estes agentes, diante da norma serão capazes de escolher por cumpri-la ou viola-la, dependendo da forma como suas estratégias para lidar com as normas foram implementadas. Nesta simulação as normas serão endereçadas apenas aos agentes guardas, os quais utilizam a estratégia *Selfish* para lidar com normas (Ver Seção 3.2.6). Por exemplo, uma norma gerada para esta simulação através da classe *RandomGenerateNorms* apresentada na Figura 4.5 foi: “não colocar a vida de civis em risco” com a seguinte estrutura: (i) o conceito deôntico da norma é uma obrigação, (ii) uma recompensa caso a norma seja cumprida é o agente obter maior velocidade no seu deslocamento e uma punição caso ele viole com a norma é a redução da sua velocidade e com isso perda do poder de efetuar prisões, (iii) a condição da norma estar ativa é ter agentes civis sendo assaltados e a desativação é quando a localidade está segura para os agentes civis, e (v) o elemento que a norma irá regular é a ação de prender criminosos (Ver Seção 3.2.4). Quando o agente guarda receber a norma acima, sendo ele *Selfish*, sua estratégia será comparar essa norma com seus objetivos individuais e ver que caso

violar a norma sofrerá sanções que podem afetar seus objetivos individuais. Já se cumprir com a norma, receberá recompensas alcançando seus objetivos individuais. Após comparar quanto ele irá ganhar por cumprir ou violar com a norma, o agente guarda toma sua decisão de acordo com seus interesses.

Os agentes guardas recebem informações de diversas áreas da cidade e eles visam um melhor resultado no combate ao crime utilizando de diferentes estratégias de movimento. Neste caso, os agentes guardas escolhem seu movimento de acordo com a situação do ambiente (Ver Seção 3.2.2). Por exemplo, se estiver acontecendo muitos assaltos em uma determinada localidade, eles agirão de forma reativa (Ver classe *Reactive* Seção 3.2.2), locomovendo-se para a localidade depois dos assaltos terem acontecido (BOSSE e GERRITSEN, 2010). Já se ocorrer um evento onde um número grande de civis estiver reunido, agirão de forma antecipada (Ver classe *Anticipatory* Seção 3.2.2), isto é, irão se locomover para esta localidade antes dos acontecer os assaltos, procurando antecipar ao deslocamento dos criminosos e deixar as pessoas em segurança nessa localidade (BOSSE e GERRITSEN, 2010). Resta ainda a possibilidade de utilizar de uma movimentação híbrida, entre reativa e antecipatória, visando um melhor resultado no combate ao crime.

Para visualizar o resultado da simulação (i) cada agente possui um identificador se é criminoso, guarda ou civil, (ii) a cidade é dividida em localidades distintas para conseguir saber onde cada agente se encontra e (iii) a taxa de assaltos é calculada a cada unidade de tempo da simulação.

A fim de lidar com os problemas supracitados e alcançar um bem estar social desejável cada agente possui: (i) um conjunto de objetivos que está ligado diretamente a sua satisfação individual, (ii) uma base de informações coletadas no ambiente de simulação para auxiliar na captura dos criminosos e (iii) uma base de estratégias utilizadas para lidar com normas (Ver Seção 3.2).

#### **4.1.2. Agentes Guardas**

A fim de adotar a estratégia *Selfish* de lidar com normas para os agentes guardas utilizou-se a classe *Selfish* (Ver classe *Selfish* na Figura 4.2), uma das extensões da classe *NormStrategy*. Na classe *Selfish* disponibilizada pelo *framework* JSAN foi implementado o método *calculate*, a qual recebe o

identificador dos agentes como parâmetro. Este método faz com que os agentes analisem a situação aonde o cumprimento de normas pode contribuir para alcançar com seus objetivos individuais (Ver Seção 3.2.6).

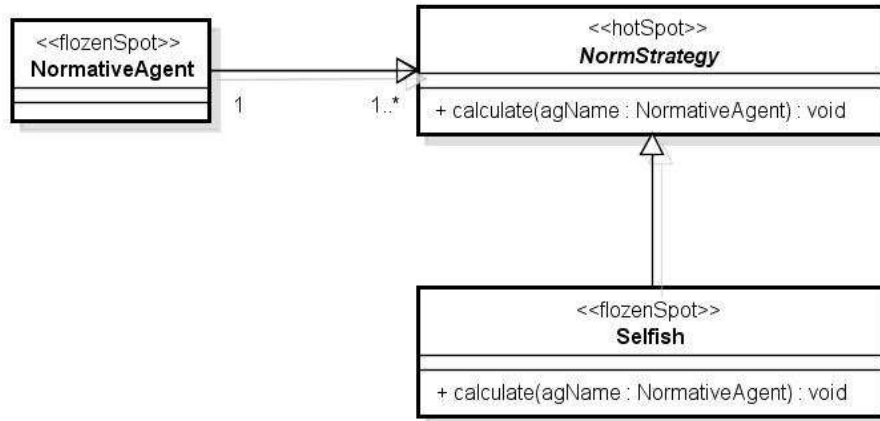


Figura 4.2 - Estratégia para lidar com normas

A classe *EnvironmentSimulation* é responsável por gerenciar um conjunto de estratégias para criação dos objetivos individuais dos agentes, estes representados pela classe abstrata *GenerateGoalStrategy*.

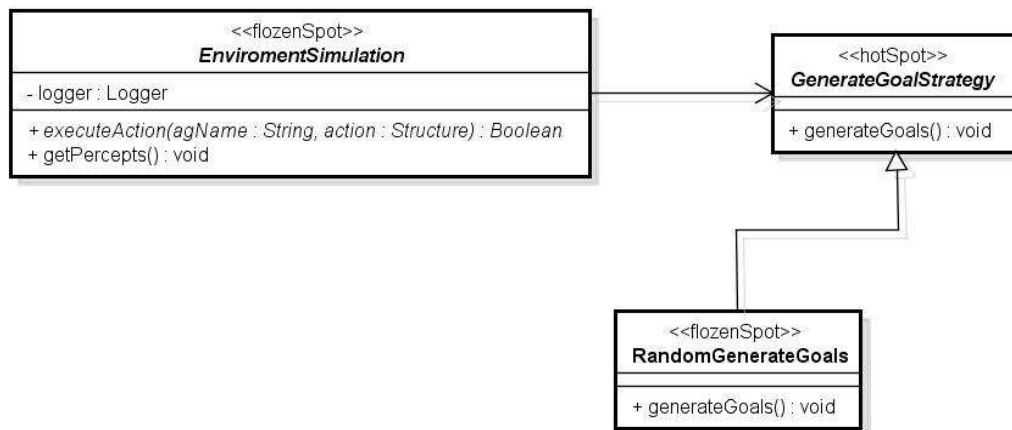


Figura 4.3 - Gerar goals dos Agentes

Utilizou-se a classe *RandomGenerateGoals* (Ver Figura 4.3) que através do método *generateGoals* e um conjunto de objetivos previamente conhecido foi capaz de escolher determinados objetivos aleatoriamente para os agentes guardas. Os objetivos gerados para a simulação foram: (i) aumentar a velocidade de

movimento para prender criminosos, (ii) não colocar a vida de civis em risco, (iii) ganhar medalha de serviços prestados a sociedade.

Na atividade *collect* (Ver Seção 3.2.5) os monitores para interceptação de mensagens (Ver método *sensor* da classe *MessageMonitor* na Figura 3.6) e informações do ambiente (Ver método *sensor* da classe *EnvironmentMonitor* na Figura 3.6) ambos disponibilizados pelo JSAN foram utilizados. O parâmetro *NormAgent* é importante para poder atribuir quais agentes receberão as informações coletadas no ambiente, neste caso os agentes guardas.

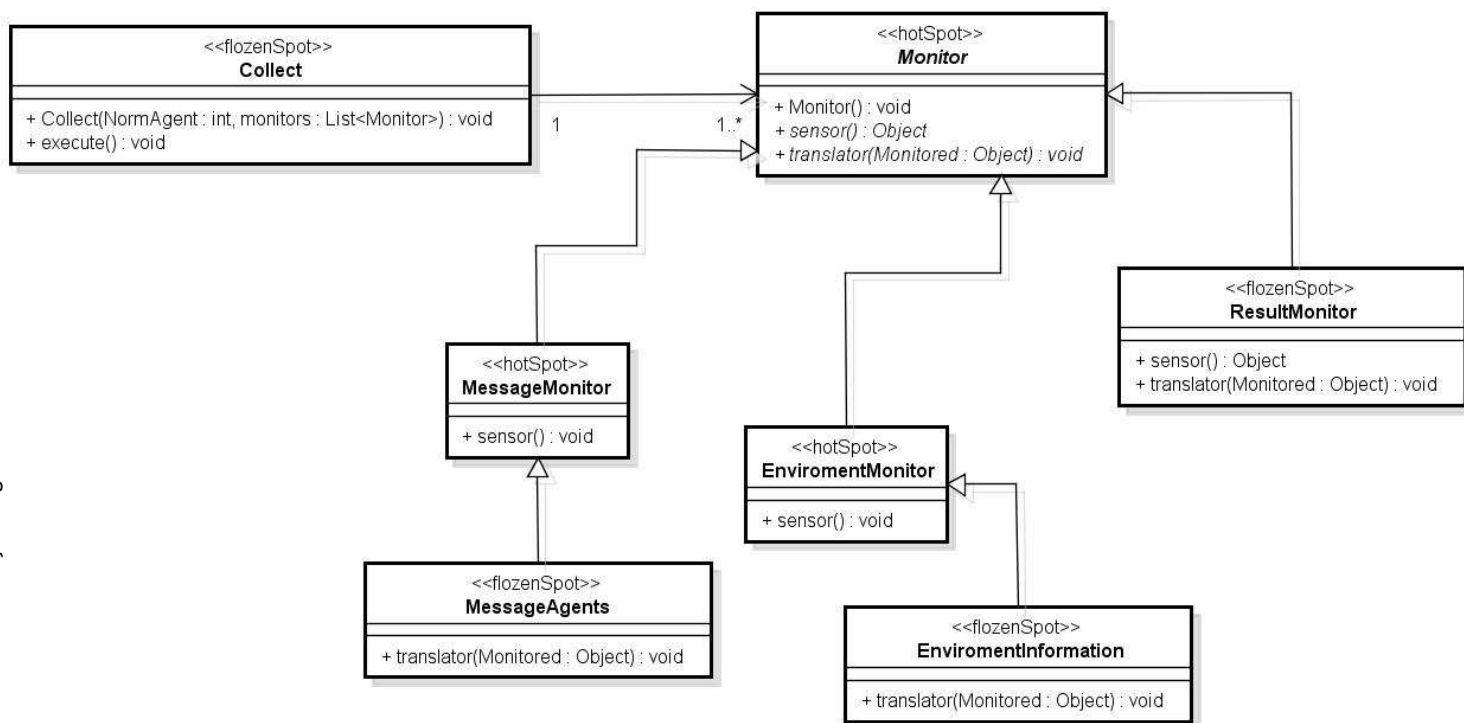


Figura 4.4 - Atividade *Collect*

O primeiro irá verificar se existem novas mensagens enviadas pelo agente Gerenciador e o segundo irá verificar se existem outras informações do ambiente. A classe *ResultMonitor* (Ver Figura 4.4) que estende a classe *Monitor* (Ver Figura 3.6) e implementa seus métodos abstratos *sensor* e *translator* foi criada. No método *sensor* foi definida uma implementação que verifica a importância de uma mensagem dado o seu tempo de duração na simulação. Já o método *translator* formata o resultado de *sensor* de forma que o agente consiga entender as mensagens.

Por fim, duas classes, *MessageAgents* e *EnvironmentInformation* (Ver Figura 4.4) foram criadas como extensão das classes *MessageMonitor* e *EnvironmentMonitor*, respectivamente, a fim de implementar o método abstrato *translator* para formatar os dados: (i) enviados pelo agente Gerenciador a partir do método *translator* da classe *MessageAgents*, e (ii) coletados a partir do monitoramento do ambiente, ou seja, informações disponibilizadas durante a simulação, por exemplo a taxa de assaltos de uma determinada localidade da cidade, a partir do método *translator* da classe *EnvironmentInformation*.

No diagrama de classes apresentado na Figura 4.5, temos a classe *EnvironmentSimulation*, responsável por gerenciar um conjunto de estratégias para geração de normas representadas pela classe abstrata *GenerateNormsStrategy*, a qual possui o método abstrato *generateNorms*.

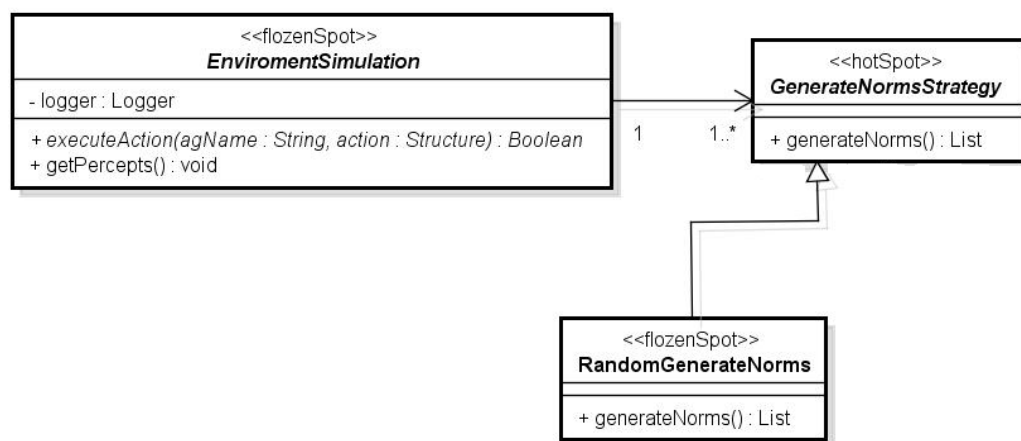


Figura 4.5 - Gerar Normas dos Agentes

Este método irá gerar normas para a simulação dado um conjunto de normas previamente conhecido, através da classe *RandomGenerateNorms* disponibilizada pelo *framework* JSAN. Um exemplo das normas geradas foi discutido na Seção 4.1.1.

No Suporte a Prevenção de Crimes diferentes estratégias com relação ao movimento dos guardas foi implementada (Ver seção 3.3.2) através de extensões da classe *MovementStrategy*, um *hotSpot* disponibilizado pelo JSAN.



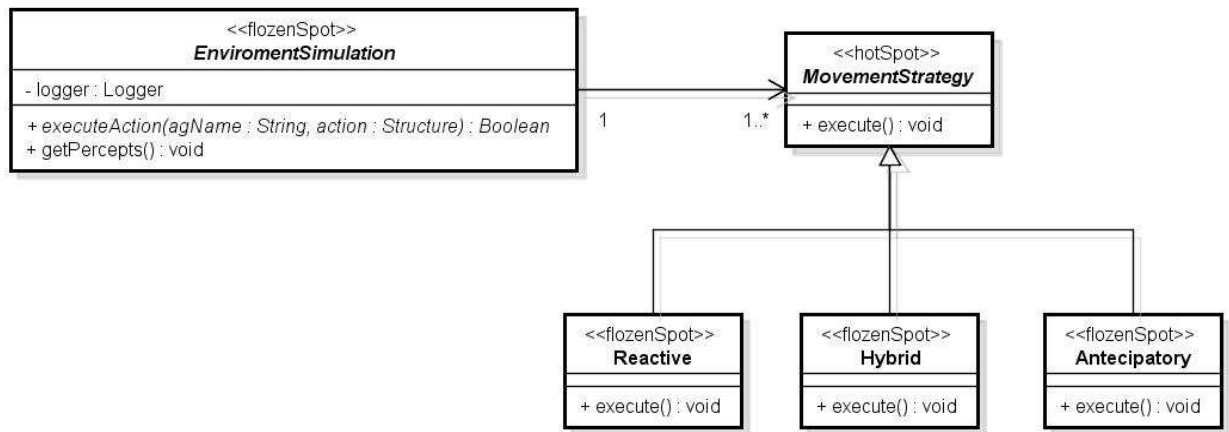


Figura 4.6 - Estratégias de Movimento dos Agentes

Foram utilizadas as classes *Reactive* e *Antecipatory* a fim de implementar o método *execute* para definir como a estratégia será utilizada. A estratégia reativa, implementada na classe *Reactive* define que a quantidade de guardas que se move para uma nova localização é proporcional à percentagem de assaltos acontecidos recentemente naquele local. Já o método *execute* da classe *Antecipatory* define que a quantidade de guardas que se move para uma nova localidade é proporcional a densidade de civis que se espera ter no futuro passando por ali. Por último, uma nova classe foi criada, a classe *Hybrid*, na qual o método *execute* define que o número de guardas que movem para uma nova localização é a média da soma dos guardas determinados pelas duas estratégias.

Na classe *Norm* (ver Figura 4.7) foram definidos atributos para que se possam criar as sanções, condições de ativação e desativação, o elemento que será regulado pela norma e qual o conceito deontico que está associado a ela (Ver Seção 3.2). Dado a norma apresentada na Seção 4.1.1: “Não colocar a vida de civis em risco”, foi associada a ela a recompensa do agente obter maior velocidade no seu deslocamento e uma punição caso ele viole com a norma é a redução da sua velocidade, assim perdendo poder de efetuar prisões.

Já nos atributos *activation* e *deactivation* da classe *Norm*, foi implementado as condições de ativação e desativação da norma: a condição da norma estar ativa é ter agentes civis sendo assaltados e a desativação é quando a localidade está segura para os agentes civis. O atributo *elementRegulated* recebeu o elemento que a norma irá regular, que é a ação de prender criminosos. Já o atributo *deonticConcept* ficou com o valor de obrigação.

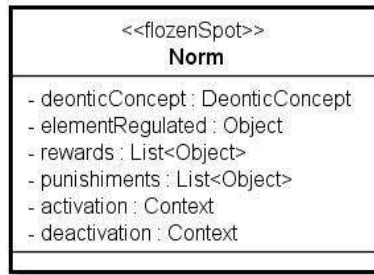


Figura 4.7- Normas

A classe *EnvironmentSimulation* também é responsável por gerenciar um conjunto de estratégias para visualização das informações sobre a simulação. A classe *ResultGraficReport* (Ver Figura 4.8) que estende *ReportyStrategy* (Ver Seção 3.2.3) e implementa seus métodos abstratos *view* e *drawAgent* foi criada. No método *view* foram definidos diversos gráficos para mostrar o desenvolvimento da simulação duração seu tempo de duração (Ver Figuras de 4.8 até 4.11). Já no método *drawAgent* foi implementado para saber qual é a posição dos agentes no ambiente de simulação.

PUC-Rio - Certificação Digital Nº 1012638/CA

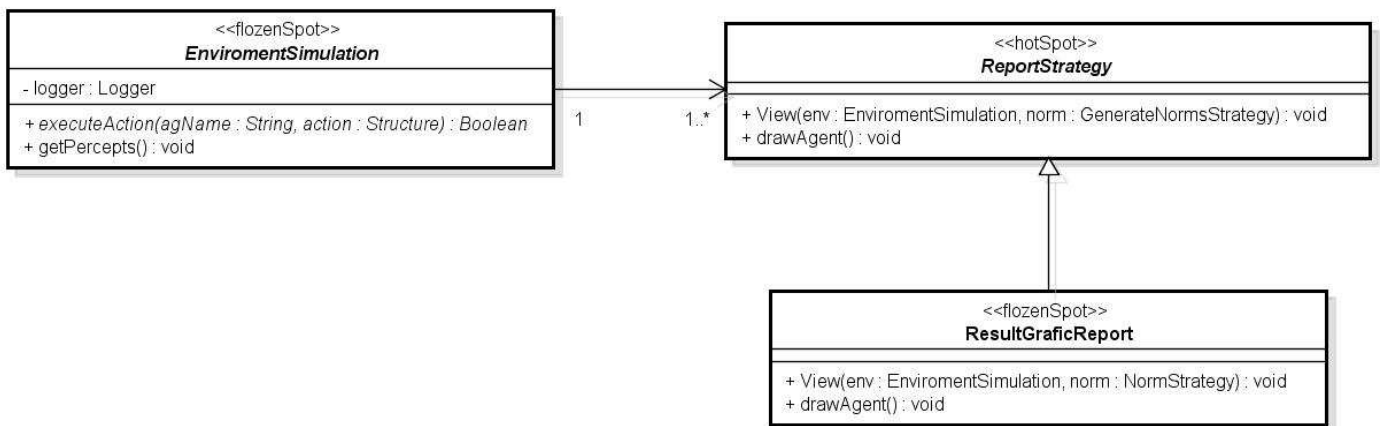


Figura 4.8 – Geração de Relatórios

Para mostrar os resultados da visualização gerada através da extensão de *ReportStrategy* foram realizadas simulações em um dado cenário, escolhido entre os cinco apresentados abaixo:

Cenário	Fase 1	Fase 2	Fase 3	Fase 4	Fase 5
1	L1=0.25 L2=0.25 L3=0.25 L4=0.25	L1=0.7 L2=0.1 L3=0.1 L4=0.1	---	---	---
2	L1=0.25 L2=0.25 L3=0.25 L4=0.25	L1=0.7 L2=0.1 L3=0.1 L4=0.1	L1=0.25 L2=0.25 L3=0.25 L4=0.25	---	---
3	L1=0.25 L2=0.25 L3=0.25 L4=0.25	L1=0.7 L2=0.1 L3=0.1 L4=0.1	L1=0.4 L2=0.4 L3=0.1 L4=0.1	L1=0.1 L2=0.7 L3=0.1 L4=0.1	---
4	L1=0.25 L2=0.25 L3=0.25 L4=0.25	L1=0.7 L2=0.1 L3=0.1 L4=0.1	L1=0.4 L2=0.4 L3=0.1 L4=0.1	L1=0.3 L2=0.3 L3=0.3 L4=0.1	L1=0.25 L2=0.25 L3=0.25 L4=0.25
5	L1=0.25 L2=0.25 L3=0.25 L4=0.25	L1=0.7 L2=0.1 L3=0.1 L4=0.1	L1=0.4 L2=0.4 L3=0.1 L4=0.1	L1=0.1 L2=0.4 L3=0.4 L4=0.1	L1=0.1 L2=0.1 L3=0.4 L4=0.4

Figura 4.9 - Tabela de Cenários

Na simulação descrita neste documento, escolheu-se um cenário entre os apresentados acima. Cada um dos cenários envolve quatro localidades (chamadas L1, L2, L3 e L4). Para reforçar o surgimento das áreas de criminalidade, em cada cenário a atratividade básica dos locais para os civis sofrem mudanças ao longo do tempo, resultando em diferentes fases. Na tabela de cenários (ver Figura 4.9), cada célula indica os valores de atratividade dos diferentes locais.

Escolheu-se como exemplo o cenário de número 2 (dois), todos os locais começam com a mesma atração base para os civis (isto é,  $L1 = L2 = L3 = L4 = 0,25$ ) e a estratégia de movimento mostrada nos gráficos abaixo é a antecipatória. Depois de um tempo, na fase dois, que inicia no instante de tempo 39, a capacidade de atração base da localidade L1 é aumentada ( $L1 = 0,7$  e  $L2 = L3 = L4 = 0,25$ ). Isto pode ser causado, por exemplo, porque um circo está chegando à cidade. Algum tempo depois, na fase três, o circo vai embora para outra cidade, no tempo 90 e a atração base para todas as localidades tornam-se iguais novamente (0,25). Os outros parâmetros foram escolhidos como se segue (para todos cenários). A população total é composta de 800 criminosos, 400 guardas e 4000 civis. Inicialmente, estes agentes são distribuídos igualmente ao longo dos quatro locais (isto é, cada localização contém 200 criminosos, 100 guardas e 1000 civis). A velocidade dos agentes é de 0,5 e o tempo total de simulação é de 100 unidades de tempo.

Para ilustrar a construção da simulação de Suporte a Prevenção de Crimes utilizando o JSAN, segue abaixo os gráficos gerados pela implementação dos métodos *view* e *drawAgent* da classe *ResultGraficReport*, a qual é uma extensão da classe *ReportStrategy* (Ver Figura 4.8).

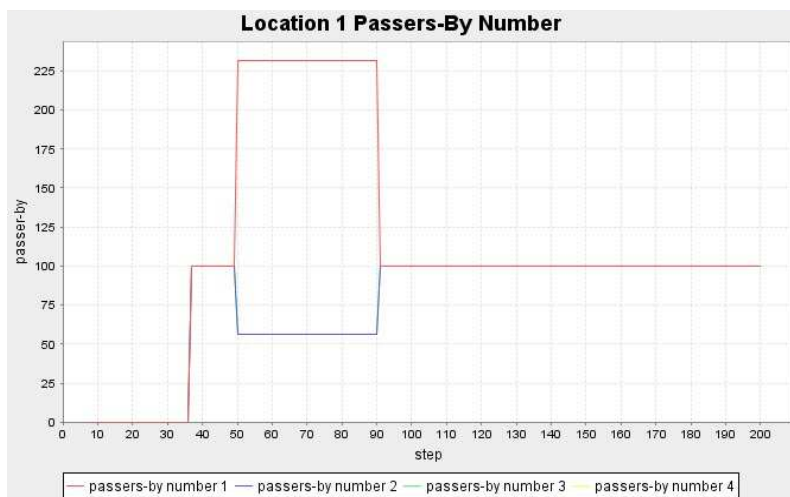


Figura 4.10 - Números de Civis

O gráfico da Figura 4.10 mostra o número de civis nas diferentes localidades. Sabendo que em todos os gráficos, a linha vermelha mostra os resultados na localidade L1, azul na L2, verde na localidade L3 e na localidade L4 é a linha amarela. No caso das Figuras 4.10, 4.11 e 4.12 as localidades L2, L3 e L4 estão na mesma posição da linha azul.

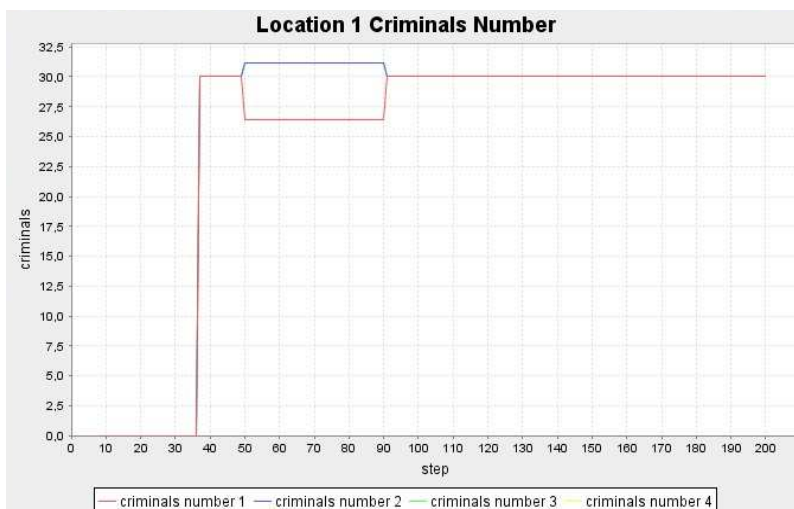


Figura 4.11 - Número de Criminosos

A Figura 4.11 mostra o número de criminosos que se deslocaram principalmente para as localidades L1, devido à chegada do circo na cidade, mas mostra a queda dos criminosos no período de tempo entre 39 e 90 devido aos agentes guardas que começaram a agir segundo as normas e passaram a se deslocar para essa localidade, como pode ser visto na Figura 4.12.

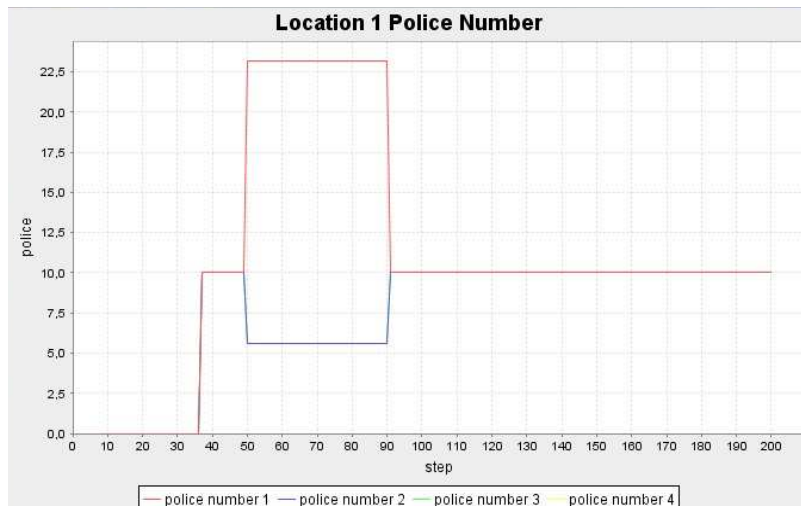


Figura 4.12 - Número de Policiais

Já na Figura 4.13 mostra a taxa de assaltos que aconteceu durante o tempo de simulação. A taxa de assalto aumenta praticamente em todas as localidades até o instante de tempo 39.

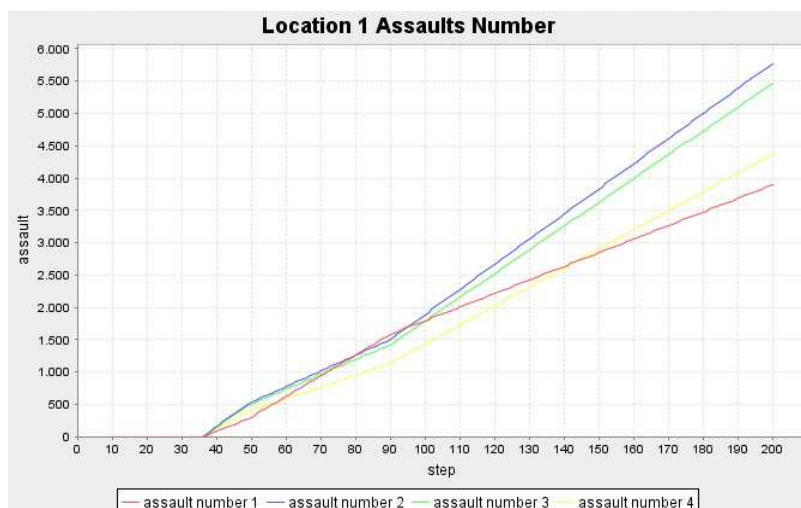


Figura 4.13 - Número de Assaltos

Depois que os guardas começam a se deslocar segundo as normas e evitar os assaltos notamos que ocorre uma diminuição na inclinação das curvas, notando-se principalmente na localidade L1, onde o maior número de civis se encontrava até o instante de tempo 90.

## 4.2. Evacuação de Civis em Áreas de Risco

Um plano de prevenção e emergência pode definir-se como a sistematização de um conjunto de normas e regras de procedimento, destinadas a evitar ou minimizar os efeitos das catástrofes que possam vir a ocorrer em determinadas áreas, gerindo de forma otimizada, os recursos disponíveis (LISBOA, 2005).

O número de óbitos e custos associados à perda de bens materiais e recuperação de áreas degradadas pelos processos de deslizamentos de massa continua bastante expressivo no Estado do Rio de Janeiro (AMARAL e FEIJO, 2004) (SANTOS NETO e LUCENA, 2010). Como exemplo, na região serrana foi registrado o maior desastre climático da história do país, envolvendo mais de 700 mortes. Custos associados a obras de contenção e recuperação das cidades somam mais de 500 milhões (DEFESA CIVIL, 2011). Tais condições estimularam o governo do estado a investir em pesquisa e desenvolvimento de sistemas de software capazes de analisar a probabilidade de ocorrência de deslizamentos de terra, assim como, realizar simulações a fim de planejar evacuações de civis em áreas de risco.

Neste contexto, surgiu o projeto GeoRisc (CERQUEIRA, SANTOS NETO, *et al.*, 2009) cujo objetivo é a construção de uma plataforma para auxiliar especialistas tanto na análise de áreas de risco, como também, no planejamento de evacuação de civis localizados nestas áreas. Tal planejamento é auxiliado por simulações, utilizando o *framework* JSAN, cujo objetivo é implementar cenários onde seja possível criar situações em que agentes civis estejam em risco e assim fornecer diferentes estratégias para os agentes bombeiros, regulados por normas, resgatar os agentes civis. Nesta simulação tem-se agentes civis, que podem estar com suas vidas em risco e agentes bombeiros, responsáveis por resgatar civis que se encontram em áreas de risco. Normas foram inseridas visando regular a ação dos agentes bombeiros, para que eles hajam com o objetivo de fazer o resgate de

civis de forma segura. Por exemplo, tais normas podem influenciar os agentes bombeiros a salvar uma pessoa após ter acontecido um deslizamento de terra, mas ele não tem os equipamentos de segurança necessário para efetuar o resgate de forma segura. Esta norma pode descrever que para o agente efetuar o resgate é necessário ele pedir ajuda aérea. Caso o agente cumpra com a norma, ele recebe a recompensa de ajuda, ou seja, outros agentes são enviados a localidade com um veículo aéreo. Já se o agente violar com a norma e colocar sua vida e de terceiros em risco, ele não recebe o apoio aéreo para fazer o resgate.

#### **4.2.1. Ideia Principal**

A simulação mencionada é um sistema multiagente normativo que recebe os dados contendo informações sobre uma área de risco, tais como condições do tempo, informação sobre a existência de agentes civis em áreas de risco, tipos de transporte para retirada de civis desses locais de risco, normas que os agentes bombeiros devem seguir durante a operação de salvamento, planos de resgates utilizados na simulação. Quando tais dados são fornecidos, um agente Gerenciador (Comandante dos Bombeiros) é responsável por recebê-los e enviá-los para os agentes bombeiros, que diante do tipo de dados fornecidos são capazes de encontrar diferentes soluções para efetuar a evacuação de civis das áreas de risco.

O grupo de bombeiros é dividido em três categorias que dizem respeito ao tipo de veículos que será utilizado: salvamento utilizando tropas, isto é, não precisa de veículos para evacuar uma área de risco, salvamento utilizando veículos terrestres e salvamento utilizando veículos aéreos.

Quando um agente bombeiro recebe o chamado para resgatar agentes civis numa determinada área de risco, ele tenta realizar esse salvamento de acordo com as normas que lhe foram endereçadas. Na simulação os bombeiros receberam a seguinte norma: “proteger a vida de civis em áreas de risco” com a seguinte estrutura: (i) seu conceito deontico é obrigatório, (ii) a recompensa caso a norma seja cumprida é o agente obter apoio aéreo ou apoio terrestre, caso seja violada o agente não receberá apoio terrestre para o resgate, (iii) a norma é ativada caso tenha alguma pessoa em risco e a norma é desativada quando todos civis estão a

salvo, e (v) o elemento que a norma regula é a ação de usar veículos aéreos (Ver Seção 3.2.4). O agente irá escolher qual estratégia irá utilizar para lidar com as normas. Caso ele seja um agente *Social* (LÓPEZ, LUCK e D'INVERNO, 2002), o qual sempre cumpre com as normas e não se preocupa com seus objetivos individuais ele irá cumprir com a norma. Entretanto, se o agente utilizar a estratégia *Rebellious* (LÓPEZ, LUCK e D'INVERNO, 2002), ele irá se preocupar em cumprir somente com seus objetivos individuais. Nesta simulação os agentes bombeiros utilizaram uma estratégia *Selfish* (LÓPEZ, LUCK e D'INVERNO, 2002), sempre irão verificar a situação aonde o cumprimento das normas irá contribuir para o alcance dos seus objetivos individuais, isto é, caso cumpra com a norma descrita acima, o envio de reforços será enviado para que o salvamento seja feito de forma rápida e segura.

A fim de lidar com os problemas supracitados e realizar as normas endereçadas a eles, os agentes bombeiros possuem: (i) um conjunto de objetivos que está ligado diretamente a sua satisfação individual, (ii) uma base de informações coletadas no ambiente de simulação para auxiliar na captura de informações sobre as localidades de risco e (iii) uma base de estratégias utilizadas para lidar com normas (Ver Seção 3.2).

#### **4.2.2. Agentes Bombeiros**

A fim de lidar com normas, utilizou-se a estratégia *Selsifh*, esta foi disponibilizada pelo *framework* JSAN. A estratégia *Selfish* foi implementada no método *calculate* da classe *Selfish* (Ver Seção 3.2.6) com o objetivo de analisar a situação aonde o cumprimento de normas irá ajudar o agente a cumprir com seus objetivos individuais, sem se esquecer dos objetivos sociais, pois o cumprimento de normas está diretamente ligado aos benefícios que ele irá receber por cumpri-la.



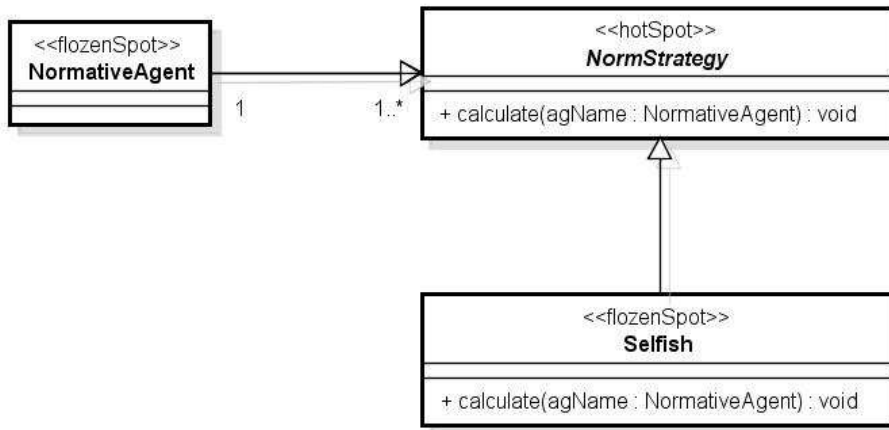


Figura 4.14 - Estratégia para lidar com normas

A classe *EnvironmentSimulation* também é responsável por gerenciar um conjunto de estratégias para criação dos objetivos individuais dos agentes, estes representados pela classe abstrata *GenerateGoalStrategy*.

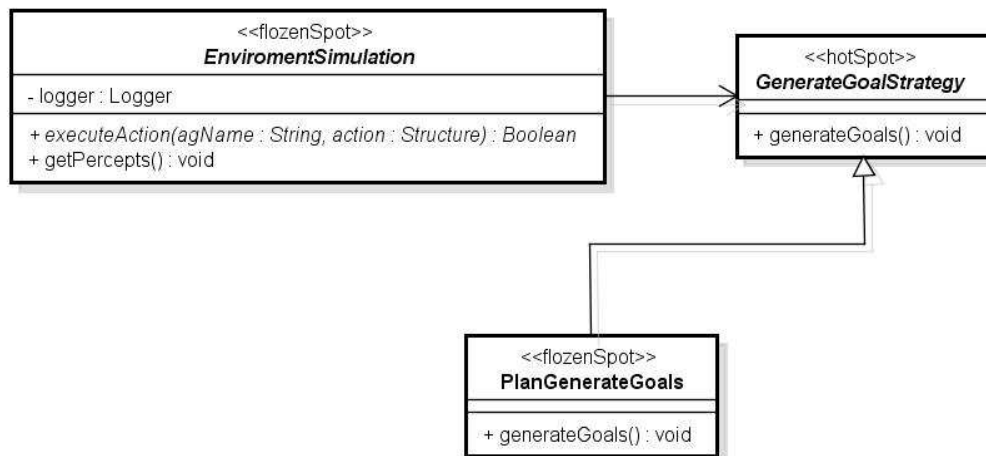


Figura 4.15 - Gerar os goals dos Agentes

Utilizou-se a classe *PlanGenerateNorms* (Ver Figura 4.15), uma extensão de *GenerateGoalsStrategy* que através do método *generateGoals* cria os objetivos dos agentes bombeiros de acordo com seus planos. Os objetivos gerados para a simulação foram: (i) salvar civis em áreas de risco, (ii) não colocar a vida de civis em risco, (iii) receber apoio de veículos aéreos.

Na atividade *collect* (Ver Seção 3.2.5) os monitores para interceptação de mensagens (Ver método *sensor* da classe *MessageMonitor* na Figura 3.6) e informações do ambiente (Ver método *sensor* da classe *EnvironmentMonitor* na Figura 3.6) ambos disponibilizados pelo JSAN foram utilizados.

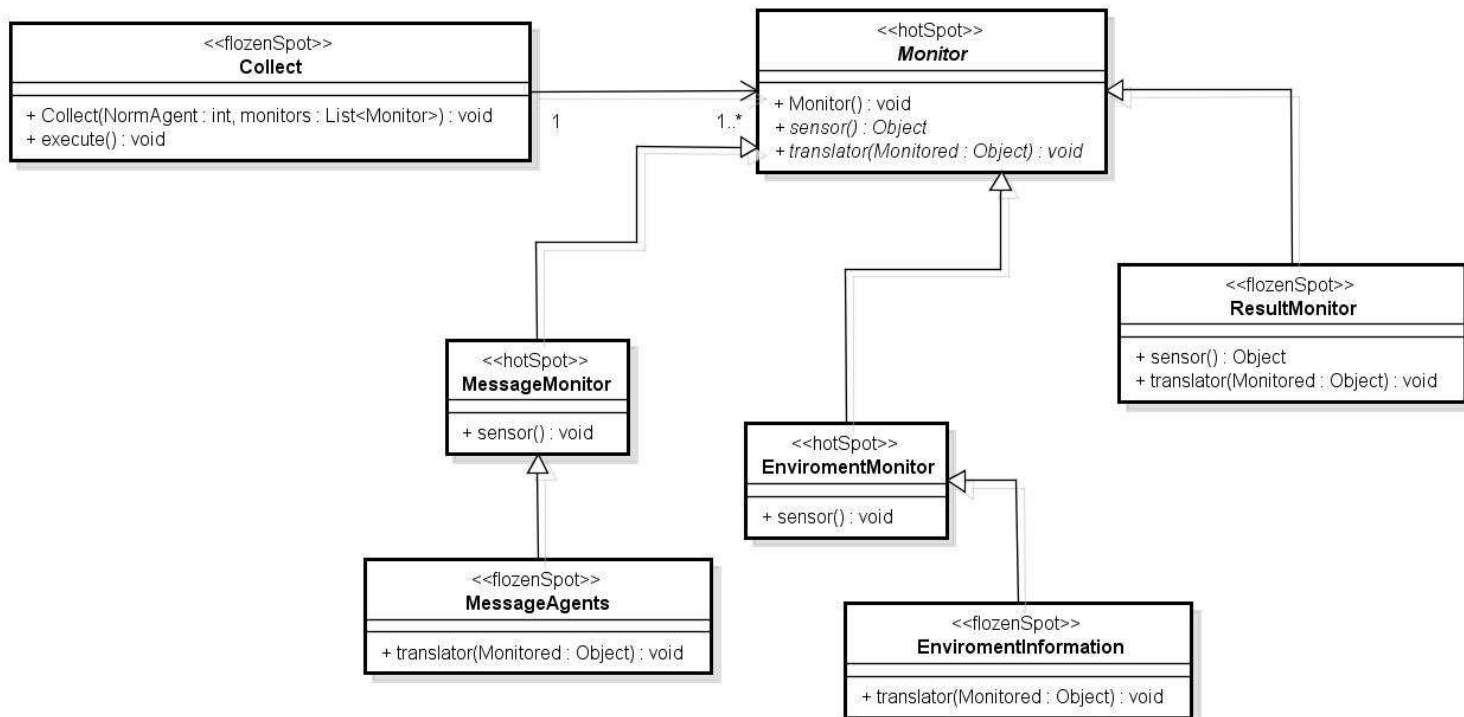


Figura 4.16 - Atividade *Collect*

O primeiro irá verificar se existem novas mensagens enviadas pelo agente Comandante dos Bombeiros e o segundo irá verificar se existem outras informações disponíveis no ambiente. A classe *ResultMonitor* (Ver Figura 4.16) que estende a classe *Monitor* (Ver Figura 3.6) e implementa seus métodos abstratos *sensor* e *translator* foi criada. No método *sensor* foi definida uma implementação que verifica a importância de uma mensagem dado o seu tempo de duração na simulação. Já o método *translator* formata o resultado de *sensor* de forma que o agente consiga entender as mensagens.

Por fim, duas classes, *MessageAgents* e *EnviromentInformation* (Ver Figura 4.16) foram criadas como extensão das classes *MessageMonitor* e *EnviromentMonitor*, respectivamente, a fim de implementar o método abstrato *translator* para formatar os dados: (i) enviados pelo agente Comandante dos Bombeiros a partir do método *translator* da classe *MessageAgents*, e (ii) coletados a partir do monitoramento do ambiente, ou seja, informações disponibilizadas durante o tempo de simulação, por exemplo as áreas de risco e quais veículos de suporte serão disponibilizados para ajudar no resgate, a partir do método *translator* da classe *EnviromentInformation*.

No diagrama de classes apresentado na Figura 4.17, temos a classe *EnvironmentSimulation*, responsável por gerenciar um conjunto de estratégias para geração de normas representadas pela classe abstrata *GenerateNormsStrategy*, a qual possui o método abstrato *generateNorms*. Este método foi utilizado para criação de normas no ambiente visando apenas os agentes bombeiros, isto é, dado um conjunto específico de agentes (Ver Seção 3.2.1) através da classe *PerceptGenerateNorms* disponibilizada pelo *framework* JSAN.

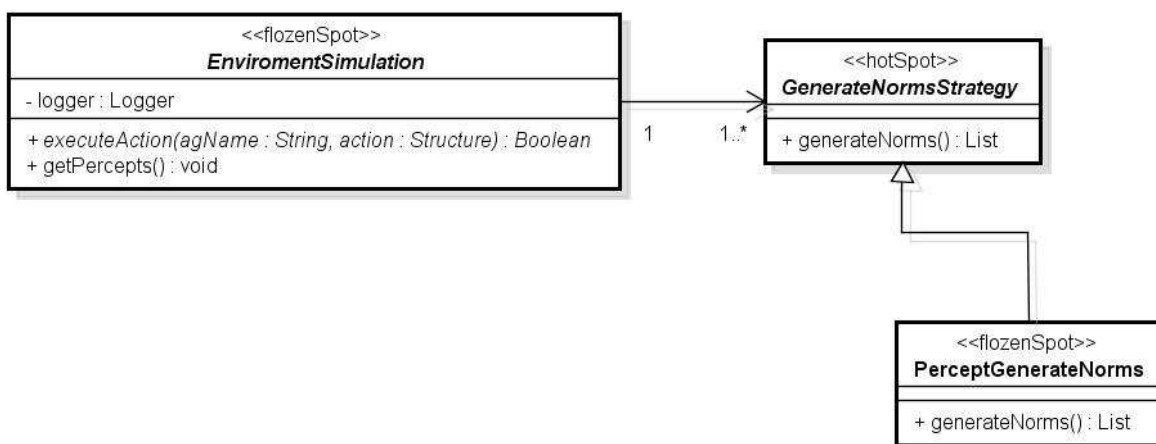


Figura 4.17 - Gerar Normas dos Agentes

Com a possibilidade de especificar a forma de movimento dos agentes bombeiros, foi estendido a classe *MovementStrategy* para implementar no método *execute* a forma específica dos agentes bombeiros se locomoverem.

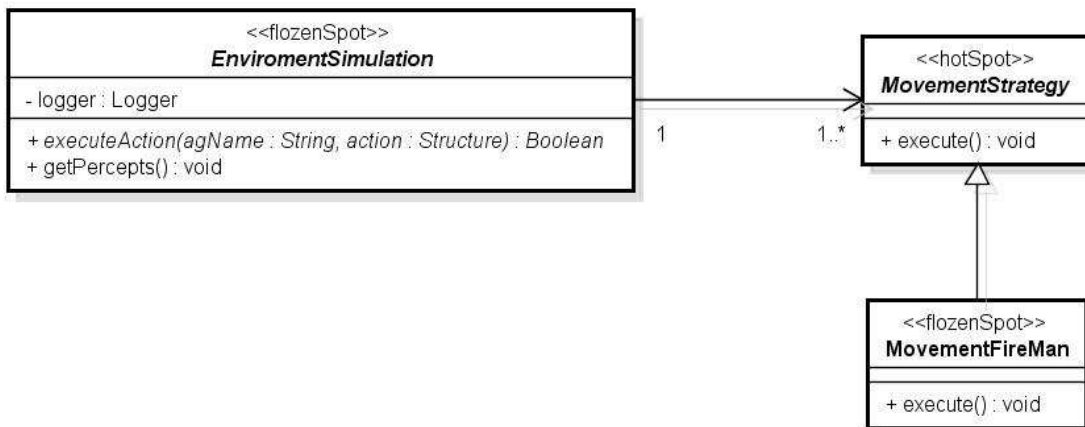


Figura 4.18 - Estratégia de movimentos dos Agentes Bombeiros

Para isso, foi criada a classe *MovementFireMan* (Ver figura 4.18), que estende classe *MovementStrategy*. A estratégia de movimento dos agentes bombeiros define que quando existe algum civil em risco eles devem ir até a localidade verificar a necessidade de salvamento dessa pessoa. Caso seja possível fazer o resgate imediatamente, ele desloca-se até o ponto e salva está pessoa. Entretanto, se não tiver como ele chegar lá sem veículos, ele precisa acionar o comando para disponibilizar outras formas de se locomover, por exemplo, através de veículos aéreos.

Na classe *Norm* (ver Figura 4.19) foram definidos atributos para que se possam criar as sanções, condições de ativação e desativação, o elemento que será regulado pela norma e qual o conceito deontico que está associado a ela (Ver Seção 3.2). Dado a norma apresentada na Seção 4.2.1: “proteger a vida de civis em áreas de risco”, foi associado a norma a recompensa do agente obter apoio de veículos aéreos ou terrestre e caso seja violada o agente não receberá apoio terrestre.

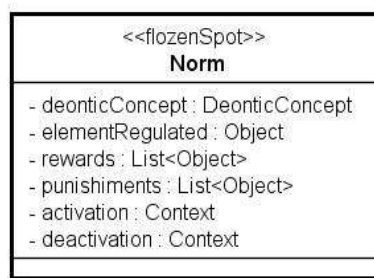


Figura 4.19 – Normas

Já nos atributos *activation* e *deactivation* da classe *Norm*, foi implementado as condições de ativação e desativação da norma: a condição da norma estar ativa é ter pessoas em áreas de risco e a desativação é quando a localidade está segura. O atributo *elementRegulated* recebeu o elemento que a norma irá regular, que é a ação de utilizar veículos aéreos. Já o atributo *deonticConcept* ficou com o valor de obrigação.

A classe *EnviromentSimulation* também é responsável por gerenciar um conjunto de estratégias para visualização das informações sobre a simulação, representados pela classe *ReportStrategy*. A estratégia de contribuição social disponibilizada pelo *framework* JSAN foi utilizada. Ela irá verificar a contribuição

social de uma norma caso seja cumprida ou violada pelos agentes bombeiros (Ver classe *SocialContributionReport* na Figura 4.20). Já em segundo verifica as informações da norma: (i) recompensas e punições atrelados a normas, (ii) conceito deôntico associado as normas e (iii) planos utilizados para cumprir com a norma (Ver classe *SpecificInformationReport* na Figura 4.20).

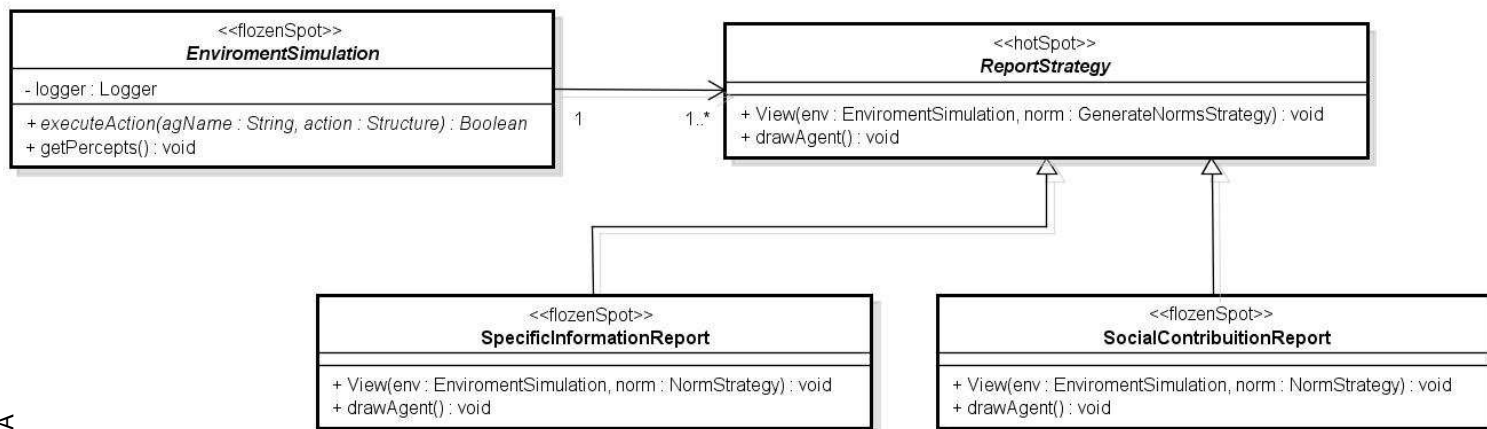


Figura 4.20 - Gerar Relatórios

Para ilustrar a simulação feita utilizando o JSAN no cenário de Evacuação de Civis em Áreas de Risco, foi gerado uma visualização através do uso das classes *SpecificInformationReport* e *SocialContributionReport* (Ver Figura 4.20), extensões de *ReportyStrategy*.

A Figura 4.21 mostra o agente Comandante dos Bombeiros recebendo uma mensagem sobre as condições do tempo, falando que o tempo está ruim e logo em seguida uma mensagem sobre a existência de civis em uma localidade de risco. Após receber as mensagens o agente Comandante dos Bombeiros envia um alerta para os agentes bombeiros que recebem as informações da existência de civis em áreas de risco.

Beginning of the simulation

```

[commanderAgent] Reported that the weather is bad
[commanderAgent] Reported that workers are in a dangerous place
[fireManAgent] Received information from workers at risk:hazardousLocation
    
```

Figura 4.21- Início da Simulação

Já a Figura 4.22, tem-se o planejamento de poder utilizar veículos aéreos, veículos terrestres, e planos caso eles precisem chamar reforços aéreos ou terrestres.

```

All Plans
-----
Plan: 1 evacuateWorkers(A)

Plan: 2 useHelicopters(A)

Plan: 3 useTroops(A)

Plan: 4 getMoreTroops(MT)

Plan: 5 getMoreHelicopters(MH)

Plan: 6 getLandBasedHelicopters(LBH)

Plan: 7 returnLandBasedHelicopters(RLBH)

Plan: 8 returnHelicopters(RH)

Plan: 9 returnTroops(RT)

```

Figura 4.22 - Planos de Resgate

Na Figura 4.23 tem-se a norma discutida na Seção 4.2.1, “proteger a vida de civis em áreas de risco”.

```

Instantiated Norm :
-----
Deontic Concept: obligation

action : useHelicopters(A)

Rewards: getMoreTroops(X)

Rewards: getMoreHelicopters(Y)

Punishments:
Deontic Concepts: obligation
action : returnTroops(X)

Norm Reward: getMoreTroops(X)
Reward Norm Contribution For Fulfilling +1: 1
Norm Reward: getMoreHelicopters(Y)
Reward Norm Contribution For Fulfilling +1: 2

```

Figura 4.23 – Uma das Normas Criadas na Simulação

Seu conceito deontico é obrigatório, (ii) uma recompensa caso a norma seja cumprida é o agente obter apoio aéreo ou apoio terrestre, caso seja violada o agente não receberá apoio terrestre para o resgate, (iii) a norma é ativada caso tenha alguma pessoa em risco e a norma é desativada quando todos civis estão a salvo, e (v) o elemento que a norma regula é a ação de usar veículos aéreos. *Norm Reward* na Figura 4.23, são as recompensas associadas a norma e *Reward Norm* mostra o quanto o agente irá ganhar caso cumpra com a norma.

Por fim, na Figura 4.24, pode ser visualizado qual dos planos os agentes bombeiros decidiram utilizar depois que a norma foi ativada devido a existência de civis em áreas de risco.

```

Plans in Select Option
-----

Activation :evacuateWorkers(A):
-----
If. useHelicopters(A)
  Contribution: contribution((-1))
-----
If. useTroops(A)
  Contribution: contribution(1)
-----

Result:
-----

{fireManAgent} FireManAgent evacuate using Troops
{fireManAgent} FireManAgent uses troops

```

Figura 4.24 - Decisão tomada pelos agentes bombeiros

Se ele optar pelo uso de veículos aéreos isto irá contribuir negativamente, já se ele optar pelo uso de veículos terrestres, sua contribuição será positiva. Como os agentes bombeiros da simulação utilizam a estratégia *Selfish* (Ver Seção 3.2.5) para lidar com normas, eles optaram por evacuar os civis que estavam nesta área de risco através do uso de veículos terrestres.