

6

Cenários

Neste capítulo descreveremos dois cenários que demonstram a aplicabilidade do AgentSpeak(L) e Jason Normativo. Na Seção 6.1 é apresentado um cenário no contexto de planejamento de missões de resgate reguladas por normas. Na Seção 6.2 um cenário no contexto de desenvolvimento de software é apresentado.

6.1

Cenário 1: Agentes de Suporte para Planejamento de Missões de Resgate Reguladas por Normas

A relação entre organizações humanitárias e militares tem mudado consideravelmente nas últimas décadas em resposta a diversas crises, tais como, desastres ambientais e guerras civis, que tem acontecido (Harmer 2009). A formação de coalizões¹ (Woolridge 2011) entre estas duas organizações tem sido motivada pelo fato de que nenhuma organização tem a capacidade e recursos necessários para cumprir todas as suas missões sozinha.

Embora tais organizações tendam a colaborar para o cumprimento de objetivos comuns, elas possuem objetivos individuais. Como discutido em (Sycara et al. 2010), a fim de influenciar cada organização participante da coalizão a colaborar, um conjunto de normas é definido. Tais normas definem obrigações e proibições que regulam o comportamento dos membros das organizações, como também podem fornecer recompensas pelo cumprimento ou estabelecer punições pela violação das normas.

Por exemplo, considere um cenário de missão de resgate, onde existem dois grupos formando uma coalizão, um grupo de ajuda humanitária (nomeado *Grupo H*) com o objetivo de resgatar e fornecer atendimento médico a feridos que estão em uma região potencialmente hostil, e um grupo militar (nomeado *Grupo M*) que tem de coordenar suas atividades para atingir seus objetivos militares, tais como combater os insurgentes, com atividades humanitárias. A situação ideal para *Grupo H* seria resgatar e prestar atendimento médico ao maior número de feridos. Para isso, *Grupo H* pode precisar de apoio do *Grupo*

¹Uma forma de organização cujos membros trabalham em conjunto.

M, tais como uma escolta para realizar resgate de feridos que estão em uma região hostil ou um auxílio na assistência médica prestada, principalmente, em casos onde o número de feridos excede a capacidade do *Grupo H*. *Grupo M*, por outro lado, tem objetivos militares que, potencialmente, podem estar em conflito com o apoio dado ao *Grupo H*.

Em suma, os grupos supracitados possuem os seguintes objetivos: (*Grupo H*) Resgatar e prestar atendimento médico ao maior número de feridos possível; e (*Grupo M*) Possui como objetivo principal atacar os insurgentes e, como objetivo de menor prioridade, contribuir para a coalizão, escoltando *Grupo H* enquanto ele resgata feridos que estão em uma região hostil e prestar assistência ao *Grupo H* no provimento de serviços médicos.

A fim de cumprir suas missões, cada grupo tem um conjunto de recursos limitados gerenciados pela *Central de Operações*(CO), por exemplo, *Grupo M* pode ter os seguintes recursos: helicópteros, helicópteros terrestres, jipes e soldados. Sendo assim, também é objetivo do *Grupo M* obter mais recursos ou manter os recursos já existentes a sua disposição.

Por fim, consideramos que neste cenário cada grupo tem associado uma taxa de risco que varia de acordo com os seus comportamentos realizados. Por exemplo, o fornecimento de escolta e assistência ao *Grupo H* no provimento de atendimento médico é importante dado que o atingimento de ambos os objetivos pode contribuir para a diminuição do risco de feridos não serem salvos dado que *Grupo H* irá realizar suas atividades com maior segurança. Logo, se o *Grupo M* fornece uma escolta ou assistência ao *Grupo H*, a taxa de risco associada ao *Grupo M* é diminuída.

Neste cenário, abordaremos como o comportamento do *Grupo M* pode ser controlado a partir de um conjunto de normas. Para tanto, normas inspiradas nas diretrizes internacionais para a cooperação humanitária/militar (Harmer 2009) foi definida a fim de influenciar *Grupo M* a expandir seu comportamento além de suas funções tradicionais e evitar comportamentos inadequados, obrigando ou proibindo *Grupo M* a realizar determinados comportamentos, provendo novos recursos ou removendo recursos à disposição do *Grupo M*, ou até mesmo tornando explícito os ganhos e perdas em relação ao risco gerado como consequência do seu comportamento.

(Norma 1): Se *Grupo H* está indo resgatar feridos em uma região hostil, *Grupo M* é obrigado a fornecer uma escolta para *Grupo H*.

(Recompensas): Se a escolta é provida:

(Recompensa 1): *Grupo M* tem uma diminuição no risco de feridos não serem salvos;

(Recompensa 2): A *Central de Operações* é obrigada disponibilizar mais soldados para *Grupo M*.

(Punições): Se a escolta não é provida:

(Punição 1): *Grupo M* tem um aumento no risco de feridos não serem salvos;

(Punição 2): *Grupo M* perde parte de seus soldados;

(Norma 2): Se *Grupo M* não é capaz de fornecer uma escolta com os recursos necessários para garantir a segurança mínima para *Grupo H* durante o resgate de feridos em uma região hostil, *Grupo M* é proibido de realizar a escolta. Isto é importante para evitar que *Grupo H* fique em uma situação perigosa, porque confiou na escolta fornecida pelo *Grupo M*.

(Punições): Se a escolta é provida:

(Punição 1): *Grupo M* tem um aumento no risco do *Grupo H* ser atacado;

(Norma 3): Se *Grupo M* está sendo obrigado a escoltar *Grupo H* durante a prestação de serviços médicos realizada pelo *Grupo H* e o número de feridos excede a capacidade de assistência do *Grupo H*. Então, *Grupo M* também é obrigado a assistir *Grupo H* na prestação do serviço.

(Recompensas): Se a assistência é provida:

(Recompensa 1): *Grupo M* tem uma diminuição no risco de feridos não serem salvos dado que *Grupo H* terá ajuda do *Grupo M* na assistência médica;

(Punições): Se a assistência não é provida:

(Punição 1): *Grupo M* tem um aumento no risco de feridos não serem salvos dado que *Grupo H* não terá ajuda do *Grupo M* e *Grupo H* não possui contingente suficiente para assistir os feridos;

(Norma 4): Se existe insurgentes, *Grupo M* é obrigado a atacá-los.

(Recompensas): Se o ataque é realizado:

(Recompensa 1): *Grupo M* tem uma diminuição no risco dos insurgentes aumentarem seu poder de atuação;

(Recompensa 2): A *Central de Operações* é obrigada a disponibilizar helicópteros para *Grupo M*.

(Punições): Se o ataque não é realizado:

(Punição 1): *Grupo M* tem um aumento no risco dos insurgentes aumentarem seu poder de atuação;

(Norma 5): Se o número de insurgentes excede o número de soldados por uma certa quantia, *Grupo M* é proibido de atacar os insurgentes, dado que ele pode colocar a tropa em perigo.

(Punições): Se o ataque é realizado:

(Punição 1): *Grupo M* tem um aumento no risco dado que a tropa está sendo posta em perigo;

(Norma 6): A fim de preservar a segurança das tropas, se o tempo estiver ruim, *Grupo M* é proibido de usar helicópteros.

(Punições): Se *Grupo M* utiliza os helicópteros:

(Punição 1): ele terá um aumento no risco da sua tropa sofrer um acidente;

(Punição 2): *Grupo M* é proibido de utilizar helicópteros terrestres por um período de tempo;

(Punições): Se os helicópteros são utilizados:

(Punição 1): *Grupo M* perde parte de seus soldados;

O planejamento de ações conjuntas em tais cenários já é um problema complexo para planejadores humanos. Ele se torna ainda mais difícil ou mesmo impossível, quando tais grupos estão sujeitos a normas organizacionais (obrigações e proibições), especialmente quando existem conflitos normativos.

A fim de lidar com tal desafio apresentaremos como agentes implementados a partir do Jason Normativo, podem ajudar ao *Grupo M* em suas atividades de planejamento normativo raciocinando sobre os interesses do *Grupo M* e as normas associadas ao mesmo.

6.1.1

Representação de Objetivos, Ações, Planos e Normas

Nesta seção descreveremos como objetivos, ações, planos disponíveis para o *Grupo M* e as normas endereçadas ao mesmo podem ser representadas utilizando o *AgentSpeak(L) Normativo*.

Como descrito anteriormente, o *Grupo M* possui um conjunto de objetivos a serem atingidos e um conjunto de recursos que podem ser utilizados. Assumindo que tais recursos são utilizados através da execução de ações, abaixo apresentamos como tais objetivos e ações podem ser descritas, como também uma possível configuração das motivações e satisfações² do *Grupo M* em atingir tais objetivos e realizar tais ações.

(*Atacar Insurgentes*): O objetivo principal do *Grupo M* é atacar os insurgentes, então assumiremos a maior motivação, igual a 10, para atingir tal objetivo que é representado como apresentado abaixo:

! *atacar(insurgentes)*

(*Escortar Grupo H*): Dado que o atingimento dos objetivos resultantes da coalizão está além das funções tradicionais do *Grupo M*, assumiremos uma menor motivação, igual a 8, para os objetivos relacionados ao fornecimento de escoltas ao *Grupo H* que é representado como descrito abaixo:

! *escortar(grupoh)*

(*Assistir Grupo H nos Serviços Médicos*): Seguindo o mesmo raciocínio de definir uma menor motivação para objetivos resultantes da coalizão, assumiremos que a motivação para assistir *Grupo H* nos serviços médicos é igual 8 e tal objetivo é representado como descrito abaixo:

! *assistir(grupoh)*

(*Ter o Risco Aumentado ou Diminuído*): Assumindo que a taxa de risco pode sofrer freqüentes variações com um aumento de 1 ou uma diminuição de -1. Definiremos a motivação do *Grupo M* por ter a taxa de risco aumentada igual a -1 ou diminuída igual 1. E tais objetivos são representados abaixo, onde o primeiro indica que o risco associado ao

²neste artigo, o valor de uma *importância* varia de -10 a 10

Grupo M foi diminuído, já o segundo indica que o risco associado ao *Grupo M* foi aumentado.

$!risco(grupom, diminuido)$
 $!risco(grupom, aumentado)$

(Disponibilização ou Perda de Recursos) :Obter mais recursos ou manter os já existentes é importante a fim de facilitar o atingimento dos objetivos do *Grupo M*. Entretanto, este não é o principal objetivo do *Grupo M*. Sendo assim, assumiremos uma motivação mediana, igual a 5, para a obtenção de novos recursos. Caso contrário, se recursos são retirados do *Grupo M*, assumiremos uma motivação de -5. Tais objetivos são representados abaixo, onde o primeiro indica que novos recursos foram disponibilizados para o *Grupo M*, já o segundo indica alguns dos recursos do *Grupo M* foram removidos.

$!disponibilizar(Recursos, grupom)$
 $!remover(Recursos, grupom)$

(Utilizar recursos): Embora os recursos tenham um custo associado à sua utilização, eles são fundamentais para *Grupo M* cumprir suas missões. Portanto, assumiremos uma satisfação de valor 2 para o *Grupo M* executar as ações para utilização dos recursos e tais ações são representadas como segue:

$satisfaction(use(Recursos)) = 2$

A fim de ilustrar como uma norma pode ser descrita utilizando o *AgentSpeak(L) Normativo*, considere *Norma 1* apresentada anteriormente.

Norma 1: "Se *Grupo H* está indo resgatar feridos numa região hostil, *Grupo M* é obrigado a escoltá-lo". A partir disto, podemos produzir os seguintes componentes da *Norma 1*:

(identifier): Considere que o identificador da *Norma 1* é *norm1*;

norm1

(addressees): *Norma 1* é endereçada ao *Grupo M*:

norm1(*grupom*

(deonticoncept): *Norma 1* é uma obrigação;

$$\begin{aligned} &norm1(\textit{grupom}, \\ &\quad \underline{OBLIGATION}, \end{aligned}$$

(activation): O contexto de ativação é satisfeito se *Grupo H* está indo resgatar feridos em uma região hostil, como apresentado abaixo:

$$\begin{aligned} &norm1(\textit{grupom}, \\ &\quad OBLIGATION, \\ &\quad \underline{\textit{regiaohostil}(\textit{grupoh})}, \end{aligned}$$

(deactivation): Assumindo que *Norma 1* está desativada se *Grupo H* não está numa região hostil, o contexto de desativação é descrito como segue:

$$\begin{aligned} &norm1(\textit{grupom}, \\ &\quad OBLIGATION, \\ &\quad \textit{regiaohostil}(\textit{grupoh}), \\ &\quad \underline{\textit{not regiaohostil}(\textit{grupoh})}, \end{aligned}$$

(behavior): *Norma 1* é uma obrigação para *Grupo M* escoltar *Grupo H*:

$$\begin{aligned} &norm1(\textit{grupom}, \\ &\quad OBLIGATION, \\ &\quad \textit{hostileregion}(\textit{grupoh}), \\ &\quad \textit{not hostileregion}(\textit{grupoh}), \\ &\quad \underline{\textit{! escoltar}(\textit{grupoh})} \end{aligned}$$

(recompensas): As recompensas da *Norma 1* são: o risco de feridos não serem salvos é diminuído e a *Central de Operações* é obrigada a disponibilizar mais soldados para *Grupo M*. Assim, podemos representar as recompensas, como segue:

(Recompensa 1): O risco é diminuído;

$$\begin{aligned}
&norm1(\textit{grupom}, \\
&\quad OBLIGATION, \\
&\quad \textit{hostileregion}(\textit{grupoh}), \\
&\quad not\ \textit{hostileregion}(\textit{grupoh}), \\
&\quad !\ \textit{escoltar}(\textit{grupoh}) \\
&\quad [!\ \textit{risco}(\textit{grupom}, \textit{diminuido}), \\
&\quad \quad norm1.1(co, \\
&\quad \quad \quad \underline{OBLIGATION},
\end{aligned}$$

(activation): O contexto de ativação é satisfeito se *Norma 1* foi cumprida:

$$\begin{aligned}
&norm1(\textit{grupom}, \\
&\quad OBLIGATION, \\
&\quad \textit{hostileregion}(\textit{grupoh}), \\
&\quad not\ \textit{hostileregion}(\textit{grupoh}), \\
&\quad !\ \textit{escoltar}(\textit{grupoh}) \\
&\quad [!\ \textit{risco}(\textit{grupom}, \textit{diminuido}), \\
&\quad \quad norm1.1(co, \\
&\quad \quad \quad OBLIGATION, \\
&\quad \quad \quad \underline{nc(FULFILLED, norm1(\neg, \neg, \neg, \neg, \neg, \neg))}),
\end{aligned}$$

(deactivation): O contexto de desativação é satisfeito se a *Central de Operações* disponibilizou os recursos para *Grupo M*;

$$\begin{aligned}
&norm1(\textit{grupom}, \\
&\quad OBLIGATION, \\
&\quad \textit{hostileregion}(\textit{grupoh}), \\
&\quad not\ \textit{hostileregion}(\textit{grupoh}), \\
&\quad !\ \textit{escoltar}(\textit{grupoh}) \\
&\quad [!\ \textit{risco}(\textit{grupom}, \textit{diminuido}), \\
&\quad \quad norm1.1(co, \\
&\quad \quad \quad OBLIGATION, \\
&\quad \quad \quad nc(FULFILLED, norm1(\neg, \neg, \neg, \neg, \neg, \neg)), \\
&\quad \quad \quad \underline{\textit{recursosdisponibilizados}(\textit{grupom})},
\end{aligned}$$

(behavior): A norma da recompensa está obrigando *CO* a atingir o objetivo que disponibiliza mais soldados para *Grupo M*;

```

norm1( grupom,
      OBLIGATION,
      hostileregion(grupoh),
      not hostileregion(grupoh),
      ! escoltar(grupoh)
      [! risco(grupom, diminuido),
       norm1.1(co,
               OBLIGATION,
               nc(FULFILLED, norm1(—, —, —, —, —, —)),
               recursosdisponibilizados(grupom),
               ! disponibilizar(soldados, grupom)

```

(recompensas e punições): A *norm1.1* não fornece recompensas nem punições;

```

norm1( grupom,
      OBLIGATION,
      hostileregion(grupoh),
      not hostileregion(grupoh),
      ! escoltar(grupoh)
      [! risco(grupom, diminuido),
       norm1.1(co,
               OBLIGATION,
               nc(FULFILLED, norm1(—, —, —, —, —, —)),
               recursosdisponibilizados(grupom),
               ! disponibilizar(soldados, grupom),
               [],
               []),

```

(punições): As punições por violar *Norma 1* é o aumento no risco e a perda de soldados;

(Punição 1): O risco é aumento:

```

norm1( grupom,
      OBLIGATION,
      hostileregion(grupoh),
      not hostileregion(grupoh),
      ! escoltar(grupoh)
      [! risco(grupom, diminuido),
       norm1.1(co,
               OBLIGATION,
               nc(FULFILLED, norm1(—, —, —, —, —)),
               recursosdisponibilizados(grupom),
               ! disponibilizar(soldados, grupom),
               [],
               []),
       [! risco(grupom, aumentado),

```

(Punição 2): Perda de soldados;

```

norm1( grupom,
      OBLIGATION,
      hostileregion(grupoh),
      not hostileregion(grupoh),
      ! escoltar(grupoh)
      [! risco(grupom, diminuido),
       norm1.1(co,
               OBLIGATION,
               nc(FULFILLED, norm1(—, —, —, —, —)),
               recursosdisponibilizados(grupom),
               ! disponibilizar(soldados, grupom),
               [],
               []),
       [! risco(grupom, aumentado),
       ! remover(soldados, grupom)]])

```

A fim de guiar o *Grupo M* no atingimento de seus objetivos, uma biblioteca de planos é disponibilizada. Por exemplo, considere os planos de 1, 2 e 3 que seguem. Os dois primeiros planos têm a mesma *condição de invocação*, ambos são acionados em resposta a eventos cujo *trigger* está relacionado a adição de um objetivo representando o desejo do *Grupo M* em escoltar o *Grupo H*. O *contexto* do plano 1 é satisfeito se existe insurgentes próximos ao *Grupo*

H e a *Norma 5*, que proíbe *Grupo M* de atacar insurgentes, está desativada. O *contexto* do plano 2 é satisfeito se *Norma 3*, que obriga *Grupo M* a fornecer assistência médica, está ativada. O corpo do Plano 1 realiza ações para utilizar soldados e helicópteros, e atingi o objetivo de atacar insurgentes. O corpo do Plano 2 realiza ações para utilizar soldados e helicópteros terrestres, e atingi o objetivo de assistir o *Grupo H* nos serviços médicos.

Plano 3 que tem uma *conditon de invocação* que é acionada em resposta a eventos cujo *trigger* está relacionado a adição de um objetivo representando o desejo do *Grupo M* em atacar insurgentes. Seu contexto é verdadeiro se a *Norma 6*, que proíbe o uso de helicópteros, foi violada. Seu corpo executa ações para utilizar soldados e jipes.

Plan 1:

```
+ ! escoltar(grupoh)
: exist(insurgentes, grupoh) &
  nc(DEACTIVATED, norm5(—, —, —, —, —, —, —))
< -action use(soldados);
  action use(helicopteros);
  ! atacar(insurgentes).
```

Plan 2:

```
+ ! escoltar(grupoh)))
: nc(ACTIVATED, norm3(—, —, —, —, —, —, —))
< -action use(soldados);
  action use(helicopterosterrestres);
  ! assistir(grupoh).
```

Plan 3:

```
+ ! atacar(insurgentes)
: nc(VIOLATED, norm6(—, —, —, —, —, —, —))
< -action use(soldados);
  action use(jipes).
```

6.1.2

Adotando Novas Normas

Após descrever os principais objetivos, ações, planos e normas do *Grupo M*. Iniciaremos a descrição do processo executado pelo agente de suporte do

Grupo M.

O primeiro passo é verificar a adoção das normas. Para tanto, considere as normas apresentadas na Seção 6.1, se *Grupo M* recebe informações sobre as normas do sistema, o Algoritmo 1 (apresentado na Seção 5.2.1 e que é responsável por verificar a adoção de normas) é executado para verificar se as normas já estão armazenadas no conjunto de normas adotadas do *Grupo M* e comparar as informações do destinatário da norma com *Grupo M*. Neste caso, as Normas 1, 2, 3, 4, 5 e 6 descritas no cenário são dirigidas ao *Grupo M* e as mesmas são adicionadas ao conjunto de normas adotadas do *Grupo M*.

6.1.3

Revisando as Normas Ativadas e Desativadas

Após realizar a adoção de normas, verificaremos quais normas foram ativadas ou desativadas utilizando o Algoritmo 2 apresentado na Seção 5.2.1. Por exemplo, considere a seguinte situação: Se *Grupo H* vai resgatar feridos em uma região hostil, o contexto de ativação da *Norma 1* é consequência lógica das crenças do agente, portanto, a *Norma 1* é adicionada a base de normas ativadas e *Grupo M* é obrigado a fornecer uma escolta para *Grupo H*. Dado que *Norma 1* é ativada e assumindo que o número de feridos excede a capacidade de assistência médica do *Grupo H*, o contexto de ativação da *Norma 3* é satisfeito. Assim, a *Norma 3* é adicionada ao conjunto de normas ativadas e *Grupo M* é obrigado a assistir *Grupo H* no resgate de feridos.

Agora, se o *Grupo H* não está em uma região hostil mas, *Norma 1* torna-se inativa, e *Norma 3* também tornar-se inativa dado que seu contexto de desativação é satisfeito se *Norma 1* está inativa. Então, ambas as normas são adicionadas ao conjunto de normas desativadas.

6.1.4

Verificando as Normas Cumpridas ou Violadas

Se durante a ativação da *Norma 1*, *Grupo M* escoltou *Grupo H*, o comportamento regulado pela *Norma 1* é satisfeito e a relação *normfulfillment* da *Norma 1* é atualizada para *fulfilled*. No entanto, se *Grupo M* não forneceu uma escolta durante a ativação da *Norma 1*, tal norma foi violada e a relação *normfulfillment* da *Norma 1* é atualizada para *violated*.

A verificação de quais normas foram cumpridas ou violadas é realizada a partir da execução do Algoritmo 3 apresentado na Seção 5.2.1.

6.1.5

Detectando e Superando Conflitos Normativos

Se as Normas 1 e 2 do nosso cenário estão ativadas, elas estão em conflito dado que a *Norma 1* está obrigando *Grupo M* a escoltar *Grupo H*, enquanto a *Norma 2* está proibindo *Grupo M* a escoltar *Grupo H*. Tal conflito é superado verificando que a influência deontica da *Norma 1* (ID1) é 8 (dado que a *Norma 1* está obrigando *Grupo M* a escoltar *Grupo H* e a motivação do *Grupo M* em fornecer escolta é igual a 8), a importância por receber as recompensas da *Norma 1* (IR1) é 6 (dado que se a *Norma 1* é cumprida o risco associado ao *Grupo M* é diminuído e a *Central de Operações* é obrigada a disponibilizar mais recursos para *Grupo M*, considerando que a motivação do *Grupo M* em ter seu risco diminuído é igual a 1 e a motivação para ter mais recursos disponíveis é igual a 5, então, a importância para o *Grupo M* em receber as recompensas da *Norma 1* é igual a 6) e por receber as punições da *Norma 1* (IP1) é -6 (dado que se a *Norma 1* é violada o risco associado ao *Grupo M* é aumentado e a *Central de Operações* remove parte dos recursos do *Grupo M*, considerando que a motivação do *Grupo M* em ter seu risco aumentado é igual a -1 e a motivação para ter parte dos recursos removidos é igual a -5, então, a importância para o *Grupo M* em receber as punições da *Norma 1* é igual a -6). A importância para receber as recompensas da *Norma 2* (IR2) é 0 (dado que a *Norma 2* não fornece recompensas) e para receber as punições da *Norma 2* (IP2) é -1 (dado que se a *Norma 2* é violada o risco associado ao *Grupo M* é aumentado e a motivação do *Grupo M* para ter seu risco aumentado é igual a -1).

Por fim, dado que:

$$\begin{aligned} ID1 + IR1 + IP2 &> IR2 + IP1 \\ 8 + 6 + (-1) &> 0 + (-6) \end{aligned}$$

Norma 1 é selecionada para ser cumprida e *Norma 2* é selecionada para ser violada.

Conflitos normativos são detectados e superados a partir da execução do Algoritmo 4 apresentado na Seção 5.2.2.

6.1.6

Selecionando Normas para serem Cumpridas ou Violadas

Após detectar e superar conflitos entre normas, é necessário escolher quais normas não conflitantes devem ser cumpridas ou violadas. Por exemplo, considere a situação onde *Norma 3* é ativada e o *Grupo M* é obrigado a assistir *Grupo H* nos serviços médicos. A influência deontica da *Norma 3* é 8 (dado

que a *Norma 3* é uma obrigação e a motivação do *Grupo M* para alcançar o objetivo "assistir(grupoh)" é 8). A importância para receber recompensas da *Norma 3* é 1 (dado que o risco será diminuído e a motivação do *Grupo M* para ter o risco diminuído é 1) e a importância de receber as punições da *Norma 3* é -1 (dado que a punição é o aumento do risco e a motivação do *Grupo M* para ter o risco aumentado é -1). Assim, *Norma 3* é selecionada para ser cumprida uma vez que a soma de sua influência deontica, que é 8, mais a importância para receber as recompensas, que é 1, é maior do que a importância para receber as punições, que é -1. No entanto, se assumirmos que a motivação do *Grupo M* para assistir *Grupo H* nos serviços médicos é -3, *Norma 3* é selecionada para ser violada.

A escolha de quais normas não conflitantes devem ser cumpridas ou violadas é realizada a partir da execução do Algoritmo 5 apresentado na Seção 5.2.2.

6.1.7

Aplicando Decisões Normativas

Após selecionar as normas que devem ser cumpridas ou violadas, é necessário efetivar tais decisões. Para tanto, lembre que a *Norma 1*, que obriga *Grupo M* a atingir o objetivo "!escoltar(grupoh)", foi selecionada para ser cumprida e a *Norma 2* que proíbe *Grupo M* de atingir tal objetivo foi selecionada para ser violada. A partir de tal decisão um novo evento cujo *trigger* representa a adição do objetivo do tipo "!escoltar(grupoh)" é adicionado a base de eventos do agente suportando o *Grupo M*.

O mesmo raciocínio é aplicado a *Norma 3* que obriga o agente a atingir o objetivo "!assistir(grupoh)" e foi selecionada para ser cumprida. Um evento cujo *trigger* representa a adição de tal objetivo é adicionado a base de eventos do agente suportando o *Grupo M*.

A efetivação de decisões normativas é realizada a partir da execução do Algoritmo 6 apresentado na Seção 5.2.2.

6.1.8

Seleção de Eventos

Agora o agente que fornece suporte ao *Grupo M* possui um conjunto de eventos na sua base. Por exemplo, lembre da Seção anterior que *Grupo M* possui eventos para serem lidados cujo *triggers* representam a adição dos objetivos "!escoltar(grupoh)" e "!assistir(grupoh)". Adicionalmente, também considere a existência de um evento cujo *trigger* está relacionado a adição do objetivo "!atacar(insurgentes)". A partir da execução do Algoritmo 7

apresentado na Seção 5.2.3 que avalia a prioridade de um trigger, a prioridade de tais *triggers* é avaliada levando em consideração a prioridade para atingir os objetivos que constituem tais *triggers* como segue:

(*escoltar(grupoh)*): Conforme descrito na Seção 6.1.1, a motivação do *Grupo M* para fornecer uma escolta para o *Grupo H* é 8. No entanto, tal objetivo é influenciado pelas Normas 1 e 2. Norma 1 é uma obrigação e a realização do objetivo "*!escoltar(grupoh)*" implica no cumprimento da *Norma 1*, isto é, receber as recompensas da *Norma 1* cuja importância é igual a 6 e não receber as punições da *Norma 1* cuja importância é igual a -6, como descrito na Seção 6.1.5. *Norma 2* é uma proibição e a realização do objetivo "*!escoltar(grupoh)*" implica na violação, isto é, receber as punições da *Norma 2* cuja importância é igual a -1 e não receber as recompensas, cuja importância é igual a 0, dado que *Norma 2* não fornece recompensas, como descrito na Seção 6.1.5. Assim, a influência normativa das Normas 1 e 2 sobre o comportamento "*!escoltar(grupoh)*" é avaliada da seguinte forma:

$$\begin{aligned} &behaviorNormativeInfluence(!escoltar(grupoh), norms) = \\ &rewardsImportance(norm1, norms) - \\ &punishmentsImportance(norm1, norms) + \\ &punishmentsImportance(norm2, norms) - \\ &rewardsImportance(norm2, norms) = \\ &6 - (-6) + (-1) + 0 = 11 \end{aligned}$$

Como resultado, a prioridade do *trigger* representando a adição do objetivo "*!escoltar(grupoh)*" é:

$$\begin{aligned} &triggerPriority(!escoltar(grupoh)) = \\ &motivation(!escoltar(grupoh)) + \\ &normativeInfluence(!escoltar(grupoh)) = 8 + 11 = 19 \end{aligned}$$

(*assistir(grupoh)*): Conforme descrito na Seção 6.1.1, a motivação do *Grupo M* para assistir *Grupo H* é 8. No entanto, este objetivo é influenciado pela *Norma 3*. *Norma 3* é uma obrigação e a realização do objetivo "*!assistir(grupoh)*" implica no cumprimento da mesma, isto é, receber as recompensas da *Norma 3* cuja importância é igual a 1 e não receber as punições cuja importância é igual a -1, como descrito na Seção 6.1.6. Assim, a influência normativa da Norma 3 sobre o objetivo "*!assistir(grupoh)*" é avaliada da seguinte forma:

$$\begin{aligned}
& \text{behaviorNormativeInfluence}(! \text{assistir}(\text{grupoh}), \text{norms}) = \\
& \text{rewardsImportance}(\text{norm3}, \text{norms}) - \\
& \text{punishmentsImportance}(\text{norm3}, \text{norms}) = \\
& 1 - (-1) = 2
\end{aligned}$$

Como resultado, a prioridade do *trigger* representando a adição do objetivo "*!assistir(grupoh)*" é:

$$\begin{aligned}
& \text{triggerPriority}(! \text{assistir}(\text{partyh})) = \\
& \text{motivation}(! \text{assistir}(\text{grupoh})) + \\
& \text{behaviorNormativeInfluence}(! \text{assistir}(\text{grupoh})) = 8 + 2 = 10
\end{aligned}$$

(*atacar(insurgentes)*): Conforme descrito na Seção 6.1.1, a motivação do *Grupo M* para atacar insurgentes é 10. No entanto, este objetivo é influenciado pela *Norma 4*. *Norma 4* é uma obrigação e a realização do objetivo "*!atacar(insurgentes)*" implica no cumprimento da mesma, isto é, receber as recompensas da *Norma 4* cuja importância é igual a 6 (dado que se a *Norma 4* é cumprida, o *Grupo M* será recompensado com a disponibilização de mais recursos, cuja motivação do *Grupo M* é igual a 5, e a diminuição na sua taxa de risco, cuja motivação do *Grupo M* é igual a 1) e não receber as punições cuja importância é igual a -1 (dado que se a *Norma 4* é violada, o *Grupo M* será punido com um aumento na sua taxa de risco, cuja motivação do *Grupo M* é igual a -1). Assim, a influência normativa da Norma 4 sobre o objetivo "*!atacar(insurgentes)*" é avaliada da seguinte forma:

$$\begin{aligned}
& \text{behaviorNormativeInfluence}(! \text{atacar}(\text{insurgentes}), \text{norms}) = \\
& \text{rewardsImportance}(\text{norm4}, \text{norms}) - \\
& \text{punishmentsImportance}(\text{norm4}, \text{norms}) = \\
& 6 - (-1) = 7
\end{aligned}$$

Como resultado, a prioridade do *trigger* representando a adição do objetivo "*!atacar(insurgentes)*" é:

$$\begin{aligned}
& \text{triggerPriority}(! \text{atacar}(\text{insurgentes})) = \\
& \text{motivation}(! \text{atacar}(\text{insurgentes})) + \\
& \text{behaviorNormativeInfluence}(! \text{atacar}(\text{insurgentes})) = 10 + 7 = 17
\end{aligned}$$

Dado que a prioridade do *trigger* relacionado a adição do objetivo "! escoltar(grupoh)" é maior que a prioridade dos *triggers* relacionado a adição dos objetivos "! assistir(grupoh)" e "! atacar(insurgentes)", o evento cujo *trigger* está relacionado a adição do objetivo "! escoltar(grupoh)" é selecionado para ser gerenciado, como descrito no Algoritmo 8 apresentado na Seção 5.2.3 que realiza a seleção de eventos.

6.1.9

Verificando Planos Relevantes e Aplicáveis

A fim de exemplificar o processo de encontrar os planos relevantes e aplicáveis. Vamos voltar ao nosso exemplo dos planos descritos na Seção 6.1 cujas condições de invocação são:

Plan1 :

+! *escoltar(grupoh)*

Plan2 :

+! *escoltar(grupoh)*

Plan3 :

+! *atacar(insurgentes)*

Lembre da Seção anterior, que o o evento cujo *trigger* está relacionado a adição do objetivo "! escoltar(grupoh)" foi selecionado para ser alcançado. Com base neste cenário, os planos 1 e 2 são retornados a partir da execução do processo *Unify Event* disponibilizado pelo Jason, dado que suas condições de invocação são mais relevantes para lidar com objetivo selecionado do que a condição de invocação do plano 3.

A fim de verificar se os planos 1 e 2 são aplicáveis, assuma que as crenças do Grupo *M* é *exist(insurgentes, grupoh)*, o que indica que existe *insurgentes* próximo ao *grupoh*. Além disto, dado que Norma 3 está ativada e Norma 5 está desativada. Nesse caso, nota-se que os contextos dos Planos 1 e 2 são satisfeitos a partir de tais informações. Portanto, os planos 1 e 2 são planos aplicáveis. Para realizar este passo o processo *Check Context* do Jason Normativo opercionalizando pelos Algoritmos 9 e 10 apresentados na Seção 5.2.4 são executados.

6.1.10

Seleção de Planos

Dado que os planos 1 e 2 foram considerados aplicáveis, é necessário selecionar um deles para se tornar uma intenção. O primeiro passo é avaliar

a importância de cada um dos planos a partir da execução do Algoritmo 13 apresentado na Seção 5.2.5, como segue:

(Plano 1) Como apresentado na Seção 6.1.1, a condição de invocação do *Plano 1* é o objetivo "*!escoltar(grupoh)*" cuja prioridade para atingir é igual a 19 (como descrito na seção 6.1.8), o corpo do plano 1 é composto pelas ações "*use(soldados)*" e "*use(helicopteros)*", e um objetivo "*!atacar(insurgents)*". A satisfação para realizar cada uma destas ações é igual 2 e a motivação para atingir o objetivo "*!atacar(insurgents)*" é igual a 10 (como descrito na Seção 6.1.1). Sendo assim, a importância principal do *Plano 1* é igual a 14. Agora, é necessário levar em consideração que o *Plano 1* sofre uma influência normativa das Normas 6 e 4.

(Influência Normativa da Norma 6) A *Norma 6* proíbe a execução da ação "*use(helicopteros)*", se tal norma é cumprida a importância por receber as recompensas é igual 0 (dado que *Norma 6* não fornece recompensas). Entretanto, se tal norma é violada a importância por receber as punições é igual a -3 (dado que o risco é aumentado e a motivação do *Grupo M* para ter o risco aumentado é igual a -1) e o *Grupo M* é proibido de utilizar helicópteros terrestres, se tal proibição é violada *Grupo M* perde parte de seus soldados, para analisar esta última punição é necessário recordar que o agente de suporte adota uma estratégia otimista para avaliar normas que fazem parte das sanções. Sendo assim, é necessário analisar se a influência deontica da proibição mais a importância por receber as recompensas é maior do que a importância por receber as punições estabelecidas pela proibição. Considerando que a proibição está proibindo *Grupo M* de executar a ação "*use(helicoptero terrestres)*" cuja satisfação é igual 2, então sua influência deontica é igual -2, a proibição não fornece recompensas e a punição por violar tal norma é a perda de parte dos soldados cuja motivação do *Grupo M* é igual a -5. Sendo assim, adotando uma estratégia otimista a importância por receber a punição 2 é igual -2. Como resultado, a influência normativa da *Norm 6* sobre a ação "*use(helicopteros)*" é igual -3.

(Influência Normativa da Norma 4) A *Norma 4* obriga o atingimento do objetivo "*!atacar(insurgentes)*", se tal norma é cumprida a importância por receber as recompensas é igual 6, dado que se *Norma 4* é cumprida o risco será diminuído, cuja motivação do *Grupo M* é igual a 1, e *CO* será obrigada a fornecer mais recursos

para o *Grupo M*, a motivação do *Grupo M* para recebimento de recursos é igual 5. Entretanto, se tal norma é violada a importância por receber as punições é igual a -1, dado que o risco é aumentado e a motivação do *Grupo M* para ter o risco aumentado é igual a -1. Como resultado, a influência normativa da *Norm 4* sobre o objetivo "*atacar(insurgentes)*" é igual 7.

Então, a influência normativa das normas sobre o *Plano 1* é igual 4, logo, a importância do *Plano 1* é igual 37.

$$\begin{aligned}
 planImportance(plan1, norms) &= triggerPriority(plan1.inv, norms) + \\
 &\quad mainImportance(plan1.body, norms) + \\
 &\quad bodyNormativeInfluence(plan1.body, norms) = \\
 &= triggerPriority(!escoltar(grupoh), norms) + \\
 &\quad mainImportance(\langle use(soldados); use(helicopters); \\
 &\quad !atacar(insurgentes) \rangle, norms) + \\
 &\quad punishmentsImportance(norm6, norms) + \\
 &\quad rewardsImportance(norm4, norms) - \\
 &\quad punishmentsImportance(norm4, norms) = \\
 &= 19 + 14 + (-3) + (7) = 37
 \end{aligned}$$

(Plano 2) Como apresentado na Seção 6.1.1, a condição de invocação do *Plano 2* é o objetivo "*escoltar(grupoh)*" cuja prioridade para atingir é igual a 19, o corpo do plano 2 é composto pelas ações "*use(soldados)*" e "*use(helicopterosterrestres)*", e um objetivo "*assistir(grupoh)*". A satisfação para realizar cada uma destas ações é igual 2 e a motivação para atingir o objetivo "*assistir(grupoh)*" é igual a 8 (como descrito na Seção 6.1.1). Sendo assim, a importância principal do *Plano 2* é igual a 12. Agora, é necessário levar em consideração que o *Plano 2* sofre uma influência normativa da Normas 3.

(Influência Normativa da Norm 3) A *Norma 3* obriga o atingimento do objetivo "*assistir(grupoh)*", se tal norma é cumprida a importância por receber as recompensas é igual 1, dado que se *Norma 3* é cumprida o risco será diminuído, cuja motivação do *Grupo M* é igual a 1. Entretanto, se tal norma é violada a importância por receber as punições é igual a -1, dado que o risco é aumentado e a motivação do *Grupo M* para ter o risco aumentado é igual a -1. Como resultado, a influência normativa da *Norm 3* sobre o objetivo "*assistir(grupoh)*" é igual 2.

Então, a influência normativa das normas sobre o *Plano 2* é igual 2, logo, a importância do *Plano 2* é igual 33.

$$\begin{aligned}
 planImportance(plan1, norms) &= triggerPriority(plan1.inv, norms) + \\
 &\quad mainImportance(plan1.body, norms) + \\
 &\quad bodyNormativeInfluence(plan1.body, norms) = \\
 &= triggerPriority(!escoltar(grupoh), norms) + \\
 &\quad mainImportance(\langle use(soldados); use(helicoptersterrestres); \\
 &\quad !assistir(grupoh) \rangle, norms) + \\
 &\quad rewardsImportance(norm3, norms) - \\
 &\quad punishmentsImportance(norm3, norms) = \\
 &= 19 + 12 + (1) - (-1) = 33
 \end{aligned}$$

Por fim, o Algoritmo 14, apresentado na Seção 5.2.5, que realiza a seleção de planos, é executado e o *Plan 1* é selecionado para se tornar uma intenção, já que a sua importância, igual a 37, é maior do que a importância do *Plan 2*, igual a 33.

6.1.11

Revisão de Intenções

Embora o agente tenha selecionado um plano para se tornar intenção, também é necessário que ele revise as intenções já existentes. O primeiro passo é avaliar cada intenção a partir da execução do Algoritmo 15, apresentado na Seção 5.2.6. Por exemplo, considere que o *Grupo M* tem uma intenção, nomeada *Intenção*, composta pelo Plano 4 que tem uma condição de invocação que é acionada em resposta a eventos cujo *trigger* está relacionado a adição de um objetivo representando o desejo do *Grupo M* em atacar insurgentes. Seu contexto é verdadeiro se *Norma 4*, que obriga *Grupo M* a atacar insurgentes, já foi cumprida. Seu corpo executa ações para utilizar soldados e helicópteros.

Plano 4:

```

+ !atacar(insurgentes)
: nc(FULFILLED, norm4(—, —, —, —, —, —, —))
< -action use(soldados);
   action use(helicopteros).

```

A fim de avaliar a importância de tal intenção é somente necessário avaliar a importância do *Plano 4*, que é avaliada como segue:

(**Plano 4**) Como apresentado na Seção 6.1.1, a condição de invocação do *Plano 4* é o objetivo "!atacar(insurgentes)" cuja prioridade para atingir é

igual a 17 (como descrito na seção 6.1.8), o corpo do plano 1 é composto pelas ações "use(soldados)" e "use(helicopteros)". A satisfação para realizar cada uma destas ações é igual 2 (como descrito na Seção 6.1.1). Sendo assim, a importância principal do *Plano 4* é igual a 4. Agora, é necessário levar em consideração que o *Plano 4* sofre uma influência normativa da Norma 6 que proíbe a execução da ação "use(helicopteros)", cuja influência normativa é igual a -3, como visto na seção 6.1.10. Então, a influência normativa das normas sobre o *Plano 4* é igual -3, logo, a importância do *Plano 1* é igual 18.

$$\begin{aligned} planImportance(plan4, norms) &= triggerPriority(plan4.inv, norms) + \\ &\quad mainImportance(plan4.body, norms) + \\ &\quad \quad bodyNormativeInfluence(plan4.body, norms) = \\ &= triggerPriority(! atacar(insurgentes), norms) + \\ &\quad mainImportance(\langle use(soldados); use(helicopters) \rangle, norms) + \\ &\quad \quad punishmentsImportance(norm6, norms) = \\ &= 17 + 4 + (-3) = 18 \end{aligned}$$

Agora podemos exemplificar como uma revisão de intenções pode ser realizada, como descrito no Algoritmo 16 apresentado na Seção 5.2.6, observando que a *Intenção* também pode ser atingida pelo *Plano 3*, cuja importância é avaliada como segue:

(Plano 3) Como apresentado na Seção 6.1, a condição de invocação do *Plano 3* é o objetivo "atacar(insurgentes)" cuja prioridade para atingir é igual a 17 (como descrito na seção 6.1.8), o corpo do plano 3 é composto pelas ações "use(soldados)" e "use(jipes)". A satisfação para realizar cada uma destas ações é igual 2. Sendo assim, a importância principal do *Plano 3* é igual a 4. O *Plano 3* não sofre influência de nenhuma norma, logo a influência normativa das normas sobre o *Plano 3* é igual 0, logo, a importância do *Plano 3* é igual 21.

$$\begin{aligned} planImportance(plan4, norms) &= triggerPriority(plan4.inv, norms) + \\ &\quad mainImportance(plan4.body, norms) + \\ &\quad \quad bodyNormativeInfluence(plan4.body, norms) = \\ &= triggerPriority(! atacar(insurgentes), norms) + \\ &\quad mainImportance(\langle use(soldados); use(jipes) \rangle, norms) = \\ &= 17 + 4 = 21 \end{aligned}$$

Desta forma, dado que a *Intenção* tem seu *status* igual a *active* e a importância do *Plano 3* é maior do que a importância do *Plano 4*, a *Intenção* tem seu plano substituído pelo *Plano 3*.

A fim de exemplificar o caso onde uma nova intenção é adicionada a base de intenções, lembre da Seção 6.1.10, que *Plano 1* foi escolhido para se tornar uma intenção, portanto, uma nova intenção composta do *Plano 1* é criada, o seu *status* é definido como *active* e ele é adicionado ao conjunto de intenções. Tal como descrito no Algoritmo 16, apresentado na Seção 5.2.6.

6.1.12

Execução e Monitoramento de Intenções

Após revisar o conjunto de intenções, uma delas é selecionada para ser executada a partir da execução do Algoritmo 17 apresentado na Seção 5.2.7. Neste caso, lembre-se que na base de intenções do agente existem duas intenções. A primeira composta pelo plano 1, cuja motivação é igual a 37 (ver Seção 6.1.10) e a segunda composta pelo plano 3 cuja motivação é igual a 21 (ver Seção 6.1.11). Sendo assim, a primeira intenção é selecionada para ser executada.

Dado que o Plano 1 é composto pelas ações "use(soldados)", "use(helicopteros)" e pelo objetivo "atacar(insurgentes)", e que a intenção selecionada para ser executada é composta pelo *Plano 1*, durante a execução do plano quando a ação "use(helicopteros)" é executada o comportamento regulado pela *Norm 6* é realizado, então a relação *behaviorrealization* de tal norma é atualizada para "REALIZED". Quando o objetivo "atacar(insurgentes)" é atingido o comportamento regulado pela *Norma 4* é realizado, então a relação *behaviorrealization* de tal norma é atualizada para "REALIZED". Este processo é operacionalizado pelos Algoritmos 18 e 19 apresentado na Seção 5.2.8 responsáveis pela execução e monitoramento de uma intenção.

6.2

Cenário 2: Desenvolvimento e Evolução de Software Suportado por Agentes Regulados por Normas

Programação eXtrema (do inglês *eXtreme Programming - XP*) é uma metodologia de desenvolvimento de software que defende a ideia de que freqüentes *releases* devem ser disponibilizadas em curtos ciclos de desenvolvimento para que haja uma melhoraria na produtividade da equipe e novos requisitos do cliente possam ser adotados (Back and Andres 2004). Um grupo de desenvolvedores que adota XP tem o objetivo comum de melhorar a qualidade do software e a capacidade de resposta à mudanças nos requisitos do cliente.

Para atingir tal objetivo durante todo o projeto, o software deve ser bem projetado e, para este propósito, *XP* utiliza um processo de contínuo melhoramento do software chamado *Refatoramento* (Fowler et al. 1999). *Re-*

fatoramento é o processo de modificar o software de forma que não altere o comportamento externo do código e ainda melhore sua estrutura interna.

Embora a atividade de refatoramento seja fortemente recomendada, é necessário levar em consideração alguns fatores:

- O principal objetivo de cada desenvolvedor é a construção do sistema, principalmente se o prazo para liberação de novas *releases* esta curto, então é necessário analisar o momento adequado para realizar atividades de refatoramento;
- Atividades de refatoramento precisam ser realizada cuidadosamente dado que se feita de maneira inadequada, novas falhas a estrutura interna do sistema podem ser adicionadas, por exemplo, um problema comum é a criação de uma *classe grande* ao aplicar o padrão de projeto *fachada*³. Como discutido em (Fowler et al. 1999), uma classe é considerada *grande* se ela contém muitas responsabilidades e (Fowler et al. 1999) defende que a existência de *classes grandes* no sistema diminuem a coesão do mesmo.

Neste contexto, a fim de regular o comportamento de desenvolvedores um conjunto de normas pode ser definido a fim de influenciá-los em relação:

- As atividades que eles devem dar maior prioridade;
- As falhas no projeto do software que eles deveriam evitar ou solucionar. Neste caso, pode-se utilizar como base (Fowler et al. 1999) que descreve falhas que devem ser evitadas num projeto de software;
- Quais padrões eles deveria aplicar (ou evitar) dado o contexto atual. Neste caso, pode-se utilizar como base (Gamma et al. 94) que descreve situações onde determinados padrões de projetos devem (ou não devem) ser aplicados a fim de solucionar falhas no projeto do software e apresenta os ganhos e perdas inerentes a aplicação de cada padrão.

Adicionalmente, um conjunto de recompensas e punições podem ser definidas a fim de tornar claro para os desenvolvedores os ganhos e perdas por cumprir ou violar cada norma. Tais recompensas e punições podem explicitar, por exemplo, os ganhos ou perdas no acoplamento ou coesão do sistema, ou até mesmo a reputação do desenvolvedor.

A fim de exemplificar tais conceitos, considere um cenário onde temos uma empresa de desenvolvimento de software que adota a metodologia XP e

³Tal padrão fornece uma interface unificada para um conjunto de interfaces no sistema e ele é utilizado principalmente para resolver problemas de alto acoplamento no sistema (Gamma et al. 94)

atualmente está responsável por implementar um sistema para ajudar a população em situações de emergência, tais como desastres ambientais. Adicionalmente, considere que a empresa tem um time de desenvolvedores (nomeado *Time A*) com o objetivo principal de implementar uma funcionalidade para notificar a população sobre áreas de risco.

Embora o principal objetivo do *Time A* seja implementar tal funcionalidade, eles podem dedicar tempo realizando atividades de melhoramento do código, mesmo que não haja problemas críticos, ou ainda solucionando falhas na estrutura do software utilizando determinados padrões de projeto. Exemplos de tais falhas de projeto são *creation sprawl* (Kerievsky 2004), esse problema ocorre quando os dados e código utilizados para instanciar uma classe estão espalhados por várias classes, e *alto acoplamento* (Kerievsky 2004), esse problema ocorre quando temos uma classe com muitas responsabilidades.

Para solucionar tais falhas de projeto, a companhia adota as soluções discutidas em (Kerievsky 2004). Onde problemas de *auto acoplamento* são solucionados através da aplicação do padrão *fachada* (Gamma et al. 94). Problemas do tipo *creation sprawl* são solucionados a partir da aplicação do padrão *fábrica* (Gamma et al. 94), que fornece uma interface para a criação de uma família de objetos relacionados ou dependentes sem especificar suas classes concretas. No entanto, como discutido em (Gamma et al. 94), existem situações onde a aplicação do padrão *fábrica* pode aumentar a complexidade do projeto desnecessariamente, por exemplo, aumentando, consideravelmente, o número de componentes do sistema. Neste caso, a empresa não recomenda o uso de tal padrão.

Além dos objetivos apresentados acima (isto é, a implementação de novas funcionalidades, solucionar falhas no projeto do software e realizar atividades de melhoramento do software, mesmo não existindo falhas no projeto) *Time A* possui outros objetivos, tais como:

- Aumentar ou manter sua reputação;
- Diminuir o acoplamento, espalhamento e a complexidade do sistema, ou ainda aumentar ou manter a coesão do sistema a fim de facilitar a realização de atividades de evolução e manutenibilidade do software; e
- Neste cenário, consideramos que cada tarefa atribuída ao *Time A* têm associada um nível de importância, quanto maior a importância de uma tarefa mais recursos (tais como, disponibilização de mais desenvolvedores para o time) podem ser disponibilizados para o time após a realização de tal tarefa. Um exemplo de uma tarefa de grande importância é a implementação da funcionalidade para notificar a população ou a

implementação da funcionalidade de autenticação, e um exemplo de uma tarefa de baixa importância é a implementação de funcionalidades relacionadas com a criação de objectos *mock* para realizar testes.

A fim de regular o comportamento do *Time A* um conjunto de normas é definido pela empresa a fim de influenciar os desenvolvedores na realização de suas atividades, algumas destas normas são descritas abaixo:

(Norma 1): Se um problema do tipo *creation sprawl* é detectado, o *Time A* é obrigado a solucionar tal problema aplicando o padrão *fábrica*.

(Recompensas): Se o desenvolvedor soluciona tal problema utilizando o padrão fábrica:

(Recompensa 1.1): Ele é premiado com uma diminuição no espalhamento de código e dados

(Punições): Se o desenvolvedor não soluciona tal problema utilizando o padrão fábrica:

(Punição 1.2): Sua reputação é diminuída dado que ele não está contribuindo para a evolução do software;

(Norma 2): Como discutido em (Fowler et al. 1999), existem situações onde a implementação do padrão fábrica pode aumentar, desnecessariamente, a complexidade do projeto, por exemplo, aumentando o número de componentes do sistema. Neste caso, o *Time A* é proibido de implementar o padrão fábrica.

(Punições): Se o desenvolvedor aplica o padrão fábrica em tais situações:

(Punição 2.1): Ele é punido com um aumento na complexidade do projeto;

(Punição 2.2): Sua reputação é diminuída.

(Norma 3): Se um problema de *auto acoplamento* é detectado, o *Time A* é obrigado a resolver tal problema aplicando o padrão *fachada*.

(Recompensas): Se o desenvolvedor soluciona tal problema utilizando o padrão fachada:

(Recompensa 3.1): Ele é premiado com uma diminuição no acoplamento.

(Recompensa 3.2): Ele é obrigado a implementar a funcionalidade de autenticação;

(Punições): Se o desenvolvedor não soluciona tal problema utilizando o padrão fachada:

(Punição 3.3): Sua reputação é diminuída

(Norma 4): Como discutido em (Fowler et al. 1999), a criação de uma fachada *grande* diminui a coesão do sistema. Então, o *Time A* está proibido de criar uma fachada *grande*.

(Punições): Se o desenvolvedor cria a fachada:

(Punição 4.1): Ele é punido com uma diminuição na coesão;

(Punição 4.2): Sua reputação é diminuída.

(Norma 5): Se o prazo para entrega do produto está curto, *Time A* é proibido de realizar qualquer atividade relacionada ao melhoramento do código.

(Recompensas): Se o time de desenvolvedores não realiza tais tipos de atividades:

(Recompensa 5.1): Ele é premiado com um aumento na sua reputação.

(Punições): Se o time de desenvolvedores realiza tais tipos de atividades:

(Punição 5.2) Sua reputação é diminuída.

(Punição 5.3) Ele é obrigado a implementar objetos *mock*

(Punições:) Se tais objetos não são implementados até um certo período;

(Punição 5.3.1): O time de desenvolvedores perde parte de seus desenvolvedores;

Dado que a construção de um sistema de software já é uma tarefa complexa, a mesma se tornar ainda mais difícil, na presença de normas, como estas supracitadas. Abaixo apresentaremos como ANA podem ajudar *Time A* na tomada de decisões considerando a importância para realizar as atividades do *Time A* e as normas do sistema. Tais decisões envolvem as funcionalidades que eles devem dar maior prioridades para implementar; as falhas de projeto que devem solucionar primeiro ou as que devem evitar, e, quais padrões eles devem aplicar ou evitar.

Assumiremos que ANA tem uma biblioteca de planos capazes de guiar o desenvolvedor ao lidar com diferentes situações inerentes ao cenário descrito acima, exemplos de planos são apresentados na Seção 6.2.1.

6.2.1

Representação de Objetivos, Ações, Planos e Normas

Nesta seção descreveremos como objetivos, ações, planos disponíveis para o *Time A* e as normas endereçadas ao *Time A* podem ser representadas utilizando o *AgentSpeak(L) Normativo*.

Como descrito anteriormente, o *Time A* possui um conjunto de objetivos a serem atingidos, tais como, a aplicação de padrões e a implementação de funcionalidades. Assumindo que padrões podem ser aplicados e funcionalidades implementadas através da execução de ações, abaixo apresentamos como tais objetivos e ações podem ser descritas, como também uma possível configuração das motivações e satisfações⁴ do *Time A* em atingir tais objetivos e realizar tais ações.

(Implementação de Funcionalidades): O *Time A* possui objetivos de implementar uma variedade de funcionalidades do sistema, abaixo descrevemos como alguns destes objetivos podem ser representados:

(Notificação): Implementação da funcionalidade para notificar a população sobre áreas de risco:

!implementar(notify_population_risk_areas)

Dado que este é o objetivo principal do *Time A*, assumiremos uma motivação igual a 10 para ele.

(Autenticação): Implementar a funcionalidade de autenticação do sistema:

!implementar(authenticacao)

Embora a implementação da funcionalidade de autenticação do sistema não seja o principal objetivo do *Time A*, implementar tal funcionalidade é uma tarefa de alta prioridade. Desta forma, definiremos uma motivação igual a 6 para tal objetivo.

(Criação de Objetos Mock): Implementar objetos *mock* para realização de testes:

⁴neste artigo, o valor de uma *importância* varia de -10 a 10

!implementar(objetos_mock)

O objetivo de criar objetos *mock* para teste é uma tarefa de baixa importância. Assumiremos uma motivação igual a -2 para a realização de tal tarefa.

(*Solucionar Falhas no Projeto*): Além de implementar funcionalidades, o *Time A* também possui o objetivo de solucionar problemas relacionados a uma lista de componentes utilizando um determinado padrão, tal objetivo é representado como descrito abaixo:

!solucionar(Problem, Components, Pattern)

Embora o objetivo de solucionar problemas seja importante para facilitar atividades de evolução e manutenibilidade do software. Este não é o principal objetivo do *Time A*. Desta forma, assumiremos uma motivação igual a 6 para realizar tal objetivo.

(*Melhorar Software*): Em alguns momentos não existe um problema crítico associado a uma lista de componentes mas, a aplicação de um determinado padrão pode melhorar a estrutura de tais componentes, objetivos relacionados ao melhoramento de componentes utilizando um determinado padrão são representados como descrito abaixo:

!melhorar(Components, Pattern)

Embora o objetivo de melhorar a estrutura do software seja importante, pois, pode facilitar o entendimento da estrutura, tais objetivos podem ser desnecessários dado que não existe um problema explícito assim como objetivo o *!solucionar(Problem, Components, Pattern)*. Desta forma, assumiremos uma motivação igual a 5 para tal objetivo.

(*Reputação, Coesão, Espalhamento, Acoplamento e Complexidade*): Objetivos relacionados ao aumento ou diminuição da reputação do *Time A*, coesão, acoplamento, espalhamento ou complexidade do sistema são representados como descrito abaixo:

!aumento(Fator)

!diminuicao(Fator)

Como discutido na seção , *Time A* está interessado no aumento da sua reputação e na coesão do sistema, e na diminuição do espalhamento, acoplamento e complexidade do sistema. Dado que tais fatores podem variar frequentemente, assumiremos que eles variam com um aumento de 1 ou uma diminuição de -1. Desta forma, podemos definir que a motivação do *Time A* para ter sua reputação ou coesão do sistema aumentada, ou o espalhamento, acoplamento ou complexidade do sistema diminuída é igual a 1. Caso contrário, assumiremos que a motivação é igual a -1.

(Recursos): O objetivo do *Time A* para obter mais desenvolvedores é representado por *!disponibilizar(empresa, desenvolvedores, timea)*, e o objetivo relacionado a remoção de parte dos desenvolvedores do *Time A* é representado por *!retirar(timea, desenvolvedores)*. Dado que obter mais desenvolvedores é tão importante quanto a não perdê-los, assumiremos que a motivação do *Time A* para a obtenção de desenvolvedores é igual a 5 e para a perda de desenvolvedores é igual a -5;

(Criação de Componentes): A fim de realizar a criação de componentes no sistema para atingir os objetivos, utilizaremos a ação *criarComponentes(Description, Components)*, que recebe uma descrição do que deve ser feito a partir de alguns componentes. Por exemplo, a criação de componentes para aplicar o padrão *fachada* com muitas responsabilidades a partir de um conjunto de componentes é realizada pela ação *criarComponentes(fachada_grande, componentes)*, neste caso a fachada é do tipo *fachada_grande* e deve ser implementada a partir dos *componentes*). Um outro exemplo, é a criação de componentes para implementar a funcionalidade de notificação utilizando o padrão *observador* a partir de um conjunto de componentes é realizada pela ação *criarComponentes(notificacao_observador, componentes)*. Embora a execução de tal ação resulte num aumento de componentes no sistema, sem a sua realização não seria possível aplicar tais padrões. Desta forma, assumiremos uma satisfação igual a 1 para executar tal ação.

A fim de guiar os desenvolvedores no atingimento dos objetivos supracitados, o *Time A* possui a sua disposição uma biblioteca de planos.

Por exemplo, considere os planos descritos abaixo, os planos 1 e 2 têm a mesma *condição de invocação*, ambos são considerados relevantes se existe um problema de *auto acoplamento* a ser resolvido, envolvendo um conjunto de componentes, utilizando o padrão de projeto *fachada*. O contexto do plano 1 é satisfeito se existe um problema de *espalhamento* relacionado a um conjunto de componentes e a *Norma 1* está ativada. Já o contexto do plano 2 é sempre

verdadeiro. O corpo do Plano 1 possui uma ação para criar uma fachada *grande* e um objetivo para solucionar um problema de espalhamento relacionado a um conjunto de componentes utilizando padrão fábrica. O corpo do Plano 2 possui uma ação para criar uma fachada *grande*.

Plano 1:

```
+ !solving(auto_acoplamento, Componentes, fachada)))
: problem(espalhamento, Comp)&
  nc(ACTIVATED, norm1(—, —, —, —, —, —, —))
< — criarComponentes(fachada_grande, Componentes);
  !solucionar(espalhamento, Comp, fabrica).
```

Plano 2:

```
+ !solucionar(auto_acoplamento, Components, fachada)
: true
< — criarComponentes(fachada_grande, Componentes).
```

Outros exemplos, são os planos 3 e 4 que possuem a mesma *condição de invocação*, ambos são considerados relevantes se o *Time A* desaja implementar a funcionalidade para notificar a população sobre áreas de risco. O contexto do *Plano 3* é satisfeito se a funcionalidade de notificação pode ser implementada a partir da aplicação do padrão *observador* utilizando um conjunto de componentes. Já o contexto do plano 4 é satisfeito se a funcionalidade de notificação pode ser implementada a partir da aplicação do padrão *observador* utilizando um conjunto de componentes e existe componentes que podem ser melhorados utilizando um determinado padrão. O corpo do *Plano 3* possui uma ação para criar um *observador* para a funcionalidade de notificação a partir de um conjunto de componentes e um objetivo para implementar objetos *mock*. O corpo do *Plano 4* possui uma ação para criar um *observador* para a funcionalidade de notificação a partir de um conjunto de componentes, um objetivo para melhorar alguns componentes utilizando um determinado padrão e implementar objetos *mock*.

Plano 3:

```
+ !!implementar(notify_population_risk_areas)
: componentes(notify_population_risk_areas, observador, Components)
< — criarComponentes(notificacao_observador, Components);
  !implementar(objetos_mock).
```

Plano 4:

```

+ !implementar(notify_population_risk_areas)
: componentes(notify_population_risk_areas, observador, Components)&
  existcompmelhorados(Comp, Pattern)
< - criarComponentes(notificacao_observador, Components);
  !melhorar(Comp, Pattern);
  !implementar(objetos_mock).

```

Como descrito no cenário apresentado na Seção 6.2, o *Time A* é regulado por um conjunto de normas, abaixo apresentamos como a *Norma 1* é representada utilizando o AgentSpeak(L) Normativo:

(Norm 1): "Se um problema do tipo *creation sprawl* é detectado, o *Time A* é obrigado a solucionar tal problema aplicando o padrão fábrica." A partir desta frase, podemos deduzir os seguintes componentes da *Norma 1*:

(idenfier): Conidere que o identificador da norma é *norm1*;

norm1(

(addressees): Ela é destinada ao time de desenvolvedores *teama*;

norm1(*timea*,

(deontic_concept): É uma obrigação

norm1(*team*, *OBLIGATION*,

(activation): Será ativada se existe um problem do tipo *creation sprawl* relacionado a um conjunto de componentes:

norm1(*team*, *OBLIGATION*,
problem(*creation_sprawl*, *Componentes*),

(deactivation): Será desativada se o problema do tipo *creation sprawl* não existe mais

norm1(*team*, *OBLIGATION*,
problem(*creation_sprawl*, *Componentes*),
not problem(*creation_sprawl*, *Components*),

(behavior): Está regulando o objetivo

!solucionar(creation_sprawl, Componentes, fabrica), então

norm1(team, OBLIGATION,
problem(creation_sprawl, Componentes),
not problem(creation_sprawl, Componentes),
!solucionar(creation_sprawl, Componentes, fabrica),

(rewards): Se o desenvolvedor soluciona tal problema utilizando o padrão fábrica, ele é premiado com uma diminuição no espalhamento.

norm1(team, OBLIGATION,
problem(creation_sprawl, Componentes),
not problem(creation_sprawl, Componentes),
!solucionar(creation_sprawl, Componentes, fabrica),
!diminuicao(espalhamento),

(punishments): Se o desenvolvedor não soluciona tal problema utilizando o padrão fábrica, ele é punido com uma diminuição na sua reputação.

norm1(team, OBLIGATION,
problem(creation_sprawl, Componentes),
not problem(creation_sprawl, Componentes),
!solucionar(creation_sprawl, Componentes, fabrica),
!diminuicao(espalhamento),
!diminuicao(reputation))

6.2.2

Adotando Novas Normas

Após descrever os principais objetivos, ações, planos e normas do *Time A*. Iniciaremos a descrição do processo executado pelo agente de suporte do *Time A*.

O primeiro passo é verificar a adoção das normas. Para tanto, considere as normas apresentadas na Seção 6.2, se o *Time A* recebe informações sobre as normas do sistema, o Algoritmo 1 apresentado na Seção 5.2.1, é executado para checar quais das normas já estão armazenadas no conjunto de normas adotadas pelo *Time A* e comparar o campo *addressees* de cada norma sensorizada com *Time A*. Neste caso, todas as normas descritas no cenário são endereçadas ao

Time A e as mesmas são adicionadas ao conjunto de normas adotadas pelo *Time A*.

6.2.3

Revisando as Normas Ativadas e Desativadas

Após realizar a adoção de normas, verificaremos quais normas foram ativadas ou desativadas utilizando o Algoritmo 2 apresentado na Seção 5.2.1. Por exemplo, considere a seguinte situação: Um problema de *auto acoplamento* é detectado, o contexto de ativação da *Norma 3* é satisfeito, ela é adicionada ao conjunto de normas ativadas e *Time A* é obrigado a implementar o padrão fachada para solucionar tal problema. Dado que *Norma 3* é ativada, o contexto de ativação da *Norma 4* também é satisfeito dado que a condição é que a *Norma 3* seja ativada. Portanto, a *Norma 4* também é adicionada ao conjunto de normas ativadas e *Time A* é proibido de criar fachadas grandes.

Se o problema de *auto acoplamento* não existe mais, então *Norma 3* é desativada e *Norma 4* também tem seu contexto de desativação satisfeito considerando que *Norma 3* esta inativa. Ambas as normas são removidas do conjunto de normas ativadas e adicionadas ao conjunto de normas desativadas.

6.2.4

Verificando as Normas Cumpridas ou Violadas

Se durante a ativação da *Norma 3*, *Time A* solucionou o problema de auto acoplamento aplicando o padrão fachada, então o comportamento regulado pela *Norma 3* foi realizado, *Time A* cumpriu com a norma e a mesma tem seu *status* de cumprimento atualizado para *FULFILLED*. Entretanto, se *Time A* não solucionou tal problema aplicando o padrão fachada, *Norma 3* foi violada e a mesma tem seu *status* de cumprimento atualizado para *VIOLATED*.

A verificação de quais normas foram cumpridas ou violadas é realizada a partir da execução do Algoritmo 3 apresentado na Seção 5.2.1.

6.2.5

Detectando e Superando Conflitos entre Normas

Se as normas 1 e 2 são ativadas, elas estão em conflito dado que *Norma 1* obriga *Time A* a implementar o padrão fábrica enquanto a *Norma 2* proíbe a aplicação do padrão. Neste caso, o Algoritmo 4 apresentado na Seção 5.2.2 detecta e supera tal conflito verificando se a influência deontica da *Norma 1* (*ID1*) que é igual a 6 (dado que *Norma 1* está obrigando *Time A* a solucionar um problema relacionado a um conjunto de componentes e a motivação do

Time A para solucionar problemas é igual a 6) mais a importância das recompensas por cumprir *Norma 1* (*IR1*) que é igual a 1 (dado que se *Time A* cumprir a *Norma 1* ele terá a taxa de espalhamento diminuída e a motivação do *Time A* para uma diminuição no espalhamento é igual 1) e a importância das punições por violar *Norma 2* (*IP2*) que é -1 (dado que se *Time A* violar *Norma 2* ele terá sua reputação diminuída e a motivação do *Time A* para uma diminuição da reputação é igual -1) é maior do que a importância por receber as recompensas por cumprir *Norma 2* (*IR2*) que é 0 (dado que *Norma 2* não fornece premiação pelo seu cumprimento) e receber as punições por violar *Norma 1* (*IP1*) que é -1 (se *norm 1* é violada o *Team A* terá sua reputação diminuída cujo motivação é -1). Finalmente, considerando que:

$$\begin{aligned} ID1 + IR1 + IP2 &> IR2 + IP1 \\ 6 + 1 + (-1) &> 0 + (-1) \end{aligned}$$

Então, *Norma 1* é selecionada para ser cumprida e *Norm 2* para ser violada.

6.2.6

Selecionando Normas para serem Cumpridas ou Violadas

Após detectar e superar conflitos entre normas, é necessário escolher quais normas não conflitantes devem ser cumpridas ou violadas. Por exemplo, considere a situação onde *Norma 3* é ativada e *Time A* é obrigado a aplicar o padrão fachada. A influência deontica da *Norma 3* é 5 (dado que *Norma 3* está obrigando *Time A* a solucionar um problema e a motivação do *Time A* para atingir tal objetivo é 5). A importância por receber as recompensas da *Norma 3* é 7 (dado que se *Norma 3* é cumprida *Time A* será premiado com uma diminuição no acoplamento cuja motivação do *Time A* é 1 e *Time A* é obrigado a implementar a funcionalidade de autenticação do sistema cujo motivação do *Time A* é 6). A importância por receber as punições da *Norma 3* é -1 (dado que se *Norma 3* é violada a reputação do *Time A* é diminuída e a motivação do *Time A* para ter a reputação diminuída é igual a -1). Então *Norma 3* é selecionada para ser cumprida dado que a soma da sua influência deontica, que é 5, mais a importância por receber as recompensas, que é 7, é maior do que a importância por receber as punições, que é -1.

Agora, considere a situação onde *Norma 5* é ativada e *Time A* é proibido de realizar atividades relacionadas a melhoramento no código. A influência deontica da *Norma 5* é -5 (dado que *Norma 5* está proibindo *Time A* a realizar melhoramentos e a motivação do *Time A* para atingir tal objetivo

é 5). A importância por receber as recompensas da *Norma 5* é 1 (dado que se *Norma 5* é cumprida *Time A* será premiado com um aumento na sua reputação e a motivação do *Time A* para ter sua reputação aumentada é igual a 1). A importância por receber as punições da *Norma 5* é -3. Neste caso, observe que a importância da *Punição 5.1* é igual a -1 dado que a reputação do *Time A* será diminuída. Para avaliar a importância da *Punição 5.2* é necessário analisar se a influência deontica da *Punição 5.2* mais a importância por receber as recompensas é maior do que a importância por receber as punições estabelecidas pela *Punição 5.2*. Dado que a *Punição 5.2* está obrigando *Time A* a implementar objetos *mock* cuja motivação é igual -2, então sua influência deontica é igual -2, *Punição 5.2* não fornece recompensas e a punição por violar tal norma é a perda de parte dos desenvolvedores cuja motivação do *Time A* é igual a -5. Sendo assim, adotando uma estratégia otimista a importância por receber a *Punição 5.2* é igual -2.

Então *Norma 5* é selecionada para ser violada dado que a soma da sua influência deontica, que é -5, mais a importância por receber as recompensas, que é 1, é maior do que a importância por receber as punições, que é -3.

A escolha de quais normas não conflitantes devem ser cumpridas ou violadas é realizada a partir da execução do Algoritmo 5 apresentado na Seção 5.2.2.

6.2.7

Aplicando Decisões Normativas

Após selecionar as normas que devem ser cumpridas ou violadas, é necessário efetivar tais decisões. Para tanto, lembre que *Norma 1* que obriga *Time A* a solucionar um problema de espalhamento, relacionado a conjunto de componentes, utilizando o padrão fábrica foi selecionada para ser cumprida e a *Norma 2* que proíbe *Time A* de solucionar tais tipos de problema, relacionado a conjunto de componentes, utilizando o padrão fábrica foi selecionada para ser violada. A partir de tal decisão um novo evento cujo *trigger* representa a adição de um objetivo para solucionar um problema de auto acoplamento, relacionado a conjunto de componentes, utilizando o padrão fábrica é adicionado a base de eventos do agente.

O mesmo raciocínio é aplicado a *Norma 3* que obriga *Time A* a solucionar um problema de auto acoplamento, relacionado a conjunto de componentes, utilizando o padrão fachada e foi selecionada para ser cumprida, e a *Norma 5* que proíbe *Time A* a realizar melhorias num conjunto de componentes utilizando um determinado padrão e foi selecionada para ser violada. Ou seja, eventos cujo *trigger* indica a adição dos objetivos regulados por tais normas

são adicionados a base de eventos do agente.

A efetivação de decisões normativas é realizada a partir da execução do Algoritmo 6 apresentado na Seção 5.2.2.

6.2.8

Seleção de Eventos

Agora o agente que fornece suporte ao *Time A* possui um conjunto de eventos na sua base. Especificamente, eventos cujo os *triggers* estão relacionados aos seguintes objetivos: "solucionar problemas de auto acoplamento, relacionados a um conjunto de *componentes*, utilizando o padrão fachada"(representado por `!solucionar(auto_acoplamento, componentes, fachada)`), "solucionar problemas de espalhamento, relacionados a um conjunto de *componentes*, utilizando o padrão fábrica"(representado por `!solucionar(espalhamento, componentes, fabrica)`) e "realizar melhorias no código relacionadas a um conjunto de *componentes* utilizando o padrão fachada"(representado por `!melhorar(componentes, fachada)`). A partir da execução do Algoritmo 7 apresentado na Seção 5.2.3 que avalia a prioridade de um trigger, a prioridade de tais *triggers* é avaliada levando em consideração a prioridade para atingir os objetivos que constituem tais *triggers* como segue:

(`!solucionar(espalhamento, componentes, fabrica)`): Conforme descrito na Seção 6.2.1, a motivação do *Time A* para atingir objetivos do tipo `!solucionar(espalhamento, componentes, fabrica)` é 6. No entanto, tal objetivo é influenciado pelas Normas 1 e 2. Norma 1 é uma obrigação e a realização do objetivo `!solucionar(espalhamento, componentes, fabrica)` implica no cumprimento da Norma 1, isto é, receber as recompensas da Norma 1 cuja importância é igual a 1 e não receber as punições da Norma 1 cuja importância é igual a -1, como descrito anteriormente. Norma 2 é uma proibição e a realização do objetivo `!solucionar(espalhamento, componentes, fabrica)` implica na sua violação, isto é, receber as punições da Norma 2 cuja importância é igual a -2 e não receber as recompensas, cuja importância é igual a 0, dado que Norma 2 não tem recompensas, como descrito anteriormente. Assim, a influência normativa das Normas 1 e 2 sobre o comportamento `!solucionar(espalhamento, componentes, fabrica)` é avaliada da seguinte forma:

$$\begin{aligned}
& \text{behaviorNormativeInfluence}(\text{!solucionar}(\text{espalhamento}, \text{componentes}, \text{fabrica})) = \\
& \text{rewardsImportance}(\text{norm1}) - \\
& \text{punishmentsImportance}(\text{norm1}) + \\
& \text{punishmentsImportance}(\text{norm2}) - \\
& \text{rewardsImportance}(\text{norm2}) = \\
& 1 - (-1) + (-2) - 0 = 0
\end{aligned}$$

Como resultado a prioridade do *trigger* relacionado ao objetivo "*!solucionar(espalhamento, componentes, fabrica)*" é:

$$\begin{aligned}
& \text{triggerPriority}(\text{!solucionar}(\text{espalhamento}, \text{componentes}, \text{fabrica})) = \\
& \text{motivation}(\text{!solucionar}(\text{espalhamento}, \text{componentes}, \text{fabrica})) + \\
& \text{behaviorNormativeInfluence}(\text{!solucionar}(\text{espalhamento}, \text{componentes}, \text{fabrica})) = \\
& 6 + 0 = 6
\end{aligned}$$

(*!solucionar(auto_acoplamento, componentes, fachada)*): Conforme descrito na Seção 6.2.1, a motivação do *Time A* para atingir objetivos do tipo "*!solucionar(auto_acoplamento, componentes, fachada)*" é 6. No entanto, tal objetivo é influenciado pela Norma 3 que obriga a realização do objetivo "*!solucionar(auto_acoplamento, componentes, fachada)*", o atingimento de tal objetivo implica no cumprimento da *Norma 3*, isto é, receber as recompensas da *Norma 3* cuja importância é igual a 7 e não receber as punições da *Norma 3* cuja importância é igual a -1, como descrito anteriormente. Assim, a influência normativa da Norma 3 sobre o comportamento "*!solucionar(auto_acoplamento, componentes, fachada)*" é avaliada da seguinte forma:

$$\begin{aligned}
& \text{behaviorNormativeInfluence}(\text{!solucionar}(\text{auto_acoplamento}, \text{componentes}, \text{fachada})) = \\
& \text{rewardsImportance}(\text{norm3}) - \\
& \text{punishmentsImportance}(\text{norm3}) = \\
& 7 - (-1) = 8
\end{aligned}$$

Como resultado a prioridade do *trigger* relacionado ao objetivo "*!solucionar(auto_acoplamento, componentes, fachada)*" é:

$$\begin{aligned}
& triggerPriority(\\
& !solucionar(auto_acoplamento, componentes, fachada)) = \\
& motivation(\\
& !solucionar(auto_acoplamento, componentes, fachada)) + \\
& behaviorNormativeInfluence(\\
& !solucionar(auto_acoplamento, componentes, fachada)) = \\
& 6 + 8 = 14
\end{aligned}$$

(*!melhorar(componentes, fachada)*): Conforme descrito na Seção 6.2.1, a motivação do *Time A* para atingir objetivos do tipo "*!melhorar(componentes, fachada)*" é 5. No entanto, tal objetivo é influenciado pela Norma 5. Norma 5 é uma proibição e a realização do objetivo "*!melhorar(componentes, fachada)*" implica na violação da Norma 5, isto é, receber as punições da Norma 5 cuja importância é igual a -3 e não receber as recompensas da Norma 5 cuja importância é igual a -1, como descrito anteriormente. Assim, a influência normativa da Norma 5 sobre o comportamento "*!melhorar(componentes, fachada)*" é avaliada da seguinte forma:

$$\begin{aligned}
& behaviorNormativeInfluence(!melhorar(componentes, fachada)) = \\
& punishmentsImportance(norm5) - rewardsImportance(norm5) = \\
& -3 - 1 = -4
\end{aligned}$$

Como resultado a prioridade do *trigger* relacionado ao objetivo "*!melhorar(componentes, fachada)*" é:

$$\begin{aligned}
& triggerPriority(!melhorar(componentes, fachada)) = \\
& motivation(!melhorar(componentes, fachada)) + \\
& behaviorNormativeInfluence(!melhorar(componentes, fachada)) \\
& = 5 - 4 = 1
\end{aligned}$$

Dado que a prioridade do *trigger* relacionado ao objetivo *!solucionar(auto_acoplamento, componentes, fachada)* é maior que a prioridade dos outros *triggers*. Então, o evento composto pelo *trigger* relacionado ao objetivo *!solucionar(auto_acoplamento, componentes, fachada)* é selecionado para ser gerenciado, como descrito no Algoritmo 8 apresentado na Seção 5.2.3 que realiza a seleção de eventos.

6.2.9

Verificando Planos Relevantes e Aplicáveis

A fim de exemplificar o processo de encontrar os planos relevantes e aplicáveis. Vamos voltar ao nosso exemplo dos planos descritos na Seção 6.2 cujas condições de invocação são:

Plano1 :

inv =!solucionar(auto_acoplamento, Componentes, fachada)

Plano2 :

inv =!solucionar(auto_acoplamento, Componentes, fachada)

Plano3 :

inv =!implementar(notify_population_risk_areas)

Plano4 :

inv =!implementar(notify_population_risk_areas)

Lembre da Seção anterior , que o evento cujo *trigger* está relacionado ao objetivo *!solucionar(auto_acoplamento, componentes, fachada)* foi selecionado para ser gerenciado. Com base neste cenário, uma vez que a condição de invocação dos planos 1 e 2 são mais relevantes para lidar com o objetivo selecionado do que as condições de invocação dos planos 3 e 4, então os planos 1 e 2 são selecionados como planos relevantes como resultado da execução do processo *Unify Event* já disponibilizado pela Jason.

A fim de verificar se os planos 1 e 2 são aplicáveis, assuma que as crenças do *Time A* são *problema(auto_acoplamento, comp)*, o que indica que existe um problema de auto acoplamento relacionado aos componentes *comp*. Além disto, relembre que Norma 1 está ativada. Nesse caso, nota-se que os contextos dos Planos 1 e 2 são satisfeitos e os mesmos são retornados como planos aplicáveis como resultado da execução do processo *Check Context* disponibilizado pelo Jason Normativo e operacionalizado pelos Algoritmos 9 e 10 apresentados na Seção 5.2.4.

6.2.10

Seleção de Planos

Dado que os planos 1 e 2 foram considerados aplicáveis, é necessário selecionar um deles para se tornar uma intenção. O primeiro passo é avaliar a importância de cada um dos planos a partir da execução do Algoritmo 13 apresentado na Seção 5.2.5, como segue:

(Plano 1): Como apresentado anteriormente, a condição de invocação do Plano 1 é o objetivo

"!solucionar(auto_acoplamento, componentes, fachada)" cuja prioridade

para atingir é igual a 14 (como avaliado anteriormente), o corpo do plano 1 é composto pela ação "*criarComponentes(fachada_grande, componentes)*", e um objetivo "*!solucionar(espalhamento, componentes, fabrica)*". A satisfação para realizar tal ação é igual 2 e a motivação para atingir o objetivo "*!solucionar(espalhamento, componentes, fabrica)*" é igual a 5 (como descrito na Seção 6.2.1). Sendo assim, a importância principal do Plano 1 é igual a 6. Agora, é necessário levar em consideração que o Plano 1 sofre uma influência normativa das Normas 1, 2 e 4. A influência normativa da *Norma 1* sobre o objetivo "*!solucionar(espalhamento, componentes, fabrica)*" é igual a 2 (como avaliado na Seção 6.2.8), A influência normativa da *Norma 2* sobre tal objetivo é igual -2 (como avaliado na Seção 6.2.8), e, finalmente, a influência normativa da *Norma 4* sobre a ação "*criarComponentes(fachada_grande, componentes)*" é igual a -2 (dado que se a *Norma 4* é violada o sistema terá um aumento no seu acoplamento e a reputação do *Time A* é diminuída). Então, a influência normativa sobre o corpo do *Plano 1* é igual a -2, logo, a importância do *Plano 1* é igual a 19.

(Plano 2): Como apresentado anteriormente, a condição de invocação do Plano 1 é o objetivo

"*!solucionar(auto_acoplamento, componentes, fachada)*" cuja prioridade para atingir é igual a 14 (como avaliado anteriormente), o corpo do plano 1 é composto pela ação "*criarComponentes(fachada_grande, componentes)*". A satisfação para realizar tal ação é igual 2 (como descrito na Seção 6.2.1). Sendo assim, a importância principal do Plano 2 é igual a 2. Agora, é necessário levar em consideração que o Plano 2 sofre uma influência normativa da Norma 4. A influência normativa da *Norma 4* sobre a ação "*criarComponentes(fachada_grande, componentes)*" é igual a -2 (como avaliado anteriormente). Então, a influência normativa sobre o corpo do *Plano 1* é igual a -2, logo, a importância do *Plano 1* é igual a 14.

Por fim, o Algoritmo 14, apresentado na Seção 5.2.5, que realiza a seleção de planos, é executado e o *Plano 1* é selecionado para se tornar uma intenção, já que a sua importância, igual a 19, é maior do que a importância do *Plano 2*, igual a 14.

6.2.11

Revisão de Intenções

Embora o agente tenha selecionado um plano para se tornar intenção, também é necessário que ele revise as intenções já existentes. O primeiro passo é avaliar cada intenção a partir da execução do Algoritmo 15, apresentado na Seção 5.2.6. Por exemplo, considere que o *Time A* contém na sua base de intenções uma intenção, nomeada *Intenção*, composta pelo *Plano 4*. Cujas importância é avaliada como segue:

(Plano 4): Como apresentado anteriormente, a condição de invocação do Plano 4 é o objetivo "*implementar(notify_population_risk_areas)*" cuja prioridade para atingir é igual a 10 (como avaliado anteriormente), o corpo do plano 4 é composto pela ação "*criarComponentes(notificacao_observador, componentes)*", um objetivo "*melhorar(comp, fabrica)*" e um outro "*implementar(objetos_mock)*". A satisfação para realizar tal ação é igual 2 (como descrito na Seção), a motivação para atingir o objetivo "*melhorar(comp, fabrica)*" é igual a 4 e "*implementar(objetos_mock)*" é igual a -2. (como descrito na Seção 6.2.1). Sendo assim, a importância principal do Plano 4 é igual a 4. Agora, é necessário levar em consideração que o Plano 4 sofre uma influência normativa da Norma 5. A influência normativa da *Norma 5* sobre o objetivo "*melhorar(comp, fabrica)*" é igual -4 (como avaliado anteriormente). Então, a influência normativa sobre o corpo do *Plano 4* é igual a -4, logo, a importância do *Plano 4* é igual a 10.

Agora podemos exemplificar como uma revisão de intenções pode ser realizada, como descrito no Algoritmo 16 apresentado na Seção 5.2.6, observando que a *Intenção* definida anteriormente também pode ser atingida pelo *Plano 3*, cuja importância é avaliada como segue:

(Plano 3): Como apresentado anteriormente, a condição de invocação do Plano 3 é o objetivo "*implementar(notify_population_risk_areas)*" cuja prioridade para atingir é igual a 10 (como avaliado anteriormente), o corpo do plano 3 é composto pela ação "*criarComponentes(notificacao_observador, componentes)*". Sendo assim, a importância principal do Plano 3 é igual a 1. O corpo do *Plano 3* não sofre influência de nenhuma norma. Então, a influência normativa sobre o corpo do *Plano 3* é igual a 0, logo, a importância do *Plano 3* é igual a 11.

Desta forma, dado que a *Intenção* tem seu *status* igual a *active* e a importância do Plano 3 é maior do que a importância do Plano 4, a *Intenção* tem seu plano substituído pelo Plano 3.

A fim de exemplificar o caso onde uma nova intenção é adicionada a base de intenções, lembre da Seção 6.2.10, que *Plano 1* foi escolhido para se tornar uma intenção, portanto, uma nova intenção composta pelo *Plano 1* é criada, o

seu *status* é definido como *active* e ela é adicionada ao conjunto de intenções. Tal como descrito no Algoritmo 16, apresentado na Seção 5.2.6.

6.2.12

Execução e Monitoramento de Intenções

Após revisar o conjunto de intenções, uma delas é selecionada para ser executada a partir da execução do Algoritmo 17 apresentado na Seção 5.2.7. Neste caso, lembre-se que na base de intenções do agente existem duas intenções. A primeira composta pelo plano 1, cuja motivação é igual a 19 (ver Seção) e a segunda composta pelo plano 3 cuja motivação é igual a 11. Sendo assim, a primeira intenção é selecionada para ser executada.

Dado que o Plano 1 é composto pelas ações "use(soldados)", "use(helicopteros)" e o objetivo "atacar(insurgentes)". Dado que a intenção selecionada para ser executada é composta pelo *Plano 1*, durante a execução do plano quando a ação "use(helicopteros) é executada o comportamento regulado pela *Norma 6* é realizado, então a relação *behaviorrealization* de tal norma é atualizada para "REALIZED". Quando o objetivo "atacar(insurgentes)" é atingido o comportamento regulado pela *Norma 4* é realizado, então a relação *behaviorrealization* de tal norma é atualizada para "REALIZED". Este processo é operacionalizado pelos Algoritmos 18 e 19 apresentado na Seção 5.2.8 responsáveis pela execução e monitoramento de uma intenção.

6.3

Considerações Finais

Este capítulo apresentou a aplicação do AgentSpeak(L) e Jason Normativo em dois diferentes cenários, ressaltando as funções normativas aplicadas para realizar as atividades do processo de raciocínio dos agentes de suporte.