

5

Estudo de Caso: Veículos Guiados Automaticamente

O sistema de Veículos Guiados Automaticamente (Automatic Guided Vehicle – AGV) consiste de um número de veículos sem tripulantes que precisam trabalhar em cooperação para transportarem cargas em um ambiente industrial, depósitos ou fábricas. O sistema de AGVs pode ser utilizado tanto para distribuição de produtos em uma fábrica como em uma etapa do processo de fabricação, transportando o produto de uma máquina para outra. As principais funcionalidades que um AGV deve desempenhar são: (i) transportar a carga de uma determinada localidade até seu destino, quando esta estiver disponível; (ii) construir sua rota para realizar a entrega; (iii) evitar colisões e situações de *deadlock*; (iv) manter as baterias recarregadas.

Os AGVs devem ser capazes de operar em um ambiente dinâmico e propício a mudanças, pois a quantidade de cargas que entram no sistema é tipicamente irregular e não preditiva, assim como outros AGVs podem entrar e sair do ambiente a qualquer momento. Vários tipos de perturbações podem ocorrer, como a inviabilidade de um trajeto, por questões de manutenção ou de carga impedindo o acesso, atraso na entrega das cargas aos AGVs, falhas nos AGVs, etc. Os AGVs são equipados com sensores para realizarem a movimentação no ambiente e utilizam redes wireless para se comunicarem. Enquanto se movimentam, os veículos seguem caminhos específicos no ambiente, como formas de navegação utilizam refletores a lasers na parede ou trilhos de imã no chão.

Uma importante característica deste problema é que todas as informações, que são necessárias para os AGVs construírem suas rotas, estão espalhadas por todo o ambiente. Por exemplo, a informação da disponibilidade de uma nova carga não está disponível diretamente ao AGV, mas em alguma parte do sistema, neste caso, nos depósitos. Por quais rotas o AGV deve se deslocar também não

está diretamente disponível. Logo, a informação é totalmente descentralizada. As necessidades de autonomia, para contornar perturbações de maneira flexível e robusta, e pela descentralização da informação, indicam que uma solução auto-organizável possa ser promissora. Assim, como forma de validação do framework, uma solução utilizando o JASOF foi construída, os detalhes são expostos na próxima seção.

5.1 Utilizando o JASOF para construção da solução

O cenário adotado neste estudo de caso representa um ambiente composto por depósitos (Warehouse), áreas de descarga (Destination), localizações (Location) e pelos AGVs (Transporter). Os depósitos recebem cargas e as tornam disponíveis para o transporte, eles estão localizados na parte esquerda do ambiente, como ilustra a **Erro! Fonte de referência não encontrada.** Compõem também o ambiente as áreas de descarga (Destination), localizadas na parte direita do ambiente, que têm como propósito o recebimento das cargas transportadas pelos AGVs. Por entre os depósitos e as áreas de descargas encontram-se as localizações, que são posições a serem trafegadas pelos AGVs quando em movimento pelo ambiente. Esses AGVs (Transporter) são capazes de um conjunto limitado de ações locais e são responsáveis pela realização do transporte de carga de um depósito até uma área de descarga. As ações locais que os AGVs realizam são: Move, para se movimentar no ambiente; Read, para colher informações do ambiente; Pick Load, para obter uma carga disponível em um depósito; e por fim, Drop Load, para despachar uma carga que está sendo carregada.

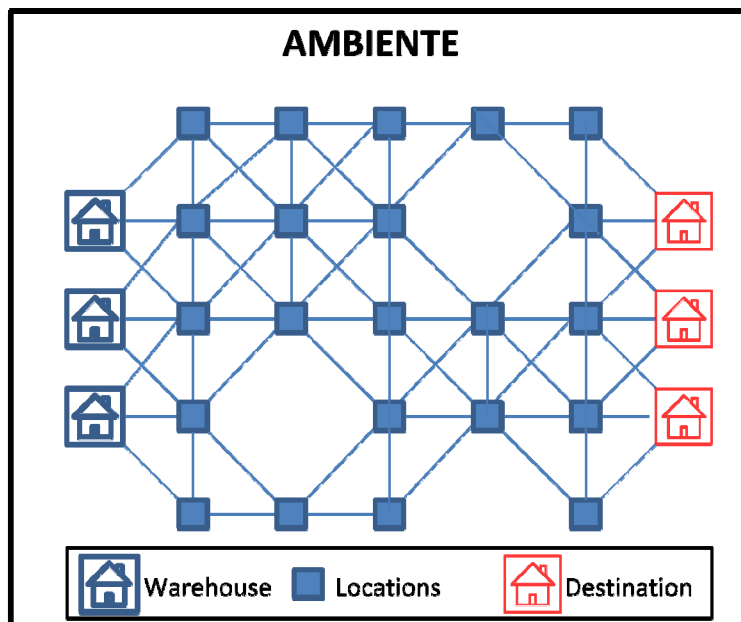


Figura 25 - Ambiente do AGV

O conceito de vizinhança seguido por este cenário é definido como: para um agente X qualquer, todos os agentes que estão conectados a X por apenas um segmento pertencem ao conjunto de agentes vizinhos de X . Conforme ilustra a Figura 26.

A dinâmica do sistema é descrita como segue: Quando o depósito recebe uma nova carga, o mesmo replica a informação às localizações vizinhas, informando que possui uma carga para ser transportada e em qual posição do ambiente ela se encontra. Esta informação é replicada periodicamente pelo depósito enquanto a carga não tiver sido recolhida. Conseqüentemente, ao receber a informação replicada, o ambiente, por meio de suas localizações, realiza um processo de difusão desta informação, ou seja, a informação é repassada aos demais vizinhos com uma modificação em sua relevância. Enquanto este processo está ocorrendo, os transportadores (AGVs) que não estão carregados, estão percorrendo o ambiente a procura de informações que possam indicar a existência de cargas a serem transportadas. Sendo assim, sempre que se movimentar pelo ambiente, o transportador irá ler as informações contidas na sua localização atual. Logo, ao percorrer uma localização que possua a informação supramencionada,

este a utilizará para a criação de uma rota que o leve até a posição na qual a carga disponível se encontra.

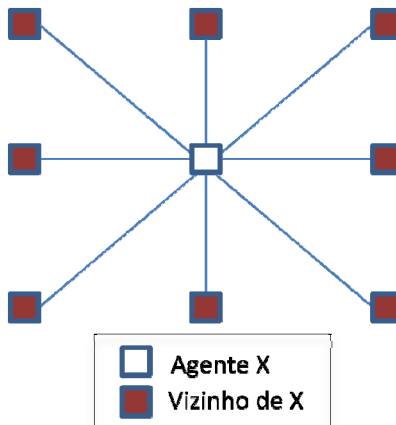


Figura 26 - Conjunto de vizinhos de um agente X

Logo após a construção da rota e o devido deslocamento até a mesma, o AGV comunica-se com o depósito para realizar a ação de coleta, Pick Load. Obtida a carga, o próximo passo do AGV é construir a rota para a realização da entrega. Este processo é similar à procura da carga, neste caso, porém, o AGV procurará informações no ambiente a respeito da posição da área de descarga. Durante o período em que estiver disponível para o recebimento de cargas, a área de descarga irá difundir no ambiente esta informação sobre sua disponibilidade, informando também a sua devida posição no ambiente. Esse procedimento é similar ao realizado pelos depósitos na disponibilidade de cargas. Assim, é através destas informações armazenadas no ambiente que os AGVs elaboram sua rota para a entrega da carga obtida. Por fim, ao chegar à área de descarga o AGV realiza a ação Drop Load, solicitando o recebimento da carga. Após a execução dessas etapas, o AGV recomeça a sua procura por novas cargas a serem transportadas.

Cada um dos componentes do cenário descritos anteriormente foi modelado como agentes do sistema. Para cada depósito há um agente responsável pela sua dinâmica e interação, assim também o é para as áreas de descarga, localizações e

para o AGV. Nas seções seguintes serão mostrados os detalhes de cada agente da solução utilizando o framework JASOF.

5.1.1

Agente Destinaton

Este é o agente responsável por receber as cargas transportadas no ambiente pelo agente Transporter (Seção 5.1.4). Como já descrito na dinâmica do sistema, este agente difunde no ambiente informações acerca da sua disponibilidade e posicionamento. Essas informações são cedidas apenas aos agentes Locations (Seção 5.1.3) da vizinhança, ou seja, àqueles que estão interligados ao Destination por um segmento. Assim, exclui o envio aos agentes Destination da vizinhança, pois a relevância que a informação possui para estes agentes é baixa. Através da ilustração da **Erro! Fonte de referência não encontrada.**, é possível visualizar que inicialmente cada agente Destination tem três vizinhos do tipo Location, que serão os receptores da mensagem, e no mínimo um vizinho do tipo Destination.

O agente Destination foi criado a partir de um agente Jadex, e para ser capaz de realizar o padrão Diffusion sua descrição (ADF) importou a *capability DiffusionPattern*, disponibilizada pelo JASOF. Como já mencionado, esta *capability* possui um plano com quatro mecanismos de extensão. Para os mecanismos *receiveRule* e *coordinationRule*, as soluções padrões fornecidas pelo framework foram adotadas, desta forma, alterações não foram necessárias. Contudo, no mecanismo *destinationRule*, encarregado pela definição dos receptores das mensagens de difusão, foi definido que os receptores seriam os vizinhos do tipo Location. Esta definição foi possível porque o ambiente permite aos agentes requisitarem quem são seus respectivos vizinhos e a qual tipo eles pertencem. Por fim, o mecanismo *propagationRule* foi estendido a fim de especificar qual tipo de informação seria passada ao receptores. Neste caso, para representar esta informação a cerca do posicionamento e disponibilidade da área de descarga, foi feita uma extensão da classe *Data*, criando-se a classe *Dispatch*, esta contendo os atributos para armazenamento das informações citadas e a implementação dos mecanismos de atuação do ambiente, como por exemplo, o

evaporate() que define como esta nova informação evapora com o passar do tempo.

Complementando a funcionalidade do agente Destination, foi criado um plano para definição do protocolo de troca de mensagens, a fim de especificar como este agente receberia as cargas do agente Transporter e como lhe comunicaria sobre a correta recepção da mesma.

5.1.2

Agente Warehouse

O agente Warehouse tem como incumbência o gerenciamento da presença de novas cargas nos depósitos. Ao chegar uma nova carga, o agente executa o padrão Replication para propagar esta informação no ambiente, informando assim que uma nova carga está disponível e pronta para ser transportada. Seguindo o mesmo raciocínio do agente Destination, a replicação da informação é feita somente aos vizinhos do tipo Location, por motivo similar ao já elucidado.

Assim como no agente Destination, os mecanismos *receiveRule* e *coordinationRule* foram mantidos conforme o padrão oferecido pelo framework, ou seja, apesar de o agente Destination estar apto a receber mensagens do tipo *replicationMsg*, ele não executará tarefas em virtude deste recebimento. Entretanto, no mecanismo *propagationRule*, foi estabelecido o conjunto de informações a serem replicadas aos vizinhos. No caso específico do agente Warehouse, a informação a ser replicada trata da disponibilidade de carga e do posicionamento desta no ambiente. A fim de representar este conjunto de informações, uma nova extensão da classe *Data* foi realizada, sendo definida como *Package*. Além do armazenamento dos dados mencionados, outro ponto chave na implementação desta classe foi o procedimento *evaporate()*, que possibilita aos agentes Locations efetuarem o padrão Evaporation. A importância da evaporação nesta etapa é vital para a dinamicidade do ambiente, pois se as informações replicadas pelos agentes Warehouses não fossem evaporadas os AGVs teriam que lidar com uma série de informações desatualizadas, desta forma aumentando consideravelmente a possibilidade de falsos positivos e tráfego desnecessário.

Por fim, no mecanismo *destinationRule* foi determinado o conjunto de vizinhos que receberiam a mensagem de replicação. Assim, conforme estabelecido previamente, somente aos agentes do tipo Location estas teriam relevância, e apenas a estes a mensagem seria enviada. Ainda, no intuito de complementar a funcionalidade dos agentes Warehouses, foi adicionado um plano em sua descrição, proporcionando aos agentes a capacidade de receber solicitações de retirada de cargas dos transportadores. Sendo assim, quando um agente Transporter, localizado na mesma posição do depósito, solicitar uma retirada de carga, o agente Warehouse liberará a carga solicitada e a retirará de sua base, assim não mais desempenhando o padrão Replication até que uma nova carga esteja disponível.

5.1.3

Agente Location

Os agentes Location são responsáveis pela gerência das localizações no ambiente e para cada localização do ambiente existe um agente Location responsável por ela. É através deste agente que é possível a execução dos padrões que atuam no ambiente, como o Evaporation e o Aggregation. Outro mecanismo exercido por este agente é o impedimento de colisões no ambiente, de forma que dois AGVs não possam ocupar a mesma posição. Sendo assim, este é o principal agente do ambiente, aquele que provê a noção de ambiente ativo ao sistema.

No cenário dos veículos não tripulados, todos os agentes Location realizam os padrões Evaporation, Aggregation e Diffusion. Como já descrito na Seção 4.2.2.2, o padrão Evaporation ocorre periodicamente, realizando o decremento do fator de relevância das informações armazenadas. Para os agentes Locations pertencentes à vizinhança de algum agente Warehouse e para aqueles que eventualmente recebam uma mensagem de difusão desses agentes Locations, o padrão Evaporation irá atuar sobre os dados encapsulados na classe *Package*. Já para os agentes Locations posicionados na vizinhança dos agentes Destination, a evaporação ocorrerá nas informações encapsuladas na classe *Dispatch*. Entretanto, qualquer outro tipo de informação armazenada nesses agentes irá sofrer o

processo de evaporação, pois, como já visto, o processo atua em toda base do conhecimento do agente.

Para o caso do padrão *Aggregation*, descrito na Seção 4.2.2.3, os agentes *Location* verificam periodicamente a base do conhecimento por quaisquer informações redundantes, e nestas aplicam o padrão *Aggregation* através do método *aggregate()*. Assim, informações similares passam a ter uma representação única na base, porém com um fator de relevância superior. Para os agentes localizados nas imediações dos agentes *Warehouses*, o processo é executado nos objetos *Packages* replicados e difundidos no ambiente. Já para os agentes localizados na vizinhança dos agentes *Destination* o processo é executado no objeto *Dispatch*.

O padrão *Diffusion* nos agentes *Location* tem duas implicações, a primeira é para que estes possam compreender a mensagem *diffusionMsg*, mediante o mecanismo *receiveRule*, e conseqüentemente exerçam a ação de coordenação por este recebimento, processo presente no mecanismo *coordinationRule*. Já a segunda implicação é decorrente do repasse das informações recebidas, que ocorre especificamente nos agentes *Location* que são vizinhos dos depósitos. A informação recebida, do tipo *Package*, mediante o padrão *Replication*, é difundida em mais um nível na vizinhança, ou seja, desta forma a mensagem é enviada para a vizinhança dos vizinhos do agente *Warehouse*. Portanto, para que estes agentes *Locations* possam construir, repassar e receber esse tipo de informação é necessária a importação do padrão *Diffusion*. Por fim, estes mesmos agentes também importam a *capability* do padrão *Replication*, pois realizam o recebimento de mensagens do tipo *replicationMsg* e as armazenam em sua base, sendo estas exercidas mediante os mecanismos *receiveRule* e *coordinationRule*, respectivamente.

Complementando o conjunto de ações do agente *Location*, está o plano para evitar colisões entre AGVs enquanto estes trafegam no ambiente. Este mecanismo é feito mediante um token que é passado para o agente *Location* pelo AGV que está prestes a se movimentar para a localização. Caso o transportador receba uma confirmação de recebimento do token, este poderá se locomover para a posição

em questão. No caso negativo, isto significa que por algum motivo a localização não está disponível para tráfego. Entre as razões que impeçam esta movimentação estão: (i) outro AGV já está ocupando a localização; (ii) manutenção da localização; (iii) evento inesperado, como uma carga impedindo o acesso.

5.1.4

Agente Transporter

O agente Transporter é o agente que representa o AGV no sistema. Este agente é responsável pelos processos de construção de rota para busca e entrega de cargas no ambiente. Como já mencionado no início da Seção 5.1, este agente possui quatro ações: Move, Read, Pick Load e Drop Load.

Através da ação Move o AGV movimenta-se no ambiente. Na seção anterior já foi mencionada a forma de movimentação, que envolve tokens no intuito de evitar colisões e contornar possíveis imprevistos. Assim, ao decidir se movimentar, o AGV solicita permissão de acesso ao agente responsável pela localização destino. Caso a localização esteja disponível, o agente Location confirma a solicitação e responde com a permissão concedida ao AGV. Conseqüentemente, o AGV retira o token da localização em que se encontra atualmente e movimenta-se para a localização pretendida. Porém, se não for possível a movimentação, o agente Location informa ao AGV que este não pode se deslocar para o local solicitado. Desta forma, o AGV poderá optar pela espera da disponibilidade da localização ou pela modificação da rota prevista.

Sempre após a movimentação no ambiente, o AGV executa a ação de leitura (Read) das informações contidas em sua nova localização. Enquanto estiver descarregado, o AGV lerá as informações em busca de direcionamentos para a obtenção de carga para transporte. Entretanto, no período em que estiver carregado, a busca do AGV será por informações que o auxiliem na construção da rota para despache.

Ao deslocar-se para um depósito que contém uma carga pronta para ser transportada, o AGV realiza sua ação Pick Load. Nesta ação o agente Warehouse recebe a solicitação de retirada da carga, repassa a confirmação ao transportador

solicitante, e por fim retira a carga da sua base. Assim, ao final da execução do Pick Load, o transportador estará carregado e pronto para iniciar o processo de entrega. Tendo feito sua rota até a área de descarga, a próxima ação a ser realizada é a de Drop Load, na qual o transportador requisita o correto recebimento da carga pelo agente Destination. Assim que recebe a confirmação do recebimento, o transportador remove a carga da sua base e reinicia o processo de busca por novas cargas.