

Simulando uma peixaria com Sistemas Multi-Agentes

Olimar Teixeira Borges

¹Escola Politécnica - Informática
Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS)
Porto Alegre – RS – Brasil

`olimar.borges@acad.pucrs.br`

Abstract. *This work simulates a small configuration of the operation of a fish-monger, contemplating the fishing of fish using boats until the transportation of the loading of these fish to the distributor. The main objective is to perform a simulation, with some different configurations, demonstrating the results of the behaviors generated at the end of each simulation.*

Resumo. *Este trabalho simula uma pequena configuração do funcionamento de uma peixaria, contemplando a pesca de peixes utilizando barcos até o transporte do carregamento destes peixes para o distribuidor. O principal objetivo é realizar uma simulação, com algumas configurações diferentes, demonstrando os resultados dos comportamentos gerados ao final de cada simulação.*

1. Introdução

Este trabalho faz parte do contexto de multi-agentes, focado na utilização do *framework* Jacamo, utilizado para a programação de ambientes multi-agentes. Como domínio proposto para este trabalho, foi escolhido o contexto de uma Peixaria, dividida em dois grupos, denominados durante este documento, como equipe de navegação e equipe de transporte.

A equipe de navegação contempla a pesca de cardumes de peixes no oceano, utilizando barcos, com tripulação a bordo e com o descarregamento dos peixes em um porto. Da mesma forma, a equipe de transporte, conta com dois integrantes dentro de um caminhão, que são responsáveis pelo transporte e carregamento de peixes do porto até o distribuidor.

Cada barco e caminhão do sistema possuirá uma capacidade pré-definida de armazenamento de peixes durante a execução da simulação. O oceano também possuirá uma quantidade pré-definida de peixes disponíveis para pesca pelos barcos. Já o porto e o distribuidor terão capacidade ilimitada de armazenamento de peixes.

A tripulação do barco é composta por três papéis distintos de agentes, sendo o capitão, o pescador e o auxiliar de pesca. Para o caminhão foram definidos dois papéis, sendo o motorista e o carregador. Para a equipe de navegação, foram criadas três tarefas que serão executadas através de três esquemas. E para a equipe de transporte, foram criadas duas tarefas, que serão executadas pelo mesmo número de esquemas.

Durante todo o trabalho, são apresentados diversos diagramas da organização, do ambiente e dos agentes. Também constam os planos e objetivos detalhados que serão executados por cada agente, como também um experimento para demonstrar o funcionamento do sistema. E por fim, segue uma breve descrição sobre o aprendizado adquirido durante o desenvolvimento, algumas dificuldades, dúvidas e observações.

2. Sistemas Multi-agentes

A área de Sistemas Multi-agentes se interessa pelo estudo de agentes autônomos em um universo Multi-agentes [Demazeau and Müller 1990]. O termo autônomo designa o fato de que os agentes têm uma existência própria, independente da existência de outros agentes. Como não existe um problema pré-definido que o sistema deve resolver, o objetivo é conceber os meios a partir dos quais pode-se assegurar que agentes desejem cooperar e efetivamente o façam, com o intuito de resolver um problema específico quando este for apresentado ao sistema [Alvares and Sichman 1997].

2.1. Jason

A linguagem interpretada por Jason é uma extensão da linguagem de programação abstrata chamada *AgentSpeak* (L), originalmente criada por Anand Rao[Rao 1996].

Jason é um intérprete para uma versão estendida do *AgentSpeak*. Ele implementa a semântica operacional desse idioma e fornece uma plataforma para o desenvolvimento de sistemas multi-agentes. Jason está disponível em código aberto, e é distribuído sob GNU LGPL [Hübner et al. 2004].

2.2. Cartago

O CArtaGO (infra-estrutura comum da *ARTifact* para ambientes *AGents Open*) é uma estrutura de propósito geral que permite programar e executar ambientes virtuais - também designados ambientes virtuais, aplicativos e/ou software - para sistemas multi-agentes¹.

O CArtaGO possibilita desenvolver e executar ambientes baseados em artefatos, estruturados em espaços de trabalho abertos (possivelmente distribuídos em toda a rede) que os agentes de diferentes plataformas podem se unir para trabalhar juntos dentro desses ambientes. Dessa forma, com o CArtaGO, os desenvolvedores de sistemas multi-agentes possuem um modelo de programação para projetar e programar o ambiente computacional do agente, composto por conjuntos dinâmicos de artefatos de diferentes tipos, além dos modelos e plataformas usados para programar agentes.

2.3. Moise

Moise é um modelo organizacional para sistemas multi-agentes baseado em noções como papéis, grupos e missões. Permite que um sistema multi-gente tenha uma especificação explícita de sua organização. Esta especificação deve ser usada pelos agentes para argumentar sobre sua organização e também como plataforma de organização que imponha quais especificações os agentes devem seguir².

Combinando as três tecnologias citadas acima (Jason, Cartago e Moise), cobrindo assim todos os níveis de abstrações que são necessárias para o desenvolvimento de sofisticados sistemas multi-agente, foi criada uma estrutura chamada JaCaMo para a programação multi-agente, sendo a plataforma escolhida para o desenvolvimento deste trabalho. A plataforma JaCaMo é distribuída sob as licenças de cada plataforma, ou seja, a fonte é aberta e está sob a licença GNU LGPL.

¹<http://cartago.sourceforge.net/>

²<http://moise.sourceforge.net/>

3. Domínio

Para o desenvolvimento deste trabalho, foi escolhido o contexto de uma empresa de peixaria, que contemplará desde a pescaria em alto mar até o transporte dos peixes em terra. Durante o processo de navegação em alto mar, haverá barcos com capacidades variadas de armazenamento de peixes, que serão responsáveis pela pesca e descarregamento de peixes no porto e que possuam redes com capacidades limitadas para a pesca. Os barcos iniciam no porto e cada um possui três tripulantes, sendo o capitão, o pescador e o auxiliar de pesca. O capitão será responsável pelo controle e navegação do barco e por determinar o local da pesca. Já o pescador possui responsabilidade de jogar e recolher as redes. O auxiliar de pesca será responsável por puxar e soltar a âncora do barco, como também retirar os peixes das redes.

Para o processo de transporte dos peixes, caminhões serão responsáveis por recolher os peixes deixados pelos barcos no porto e levar até o distribuidor. Os caminhões também possuem capacidades variadas de armazenamento de peixes e cada caminhão possui um motorista e um carregador. O motorista será responsável pelo controle e direção do caminhão, verificando a rota até o distribuidor pelo GPS. Já o carregador, possui como responsabilidades localizar o carregamento de peixes no porto e carregar o caminhão.

O detalhamento do processo de desenvolvimento deste domínio, será detalhado nas próximas seções, com diagramas da organização, especificações das missões, normas, artefatos e agentes.

4. Desenvolvimento

Serão apresentados diferentes cenários, com configurações diversas para exemplificar alguns comportamentos, de acordo com as quantidades de artefatos e demais configurações realizadas nos mesmos. Alguns diagramas serão apresentados, com a estrutura do sistema, sua organização e demais informações pertinentes para um melhor entendimento sobre o problema proposto.

4.1. Organização (Moise)

Conforme pode ser observado na Figura 1, foi criado um grupo denominado de peixaria e dois subgrupos denominados de transporte e embarcação. No arquivo XML de configuração da organização, o grupo possui o identificador “peixaria_group” e os subgrupos de “equipe_navegacao” e de “equipe_transporte”. O subgrupo transporte possui dois papéis para agentes, carregador e motorista, já o subgrupo embarcação, possui três papéis, capitão, pescador e auxiliar de pesca.

O grupo de navegação possui três esquemas sendo, “Procurar peixes”, “Realizar a pescaria” e “Descarregar peixes no porto”. Seguindo o modelo de diagramas denominado de *Open AEOLus*³, criado por Daniela Uez, foram criados um diagrama para cada esquema. Este modelo foi utilizado para a criação de todos os diagramas que constam neste trabalho, nas três camadas dos sistemas multi-agentes.

Para atingir aos objetivos, foram definidos cinco papéis no sistema, conforme observado na Figura 2. As setas com a ponta sem preenchimento, representam a herança

³<http://www.uez.com.br/aeolus>

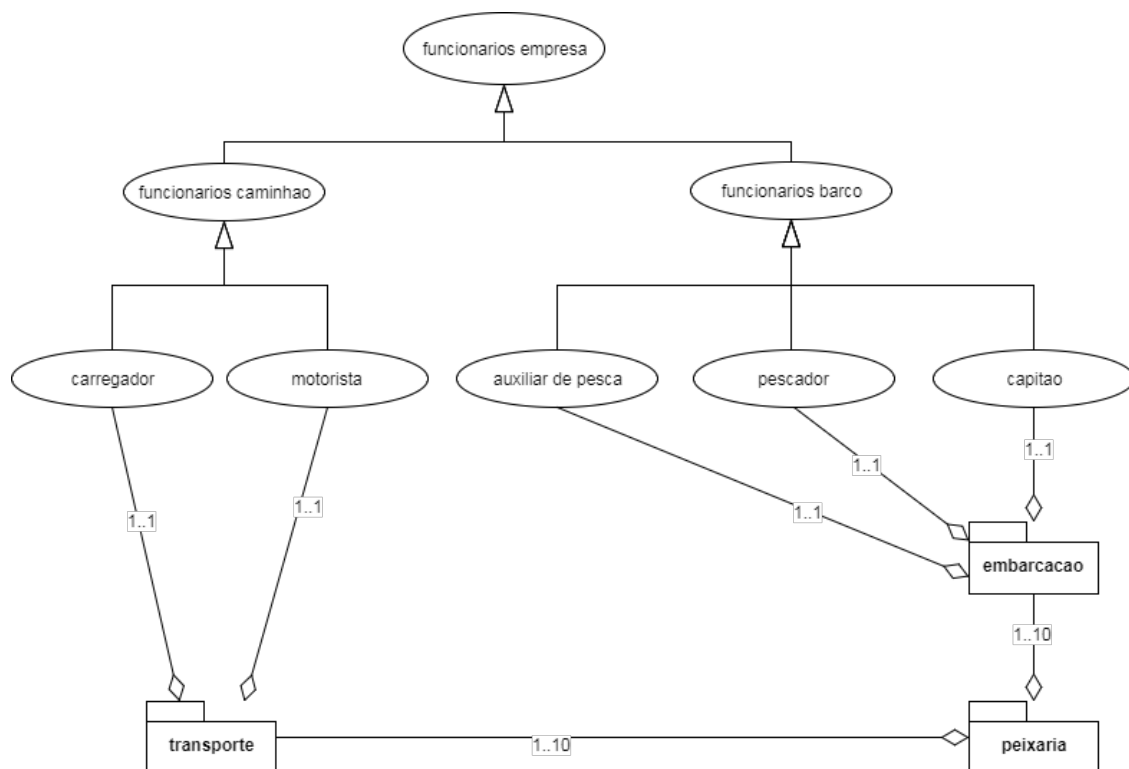


Figura 1. Diagrama geral do sistema - Moise

de cada papel. As setas com ponta preenchida, representam a hierarquia entre os papéis. Já as setas com ponta arredondada, apresentam a comunicação entre os papéis. Como por exemplo, o papel Capitao tem autoridade em cima dos papéis Pescador e Auxiliar de Pesca, enquanto herda de Funcionários Barco, que possui um *link* de comunicação e que por sua vez, também herda de Funcionários Empresa.

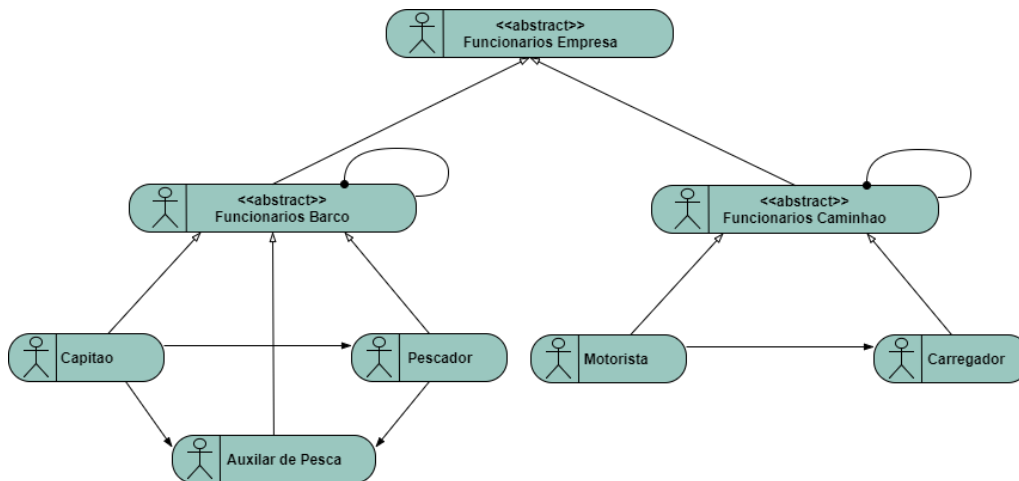


Figura 2. Diagrama geral dos papéis do sistema

A Figura 3 apresenta a estrutura do esquema Procurar peixes, a Figura 4 do esquema Realizar a pescaria e a Figura 5 do esquema Descarregar peixes no porto.

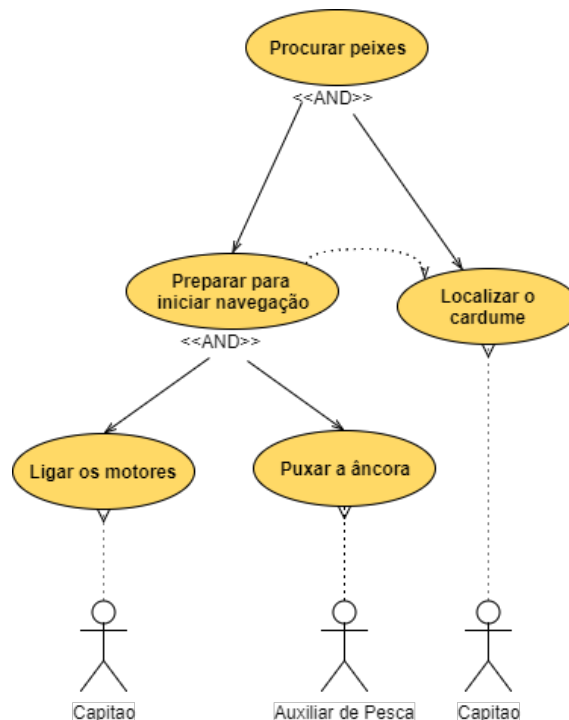


Figura 3. Diagrama do esquema Procurar peixes - Moise

Para o esquema Procurar peixes, foram criadas duas missões, a “m_navegacao_t1”, responsável pela execução dos objetivos “Ligar os motores” e “Localizar o cardume” e a

“m_auxilio_t1”, responsável pelo objetivo “Puxar a âncora”.

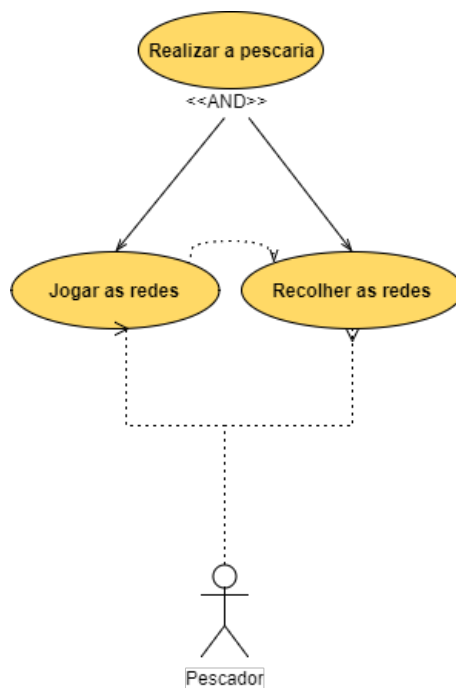


Figura 4. Diagrama do esquema Realizar a pescaria - Moise

Para o esquema Realizar a pescaria, foi criada uma missão, a “m_pescaria_t2”, responsável pela execução dos objetivos “Jogar as redes” e “Recolher as redes”.

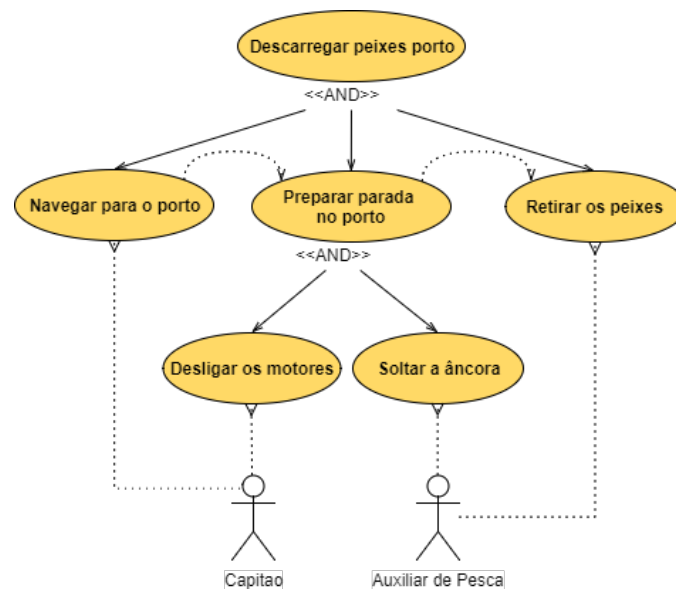


Figura 5. Diagrama do esquema Descarregar peixes no porto - Moise

Para o esquema Descarregar peixes no porto, foram criadas duas missões, a “m_navegacao_t3”, responsável pela execução dos objetivos “Navegar para o porto” e “Desligar os motores” e a “m_auxilio_t3”, responsável pelos objetivos “Retirar os peixes” e “Soltar a âncora”.

O grupo de transporte possui dois esquemas sendo, “Recolher os peixes” e “Levar os peixes ao distribuidor”. As Figuras 6 e 7 apresentam as estruturas dos esquemas, respectivamente.

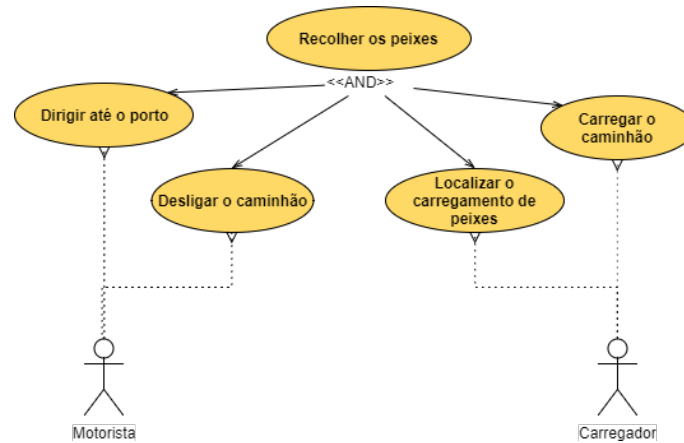


Figura 6. Diagrama do esquema Recolher os peixes - Moise

Para o esquema Recolher os peixes, foram criadas duas missões, a “m_deslocamento_t1”, responsável pela execução dos objetivos “Dirigir até o porto” e “Desligar o caminhão” e a “m_operacional_t1”, responsável pelos objetivos “Localizar o carregamento de peixes” e “Carregar caminhão”.

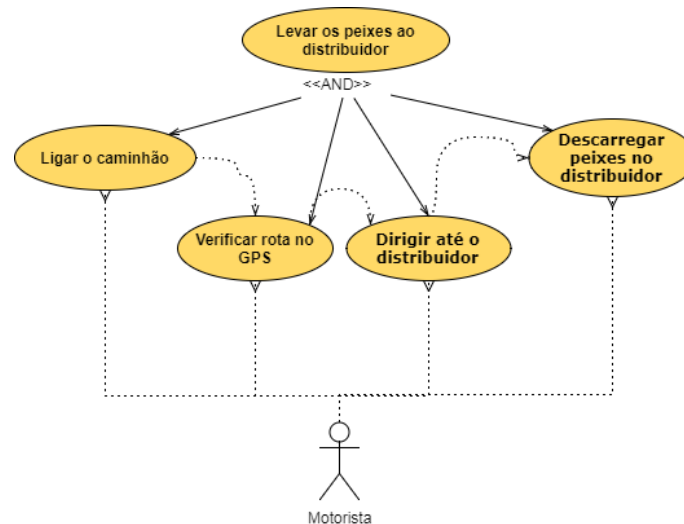


Figura 7. Diagrama do esquema Levar os peixes ao distribuidor - Moise

Para o esquema Levar os peixes ao distribuidor, foi criada uma missão, a “m_deslocamento_t2”, responsável pela execução dos objetivos “Ligar o caminhão”, “Verificar rota no GPS”, “Dirigir até o distribuidor” e “Descarregar peixes no distribuidor”.

Para organizar a distribuição das tarefas, cada missão foi atribuída a um papel responsável para atingir aos objetivos vinculados a missão. Para isso foram criadas oito normas, todas do tipo *obligation*, conforme pode ser observado a seguir:

- n1: missão m_navegacao_t1, papel responsável capitao.
- n2: missão m_auxilio_t1, papel responsável aux_pesca.
- n3: missão m_pescaria_t2, papel responsável pescador.
- n4: missão m_navegacao_t3, papel responsável capitao.
- n5: missão m_auxilio_t3, papel responsável aux_pesca.
- n6: missão m_deslocamento_t1, papel responsável motorista.
- n7: missão m_operacional_t1, papel responsável carregador.
- n8: missão m_deslocamento_t2, papel responsável motorista.

4.2. Ambiente (Cartago)

Conforme definido no domínio, barcos são responsáveis pela pesca, desta forma, foi criado um artefato denominado Barco, que proporcionará aos agentes consultar informações e executar ações para que consigam atingir seus objetivos. Na Figura 8, são detalhadas quais ações os artefatos Barco e Oceano possuem, e as informações que os artefatos enviarão como crenças para os agentes.

O artefato Oceano foi criado com o intuito de servir como parte auxiliar ao Barco, no qual ficarão armazenados uma quantidade variável de peixes, nos quais as instâncias de Barco poderão retirar peixes. Para possibilitar essa troca de peixes entre o Oceano e os Barcos, foi criada uma operação de *link* no Barco, chamada de “recolher redes”, que recebe por parâmetro o identificador do artefato Oceano e retira a quantidade que uma rede possui como capacidade máxima. Essa operação executa outra dentro do artefato Oceano, chamada de “pescar cardume”, que realiza algumas validações de quantidades de peixes que ainda constam no Oceano e após envia a quantidade solicitada ao Barco e diminui este valor de sua base de peixes. Por exemplo, se um Barco com capacidade de armazenamento de 1000 peixes e com redes com capacidade de pesca de 25 peixes por vez que a rede é jogada no mar, terá que executar esta ação de pesca 40 vezes, até atingir a capacidade máxima de peixes possíveis de armazenamento. Quando esse limite do Barco é atingido, ele pára com a pesca e se dirige ao Porto para descarregar.

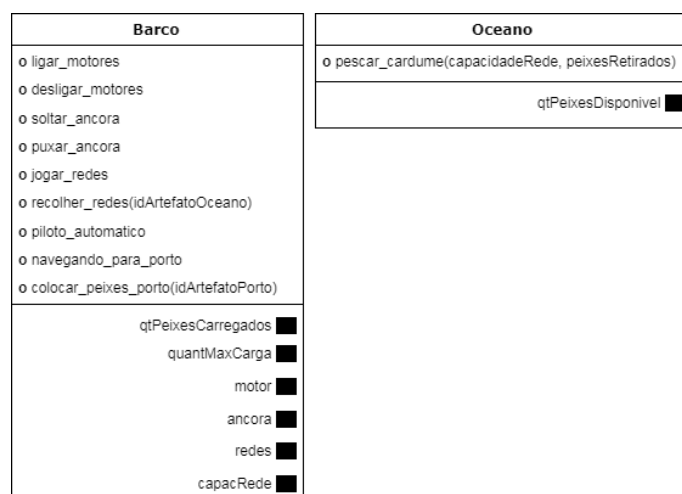


Figura 8. Diagramas dos artefatos de navegação

Dentro do domínio proposto, existem duas equipes, uma responsável pela pescaria e outra pelo transporte. Para que os peixes pescados possam ser transferidos da equipe de pesca para a equipe de transporte, foi criado um artefato chamado Porto, que pode ser observado na Figura 9. Ele é responsável por fazer a intermediação entre o artefato Barco e o artefato Caminhão. O Porto pode receber uma quantidade indeterminada de peixes dos Barcos e os Caminhões retiram esses peixes, de acordo com suas capacidades de armazenamento. Da mesma forma que o Barco e o Oceano, o Porto também possui operações de *links* com os Barcos e com os Caminhões. Para a troca de peixes com os Barcos foi criada uma operação chamada “receber peixes barco”, no qual o Barco passa para o Porto todos os peixes que possui armazenado. Enquanto que a troca de peixes com o Caminhão ocorre através da operação “carregar peixes caminhão”.

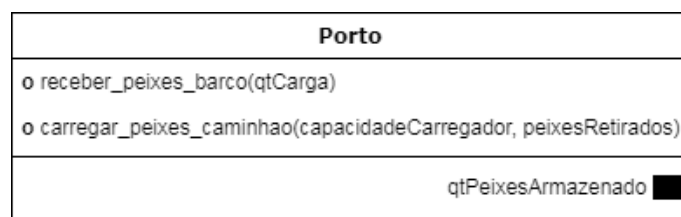


Figura 9. Diagrama do artefato Porto

A equipe de transporte possui dois artefatos, o Caminhão e o Distribuidor, conforme pode ser observado na Figura 10. O Caminhão possui um limite pré-definido de capacidade de armazenamento e é responsável por retirar do Porto a quantidade máxima que pode armazenar. Após sua capacidade máxima ser atingida, o Caminhão sai do Porto e vai em direção ao artefato Distribuidor, que também possui uma operação de *link* chamada “receber carregamento de peixes”, que serve para esvaziar o Caminhão e armazenar os peixes para a distribuição.

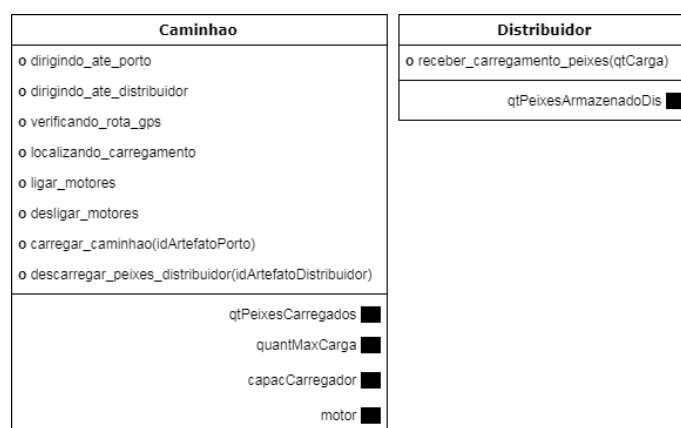


Figura 10. Diagramas dos artefatos de transporte

Os cinco artefatos criados neste sistema, servem para auxiliar os agentes durante a execução de seus objetivos e serão utilizados durante todo o processo de execução do sistema.

4.3. Agentes (Jason)

Os agentes do sistema seguirão os papéis já estipulados anteriormente. Conforme definido na subseção da Organização [4.1], cada papel será responsável por executar algumas missões e por consequência, os objetivos vinculados a cada missão.

A Figura 11 mostra quais objetivos cada papel, da equipe de navegação, será responsável por atingir.

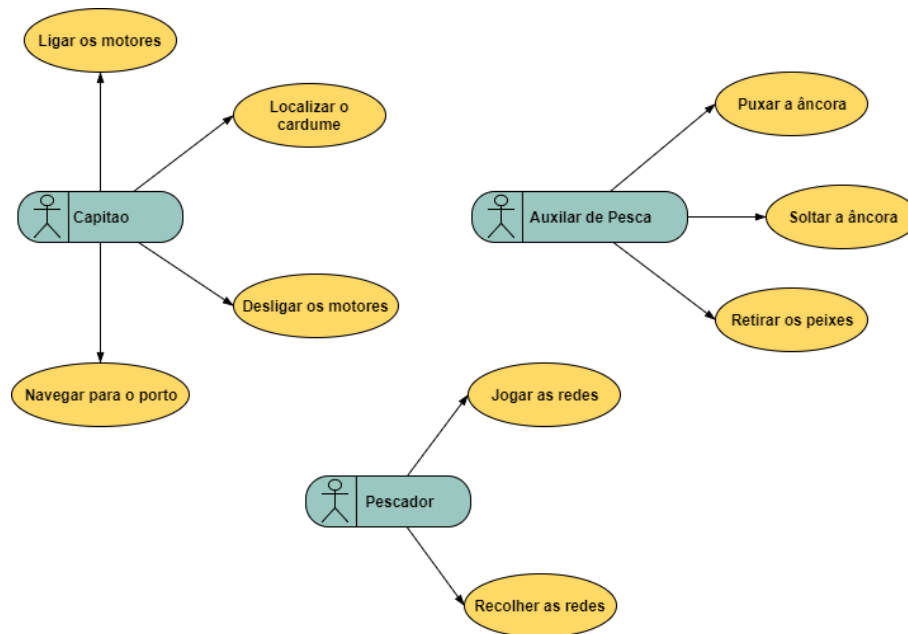


Figura 11. Diagramas das responsabilidades dos papéis da equipe de navegação

Os agentes do tipo Capitão serão responsáveis por iniciarem a execução das simulações da equipe de navegação, através da execução da instância do primeiro esquema do subgrupo de navegação. O Capitão possui os seguintes planos:

- !criarEsquema: responsável por criar os esquemas do subgrupo da equipe de navegação, que serão instanciados ao longo do sistema pelos agentes desta equipe. Nele também é instanciado o primeiro esquema da equipe, denominado “procPeixes”.
- !g1t1_ligar_motor: responsável por ligar os motores do barco para iniciar a navegação em mar aberto, por meio da execução da operação “ligar motores” do artefato Barco.
- !g1t1_localizar_cardume: responsável por localizar os cardumes no Oceano. Nele é acionada a operação “piloto automático” do artefato Barco e também a instancia do segundo esquema denominado “realizarPesca”.
- !g1t3_nav_porto: responsável por navegar em direção ao Porto para descarregar, por meio da execução da operação “navegando para porto” do artefato Barco.
- !g1t3_desligar_motor: responsável por desligar os motores do Barco, por meio da execução da operação “desligar motores” do artefato Barco.

Os agentes do tipo Pescador serão responsáveis pela execução da pescaria. O Pescador possui os seguintes planos:

- !gl1t2_jogar_redes: responsável por jogar as redes nos cardumes, por meio da execução da operação “jogar redes” do artefato Barco.
- !gl1t2_recolher_redes: responsável por recolher as redes e se dirigir em direção ao Porto. Nesta primeira versão do plano, é verificado na base de crenças do agente se a capacidade de armazenamento do Barco ainda não foi atingida, caso contrário, o Barco suspende a pescaria e instancia o terceiro esquema denominado “descarrPeixes”.
- !gl1t2_recolher_redes: Nesta segunda versão do plano, é verificado na base de crenças do agente se o Oceano ainda possui peixes disponíveis para pesca, caso contrário, o Barco suspende a pescaria e instancia o terceiro esquema denominado “descarrPeixes”.
- !gl1t2_recolher_redes: Nesta terceira versão do plano, é verificado na base de crenças do agente se a quantidade de peixes armazenados ainda não foi atingida, caso contrário, o Barco continua jogando as redes no Oceano até atingir a capacidade máxima de armazenamento. É neste plano que será executado a operação de *link* entre o Barco e o Oceano.

Os agentes do tipo Auxiliar de Pesca serão responsáveis por retirarem os peixes do Barco e descarregar no Porto e por puxar e soltar a âncora do Barco. O Auxiliar de Pesca possui os seguintes planos:

- !gl1t1_puxar_ancora: responsável por puxar a âncora para iniciar a navegação em mar aberto, por meio da execução da operação “puxar âncora” do artefato Barco.
- !gl1t1_soltar_ancora: responsável por soltar a âncora para iniciar a parada no Porto, por meio da execução da operação “soltar âncora” do artefato Barco.
- !gl1t1_retir_peixes: responsável por retirar os peixes do Barco e descarregar no Porto, por meio da execução da operação “colocar peixes porto” do artefato Barco. É neste plano que será executado a operação de *link* entre o Barco e o Porto.

Para a equipe de transporte, a Figura 12 apresenta quais objetivos cada papel será responsável por atingir.

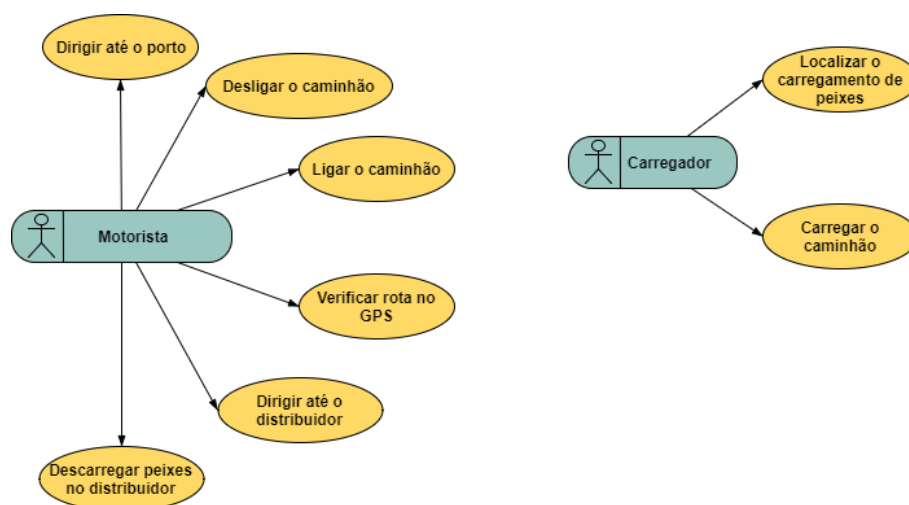


Figura 12. Diagramas das responsabilidades dos papéis da equipe de transporte

Os agentes do tipo Motorista serão responsáveis por iniciarem a execução das simulações da equipe de transporte, através da execução da instância do primeiro esquema do subgrupo de transporte. O Motorista possui os seguintes planos:

- !criarEsquema: responsável por criar os esquemas do subgrupo da equipe de transporte, que serão instanciados ao longo do sistema pelos agentes desta equipe. Nele também é instanciado o primeiro esquema da equipe, denominado “recolherPeixesPorto”.
- !g2t1_dirigir_ate_porto: responsável por se locomover até o Porto para buscar o carregamento de peixes, por meio da execução da operação “dirigindo ate porto” do artefato Caminhão.
- !g2t1_desligar_caminhao: responsável por desligar o caminhão, por meio da execução da operação “desligar motores” do artefato Caminhão.
- !g2t2_ligar_caminhao: responsável por ligar os motores do caminhão, por meio da execução da operação “ligar motores” do artefato Caminhão.
- !g2t2_verificar_rota_gps: responsável por verificar o melhor caminho até o Distribuidor, por meio da operação “verificando_rota_gps” do artefato Caminhão.
- !g2t2_dirigir_ate_distribuidor: responsável por dirigir até o Distribuidor, por meio da execução da operação “dirigindo ate distribuidor” do artefato Caminhão.
- !g2t2_descarr_peixes: responsável por retirar os peixes do Caminhão e descarregar no Distribuidor, por meio da execução da operação “descarregar peixes distribuidor” do artefato Caminhão. É neste plano que será executado a operação de *link* entre o Caminhão e o Distribuidor.

Os agentes do tipo Carregador serão responsáveis pelo carregamento de peixes no Caminhão. O Carregador possui os seguintes planos:

- !g2t1_localizar_carregamento: responsável por localizar o carregamento de peixes no Porto, por meio da execução da operação “localizando carregamento” do artefato Caminhão.
- !g2t1_carregar_caminhao: responsável por carregar o Caminhão e se dirigir em direção ao Distribuidor. Nesta primeira versão do plano, é verificado na base de crenças do agente se a capacidade de armazenamento do Caminhão ainda não foi atingida, caso contrário, o Caminhão suspende o carregamento e instancia o segundo esquema denominado “levarPeixesDistribuidor”.
- !g2t1_carregar_caminhao: Nesta segunda versão do plano, é verificado na base de crenças do agente se o Porto ainda possui peixes disponíveis para carregamento, caso contrário, o Caminhão suspende o carregamento e instancia o segundo esquema denominado “levarPeixesDistribuidor”.
- !g2t1_carregar_caminhao: Nesta terceira versão do plano, é verificado na base de crenças do agente se a quantidade de peixes armazenados ainda não foi atingida, caso contrário, o Caminhão continua carregamento peixes do Porto até atingir a capacidade máxima de armazenamento. É neste plano que será executado a operação de *link* entre o Caminhão e o Porto.

Todos os agentes do sistema possuem a *trigger* denominada “+play”. Nela os agentes recebem por parâmetro o nome do subgrupo ao qual estão vinculados, que é o mesmo nome da instância do artefato Barco ou Caminhão. Dentro deste plano, os agentes realizam o foco nos artefatos que irão visualizar dentro do sistema. É neste momento que

os agentes focam apenas em um Barco ou em um Caminhão, de acordo com a equipe que pertencem, sem que ocorra o problema de um agente focar em mais de um artefato ao mesmo tempo, garantindo que as cardinalidades definidas no Moise, sejam respeitadas.

5. Experimento

Para demonstrar o resultado do desenvolvimento deste trabalho, foi realizado um experimento que apresenta uma pequena simulação de uma configuração do funcionamento de uma peixaria.

Neste experimento foram criados dois barcos, dois caminhões, um porto, um oceano e um distribuidor. Para os barcos foram criados três agentes para cada e para os caminhões foram criados dois agentes para cada, conforme pode ser observados na Figura 13.

```
//Barco 01 #####
agent pedroCap1 : capitao.asl{ //capitao
  focus: oceano, porto
}
agent pedroPesc1 : pescador.asl{ // pescador
  focus: oceano, porto
}
agent pedroAux1 : aux_pesca.asl{ //aux pesca
  focus: oceano, porto
}

//Barco 02 #####
agent pedroCap2 : capitao.asl{ //capitao
  focus: oceano, porto
}
agent pedroPesc2 : pescador.asl{ // pescador
  focus: oceano, porto
}
agent pedroAux2 : aux_pesca.asl{ //aux pesca
  focus: oceano, porto
}

//Caminhão 01 #####
agent samuelMot1 : motorista.asl{ //capitao
  focus: distribuidor, porto
}
agent samuelCarr1 : carregador.asl{ // pescador
  focus: distribuidor, porto
}

//Caminhão 02 #####
agent samuelMot2 : motorista.asl{ //capitao
  focus: distribuidor, porto
}
agent samuelCarr2 : carregador.asl{ // pescador
  focus: distribuidor, porto
}
```

Figura 13. Definição dos agentes da simulação

O barco 1 possuirá uma capacidade de armazenamento de 500 peixes e uma capacidade de suas redes para pesca de apenas 25 peixes por vez. O barco 2 possuirá uma capacidade de armazenamento de 1000 peixes e uma capacidade de suas redes para pesca de apenas 50 peixes por vez. Da mesma forma, o Caminhão 1 possuirá uma capacidade máxima de armazenamento de 400 peixes e de 25 peixes possíveis de carregamento por vez. O Caminhão 2 possuirá uma capacidade máxima de armazenamento de 1000 peixes e de 30 peixes possíveis de carregamento por vez. O oceano terá uma quantidade de 100.000 peixes disponíveis para pesca, enquanto que o Porto e o Distribuidor terão capacidade ilimitada de armazenamento de peixes. A Figura 14 demonstra a configuração citada.

```

//Artefatos
workspace peixaria{

    //Barco 01: capacidade máxima de peixes na rede por pescador = 25 /
    //quantidade máxima de carga permitida de peixes = 500
    artifact barco_01: peixaria.Barco(25, 500)
    //Barco 02: capacidade máxima de peixes na rede por pescador = 50 /
    //quantidade máxima de carga permitida de peixes = 1000
    artifact barco_02: peixaria.Barco(50, 1000)

    //Caminhao 01: capacidade máxima de peixes que o carregador consegue carregar por vez = 25 /
    //quantidade máxima de carga permitida de peixes = 400
    artifact caminhao_01: peixaria.Caminhao(25, 400)
    //Caminhao 02: capacidade máxima de peixes que o carregador consegue carregar por vez = 25 /
    //quantidade máxima de carga permitida de peixes = 1000
    artifact caminhao_02: peixaria.Caminhao(30, 1000)

    //Oceano: quantidade de peixes disponível no mar = 100000
    artifact oceano: peixaria.Oceano(100000)
    artifact porto: peixaria.Porto()
    //Distribuidor: não há restrições de quantidade de armazenamento de peixes
    artifact distribuidor: peixaria.Distribuidor()
}

```

Figura 14. Definição dos artefatos da simulação

A configuração da organização pode ser visualizada na Figura 15. Ela apresenta a criação dos subgrupos e a definição dos papéis para cada agente.

```

organisation opeixaria: empresaPeixaria-os.xml {
    // #####
    // EQUIPE DE NAVEGAÇÃO #####
    // #####
    group barco_01: equipe_navegacao {
        players: pedroCap1 capitao
                pedroPesc1 pescador
                pedroAux1 aux_pesca
    }
    group barco_02: equipe_navegacao {
        players: pedroCap2 capitao
                pedroPesc2 pescador
                pedroAux2 aux_pesca
    }

    // #####
    // EQUIPE DE TRANSPORTE #####
    // #####
    group caminhao_01: equipe_transporte {
        players: samuelMot1 motorista
                samuelCarr1 carregador
    }
    group caminhao_02: equipe_transporte {
        players: samuelMot2 motorista
                samuelCarr2 carregador
    }
}

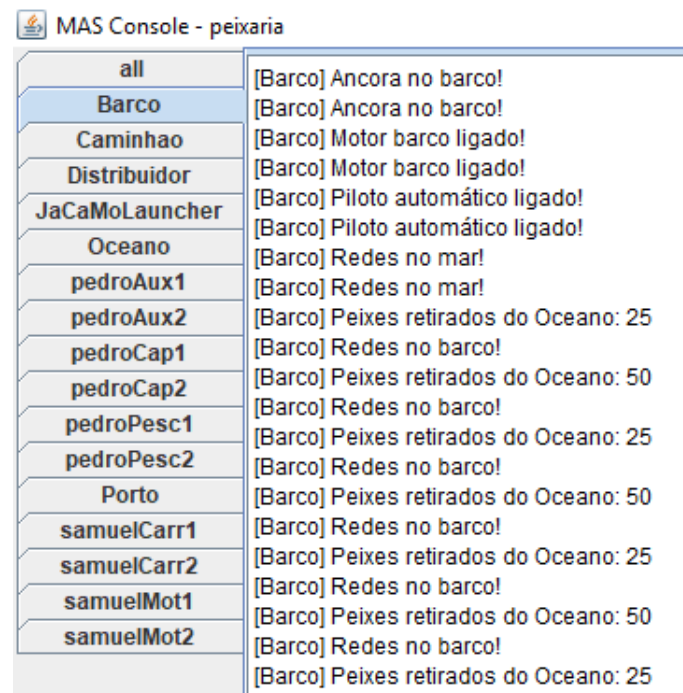
```

Figura 15. Definição da organização da simulação

Após a execução da simulação, pode ser observado que os subgrupos de cada equipe são criados, enquanto os agentes iniciam a ação de focar nos artefatos Barco e Caminhão, que são responsáveis por agrupá-los. Na sequência são executadas as ações para

se responsabilizar por cada missão, de acordo com seus papéis. São apresentados *prints* de cada operação realizada nos artefatos, como também dos planos que os agentes vão executando. A seguir seguem algumas figuras que demonstram a execução da simulação nas configurações pré-definidas acima.

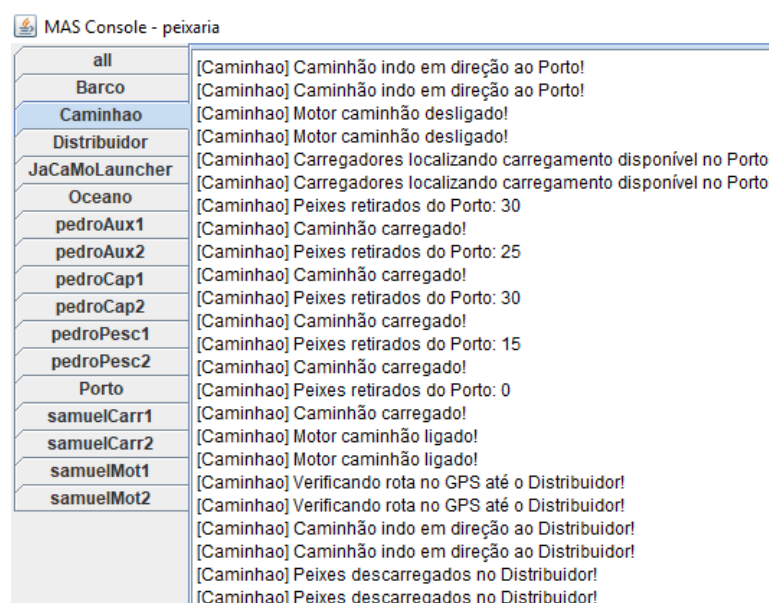
A Figura 16 apresenta os resultados das execuções realizadas pelos Barcos.



MAS Console - peixaria	
all	[Barco] Ancora no barco!
Barco	[Barco] Ancora no barco!
Caminhao	[Barco] Motor barco ligado!
Distribuidor	[Barco] Motor barco ligado!
JaCaMoLauncher	[Barco] Piloto automático ligado!
Oceano	[Barco] Piloto automático ligado!
pedroAux1	[Barco] Redes no mar!
pedroAux2	[Barco] Peixes retirados do Oceano: 25
pedroCap1	[Barco] Redes no barco!
pedroCap2	[Barco] Peixes retirados do Oceano: 50
pedroPesc1	[Barco] Redes no barco!
pedroPesc2	[Barco] Peixes retirados do Oceano: 25
Porto	[Barco] Redes no barco!
samuelCarr1	[Barco] Peixes retirados do Oceano: 50
samuelCarr2	[Barco] Redes no barco!
samuelMot1	[Barco] Peixes retirados do Oceano: 25
samuelMot2	[Barco] Peixes retirados do Oceano: 50
	[Barco] Redes no barco!
	[Barco] Peixes retirados do Oceano: 25

Figura 16. Resultados dos artefatos Barcos

A Figura 17 apresenta os resultados das execuções realizadas pelos Caminhões.



MAS Console - peixaria	
all	[Caminhao] Caminhão indo em direção ao Porto!
Barco	[Caminhao] Caminhão indo em direção ao Porto!
Caminhao	[Caminhao] Motor caminhão desligado!
Distribuidor	[Caminhao] Motor caminhão desligado!
JaCaMoLauncher	[Caminhao] Carregadores localizando carregamento disponível no Porto!
Oceano	[Caminhao] Carregadores localizando carregamento disponível no Porto!
pedroAux1	[Caminhao] Peixes retirados do Porto: 30
pedroAux2	[Caminhao] Caminhão carregado!
pedroCap1	[Caminhao] Peixes retirados do Porto: 25
pedroCap2	[Caminhao] Caminhão carregado!
pedroPesc1	[Caminhao] Peixes retirados do Porto: 30
pedroPesc2	[Caminhao] Caminhão carregado!
Porto	[Caminhao] Peixes retirados do Porto: 15
samuelCarr1	[Caminhao] Caminhão carregado!
samuelCarr2	[Caminhao] Peixes retirados do Porto: 0
samuelMot1	[Caminhao] Caminhão carregado!
samuelMot2	[Caminhao] Motor caminhão ligado!
	[Caminhao] Motor caminhão ligado!
	[Caminhao] Verificando rota no GPS até o Distribuidor!
	[Caminhao] Verificando rota no GPS até o Distribuidor!
	[Caminhao] Caminhão indo em direção ao Distribuidor!
	[Caminhao] Caminhão indo em direção ao Distribuidor!
	[Caminhao] Peixes descarregados no Distribuidor!
	[Caminhao] Peixes descarregados no Distribuidor!

Figura 17. Resultados dos artefatos Caminhões

A Figura 18 apresenta os resultados das execuções realizadas pelo Oceano.

MAS Console - peixaria	
all	[Oceano] Achou Cardume!
Barco	[Oceano] Peixes disponíveis no mar: 99975
Caminhao	[Oceano] Achou Cardume!
Distribuidor	[Oceano] Peixes disponíveis no mar: 99925
JaCaMoLauncher	[Oceano] Achou Cardume!
Oceano	[Oceano] Peixes disponíveis no mar: 99900
pedroAux1	[Oceano] Achou Cardume!
pedroAux2	[Oceano] Peixes disponíveis no mar: 99850
pedroCap1	[Oceano] Achou Cardume!
pedroCap2	[Oceano] Peixes disponíveis no mar: 99825
pedroPesc1	[Oceano] Achou Cardume!
pedroPesc2	[Oceano] Peixes disponíveis no mar: 99775
Porto	[Oceano] Achou Cardume!
samuelCarr1	[Oceano] Peixes disponíveis no mar: 99750
samuelCarr2	[Oceano] Achou Cardume!
samuelMot1	[Oceano] Peixes disponíveis no mar: 99700
samuelMot2	[Oceano] Achou Cardume!
	[Oceano] Peixes disponíveis no mar: 99675
	[Oceano] Achou Cardume!
	[Oceano] Peixes disponíveis no mar: 99625

Figura 18. Resultados do artefato Oceano

A Figura 19 apresenta os resultados das execuções realizadas pelo Porto.

MAS Console - peixaria	
all	[Porto] Achou carregamento de peixes!
Barco	[Porto] Peixes disponíveis ainda no Porto: 70
Caminhao	[Porto] Achou carregamento de peixes!
Distribuidor	[Porto] Peixes disponíveis ainda no Porto: 45
JaCaMoLauncher	[Porto] Achou carregamento de peixes!
Oceano	[Porto] Peixes disponíveis ainda no Porto: 15
pedroAux1	[Porto] Achou carregamento de peixes!
pedroAux2	[Porto] Peixes disponíveis ainda no Porto: 0
pedroCap1	[Porto] Achou carregamento de peixes!
pedroCap2	[Porto] Peixes disponíveis ainda no Porto: 0
pedroPesc1	[Porto] Recebendo carregamento de peixes do barco!
pedroPesc2	[Porto] Quantidade total de peixes no Porto: 500
Porto	[Porto] Recebendo carregamento de peixes do barco!
samuelCarr1	[Porto] Quantidade total de peixes no Porto: 1500
samuelCarr2	
samuelMot1	
samuelMot2	

Figura 19. Resultados do artefato Porto

A Figura 20 apresenta os resultados das execuções realizadas pelo Distribuidor.

MAS Console - peixaria	
all	[Distribuidor] Recebendo carregamento de peixes do caminhão!
Barco	[Distribuidor] Quantidade total de peixes no Distribuidor: 400
Caminhao	[Distribuidor] Recebendo carregamento de peixes do caminhão!
Distribuidor	[Distribuidor] Quantidade total de peixes no Distribuidor: 1400
JaCaMoLauncher	
Oceano	
pedroAux1	
pedroAux2	
pedroCap1	
pedroCap2	
pedroPesc1	
pedroPesc2	
Porto	
samuelCarr1	
samuelCarr2	
samuelMot1	
samuelMot2	

Figura 20. Resultados do artefato Distribuidor

A Figura 21 apresenta os resultados das execuções realizadas por um dos Capitães.

MAS Console - peixaria	
all	[pedroCap2] Focando no Barco: barco_02...
Barco	[pedroCap2] I am obliged to commit to m_navegacao_t1 on procPeixesbarco_02... doing so
Caminhao	[pedroCap2] ligando os motores para iniciar a navegacao em mar aberto...
Distribuidor	[pedroCap2] se preparando para localizar os cardumes no mar...
JaCaMoLauncher	[pedroCap2] cardume encontrado...
Oceano	[pedroCap2] I am obliged to commit to m_navegacao_t3 on descarrPeixesbarco_02... doing so
pedroAux1	[pedroCap2] navegando em direção ao Porto para descarregar...
pedroAux2	[pedroCap2] desligando os motores...
pedroCap1	
pedroCap2	
pedroPesc1	
pedroPesc2	
Porto	
samuelCarr1	
samuelCarr2	
samuelMot1	
samuelMot2	

Figura 21. Resultados do Capitão pedroCap2

A Figura 22 apresenta os resultados das execuções realizadas por um dos Pescadores.

MAS Console - peixaria	
all	[pedroPesc2] Quantidade de peixes pescados neste lançamento de redes: 50
Barco	[pedroPesc2] recolhendo as redes com os peixes pescados...
Caminhao	[pedroPesc2] Quantidade de peixes pescados neste lançamento de redes: 50
Distribuidor	[pedroPesc2] recolhendo as redes com os peixes pescados...
JaCaMoLauncher	[pedroPesc2] Quantidade de peixes pescados neste lançamento de redes: 50
Oceano	[pedroPesc2] recolhendo as redes com os peixes pescados...
pedroAux1	[pedroPesc2] Quantidade de peixes pescados neste lançamento de redes: 50
pedroAux2	[pedroPesc2] recolhendo as redes com os peixes pescados...
pedroCap1	[pedroPesc2] Quantidade de peixes pescados neste lançamento de redes: 50
pedroCap2	[pedroPesc2] recolhendo as redes com os peixes pescados...
pedroPesc1	[pedroPesc2] Quantidade de peixes pescados neste lançamento de redes: 50
pedroPesc2	[pedroPesc2] recolhendo as redes com os peixes pescados...
Porto	[pedroPesc2] Quantidade de peixes pescados neste lançamento de redes: 50
samuelCarr1	[pedroPesc2] recolhendo as redes com os peixes pescados...
samuelCarr2	[pedroPesc2] Quantidade de peixes pescados neste lançamento de redes: 50
samuelMot1	[pedroPesc2] recolhendo as redes com os peixes pescados...
samuelMot2	[pedroPesc2] Quantidade de peixes pescados neste lançamento de redes: 50

Figura 22. Resultados do Pescador pedroPes2

A Figura 23 apresenta os resultados das execuções realizadas por um dos Auxiliares de Pesca.

MAS Console - peixaria	
all	[pedroAux1] Focando no Barco: barco_01...
Barco	[pedroAux1] I am obliged to commit to m_auxilio_t1 on procPeixesbarco_01... doing so
Caminhao	[pedroAux1] puxando ancora para iniciar a navegacao em mar aberto...
Distribuidor	[pedroAux1] I am obliged to commit to m_auxilio_t3 on descarrPeixesbarco_01... doing so
JaCaMoLauncher	[pedroAux1] soltar ancora para iniciar a parada...
Oceano	[pedroAux1] retirando peixes do Barco...
pedroAux1	
pedroAux2	
pedroCap1	
pedroCap2	
pedroPesc1	
pedroPesc2	
Porto	
samuelCarr1	
samuelCarr2	
samuelMot1	
samuelMot2	

Figura 23. Resultados do Auxiliar de Pesca pedroAux1

A Figura 24 apresenta os resultados das execuções realizadas por um dos Motoristas.

MAS Console - peixaria	
all	[samuelMot1] Focando no Caminhao: caminhao_01...
Barco	[samuelMot1] I am obliged to commit to m_deslocamento_t1 on recolherPeixesPortocaminhao_01... doing so
Caminhao	[samuelMot1] dirigindo ate o porto para buscar carregamento de peixes...
Distribuidor	[samuelMot1] desligando o caminhao...
JaCaMoLauncher	[samuelMot1] I am obliged to commit to m_deslocamento_t2 on levarPeixesDistribuidorcaminhao_01... doing so
Oceano	[samuelMot1] ligando o motor do caminhao...
pedroAux1	[samuelMot1] verificando o melhor caminho ate o distribuidor...
pedroAux2	[samuelMot1] dirigindo ate o distribuidor...
pedroCap1	[samuelMot1] retirando peixes do Caminhão e descarregando no Distribuidor...
pedroCap2	
pedroPesc1	
pedroPesc2	
Porto	
samuelCarr1	
samuelCarr2	
samuelMot1	
samuelMot2	

Figura 24. Resultados do Motorista samuelMot1

A Figura 25 apresenta os resultados das execuções realizadas por um dos Carregadores.

MAS Console - peixaria	
all	[samuelCarr2] Focando no Caminhão: caminhao_02...
Barco	[samuelCarr2] I am obliged to commit to m_operacional_t1 on recolherPeixesPortocaminhao_02... doing so
Caminhao	[samuelCarr2] localizando o carregamento de peixes no porto...
Distribuidor	[samuelCarr2] carregando o caminhao com o carregamento de peixes...
JaCaMoLauncher	[samuelCarr2] Quantidade de peixes retirados do Porto e recolhidos para este Caminhao: 30
Oceano	[samuelCarr2] carregando o caminhao com o carregamento de peixes...
pedroAux1	[samuelCarr2] Quantidade de peixes retirados do Porto e recolhidos para este Caminhao: 30
pedroAux2	[samuelCarr2] preparando o Caminhao para ir em direção ao Distribuidor...
pedroCap1	
pedroCap2	
pedroPesc1	
pedroPesc2	
Porto	
samuelCarr1	
samuelCarr2	
samuelMot1	
samuelMot2	

Figura 25. Resultados do Carregador samuelCarr2

Conforme pode ser observado nos resultados da simulação, todos os agentes executam corretamente suas obrigações, as regras da organização são respeitadas, como o limite de agentes por missão e os artefatos do ambiente são utilizados e demonstram seu comportamento durante toda a simulação. Demais informações podem ser consultadas diretamente no sistema.

6. Considerações Finais

Durante o desenvolvimento deste trabalho, foram encontradas diversas dificuldades durante a criação da organização, ambiente e agentes. O pouco material existente para a criação da organização foi a maior das dificuldades. Todos os exemplos encontrados durante a pesquisa, eram exemplos ultrapassados para os requisitos exigidos para este trabalho. Durante a criação do arquivo JCM, foram encontrados diversos erros na sintaxe da programação, não havendo material que explicasse como este arquivo deveria ser criado. Os exemplos encontrados, faziam toda a criação dos esquemas, grupos, missões e etc, dentro dos próprios agentes. A ideia da criação do arquivo JCM para unificar as criações dos agentes, grupos, subgrupos e esquemas, é conveniente e na teoria, facilitaria o desenvolvimento, no entanto, pela falta de documentação de como criá-lo, não foi possível usufruir desta facilidade.

O desenvolvimento também trouxe grandes aprendizados para o autor na área e com isso, foi possível identificar alguns comportamentos divergentes, como quando uma missão é colocada como tipo “permission” para algum papel do sistema, esta missão nunca é executada, sendo necessário colocar sempre “obligation”, para que algum papel a execute.

Referente a utilização dos diagramas criados pela Daniela Uez, houve muita dificuldade em entender alguns deles, pois o material disponibilizado não detalhava o uso da grande maioria e quando foi solicitado contato para explicações, não houve retorno. Algumas observações encontradas durante a criação dos diagramas foram:

- Diagrama Atores - Objetivos: faltou a representação para objetivos que ocorrem em paralelo/ao mesmo tempo.
- Diagrama de artefatos: não ficou claro como que são representados/identificamos artefatos que são criados em tempo de execução.

Com tudo, a curva de aprendizado durante este desenvolvimento foi demasiadamente grande, pois o autor deste trabalho não é da área de multi-agentes e não possuía conhecimento nas linguagens de programação orientadas a agentes. Devido a pouca documentação existente para o auxílio no desenvolvimento, foi necessário um apoio constante de um integrante do grupo de pesquisa relacionado a multi-agentes, que já possuía algum conhecimento do processo. No entanto, o mesmo também não sabia responder a todos questionamentos, mas que pesquisando juntamente com o autor deste trabalho, conseguiu resolver grande parte dos requisitos propostos pelo domínio criado para este estudo.

Referências

- Alvares, L. O. and Sichman, J. S. (1997). Introdução aos sistemas multiagentes. In *XVII Congresso da SBC-Anais JAI'97*.
- Demazeau, Y. and Müller, J.-P. (1990). *Decentralized Ai*, volume 2. Elsevier.
- Hübner, J. F., Bordini, R. H., and Vieira, R. (2004). Introdução ao desenvolvimento de sistemas multiagentes com jason. *XII Escola de Informática da SBC*, 2:51–89.
- Rao, A. S. (1996). Agentspeak (I): Bdi agents speak out in a logical computable language. In *European Workshop on Modelling Autonomous Agents in a Multi-Agent World*, pages 42–55. Springer.