UNIVERSIDADE FEDERAL DE PELOTAS
INSTITUTO DE FÍSICA E MATEMÁTICA
DEPARTAMENTO DE INFORMÁTICA
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

UTILIZAÇÃO DA ABORDAGEM DE SISTEMAS MULTIAGENTES NO PROBLEMA DO CONTROLE DO ESPAÇO AÉREO

GUSTAVO OLIVEIRA DE ALMEIDA

PELOTAS

2007

GUSTAVO OLIVEIRA DE ALMEIDA

UTILIZAÇÃO DA ABORDAGEM DE SISTEMAS MULTIAGENTES NO PROBLEMA DO CONTROLE DO ESPAÇO AÉREO

Trabalho acadêmico apresentado ao Curso de Bacharelado em Ciência da Computação da Universidade Federal de Pelotas, como requisito parcial à obtenção do título de Bacharel em Ciência da Computação.

Orientador: Prof. Anderson Priebe Ferrugem, MSc.

PELOTAS

2007

Dados de catalogação na fonte:
Ubirajara Buddin Cruz – CRB-10/901
Biblioteca de Ciência & Tecnologia - UFPel

A447u **Almeida, Gustavo Oliveira de**

Utilização da abordagem de sistemas multiagentes no problema do controle do espaço aéreo / Gustavo Oliveira de Almeida ; orientador Anderson Priebe Ferrugem. – Pelotas, 2007. – 83f : il. color. - Monografia (Conclusão de curso). Curso de Bacharelado em Ciência da Computação. Departamento de Informática. Instituto de Física e Matemática. Universidade Federal de Pelotas. Pelotas, 2007.

1.Informática. 2.Inteligência artificial. 3.Sistemas multiagentes. 4.Prometheus. 5.Arquitetura BDI. 6.Controle de tráfego aéreo. 7.PDT. I.Ferrugem, Anderson Priebe. II.Título.

CDD: 006.3

BANCA EXAMINADORA:

_____

Profa. Eliane A. Diniz.

_____

Prof. Lucas F. Oliveira.

_____

Prof. Paulo Ricardo Prestes Porto.

_____

Prof. Anderson Priebe Ferrugem.

Dedicatória

Dedico a realização deste trabalho:

A Deus;

Aos meus pais Mário e Tânia;

Aos meus irmãos Leonardo e Débora;

A minha noiva Graciane.

Agradecimentos

Agradeço a Deus por me guiar nos momentos de dificuldade;

Ao professor Andersom Ferrugem pelo incentivo e orientação;

Ao meu pai Mário pelos ensinamentos;

A minha mãe Tânia por sempre zelar pelos meus estudos;

A minha noiva Graciane pelas palavras de conforto e incentivo;

Aos meus irmãos Leonardo e Débora pela paciência;

Ao Sr. Wladimir Feijó Pedrotti, coordenador do Aeroporto Internacional de Pelotas, e ao Sr. Airton José Mendes da Silva pela atenção e auxilio para o desenvolvimento deste trabalho.

“Envelhecer é obrigatório, Mas crescer é opcional!”

RESUMO

Sistemas Multiagentes, subárea da inteligência artificial distribuída, propõem a utilização de sociedades de agentes na solução de problemas complexos. Este trabalho tem como objetivo modelar um subconjunto do sistema de controle do espaço aéreo brasileiro, utilizando essa técnica. É aplicada a abordagem de sistemas multiagentes cognitivos, sendo a estrutura interna dos agentes baseada na arquitetura BDI. A aplicação foi feita utilizando a metodologia Prometheus, orientada a agentes, e a ferramenta Prometheus Design Tool para modelar o controle do espaço aéreo. A verificação do modelo se deu através do estudo de campo e modelos pré-concebidos.

Palavras-chave: Sistemas Multiagentes. Prometheus. Arquitetura BDI. Controle de tráfego aéreo. PDT.

ABSTRACT

Multi-agents Systems, sub-area of distributed artificial intelligence, consider the use of society of agents in the solution of complex problems. This work has as objective shape a subgroup of the Brazilian Airspace Control System, using this technique. The boarding of cognitivos multi-agent systems is applied, being the internal structure of the agents based in architecture BDI. The application was made using the Prometheus methodology, oriented-agent, and the tool Prometheus Design Tool shape the control of the airspace. The verification of the model if gave through the study of field and daily pay-conceived models.

Keywords: Multi-Agents Systems. Prometheus. BDI. Control of air traffic. PDT.

Lista de figuras

Figura 1 – Agente interage com o ambiente 19
Figura 2 – Uma Arquitetura de Agente BDI (WEISS, 1999). 25
Figura 3 – Pontos de vista de uma organização SMA 26
Figura 4 – interação por troca de mensagens 29
Figura 5 – Regiões de Informação de Vôo monitoradas pelos CInDACTAs 39
Figura 6 – Fluxo de comunicação dos agentes inseridos no ambiente 44
Figura 7 – Prometheus Design Tool (PDT) 36
Figura 8 – Notação PDT 37
Figura 9 – Código JACK gerado pela ferramenta PDT 37
Figura 10 – Diagrama de cenários 47
Figura 11 – Diagrama da visão geral dos Objetivos 49
Figura 12 – Diagrama dos Papéis do Sistema 50
Figura 13 – Diagrama de acoplamento de dados 54
Figura 14 – Diagrama de Agrupamento Agente-Papel. 56
Figura 15 – Diagrama de conhecimento dos agentes 57
Figura 16 – Diagrama Visão geral do Sistema 58
Figura 17 – Diagrama Comunicação 59
Figura 18 – Arquitetura interna do Agente Aeronave 61
Figura 19 – Arquitetura interna da capacidade Gerar Plano de Vôo 63
Figura 20 – Arquitetura interna da capacidade Definição NC. 69
Figura 21 – Arquitetura interna do Agente Controle 71
Figura 22 – Arquitetura interna da capacidade Aprovar Plano de vôo 73
Figura 23 – Arquitetura interna do Agente Aeroporto 76
Figura 24 – CrossCheck. 77

Lista de tabelas

Tabela 1 – Níveis de Cruzeiro 41
Tabela 2 – comparação das metodologias orientadas a agentes 31
Tabela 3 – comparação das metodologias orientadas a agentes 32
Tabela 4 – Painel de Mensagens 60

Lista de abreviaturas e siglas

BDI – *Beliefs, Desires and Intentions*.

PDT – *Prometheus Design Tool.*

IAD – Inteligência Artificial Distribuída.

IA – Inteligência Artificial.

RDP – Resolução Distribuída de Problemas.

SMA – Sistemas Multiagentes.

FIR – *Flight Information Region*.

CINDACTA – Centro Intergrado de Defesa Aérea e Controle de Tráfego Aéreo.

VFR – *Visual Flight Rules*.

IFR – *Instrument Flight Rules*.

FL – *Flight Level*.

RNAV – Rotas de Navegação de Área.

MASUP – *Multi-Agent System Unified Process*.

UML – *Unified Modeling Language*.

FIPA – *Fundation For Intelligente Physical Agents*.

ACL – *Agent Communication Language*.

KQML – *Knowledge Query Manipulation Language*.

RMIT – *Royal Melbourne Institute of Technology*.

AOS – *Agent-Oriented Software*.

RM – Rumo Magnético.

NC – Nível de Cruzeiro.

AUML – *Agent-based Unified Modeling Language.*

PV – Plano de Vôo.

JDE – *Jack Development Environment*

Sumário

1 Introdução..... 14
1.1 Motivação..... 15
1.2 Objetivos..... 16
1.3 Organização do Trabalho..... 16
2 Sistemas Multiagentes..... 18
2.1 Agentes..... 18
2.2 A abordagem de Sistemas Multiagentes..... 20
2.3 Sistemas Multiagentes Reativos..... 21
2.4 Sistemas Multiagentes Cognitivos..... 22
2.4.1 Arquitetura de Agentes..... 23
2.4.1.1 Arquitetura BDI..... 24
2.4.2 Organizações de agentes..... 25
2.4.2.1 Pontos de Vista Para uma Organização SMA..... 26
2.4.3 Interação entre agentes..... 27
2.4.3.1 Métodos de comunicação entre agentes..... 28
2.4.3.1.1 Quadro de avisos..... 28
2.4.3.1.2 Troca de mensagens..... 29
3 Metodologia Orientada a Agentes..... 31
3.1 Metodologia Prometheus..... 33
3.1.1 Prometheus Design Tools..... 35
4 Controle do Espaço Aéreo Brasileiro..... 39
5 Descrição do Problema Aéreo..... 43
5.1 Aplicação da abordagem de Sistemas Multiagentes no problema..... 43
5.2 Breve histórico de outras aplicações usadas neste problema..... 37
6 Modelagem do Controle do Espaço Aéreo..... 46
6.1 Especificação do Sistema..... 46
6.1.1 Cenários..... 46
6.1.2 Visão Geral dos Objetivos..... 48
6.1.3 Funcionalidades do Sistema..... 50
6.2 Desenvolvimento da Arquitetura..... 53

6.2.1 Acoplamento de Dados..... 53
6.2.2 Agrupar Agente/Papel..... 56
6.2.3 Conhecimento do Agente..... 57
6.2.4 Visão Geral do Sistema..... 57
6.3 Desenvolvimento detalhado..... 60
6.3.1 Agente Aeronave..... 60
6.3.2 Agente Controle..... 71
6.3.3 Agente Aeroporto..... 76
6.4 Crosscheck..... 77
7 Conclusão..... 79
7.1 Trabalhos futuros..... 81
Referências..... 82
Anexos..... 84

1 Introdução

> Inteligência Artificial Distribuída (IAD) surgiu como uma subárea de Inteligência Artificial (IA). A IAD está relacionada a sistemas que consistem em múltiplas entidades independentes que interagem em um domínio. Tradicionalmente, IAD foi dividido em duas subáreas: Resolução Distribuída de Problemas (RDP) e Sistemas Multiagentes (SMA). (STONE; VELOSO, 2006).

A abordagem de Sistemas Multiagentes (SMA) é uma subárea da Inteligência Artificial Distribuída (IAD) que é baseada em estudos de comportamento social, destacando a comunicação e interação entre agentes, tendo uma concepção de natureza sociológica[1] e ou etológica[2] (ALVARES; SICHMAN, 1997).

Tem-se na vida real algumas situações onde pode-se dizer que há um comportamento inteligente de entidades coletivas, como, por exemplo, uma empresa competitiva num determinado segmento de mercado, uma equipe de futebol que vence a maioria de seus jogos ou uma colônia de formigas em busca de alimento.

Quando se quer resolver problemas complexos e grandes, que necessitam de conhecimento de vários domínios distintos e que algumas vezes envolvam coleta de dados fisicamente distribuídos, como, por exemplo, em sistemas de controle de espaço aéreo, torna-se desejável utilizar uma abordagem SMA. Do ponto de vista estritamente computacional, uma abordagem distribuída permite o controle da complexidade e suporte para evoluções e eventuais replicações como diz Chandrasekaran (1981 apud ALVARES; SICHMAN, 1997).

---

[1] Sociologia é o ramo da ciência que estuda o comportamento humano em função do meio e os processos que interligam o indivíduo em associações, grupos e instituições. (SOCIOLOGIA, 2007).

[2] Etologia (do grego *éthos:* costume, uso, hábito) é a ciência que se ocupa com o estudo do comportamento social e individual dos animais. Segundo Ardans (1996): "Pode-se definir a etologia como a aplicação, ao estudo do comportamento, de tudo o que foi obtido pela investigação biológica desde Darwin".

Foi proposta uma decomposição básica de um sistema segundo uma metodologia de Inteligência Artificial Distribuída por Demazeau (1995), cujo, os principais conceitos desta decomposição são dados a seguir:

> Dado um determinado sistema, denomina-se *agente* cada uma de suas entidades ativas. O conjunto de agentes forma uma *sociedade*. As entidades passivas serão designadas pelo termo *ambiente.* Um agente raciocina sobre o ambiente, sobre os outros agentes e decidem racionalmente quais objetivos deve perseguir, quais ações devem tomar etc. (PLEIAD, 1992 apud ALVARES; SICHMAN, 1997).

Sendo que o termo ativo tem como objetivo sinalizar que o conceito de agente não corresponde a noções estáticas como módulos, conjunto de regras e bases de conhecimento, sem que a estas estejam intimamente relacionado um mecanismo de controle para a sua ativação.

A troca de informações que ocorrem entre os agentes dá-se o nome de interação. Podendo essas trocas serem realizadas de forma direta ou indireta.

Tendo por fim, o conceito de organização que exprime as restrições aplicadas aos agentes pertencentes a uma determinada sociedade, ou seja, é como os projetistas de tais sistemas podem garantir que cada agente desejará realizar e realizará o que pretende e no momento certo.

Segundo Jonker, Meyer e Dignum (2005), O planejamento do Controle de tráfego aéreo é uma área de pesquisa muito complexa na qual há uma grande necessidade de novas e eficientes técnicas de coordenação. A estreita conexão entre várias entidades com interesses diferentes faz desta uma área típica e desafiadora para a aplicação de técnicas de Sistemas Multiagentes.

1.1 Motivação

Agregar tecnologia de Sistema Multiagentes no controle do tráfego aéreo é uma forma de produzir ganhos de produtividade para o ambiente. Além do mais, esse trabalho consolida conhecimentos adquiridos no curso de Ciência da Computação tais como Estrutura de dados, protocolos de comunicação, técnicas de descrição de problemas, banco de dados, engenharia de software, análise de requisitos, análise de sistemas entre outros.

Pelo fato de vários pesquisadores da área de Inteligência Artificial terem um largo interesse pela subárea de Sistemas Multiagentes, faz com que essa abordagem seja de extrema relevância e importante na resolução de problemas de sistemas muito complexos.

1.2 Objetivos

O objetivo principal deste trabalho é utilizar a abordagem de sistemas Multiagentes, para fazer a modelagem parcial do controle do espaço aéreo do tipo centralizado usado no Brasil, para evitar colisões entre as aeronaves.

Analisar a ferramenta Prometheus Design Tool (PDT) e a metodologia orientada a agentes Prometheus que foram utilizadas para modelar o sistema para resolver o problema do controle do espaço aéreo.

Com a conclusão deste trabalho espera-se chegar a uma convergência ainda maior entre a tecnologia de sistemas Multiagentes e o controle do espaço aéreo, servindo como base para futura simulação ou implementação (parcial ou completa) utilizando sistemas Multiagentes no tratamento do problema do controle do espaço aéreo.

1.3 Organização do trabalho

O próximo capítulo apresenta aspectos da área de Sistemas Multiagentes, bem como conceitos sobre agentes. Destacando características da arquitetura BDI, organização e comunicação entre agentes.

O capítulo 3 exemplifica como se dá o controle do espaço aéreo do tipo centralizado como é feito no Brasil e suas características bem como a descrição do problema do controle do espaço aéreo.

O capítulo 4 apresenta como é feita a aplicação da abordagem de sistemas multiagentes no problema do controle do espaço aéreo, bem como suas convergências.

O capítulo 5 apresenta algumas propostas de aplicações da abordagem de sistemas multiagentes no problema do controle do espaço aéreo descentralizado, utilizando-se do modelo de vôo livre.

No capítulo 6 mostra a comparação de metodologias orientadas a agentes, destacando o detalhamento da metodologia Prometheus que foi utilizada para a realização do trabalho.

No capítulo 7 apresenta a ferramenta Prometheus Design Tool que foi utilizada para a modelagem do sistema, bem como suas notações e funcionamento.

No capítulo 8 mostra a modelagem do sistema de acordo com a metodologia Prometheus, destacando suas fases e o detalhamento dos agentes e entidades modeladas.

No capítulo 9 são apresentadas as conclusões geradas por este trabalho, bem como as considerações finais e os objetivos alcançados.

2 Sistemas Multiagentes

Segundo Demazeau e Müller (1990 apud ALVARES; SICHMAN, 1997), “A área de Sistemas Multiagentes (SMA) se interessa pelo estudo de agentes *autônomos* em um universo Multiagentes”. Autonomia significa que os agentes operam sem intervenção do usuário e possuem controle sobre suas próprias ações.

2.1 Agentes

Para entendermos a abordagem de Sistema Multiagentes é preciso ter conhecimento de alguns conceitos de agentes. Não existe uma definição única do conceito de agentes que seja aceita por todos os pesquisadores da área de Inteligência Artificial, porque algumas características de um agente têm sua importância infiltrada ao contexto de sua aplicação. Com isso, encontra-se a seguir duas definições genéricas que abordam aspectos complementares deste conceito:

> Um agente é uma entidade *real* ou *virtual*, imersa num *ambiente* sobre o qual ela é capaz de agir, que dispõe de uma capacidade de *percepção* e de *representação* parcial deste ambiente, que pode se *comunicar* com outros agentes, e que possui um *comportamento autônomo*, conseqüência de suas observações, de seu conhecimento e das suas interações com os outros agentes (FERBER; GASSER, 1991).
>
> Um agente é uma entidade à qual nós podemos associar uma *identidade única,* e que é capaz de executar um processamento de cálculo. Um agente pode ser considerado como um *meio* que produz um certo número de ações, a partir de seu conhecimento e dos mecanismos internos que lhe são próprios (GASSER, 1992).

Um agente possui sensores que percebem ação do ambiente e agem através de atuadores de acordo com sua percepção e de algumas regras de condição, como ilustra a Fig. 1.

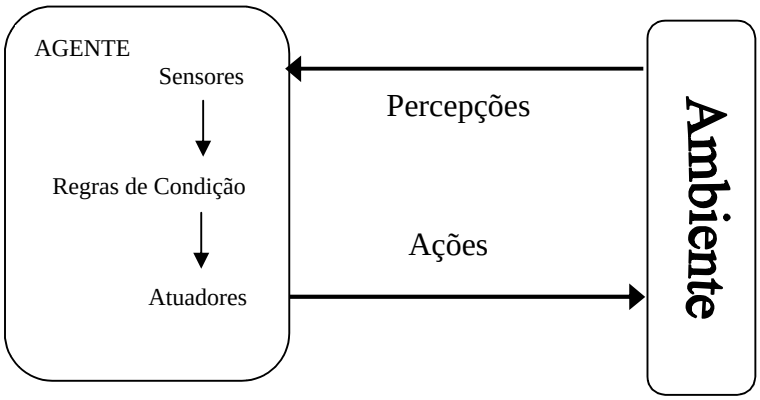


Figura 1 – Agente interage com o ambiente

Os agentes possuem características de acordo com seu comportamento e dependendo da forma com que esses agentes estão inseridos no contexto, essas características podem aparecer em menor ou maior grau. Tendo assim tais características:

- Adaptabilidade: Um agente pode mudar seu comportamento para melhor se adaptar a seu ambiente.
- Autonomia: Capacidade de o agente operar sem intervenção direta de um usuário, e também possuir um certo controle sobre suas ações.
- Interatividade: Interação entre agentes ou de agentes com os usuários através de uma linguagem de comunicação de agentes.
- Mobilidade: Um agente pode ser capaz de migrar de uma plataforma para outra.
- Pró-Atividade: Agentes devem ser capazes de planejar seus atos e não simplesmente reagir a estímulos externos.
- Racionalidade: Entre diversas opções de ações a disposição de um agente em um determinado momento, ele deve ser capaz de escolher a que julgar melhor.
- Reatividade: Agentes percebem os estímulos externos e respondem com uma ação respeitando seu comportamento.
- Sociabilidade: Define se um agente é cooperativo (benevolente, altruísta) ou competitivo (egoísta).

Muitos desses conceitos são válidos quando analisados no contexto de sociedade de agentes, ou seja, Sistemas Multiagentes. Podendo em determinada

aplicação uma característica ser mais importante que as outras, fazendo com que um agentes não precise conter todas essas características.

2.2 A abordagem de Sistemas Multiagentes

Um Sistema Multiagente consiste em vários agentes autônomos que interagem entre si e trabalham juntos para alcançarem um objetivo em comum.

Os agentes inseridos nesse sistema podem ser homogêneos, utilizam a mesma arquitetura sendo que replicada para cada agente, ou heterogêneos utilizam arquiteturas distintas.

Do ponto de vista de concepção de sistemas a abordagem de Sistemas Multiagentes apresenta as seguintes características, segundo Alvares e Sichman, (1997):

a) Os agentes são concebidos independentemente de um problema particular a ser resolvido. O projeto de um agente deve resultar numa entidade capaz de realizar um determinado processamento, e não numa entidade capaz de realizar este processamento exclusivamente no contexto de uma aplicação alvo particular;
b) A concepção das interações também é realizada independentemente de uma aplicação alvo particular. Busca-se desenvolver protocolos de interação genéricos, que possam ser reutilizados em várias aplicações similares. Um exemplo de um protocolo seria, por exemplo, um protocolo de apresentação de um agente quando este ingressa numa sociedade. Obviamente, um protocolo deverá ser instanciado com dados do domínio do problema para poder ser efetivamente utilizado numa aplicação;
c) A mesma filosofia anterior pode ser estendida ao projeto das organizações. Normalmente, se distinguem as funcionalidades necessárias a uma resolução particular dos agentes que irão efetivamente implementar tais funcionalidades;
d) Durante a resolução, os agentes utilizam suas representações locais dos protocolos de interação e das organizações para raciocinar e agir.

> Deste modo, não existe um controle global do sistema, este é implementado de forma totalmente descentralizada nos agentes.

Conforme Alvares e Sichman, (1997), “do ponto de vista de concepção do sistema, os agentes, suas interações e sua organização no seio da sociedade são concebidas independentemente de um problema particular a ser resolvido”. Podendo reutilizar tais componentes para projetar aplicações similares. Quando um problema for apresentado ao sistema os agentes irão instanciar dinamicamente as organizações e interações

Segundo Bond e Gasser (1988), os principais problemas na área de Sistemas Multiagentes são os seguintes:

- *Descrição, decomposição e alocação de tarefas*: como se deve descrever e decompor uma tarefa complexa em sub-tarefas;
- *Interação, linguagem e comunicação*: quais primitivas básicas de comunicação devem ser utilizadas para exprimir os conceitos semânticos envolvidos num trabalho cooperativo e como estas primitivas podem ser combinadas em protocolos de resolução de problemas;
- *Coordenação, controle e comportamento coerente*: como se pode garantir um comportamento global coerente numa coleção de agentes, cada qual com seus próprios objetivos e habilidades, e como um controle de tal sistema pode ser projetado;
- *Conflito e incerteza*: considerando que os agentes não têm uma informação completa sobre o seu ambiente, como os conflitos que surgem a partir de visões locais podem ser detectados e resolvidos e como dados incertos e incompletos devem ser manipulados de modo a garantir resultados globais coerentes;

2.3 Sistemas Multiagentes Reativos

Os agentes reativos não possuem representação de seu ambiente e são muito simples. Assim, todas as informações relativas ao seu comportamento estão no ambiente e suas reações dependem unicamente de sua percepção deste

ambiente. Os agentes, por não possuírem registros de suas ações não podem planejar o futuro. Dando, com isso, mais ênfase ao trabalho em grupo, fazendo com que juntos realizem trabalhos que não poderiam fazer sozinhos.

Segundo Alvares e Sichman, (1997) têm os agentes e sistemas Multiagentes reativos algumas características, tais como:

a) Não há representação explícita de conhecimento - o conhecimento dos agentes é implícito e se manifesta através do seu comportamento;
b) Não há representação do ambiente - o seu comportamento se baseia no que é percebido a cada instante do ambiente, mas sem uma representação explícita deste;
c) Não há memória das ações - os agentes reativos não mantém um histórico de suas ações, de forma que o resultado de uma ação passada não exerce nenhuma influência sobre as suas ações futuras;
d) Organização etológica - a forma de organização dos agentes reativos é similar a dos animais;
e) Grande número de membros - os sistemas Multiagentes reativos têm, em geral, um grande número de agentes, da ordem de dezenas, centenas ou mesmo milhões de agentes.

Nos modelos de sistemas Multiagentes reativos, cada agente tem seus próprios objetivos individuais, mas possuem interações entre eles.

2.4 Sistemas Multiagentes Cognitivos

As organizações humanas, e suas hierarquias, dinâmica de mercados e grupos são a inspiração estrutural para os Sistemas Multiagentes Cognitivos.

Tendo, segundo Ferber e Gasser (1991), os agentes cognitivos, as seguintes características:

- Mantêm uma representação *explícita* de seu ambiente e dos outros agentes da sociedade;

- Podem manter um *histórico* das interações e ações passadas, isto é, têm memória do passado;
- A comunicação entre os agentes é feita de modo *direto*, através do envio e recebimento de mensagens;
- Seu mecanismo de controle é *deliberativo*, ou seja, tais agentes raciocinam e decidem sobre quais objetivos devem alcançar, que planos seguir e quais ações devem ser executadas num determinado momento;
- Seu modelo de organização é baseado em modelos *sociológicos*, como as organizações humanas;
- Uma sociedade contém tipicamente poucos agentes, na ordem de uma dezena.

De acordo com o aumento da complexidade do tratamento interno e dos conceitos que se encontram explicitamente representados no interior das diversas classes de agentes cognitivos, são descritas, a seguir, em ordem de complexidade decrescente e inclusiva de acordo com Demazeau (1995), tais classes:

- Agentes *organizados*: mantém múltiplas perspectivas sobre um determinado problema, obedecem a leis e regras sociais;
- Agentes *negociantes*: contemplam um processo de resolução de conflitos por negociação;
- Agentes *intencionais*: representam internamente noções como intenções, engajamentos, e planos parciais;
- Agentes *cooperativos:* contém representações mútuas uns dos outros, existe um esquema de alocação de tarefas;
- Módulos comunicantes: utilizam protocolos de comunicação;
- Atores, processos: implementam primitivas de comunicação.

2.4.1 Arquitetura de Agentes

A estrutura interna básica de uma arquitetura de um agente cognitivo deve contemplar ao menos módulos responsáveis pelos mecanismos de comunicação,

percepção, raciocínio e ação e como esses módulos devem interagir. Sendo que o módulo de raciocínio envolve dois processos importantes: decidir que metas serão alcançadas, e como serão alcançadas estas metas. Os módulos e suas interações fazem com que os dados recebidos do ambiente e o estado interno do agente determinem suas ações (MAE, 1995 apud WOOLDRIDGE; JENNINGS, 1995). E o agente fará o mapeamento de uma percepção em uma ação conforme sua arquitetura.

2.4.1.1 Arquitetura BDI

A Arquitetura BDI – *Beliefs* (crenças)*, Desires* (desejos) *and Intentions* (intenções) é baseada em estados mentais. Os estados mentais humanos possuem um vínculo com o mundo através do qual estabelecem sua existência e significância.

Como dizem Russel e Norvig (1995), tal arquitetura tem como raciocínio o processo de decidir antes do momento de executar as ações adiantando as próprias metas. Essa arquitetura é baseada nas Crenças, Desejos e Intenções dos agentes. No ponto de vista de sistemas, a ação se dá de acordo com a percepção (estimulo do ambiente) e com raciocínio, que é uma manipulação da estrutura de dados que representam tais Crenças, Desejos e Intenções.

A *crença* é uma visão geral do ambiente em que o agente está inserido, esta visão é usada para expressar as expectativas do estado futuro desse ambiente. A *crença* é como um informativo do sistema em um determinado estado, ou seja, podem ser vistas como um provável estado de um ambiente. Diferentemente dos *desejos*, que representam os estados desejáveis que o sistema poderia representar, levando os agentes a agir de forma a realizar metas, tais ações são realizadas através das *intenções* causadas pelos *desejos*. Assim as *intenções* são um subconjunto dos *desejos*. Sendo a *intenção* uma meta que o agente *deseja* seguir.

Nessa arquitetura as *Intenções* fazem um papel crucial no processo de raciocínio, levando tais intenções a terem a tendência de conduzir a ação do agente.

A Arquitetura BDI é descrita a seguir de acordo com Weiss (1999), tendo como base as crenças atuais do agente e a informação atual do ambiente:

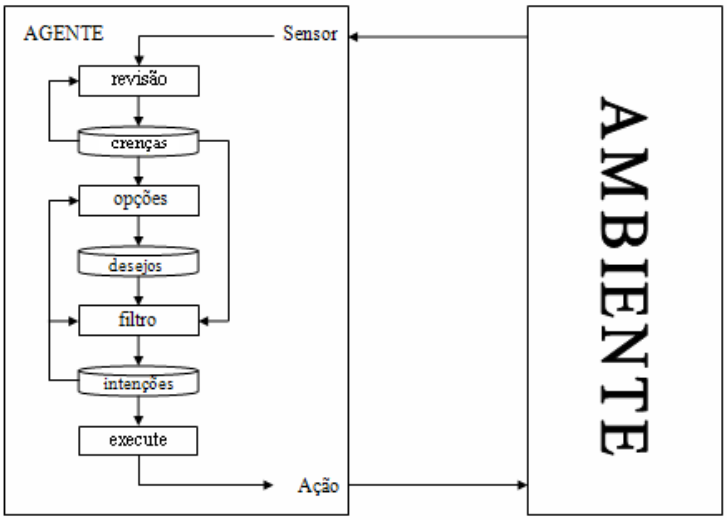


Figura 2 – Uma Arquitetura de Agente BDI (WEISS, 1999).

- Uma função de revisão de Crenças (revisão), que tem como base a percepção atual do ambiente e as Crenças atuais do agente, podendo determinar novas Crenças;
- Uma função de geração de opção (opções), que determina as opções disponíveis para o agente (seus desejos), em base de suas Crenças atuais sobre seu ambiente e suas intenções atuais;
- Um conjunto de opções atuais, representando possíveis escolhas de ações, disponível para o agente;
- Uma função de filtro (filtro) que representa o processo de deliberação do agente e que determina as intenções do agente tendo como base suas Crenças, desejos e intenções, atuais;
- Um conjunto de intenções atuais, representando o foco.
- Uma função de seleção de ação (execute) que determina uma ação para executar com base nas intenções atuais.

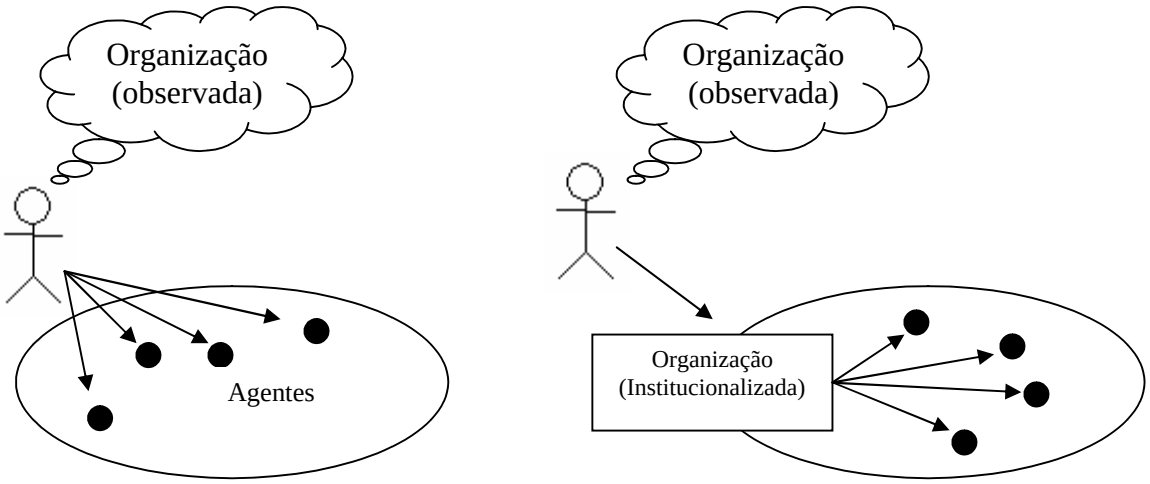
2.4.2 Organizações de agentes

O conceito de Organização condiz com as limitações aplicadas a um agente que pertence a uma determinada sociedade, ou seja, os meios, utilizados pelos projetistas de um determinado sistema, que garantem o que cada agente quer ou deseja realizar em um determinado momento.

Segundo Alvares e Sichman (1997), “uma organização de agentes é concebida para restringir o comportamento destes agentes e tal organização, na maioria dos casos, é completamente definida durante a fase de concepção do sistema.”

2.4.2.1 Pontos de Vista Para uma Organização SMA

Uma organização, observada por um agente ou um observador externo, pode ter: uma visão centrada nos agentes ou uma visão centrada na organização, conforme sugere Lemaitre e Excelente (1998 apud HÜBNER, 2003).



(a) Visão centrada nos agentes (b) Visão centrada na organização

Figura 3 – Pontos de vista de uma organização SMA

Na visão centrada nos agentes existe uma organização somente no interior dos agentes, não tendo assim o observador um acesso a ela, ou seja, a organização não é representada explicitamente, podendo com isso, apenas criar uma visão subjetiva da organização através da observação do comportamento do sistema. Sendo essa descrição subjetiva da organização chamada de O*rganização Observada*.

Já na visão centrada na organização, além da organização observada, tem-se uma representação externa da organização, ou seja, ela é representada explicitamente sem precisar observar o comportamento de determinada sociedade

ou dos agentes que a compõem. Sendo essa descrição mais objetiva de organização chamada de *Organização Institucionalizada*.

> Se do ponto de vista de um observador externo ao SMA podem existir duas situações, do lado dos agentes que formam a sociedade também podem ser concebidas duas situações: uma onde os agentes são capazes de explicitamente representar uma organização e outra onde os agentes não são capazes de fazê-lo. (HÜBNER, 2003).

Portanto, para conhecer a organização de um sistema o observador pode consultar a descrição da mesma ou construir uma descrição subjetiva através de observação.

2.4.3 Interação entre agentes

Na maioria dos trabalhos a comunicação entre agentes cognitivos se baseia na teoria de atos de fala (“speech act theory”). Essa teoria diz que, comunicar é agir: trata-se de uma ação regular como qualquer outra que os agentes devem gerar e processar (SEARLE, 1969 apud ALVARES; SICHMAN, 1997).

Tais atos de fala podem ser classificados segundo três aspectos:

- Ato *locucionário*: emissão de palavras e sentenças com algum significado;
- Ato *ilocucionário*: corresponde à *intenção* da emissão: pedido, ordem, pedido de desculpas, afirmação;
- Ato *perlocucionário*: corresponde ao resultado (desejado) da emissão: convença, insulte, amedronte.

A estrutura dos atos de fala se dá através de protocolos de comunicação que permitem os agentes trocar e entender mensagens. Esses protocolos permitem os agentes terem conversações que são trocas de mensagens estruturadas.

Com isso, pode-se especificar os seguintes tipos de protocolo de comunicação entre agentes:

- Propor uma ação;
- Aceitar uma ação;
- Rejeitar uma ação;
- Discordar com uma ação proposta.

Baseado nisso, é apresentado um exemplo a seguir de um protocolo de interação de negociação entre dois agentes:

Agente1: Propõe uma ação ao Agente2,

Agente2: Avalia a proposta e envia aceitação ao Agente1,

ou,

envia contra proposta ao Agente1,

ou,

envia discordância ao Agente1,

ou,

envia rejeição ao Agente1.

Os protocolos são concebidos com um caráter genérico, podendo assim, um determinado protocolo desenvolvido para uma aplicação poder ser utilizado em outras aplicações.

2.4.3.1 Métodos de comunicação entre agentes

Em uma sociedade de agentes, a comunicação entre eles é de extrema importância, pois cada um tem uma tarefa específica e o objetivo final depende da junção de todas as operações realizadas. Sendo assim é imprescindível que se tenha uma forma para que possam transmitir suas informações e avisos.

Segundo Brenner (1998), os métodos de comunicação podem ser divididos em dois grupos: quadro de avisos e troca de mensagens.

2.4.3.1.1 Quadro de Avisos

Nesse método os agentes podem trocar informações, conhecimentos e dados através de um local compartilhado por todos. É também conhecido como quadro negro.

Inicia-se uma ação sempre que um agente inserir uma informação nesse local compartilhado por todos, ou seja, quando uma mensagem for escrita no quadro

negro. Podendo os agentes acessar essas informações a qualquer momento e verificando se há alterações ou novas informações.

Um agente filtra e extrai todas informações que afetam suas ações, pois ele não lê todas as mensagens que estão inseridas ou escritas no quadro de avisos.

Um SMA pode conter vários quadros negros que podem ser administrados com acesso restrito a algum grupo de agentes que estão registrados como não autorizados a acessar as informações.

“Quando um sistema possui um grande número de agentes, a quantidade de informações escritas no quadro negro cresce exponencialmente e, portanto, a busca por informações relevantes se torna cada vez mais pesada” (BRENNER, 1998).

Para resolver isso, dividem-se as áreas dos quadros negros em sub-áreas, cada qual alocada a um agente ou a um grupo deles. Sendo assim, um agente precisará monitorar apenas a região a ele alocada.

Segundo Brenner (1998), uma sobrecarga do sistema pode ser causada pela leitura de informações no quadro de avisos.

Com isso, os sistemas baseados em troca de mensagens vêem sendo mais utilizados.

2.4.3.1.2 Troca de Mensagens

A Fig. 4 mostra o conceito básico do sistema de troca de mensagens. Um agente, emissor, envia uma mensagem direta a um outro agente, o receptor.

Na troca de mensagens as informações são passadas de forma direta entre os agentes, diferentemente do método do quadro negro.

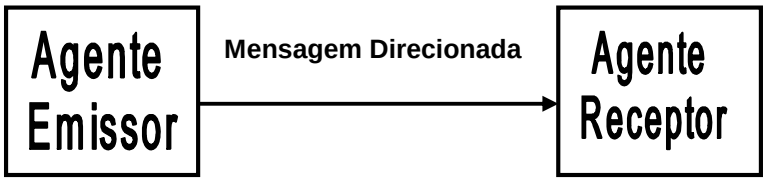


Figura 4 – interação por troca de mensagens

Para Finin (1997) a troca de mensagens pode ser feita de três formas:

- *Broadcast*: A mensagem é enviada para todos os agentes da sociedade. Nessa abordagem, um agente pode ser substituído por outro equivalente

e o processamento do sistema será inalterado. A passagem de mensagens por broadcast não é segura, já que qualquer agente pode examinar o conteúdo de qualquer mensagem.

- *Multicast*: A sociedade de agente é dividida em grupos. Desta forma as mensagens podem ser enviadas somente aos agentes de determinados grupos.
- Ponto-a-ponto: As mensagens são enviadas para um agente específico que deve ser conhecido pelo emissor. A vantagem dessa abordagem é que o agente emissor sempre sabe para onde uma mensagem está sendo enviada, permitindo controles de segurança.

A maioria dos SMAs utilizam os sistemas baseados em troca de mensagens, por proporcionarem mais flexibilidade e dinamismo do que a arquitetura do quadro de avisos apresentada anteriormente.

3 Metodologia Orientada a Agentes

A metodologia escolhida foi baseada em uma pesquisa realizada por Santos, Ribeiro e Bastos (2006), que fizeram a comparação de quatro metodologias, Multi-Agent Systems Unified Process (MASUP), de Bastos e Ribeiro (2005 apud SANTOS; RIBEIRO; BASTOS, 2006), Tropos, de Castro et al.(2001 apud SANTOS; RIBEIRO; BASTOS, 2006), Multi-agent Systems Engineering (MaSE), de Deloach et al. (2001 apud SANTOS; RIBEIRO; BASTOS, 2006) and Prometheus, de Padgham e Winikoff (2002 apud SANTOS; RIBEIRO; BASTOS, 2006), para desenvolvimento de sistemas Multiagentes. Esta comparação foi feita baseada no resultado de experiências utilizando tais metodologias.

Essas metodologias foram aplicadas para desenvolver dois sistemas de agentes diferentes e logo a seguir foram feitas as comparações de acordo com alguns critérios, como descrito na Tab. 2 e na Tab. 3.

Tabela 1 – comparação das metodologias orientadas a agentes

Característica	**MASUP**	**Tropos**	**MaSE**	**Prometheus**
Definição de processos.	Satisfaz.	Satisfaz parcialmente. Processo evolutivo, incremento parcial, não interage e pouco definida.	Satisfaz.	Satisfaz.
Uso de UML.	Satisfaz.	Satisfaz parcialmente. Uso do diagrama de estereótipo de uma forma inadequada.	Não satisfaz.	Não satisfaz.
Agencia as características de captura.	Não satisfaz.	Não satisfaz.	Não satisfaz.	Não satisfaz.

Tabela 2 – comparação das metodologias orientadas a agentes

Característica	MASUP	Tropos	MaSE	Prometheus
Estrutura organizacional.	Satisfaz parcialmente. Habilita relações de hierarquia fixas entre agentes, mas só em um nível de hierarquia, como relações de autoridade.	Satisfaz parcialmente. A estrutura organizacional depende do estilo de arquitetura escolhida e de algum padrão social que será usado.	Satisfaz parcialmente. Relações de hierarquias não são tratadas.	Satisfaz parcialmente. Relações de hierarquias não são tratadas.
Traçado de modelos.	Satisfaz parcialmente. A consistência é mantida entre as fases, mas não existe um processo detalhado que garanta a consistência.	Não satisfaz.	Não satisfaz.	Satisfaz.
Capturam exigências.	Satisfaz.	Satisfaz.	Satisfaz.	Satisfaz.
Identificação de agente.	Não satisfaz.	Não satisfaz.	Não satisfaz.	Não satisfaz.
Interação de agentes.	Satisfaz.	Satisfaz.	Satisfaz.	Satisfaz.
Modelo interno de agentes.	Não satisfaz.	Não satisfaz.	Não satisfaz.	Satisfaz.
Plataforma de implementação.	Satisfaz. Não faz ligação do processo a uma plataforma de implementação.	Satisfaz. Não faz ligação do processo a uma plataforma de implementação.	Satisfaz. Não faz ligação do processo a uma plataforma de implementação.	Satisfaz. Não faz ligação do processo a uma plataforma de implementação.
Processo organizacional Workflow.	Não satisfaz.	Não satisfaz.	Não satisfaz.	Não satisfaz.
Estrutura de mensagem.	Satisfaz. Usa FIPA ACL[3].	Satisfaz. Sugere uso de padrões de mensagem como FIPA ACL e KQML[4].	Satisfaz parcialmente. Uso de um padrão semelhante ao FIPA ACL.	Não satisfaz.

[3] FIPA-ACL: (FIPA – *Agent Communication Language*), linguagem de comunicação de agentes.
[4] KQML (*Knowledge Query and Manipulation Language*) que é uma linguagem e um protocolo para troca de informações e conhecimento entre agentes.

O trabalho foi desenvolvido utilizando a metodologia Prometheus, por ter característica de modelo interno de agentes. E conforme Santos, Ribeiro e Bastos (2006), “Prometheus é focalizado em sistemas que usam arquitetura de agentes BDI.”

Além de ser mais detalhada e completa, pois cobre todos os estágios de desenvolvimento de software aplicado a sistemas baseados em agentes.

3.1 Metodologia Prometheus

A metodologia Prometheus foi desenvolvida por Lin Padgham e Michael Winikoff (RMIT University, Austrália) em colaboração com o Grupo de Software Orientado a Agentes (AOS), e foi idealizada para se obter um processo que pudesse ser utilizado para desenvolver agentes inteligentes (SANTOS; RIBEIRO; BASTOS, 2006).

Características dessa metodologia:

- Suporta o desenvolvimento de Agentes inteligentes;
- Suporta todo o processo de desenvolvimento;
- Evoluiu a partir de experiências práticas e pedagógicas;
- Tem sido utilizada tanto no meio acadêmico como industrial;
- Detalhada e Completa;
- Possui ferramentas que possibilitam o *cross checking* entre modelos.

A metodologia Prometheus consiste em três fases: Especificação do Sistema, Desenvolvimento da Arquitetura e Desenvolvimento Detalhado.

A fase de Especificação de Sistema identifica as funcionalidades básicas do sistema, as entradas, as saídas e as fontes de dados compartilhados que tenham importância. Esta fase consiste das seguintes atividades:

- Desenvolver cenários de casos de uso: os cenários dão uma ampla visão do sistema. A parte central de um cenário é uma seqüência de passos que descreve o sistema em operação;
- Identificar objetivos: definição dos objetivos e sub-objetivos do sistema;

- Identificar funcionalidades: as funcionalidades se referem a aspectos particulares do sistema e devem estar relacionadas aos objetivos e sub-objetivos. As informações de entrada e saída de cada funcionalidade devem ser explicitadas;
- Identificar a interface do sistema: definição da interface do sistema com o ambiente em termos de ações, percepções e dados externos;
- Preparar esquemas: tanto para as funcionalidades como para os objetivos, esquemas devem ser criados incluindo suas características e informações sobre interação com outras funcionalidades e objetivos.

A fase de Desenvolvimento da Arquitetura utiliza os resultados obtidos na fase anterior e tem por finalidade determinar os agentes que deverão existir e como eles irão interagir. Esta fase possui as seguintes atividades:

- Agrupar funcionalidades para determinar os tipos de agentes: este processo analisa as razões do agrupamento de funcionalidades em particular. Funcionalidades que lidam com os mesmos dados podem ser compostas por um mesmo agente. Já funcionalidades claramente exclusivas não deverão ser agrupadas.
- Definição dos tipos de agentes e desenvolvimento dos descritores de agentes: ao definir os tipos de agentes deve-se buscar uma forte coesão interna e um baixo acoplamento com os outros tipos. Definindo os tipos deve-se escrever os descritores contendo informações de alto nível como: número de agentes deste tipo; a quais eventos o agente irá reagir; dados que devem ser monitorados; tempo de vida do agente; requisitos de inicialização; requisitos de terminação e mais algumas informações extraídas das funcionalidades que compõe o agente como: objetivos, eventos, percepções, ações e etc.
- Produção de um diagrama da visão geral da sua estrutura do sistema: nesta produção há o relacionamento dos agentes, eventos e objetos de dados compartilhados. Através deste diagrama tem-se uma compreensão geral do funcionamento do sistema como um todo, incluindo como os agentes interagem.

- Desenvolvimento de protocolos de interação a partir dos cenários de casos de uso: no final desta fase é feita a especificação completa das interações entre os agentes. Os protocolos de interação são definidos a partir dos Diagramas de interação que devem ser criados.

E, na terceira e última fase, a de Desenvolvimento Detalhado vai definir detalhadamente como e de que forma cada agente irá resolver as tarefas a ele atribuída no sistema. Esta fase consiste das seguintes atividades:

- Produção de diagramas de visão geral dos agentes: estes diagramas mostram as habilidades dentro de um agente ao invés de mostrar os agentes dentro de um sistema como é mostrado nos diagramas de visão geral do sistema.
- Redefinir habilidades: um processo para refinar as habilidades do agente que são expressas em termos de planos, eventos e dados.
- Introduzir planos para tratar eventos: um plano é acionado por um evento e define qual o comportamento do agente diante desse evento.
- Definir detalhes dos eventos, planos e dados: para a implementação do sistema é preciso por fim definir os detalhes para os eventos, planos e dados.

Ainda encontra-se uma ferramenta para modelar sistemas baseados em agentes inteligentes, que possibilita a verificação desses sistemas e que segue as etapas e fases da metodologia Prometheus, que veremos a seguir.

3.1.1 Prometheus Design Tool (PDT)

Para fazer a modelagem foi utilizada a ferramenta Prometheus Design Tool (PDT) que foi desenvolvida em Java e é um Protótipo (*Freeware*) criado pelos criadores da metodologia Prometheus.

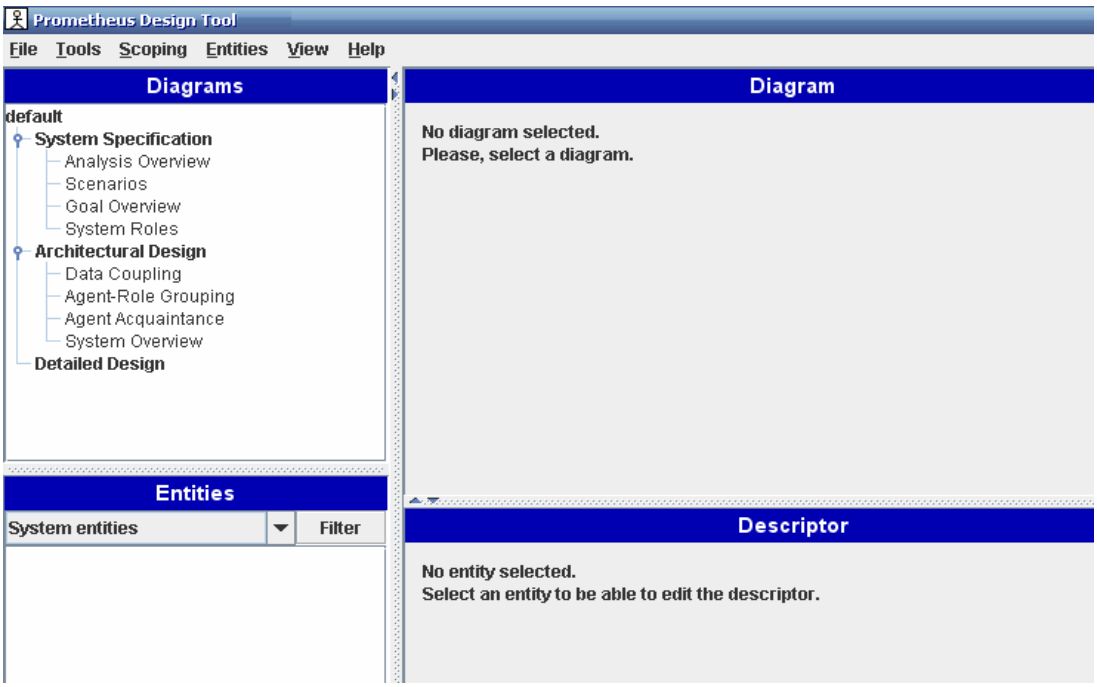


Figura 5 – Prometheus Design Tool (PDT)

A Prometheus Design Tool (PDT) é uma ferramenta que permite um usuário entrar e editar uma especificação seguindo a metodologia Prometheus.

Essa ferramenta auxilia na especificação para uma variedade de possíveis inconsistências, gera automaticamente uma variedade de diagramas de acordo com a metodologia e gera automaticamente uma descrição da especificação (em formato LaTeX) isso inclui descrições para cada entidade de especificação, um dicionário de especificação, e os diagramas anteriormente gerados.

> PDT provê uma interface gráfica que permite o usuário entrar em elementos da especificação (capacidades, eventos, agentes, bancos de dados e planos). Cada elemento tem uma informação associada as descrições de Prometheus. Por exemplo, eventos têm um campo de cobertura, um subtipo de campo (por exemplo, ação, percepção, mensagem, evento), um nome e uma descrição. (PADGHAM; WINIKOFF, 2002).

A PDT gera um diagrama de seqüência e de atividades de acordo com o código AUML (*Agent Unified Modeling Language*).

A AUML é uma extensão do padrão UML para a especificação de protocolos de interação entre agentes e outras noções de sistemas multiagentes. AUML fornece ferramentas para especificar protocolos de interações entre agentes e representar o comportamento interno de um agente.

A notação do fluxo de dados e das entidades – ator, agente, objetivo, cenários de casos de usos, mensagem percepção ação, base de dados, protocolo, regras – que podem ser descritas pela ferramenta, é demonstrada na Fig. 8.

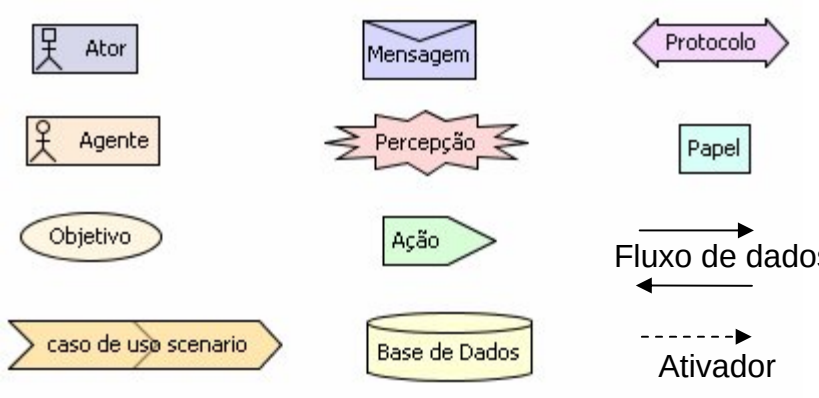


Figura 6 – Notação PDT

A ferramenta Prometheus Design Tool gera código da linguagem JACK.

A linguagem de programação JACK é uma extensão orientada a agentes da linguagem JAVA que trata: agentes, potencialidades, eventos, planos, bases de conhecimentos do agente (base de dados) e gerência de recursos e simultaneidade.

```
package plans;
import events.*;
import data.*;

public plan AprovaNC extends Plan
{
Plan Name: Aprova NC    UniqueID: 225
Description: Atualizar a Base de dados Níveis de Cruzeiro e
aprovar o plano de vôo
Procedure: é ativado pela mensagem Disponível NC;
coloca a Rota e o Tipo Avião na base de dados Níveis de
Cruzeiro Disponíveis;
envia uma mensagem Plano de Vôo aprovado com o Plano de Vôo.
Failure:
Failure Recovery:
   #handles event DisponívelNC disponívelnc_h;
   #posts event planodevôoaprovado planodevôoaprovado_p;
   #modifies data
intervalodetempoqueosníveisdecruzeirodasaeroviasestãodisponí
veis Níveis de Cruzeiro Disponíveis_dat;
```

Figura 7 – Código JACK gerado pela ferramenta PDT

Os agentes em JACK são componentes de software autônomos que apresentam objetivos a serem alcançados ou eventos a serem tratados. São programados com um conjunto de planos para torná-los capazes de alcançar seus objetivos.

A linguagem JACK possui um ambiente de programação o Jack Development Environment – JDE, que inclui uma ferramenta de modelagem para a construção dos diagramas, resultando na geração do código JACK. O JDE dá suporte à metodologia Prometheus pelo fato dos conceitos utilizados por JACK corresponderem aos artefatos gerados na fase de projeto detalhado da metodologia.

4 Controle do Espaço Aéreo Brasileiro

Foi feita uma pesquisa de campo no Aeroporto Internacional de Pelotas, junto a especialistas do tráfego aéreo, que serviu de subsídio para compreender o funcionamento do controle do espaço aéreo brasileiro.

Segundo o Sr. Airton do aeroporto internacional de Pelotas o Brasil é dividido em quatro regiões de informações de vôos (FIR – Flight Information Region) monitoradas pelos CInDACTAs (Centro Integrado de Defesa Aérea e Controle de Tráfego Aéreo) I, II, III, IV.

Ainda conforme diz o Sr. Airton, cada unidade de monitoramento (CInDACTA) é composta por uma rede de radares e estações de telecomunicação que detectam as aeronaves que sobrevoam as áreas de cobertura. Fazendo com que essas aeronaves sejam monitoradas pelos controladores de vôo e somente as que possuem transponder[5], é que são identificadas.

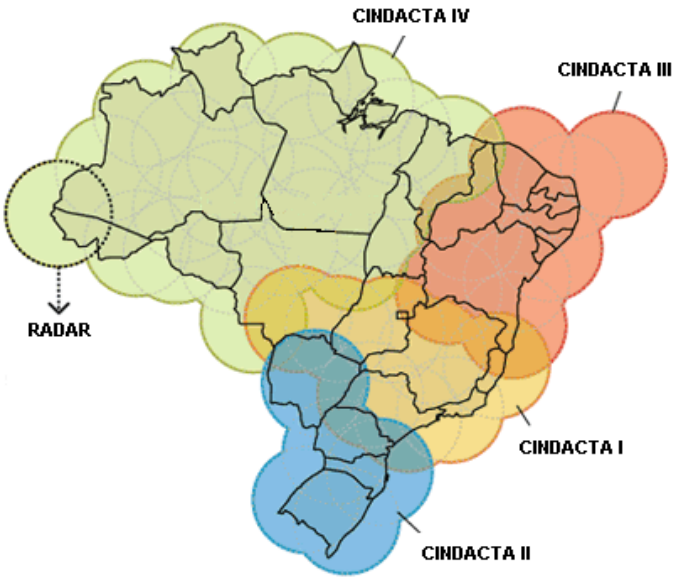


Figura 8 – Regiões de Informação de Vôo monitoradas pelos CInDACTAs

[5] **Transponder:** Dispositivo eletrônico de radiocomunicação entre o avião e o radar, que informa dados de telemetria da aeronave, entre eles, o número do vôo, altitude, velocidade e direção.

Segundo Sales (2007), para definir a rota que uma aeronave irá sobrevoar é preciso primeiro definir o Rumo Magnético[6] e logo após a aerovia que irá sobrevoar.

O piloto define as chamadas regras de vôo de acordo com a experiência de já ter trafegado a rota que irá trafegar e as condições de vôo.

As regras de vôo se dão de duas formas:

Visual (VFR – *Visual Flight Rules*) – Regras de Vôo à Vista, o piloto é o principal responsável por ver e evitar obstáculos ou tráfego e tem o solo como referência para o vôo.

Instrumental (IFR – *Instrument Flight Rules*) – Regras de Vôo por Instrumentos, o piloto não possui visualização ou desconhece a região onde está sobrevoando, sendo conduzido por instrumentos que são pontos de comunicação que enviam sinais de código Morse e as aeronaves sobrevoam auxiliadas por instrumentos, indo em direção a esses sinais.

As aeronaves trafegam pelas aerovias. As aerovias são áreas monitoradas pelos controladores e podem ser:

Aerovias superiores – possuem a letra “U”, de Upper, em inglês, como a primeira letra de sua identificação (Ex.: UW33, UL309, UB750 e etc.).

As aerovias superiores possuem as seguintes características:

a) limite vertical superior - ilimitado;

b) limite vertical inferior – FL 245[7] exclusive; e

c) limites laterais – 80 Km.

Aerovias Inferiores – não possuem nem uma letra de identificação, não possuindo a letra “U” como a primeira letra. (Ex.:W10, G449, W52 e etc.).

As aerovias inferiores possuem as seguintes características:

a) limite vertical superior – FL 245 inclusive;

b) limite vertical inferior - solo ou água; e

c) limites laterais – 30 Km.

Identificação das Aerovias:

---

[6] Rumo Magnético: direção dada pela bússola e depende da direção em que a aeronave irá sobrevoar. O rumo magnético é dado pelo ângulo em que o aeroporto de destino se encontra do aeroporto de origem. De acordo com o rumo magnético tem-se os níveis de cruzeiros específicos (Tab. 3).

[7] FL, do inglês *flight level*, indica o nível de vôo em que uma aeronave pode se encontrar. É a altitude nominal de uma aeronave, baseada em um padrão de pressão mundial preestabelecido de 1013.25 mbar.

Cada rota é identificada pelas letras A, B, G, L, R, W ou Z, seguido de um número. Ex.: (G678).

Conforme Sales (2007), existem algumas Observações, como descritas a seguir:

1) As letras W e Z são usadas na identificação de rotas domésticas e as demais, nas rotas internacionais.
2) As letras M, N, L e Z são usadas em particular nos designadores de Rotas de Navegação de Área (RNAV).

Um designador de rota poderá ser acompanhado das letras U, S e K, com os seguintes significados:

a) U (UPPER): indicando rota pertencente ao espaço aéreo superior;
b) S (SUPERSONIC): indicando rota específica para vôos supersônicos; e
c) K (KOPTER): indicando rota específica para vôo de helicóptero.

Qualquer aeronave pode trafegar por qualquer aerovia, mas dependendo das características de uma determinada aeronave é mais conveniente que essa trafegue em uma aerovia que seja mais adequada as suas características.

O piloto da aeronave é quem define a aerovia e o nível em que a aeronave irá trafegar na aerovia e alguém (controlador de vôo) irá autorizar ou não o tráfego da aeronave pode ou não trafegar pela aerovia no nível estipulado pelo piloto.

As aerovias possuem níveis de cruzeiro (“altura”) que podem ser pares ou ímpares. Para determinar se o nível é par ou ímpar, depende do Rumo Magnético que o vôo de determinada aeronave se dará e da regra de vôo (Visual ou Instrumental). Como mostra a Tab. 3.

Tabela 3 – Níveis de Cruzeiro

	Rumo Magnético	Níveis de Cruzeiro
VFR	de 0° a 179°	035, 055, 075, 095, 115, 135.
	de 180° a 359°	045, 065, 085, 105, 125, 145.
IFR	de 0° a 179°	030, 050, 070, 090, 110, 130, 150, 190, 210, 230, 250, 270, 290, 330, 370, 410, 450, etc.
	de 180° a 359°	020, 040, 060, 080, 100, 120, 140, 160, 180, 200, 220, 240, 260, 280, 310, 350, 390, etc.

Fonte – BASTOS; BAUM, 2007.

Diante disso o Piloto irá definir um plano de vôo e enviar para o controle. Esse plano de vôo é composto por vários itens (ver Anexo A), dentre eles os mais importantes e relevantes para que seja evitada a colisão do avião com outras aeronaves são:

A matricula da aeronave;
A regra de Vôo (VFR ou IFR);
O Código do aeródromo de partida;
O Código do aeródromo de Destino;
O Código do aeródromo de alternativa;
Nível (par ou ímpar);
Rota (aerovias que irá sobrevoar);
Hora de saída/chegada.

O plano de vôo é feito pelo piloto, que o envia para o CInDACTA que faz o monitoramento da área em que a aeronave se encontra e irá decolar. Segundo os especialistas do Aeroporto Internacional de Pelotas, esse plano de vôo é recebido em uma sala que é chamada de sala de aprovação de plano de vôo e se o plano de vôo for aprovado é passado para os controladores de vôo os dados do avião, e este é autorizado a decolar e passa a ser monitorado pelos controladores de vôo.

5 Descrição do problema aéreo

O problema do controle do espaço aéreo se dá em estabelecer a comunicação entre a central de controle com as aeronaves e os aeroportos para que o avião chegue ao seu destino em segurança, livre de incidentes e evitando colisões entre as aeronaves.

O operador da aeronave envia o plano de vôo para a central e o controlador de acordo com algumas regras, prioridades, características operacionais da aeronave e com as condições de vôo no momento, autoriza ou indica novas rotas à aeronave fazendo a negociação com os aviões solucionando possíveis conflitos.

5.1 Aplicação da Abordagem de Sistemas Multiagentes no Problema

Para aplicar a abordagem de sistemas Multiagentes sobre o problema do controle do espaço aéreo deve-se considerar que:

Uma sociedade que dispõe de um agente central, um agente aeronave e um agente aeroporto, pode-se dizer que estes agentes possuem uma representação explícita e homogênea, pois os agentes possuem características diferentes uns dos outros, mas utilizarão a mesma arquitetura. Adotando que estes agentes possuem uma memória do passado, as comunicações entre os agentes se dão através de mensagens diretas e possuem controles deliberativos, ou seja, as ações são planejadas por algumas regras de condições, que fazem com que esses agentes sejam caracterizados como cognitivos.

Havendo essa convergência entre o problema do controle do espaço e a abordagem de sistemas Multiagentes cognitivos e considerando que esses agentes possuem uma estrutura interna baseada na arquitetura BDI, pode-se dizer que esses agentes são organizados, pois obedecem a regras de condições, são negociantes e resolvem conflitos (dois aviões não cruzam a mesma rota ao mesmo

Efeitos produzidos: mostra na tela a mensagem "Dados Enviados".

Ação: Atualiza Dados.
Descrição: vai mostra uma mensagem de dados atualizados.
Parâmetros: Dados Atualizados.
Efeitos produzidos: mostra na tela a mensagem "Dados Atualizados".

6.2 Desenvolvimento da Arquitetura

Nesta fase, com os resultados obtidos na fase anterior foi definida a existência do agente Aeronave, agente Aeroporto e agente Controle e como ocorrerá a interação entre eles. Nesta fase foram definidos também a utilização e o compartilhamento entre os agentes das bases de dados, o que define as crenças dos agentes.

6.2.1 Acoplamento de Dados

Nesta primeira etapa da segunda fase da metodologia seguida foram definidas as Bases de Dados e a ligação dessas com as funcionalidades. Foi definida que funcionalidade lê e/ou escreve nas bases de dados.

A funcionalidade *Solicitação para Decolar* lê informações das bases de dados *BaseDadosRM, BaseDadosAerovias, Base Dados Aeroportos, BaseDadosNC* e *Rotas* e escreve informações na base de dados *Rotas*. Já a funcionalidade *Comunicação de Pouso* não lê nem escreve informações em nenhuma base de dados.

A funcionalidade *Permissão de Decolagem* lê e escreve informações na base de dados *Níveis de Cruzeiro Disponíveis* e a funcionalidade *Atualização da Base de Dados de Pouso* escreve informações na base de dados *Pousos*.

Não há funcionalidades que lidam com os mesmos dados como é mostrado na Fig. 13, ou seja, os dados não são compartilhados entre os agentes que desempenham as funções do sistema.

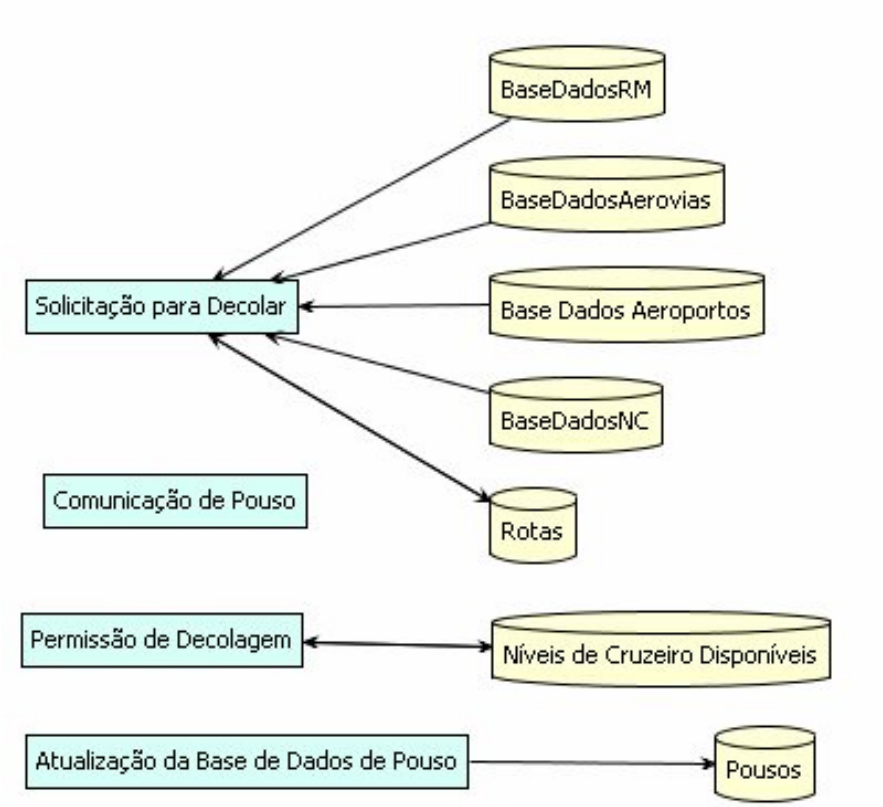


Figura 13 – Diagrama de acoplamento de dados

A seguir é feita a descrição das Bases de dados, que representam as crenças dos agentes:

Dados: BaseDadosRM (externo ao sistema).
Descrição: esta base de dados possui os rumos magnéticos de acordo com os aeroportos.
Tipo de dados: Rumos Magnéticos de 0° a 179° e Rumos Magnéticos de 180° a 359°.
Conexões: BaseDadosRM é lido do plano DefinirRM.
Quando é usado: para definir o rumo magnético.

Dados: BaseDadosAerovias (externo ao sistema).
Descrição: nesta base de dados possui as informações de todas aerovias.
Tipo de dados: Aerovias superiores e inferiores.

Conexões: BaseDadosAerovias lido do plano Definir Aerovia.
Quando é usado: quando o Agente Aeronave for definir a aerovia.

Dados: BaseDadosAeroportos (externo ao sistema).
Descrição: nesta base de dados possui identificação e a localização geográfica dos aeroportos.
Tipo de dados: Aeroporto.
Conexões: BaseDadosAeroportos lido dos planos Definir RM, Definir Aerovia, Define Hora de Chegada.
Quando é usado: quando o Agente Aeronave quiser ter alguma informação sobre um determinado aeroporto.

Dados: BaseDadosNC (externo ao sistema).
Descrição: essa base de dados contém os níveis de cruzeiro divididos em grupos de acordo com o tipo de vôo e o rumo magnético.
Tipo de dados: Níveis de cruzeiro
Conexões: BaseDadosNC é lido dos planos Gerar Novo NC, Nível de Cruzeiro
Quando é usado: quando for necessário saber um determinado nível de cruzeiro de acordo com o rumo magnético e o tipo de vôo.

Dados: Rotas.
Descrição: possui todas as rotas sobrevoadas pela aeronave.
Tipo de dados: Rota.
Conexões: Rotas é escrito pelo plano Atualiza Rotas e é lido do plano Definir Tipo de Vôo.
Iniciação: quando for aprovado o primeiro plano de vôo.
Quando é usado: para definir o tipo de vôo e atualizado quando a rota não foi percorrida.

Dados: Níveis de Cruzeiro Disponíveis.
Descrição: nesta base de dados contém a disponibilidade dos Níveis de cruzeiro das Aerovias.
Tipo de dados: intervalo de tempo em que os níveis de cruzeiro das aerovias estão disponíveis.

Conexões: Níveis de Cruzeiro Disponíveis é escrito pelo plano Aprova NC e lido do plano Ver Disponibilidade de NC.
Iniciação: quando for aprovado o primeiro plano de vôo.
Quando é usado: para ver a disponibilidade de níveis de cruzeiro.

Dados: Pousos.
Descrição: contém os horários de pouso e a identificação dos Aviões.
Tipo de dados: Tipo Avião, Hora de pouso.
Conexões: Pousos é escrito do plano Atualizar Dados.
Iniciação: quando for informado o primeiro pouso.
Quando é usado: quando um pouso for autorizado.

6.2.2 Agrupar Agente/Papel

Nesta etapa foram definidos os agentes e foram feitos o agrupamento dos agentes com suas funcionalidades. Os agentes desempenham algumas funções, ou seja, cada um deles têm um ou mais papéis a serem desenvolvidos no sistema.

Foram definidos três agentes: Aeronave, Controle e Aeroporto. O detalhamento de cada agente foi descrito na fase seguinte da modelagem.

A função do agente Aeronave é solicitar a decolagem do avião com o papel Solicitação para Decolar. Já o agente Controle possui duas funções, a de permitir ou não a decolagem do avião e comunicar o aeroporto de destino a aterrissagem.

O agente Aeroporto tem a função de atualizar a base de dados de aterrissagem do aeroporto de destino dos aviões com o papel Atualização da Base de Dados de Pouso.

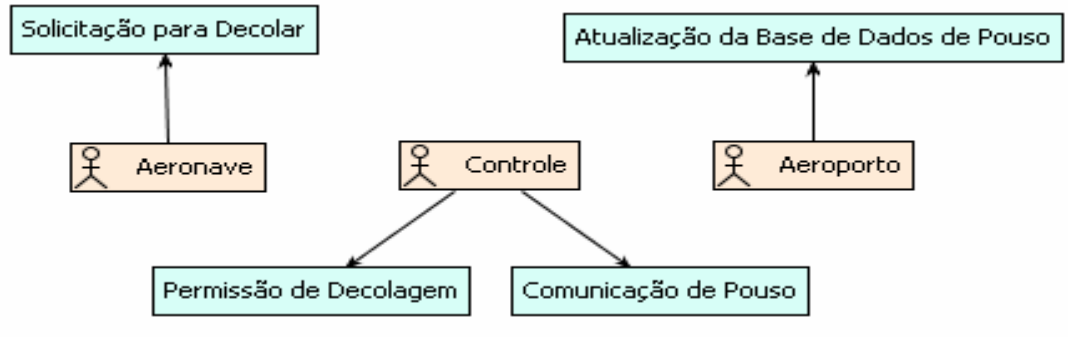


Figura 14 – Diagrama de Agrupamento Agente-Papel.

A Fig. 14 demonstra quais as funções de cada agente, ou seja, qual o papel que cada agente desempenha no sistema.

6.2.3 Conhecimento do Agente

Nesta etapa é demonstrado o fluxo de comunicação entre os agentes que compõem o sistema.

As interações dos agentes se dão de forma direta baseadas na forma de Troca de Mensagem, ou seja, as mensagens são direcionadas de um agente para outro.

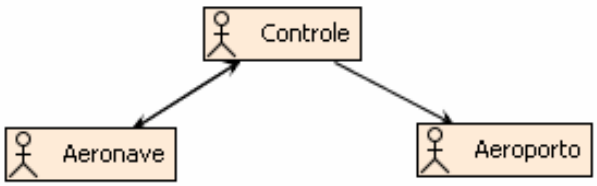


Figura 15 – Diagrama de conhecimento dos agentes

Como mostra a Fig. 15 foram definidos três agentes:

O *Agente Aeronave* que envia e recebe informações do Agente Controle;

O *Agente Controle* que interage com o Agente Aeronave enviando e recendo informações e com o Agente Aeroporto somente enviando informações.

E, por fim, o *Agente Aeroporto* que interage com o Agente Controle somente recebendo informações.

6.2.4 Visão Geral do Sistema

Nesta etapa pode-se ter uma idéia do funcionamento do sistema mostrando as Ações, Percepções e a Interação dos agentes.

Foi produzido um diagrama da estrutura geral do sistema, com todas suas funcionalidades e o protocolo de comunicação entre os agentes, descrito a seguir:

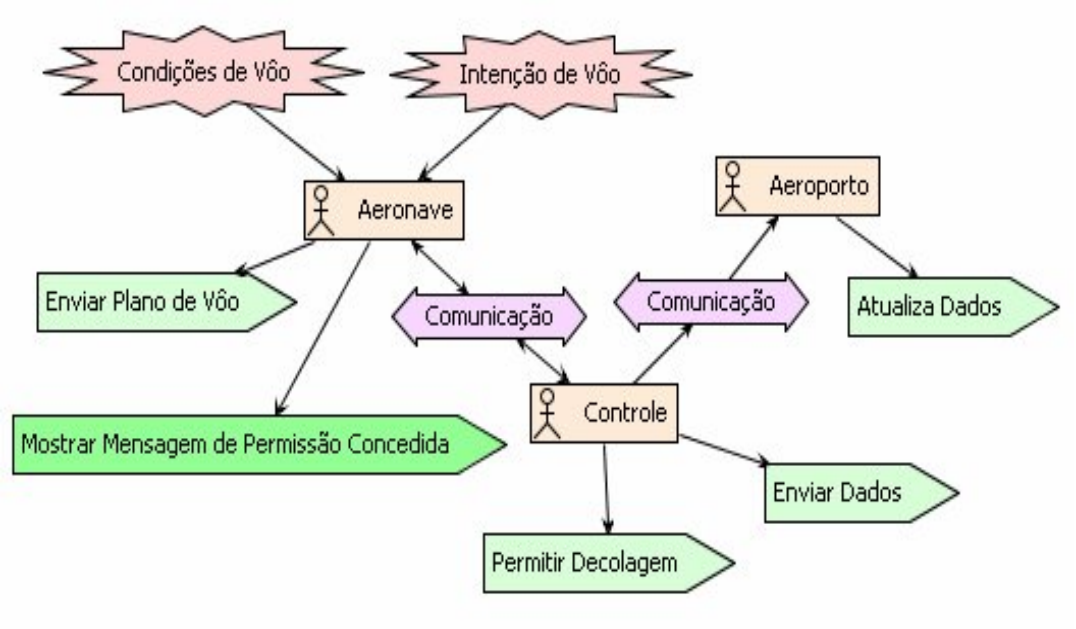


Figura 16 – Diagrama Visão geral do Sistema

A Fig. 16 demonstra as percepções (Condições de Vôo e Intenção de Vôo) e ações (Enviar Plano de Vôo e Mostrar Mensagem de Permissão Concedida) do Agente Aeronave, as Ações (Permitir Decolagem e Enviar Dados) do Agente Controle, a Ação (Atualiza Dados) do Agente Aeroporto e também a interação entre os Agentes, ou seja, o protocolo de comunicação entre eles.

A seguir é feita a descrição do protocolo de comunicação:

Protocolo: Comunicação.
Descrição: faz a comunicação, ou seja, a interação entre os agentes.
Mensagens Incluídas:

Permissão para decolar: Aeronave para Controle;

Pouso autorizado: Controle para Aeroporto;

Permissão de decolagem concedida: Controle para Aeronave;

Permissão de decolagem não concedida: Controle para Aeronave.

Agentes Envolvidos: Aeronave, Controle, Aeroporto.

AUML:

```
start Comunicação
agent A Aeronave
agent B Controle
agent C Aeroporto
message A B Permissão para decolar
box alt
message B C Pouso autorizado
message B A Permissão de decolagem concedida
next
message B A Permissão de decolagem não concedida
end alt
finish
```

Com o código AUML descrito foi gerado automaticamente pela ferramenta PDT um diagrama do protocolo Comunicação e um painel de mensagens.

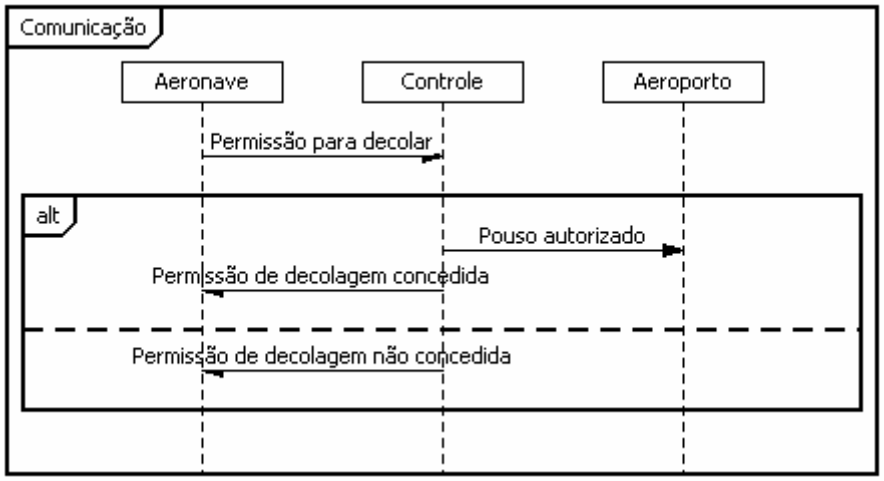


Figura 17 – Diagrama Comunicação

O Protocolo Comunicação demonstrado na Fig. 17 é feito da seguinte maneira:

O Agente Aeronave envia a mensagem Permissão para Decolar para o Agente Controle;

O Agente Controle envia a mensagem Pouso Autorizado para o Agente Aeroporto e a mensagem Permissão de decolagem concedida ou envia a mensagem Permissão de decolagem não concedida.

	MENSAGEM	Do Agente	Para o Agente
1	Permissão para decolar	Aeronave	Controle
2	Pouso autorizado	Controle	Aeroporto
3	Permissão de decolagem...	Controle	Aeronave
4	Permissão de decolagem...	Controle	Aeronave

Tabela 4 – Painel de Mensagens

A Tab. 4 demonstra um painel com quantas e quais mensagens estão contidas no protocolo Comunicação bem como os Agentes que estão enviando e os Agentes que estão recebendo a mensagem.

6.3 Desenvolvimento detalhado

Nesta terceira e última fase, de acordo com a metodologia seguida, foram desenvolvidos os detalhes internos dos agentes que compõem o sistema.

Esses detalhes descritos foram definidos a arquitetura interna dos agentes de acordo com suas crenças, desejos e intenções.

A seguir são demonstradas as estruturas internas dos três agentes (Aeronave, Controle e Aeroporto) modelados:

6.3.1 Agente Aeronave

Este agente, que cuja função é solicitar a decolagem de uma aeronave, gera um plano de vôo. As informações desse plano de vôo são baseadas nas informações dos planos de vôos que são feitos pelos pilotos nos aeroportos brasileiros. O plano de vôo gerado pelo sistema é formado somente com as informações que são relevantes para que seja alcançado o objetivo principal do

sistema que é evitar colisões, abstraindo as informações complementares de um plano de vôo.

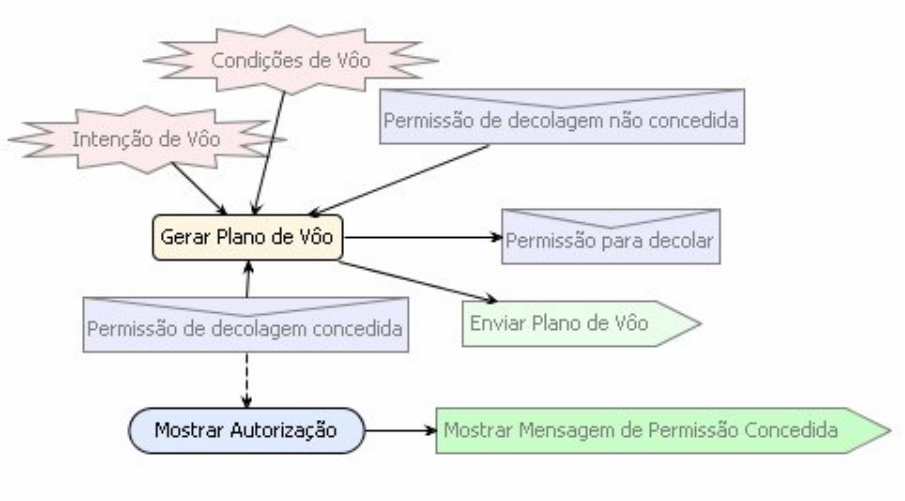


Figura 18 – Arquitetura interna do Agente Aeronave

Descrição: como demonstra a Fig. 18 o Agente Aeronave percebe as condições de vôo e a intenção de vôo e através dessas percepções gera um plano de vôo enviando-o para o Agente Controle pedindo permissão para decolar, se o plano de vôo não for aprovado pelo Agente Controle ele recebe uma mensagem Permissão não concedida do Agente Controle com os níveis de cruzeiro disponíveis para o Agente Aeronave gerar novo plano de Vôo, enviando-o novamente para o Agente Controle, pedindo permissão para decolar novamente, assim até que o plano de vôo seja aprovado e a permissão concedida, mostrando na tela a mensagem de permissão concedida.

Cardinalidade mínima: 1.

Cardinalidade máxima: 1.

Tempo de vida: desde a percepção da intenção de vôo até quando o Plano de Vôo gerado for aprovado.

Inicialização: quando percebe uma intenção de vôo.

Término: Plano de vôo aprovado.

Mensagens recebidas:

Permissão de decolagem concedida: Controle para Aeronave;

Permissão de decolagem não concedida: Controle para Aeronave.

Mensagens enviadas:

Permissão para decolar: Aeronave para Controle.

Percepções: Intenção de Vôo, Condições de Vôo.

Ações: Enviar Plano de Vôo, Mostrar Mensagem de Permissão Concedida.

Objetivo: Evitar Colisões.

Papel: Solicitação para Decolar.

Protocolo: Comunicação.

Planos incluídos: Mostrar Autorização.

Capacidades incluídas: Gerar Plano de Vôo.

A seguir são descritas as entidades contidas no Agente Aeronave de acordo com a Fig. 18:

Plano: Mostrar Autorização

Descrição: mostrar que a permissão para decolagem foi concedida.

Ativador: Permissão de decolagem concedida: Controle para Aeronave (Mensagem).

Ação: Mostrar Mensagem de Permissão Concedida.

Procedimento:

É ativado pela mensagem Permissão de decolagem concedida;

Executa a ação Mostrar Mensagem de Permissão Concedida que coloca na tela "Permissão de decolagem Concedida"

Mensagem: Permissão de decolagem concedida.

Descrição: mostra que o plano de vôo foi aprovado.

Distribuição: Controle para Aeronave.

Finalidade: permitir a decolagem.

Informação contida: Aeroporto de Origem, Aeroporto de destino, Hora de saída, Hora de Chegada, Rota, Rumo Magnético, Tipo de Vôo, Tipo Avião.

Mensagem: Permissão de decolagem não concedida.

Descrição: é uma mensagem para dizer que o plano de vôo não foi aprovado, que contém os níveis de cruzeiro no intervalo de tempo disponível diferentes da Rota do plano de vôo não aprovado.

Distribuição: Controle para Aeronave.

Finalidade: avisar que o plano de vôo não foi aprovado.

Informação contida: níveis de cruzeiro disponíveis no intervalo de tempo em que os níveis de cruzeiro, da Rota do plano de vôo não aprovado, estão ocupados.

Mensagem: Permissão para decolar.

Descrição: uma mensagem pedindo permissão para decolar de acordo com o plano de vôo gerado.

Distribuição: Aeronave para Controle.

Finalidade: pedir permissão para decolar.

Informação contida: Aeroporto de Origem, Aeroporto de destino, Hora de saída, Hora de Chegada, Rota, Rumo Magnético, Tipo de Vôo, Tipo
Avião.

Capacidade Gerar Plano de Vôo:

A capacidade de uma agente é o potencial que este tem de desempenhar uma determinada função internamente desenvolvendo alguma ação; e a capacidade do agente Aeronave é de gerar um plano de vôo assim que o ele for ativado.

O Agente Aeronave tem essa capacidade partindo da percepção *Intenção de Vôo* e passando por outra capacidade *Definição NC* que define os níveis de cruzeiro e gera a *Rota* a ser percorrida e define um plano de vôo.

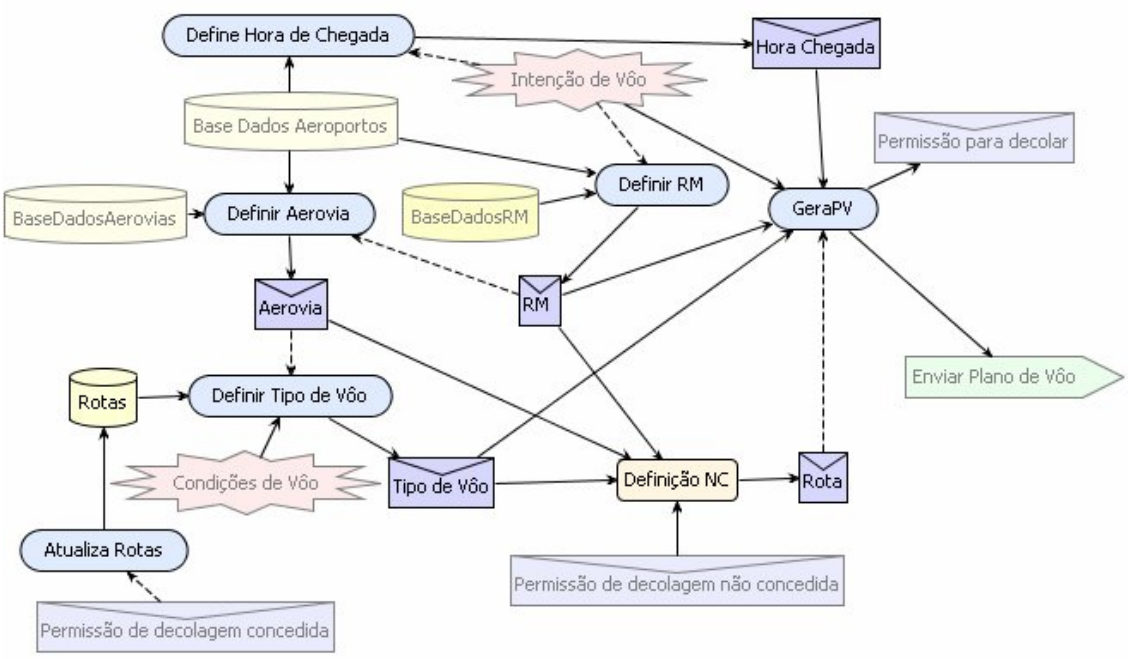


Figura 19 – Arquitetura interna da capacidade Gerar Plano de Vôo

A seguir é feita a descrição da capacidade Gerar Plano de Vôo:

Objetivo: Solicita Decolagem

Processos:

Como demonstra a Fig. 19 através da Intenção de vôo é definida a hora de chegada, com o plano *Define Hora de Chegada* que através da localização dos aeroportos lidos na base de dados *Base Dados Aeroportos* calcula a hora que a aeronave irá chegar ao seu destino e envia uma mensagem *Hora Chegada* para o plano *GeraPV*, e o rumo magnético, através do plano *Definir RM* que lê na base de dados *Base Dados Aeroportos* a localização dos aeroportos de origem e destino e com isso lê o rumo magnético através destas localizações na base de dados *BaseDadosRM* e envia uma mensagem *RM* com o rumo magnético para os planos *Definir Aerovia* e *GerarPV* e *Definição NC*. O plano *Definir Aerovia* é ativado pela mensagem *RM* e através do rumo magnético o aeroporto de origem e o aeroporto de destino, contido na mensagem, vai ler na base de dados Base Dados Aeroportos a localização dos aeroportos e ler na base de dados *BaseDadosAerovias* e definir qual aerovia será percorrida e envia uma mensagem *Aerovia* com a aerovia definida para o plano *Definir Tipo de Vôo* e para *Definição NC.* O plano *Definir Tipo de Vôo* é ativado pela mensagem *Aerovia* e através da aerovia contida na mensagem ele vai pesquisar na base de dados *Rotas* se essa aerovia já foi percorrida por ele. Com isso, e de acordo com as condições de vôo percebidos pela percepção *Condições de Vôo* vai definir qual o tipo de vôo (VRF ou IRF) contido na mensagem *Tipo de Vôo* enviada para *Definir NC* e para o plano *GeraPV*.

O plano *Atualiza Rota* é ativado pela mensagem *Permissão de Decolagem Concedida* que contém a rota autorizada pelo Agente Controle, e atualiza a base de dados *Rotas* com a rota recebida.

A capacidade *Definição NC* recebe as mensagens *RM*, *Aerovia*, *Tipo de Vôo* e é ativado pela mensagem *Tipo de Vôo* ou pela mensagem *Permissão de Decolagem não Concedida*, que contém os níveis de cruzeiro disponíveis em um determinado intervalo de tempo, e envia a mensagem *Rota* com a rota definida.

O plano *GeraPV* é ativado pela mensagem *Rota* e através de todas mensagens recebidas gera um plano de vôo e envia-o na mensagem *Permissão para Decolar* para o Agente Controle e executa a ação *Enviar Plano de Vôo.*

Mensagens recebidas:

Permissão de decolagem não concedida: Controle para Aeronave;

Permissão de decolagem concedida: Controle para Aeronave.

Mensagens enviadas:

Permissão para decolar: Aeronave para Controle.

Mensagens internas:

RM: Definir RM para Definir Aerovia, Definição NC, GeraPV;

Aerovia: Definir Aerovia para Definir Tipo de Vôo, Definição NC;

Tipo de Vôo: Definir Tipo de Vôo para Definição NC, GeraPV;

Rota: Definição NC para GeraPV;

Hora Chegada: Define Hora de Chegada para GeraPV.

Percepções: Intenção de Vôo, Condições de Vôo.

Ações: Enviar Plano de Vôo.

Leitura de dados internos: BaseDadosRM, BaseDadosAerovias, Base Dados Aeroportos, Rotas.

Escrita de dados internos: Rotas.

Planos incluídos: Definir RM, Definir Aerovia, Definir Tipo de Vôo, Define Hora de Chegada, GeraPV, Atualiza Rotas.

Capacidades incluídas: Definição NC.

A seguir são detalhadas as entidades que fazem parte da Capacidade Gerar Plano de Vôo de acordo com a Fig. 19:

Plano: Define Hora de Chegada.

Descrição: De acordo com o aeroporto de origem e o aeroporto de destino, vai pesquisar a localização dos aeroportos na Base Dados Aeroportos e ver a distância entre os dois e calcular a Hora de Chegada.

Ativador: Intenção de Vôo (Percepção).

Mensagem enviada:

Hora Chegada: Define Hora de Chegada para GeraPV

Procedimento:

ver o aeroporto de origem;

ver a localização do aeroporto de origem na BaseDadosAeroportos;

ver o aeroporto de destino;

ver a localização do aeroporto de destino na BaseDadosAeroportos;

ver a distância entre os dois;

calcular o tempo para percorrer está distância;

ver a hora de saída e calcular a hora de chegada;

enviar uma mensagem Hora Chegada com a hora de chegada para o plano GerarPV.

Plano: Definir RM.

Descrição: Define o Rumo Magnético de acordo com a localização dos aeroportos de origem e de destino que está na Base Dados Aeroportos

Ativador: Intenção de Vôo (Percepção).

Mensagem enviada:

RM: Definir RM para Definir Aerovia, Definição NC e GeraPV

Procedimento:

Vai perceber a intenção de vôo;

ver o aeroporto de origem;

verificar na base de dados a localização do aeroporto de origem;

ver o aeroporto de destino.;

verificar na base de dados a localização do aeroporto de destino;

e ver, partindo do aeroporto de origem, qual o rumo magnético do vôo, para chegar no aeroporto de destino.

Enviar a mensagem RM para o plano Definir Aerovia, Definição NC e GeraPV.

Plano: Definir Aerovia.

Descrição: vai definir a Aerovia de acordo com o rumo magnético e consultando a BaseDadosAerovias.

Ativador: RM: Definir RM para Definir Aerovia, Definição NC, GeraPV (Mensagem)

Mensagem enviada:

Aerovia: Definir Aerovia para Definir Tipo de Vôo, Definição NC.

Procedimento:

É ativado pela mensagem *RM*;

Ver aeroporto de destino e o aeroporto de origem contidos na mensagem;

Ver na base de dados Base Dados Aeroportos a localização deles

ver o rumo magnético contido na mensagem;

pesquisar na BaseDadosAerovias as aerovias que coincidem com o rumo magnético e com a localização dos aeroportos de origem e destino;
definir a melhor aerovia;
Enviar a mensagem *Aerovia*

Plano: Definir Tipo de Vôo.
Descrição: Vai definir qual o tipo de vôo de acordo com a aerovia, a rota lida na base de dados *Rota* e as condições de vôo.
Ativador: Aerovia: Definir Aerovia para Definir Tipo de Vôo, Definição NC (Mensagem).
Mensagem enviada:
Tipo de Vôo: Definir Tipo de Vôo para Definição NC, GeraPV.
Percepções: Condições de Vôo.
Procedimento:
É ativado pela mensagem *Aerovia,* que ontem a aerovia definida;
verifica se a aerovia já foi percorrida, na base de dados *Rota*;
se já foi percorrida então vê a condição de vôo
se a condição é normal então o tipo de vôo é VRF
senão o tipo de vôo é IRF.
Envia a mensagem *Tipo de Vôo.*

Plano: Atualiza Rotas.
Descrição: vai atualizar a base de dados Rota com a aerovia que irá percorrer.
Ativador: Permissão de decolagem concedida: Controle para Aeronave (Mensagem).
Procedimento:
É ativado quando receber a mensagem *Permissão de decolagem concedida;*
Ver a rota contida na mensagem;
Atualiza a base de dados *Rota* com a aerovia contida na rota.

Plano: GeraPV.
Descrição: carrega todas as informações para descrever um plano de vôo e gerá-lo.
Ativador: Rota: Definição NC para GeraPV (Mensagem).

Mensagens recebidas:

RM: Definir RM para Definir Aerovia, Definição NC, GeraPV;

Tipo de Vôo: Definir Tipo de Vôo para Definição NC, GeraPV;

Hora Chegada: Define Hora de Chegada para GeraPV.

Mensagem enviada:

Permissão para decolar: Aeronave para Controle

Percepção: Intenção de Vôo.

Ação: Enviar Plano de Vôo.

Procedimento:

É ativado com a mensagem *Rota*;

Percebe as seguintes informações: Hora de Saída; Aeroporto de Destino; Aeroporto de Origem; Tipo Avião.

Ver as mensagens recebidas: Rumo Magnético; Rota; Tipo de Vôo; Hora de Chegada.

Gera o Plano de Vôo com: Hora de Saída; Aeroporto de Destino; Aeroporto de Origem; Tipo Avião; Rumo Magnético; Rota: Tipo de Vôo; Hora de Chegada.

Envia a mensagem *Permissão para decolar* e executa a ação *Enviar Plano de Vôo.*

Mensagem: Hora Chegada.

Descrição: mensagem que contém a hora de chegada do avião no aeroporto de destino.

Distribuição: Define Hora de Chegada para GeraPV.

Finalidade: enviar a hora de chegada para gerar o plano de vôo.

Informação contida: Hora de Chegada.

Mensagem: RM.

Descrição: nesta mensagem vai o Rumo Magnético, a aeroporto de origem e o aeroporto de destino para que sejam definidos a aerovia, a rota, e o plano de vôo.

Distribuição: Definir RM para Definir Aerovia, Definição NC, GeraPV.

Finalidade: definir a aerovia, para a definição da rota, e gerar o plano de vôo.

Informação contida: Rumo Magnético, aeroporto de origem e aeroporto de destino.

Mensagem: Aerovia.
Descrição: essa mensagem possui a aerovia definida anteriormente.
Distribuição: Definir Aerovia para Definir Tipo de Vôo, Definição NC.
Finalidade: ativar e definir o tipo de vôo, para definir a rota e gerar o plano de vôo.
Informação contida: Aerovia.

Mensagem: Tipo de Vôo.
Descrição: a mensagem possui o tipo de vôo se é IRF ou VRF, para definir o nível de cruzeiro e gerar o plano de vôo.
Distribuição: Definir Tipo de Vôo para Definição NC, GeraPV.
Finalidade: ativar e definir nível de cruzeiro e gerar plano de vôo.
Informação contida: Tipo de Vôo (IRF ou VRF).

Mensagem: Rota.
Descrição: uma mensagem que contém rota a ser percorrida.
Distribuição: Definição NC para GeraPV.
Finalidade: ativar GerarPV e gerar o plano de vôo.
Informação contida: Níveis de Cruzeiro da aerovia (rota).

Capacidade Definição NC:

Esta capacidade faz parte da estrutura interna da capacidade *Gerar Plano de Vôo*. E tem a finalidade de definir os níveis de cruzeiro que será percorrido na aerovia para definir a rota que a aeronave irá percorrer.

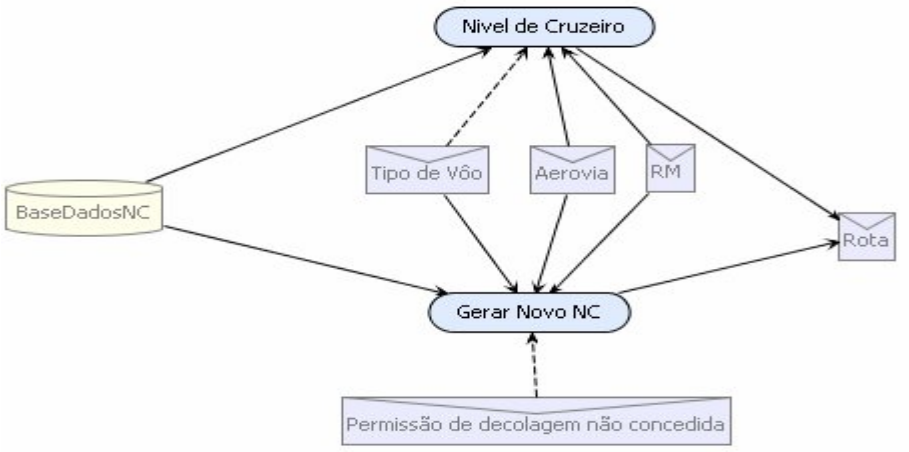


Figura 20 – Arquitetura interna da capacidade Definição NC.

A seguir é feita a descrição da capacidade Definição NC:

Objetivo: solicita decolagem.

Processos:

A estrutura interna dessa capacidade é mostrado na Fig. 20 que possui o plano *Nível de Cruzeiro*, que vai gerar o nível de cruzeiro do primeiro plano de vôo que será gerado para uma determinada solicitação de decolagem, é ativado pela mensagem *Tipo de Vôo* e de acordo com essa mensagem e com as mensagens *Aerovia* e *RM* vai ler na base de dados *BaseDadosNC* o nível de cruzeiro e enviar uma mensagem *Rota* com a aerovia e o nível de cruzeiro dessa aerovia que será percorrido.

Já o plano *Gerar Novo NC* é ativado, caso o plano de vôo enviado anteriormente não tenha sido aprovado, pela mensagem *Permissão de decolagem não concedida* e de acordo com os níveis de cruzeiro contidos nessa mensagem e com as mensagens Tipo de Vôo, Aerovia e RM vai ler na base de dados BaseDadosNC os níveis de cruzeiro e vai definir novos níveis de cruzeiro e enviar uma mensagem Rota com a aerovia e os níveis de cruzeiro dessa aerovia que serão percorridos.

Mensagens recebidas:

Tipo de Vôo: Definir Tipo de Vôo para Definição NC, GeraPV;

RM: Definir RM para Definir Aerovia, Definição NC, GeraPV;

Aerovia: Definir Aerovia para Definir Tipo de Vôo, Definição NC;

Permissão de decolagem não concedida: Controle para Aeronave.

Mensagem enviada:

Rota: Definição NC para GeraPV.

Leitura de dados internos: BaseDadosNC.

Planos incluídos: Gerar Novo NC, Nivel de Cruzeiro.

6.3.2 Agente Controle

A função do agente Controle é de aprovar o plano de vôo recebido e comunicar ao aeroporto de destino a hora da aterrissagem.

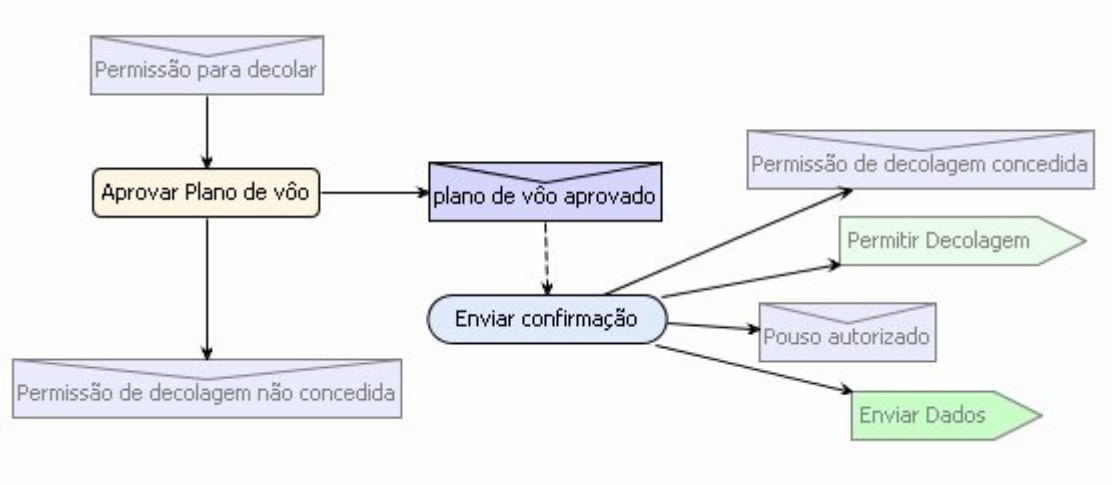


Figura 21 – Arquitetura interna do Agente Controle

Descrição: a Fig. 21 demonstra que o Agente Controle recebe a *mensagem Permissão para decolar* com o plano de vôo, com isso, tem a capacidade *Aprovar Plano de vôo* que verifica se o plano de vôo é válido ou não, se sim, então envia a mensagem *plano de vôo aprovado*, com o plano de vôo, para o plano *Enviar confirmação*, senão envia a mensagem *Permissão de decolagem não concedida*, com os níveis de cruzeiro disponíveis para o Agente Aeronave.

O plano *Enviar confirmação* é ativado pela mensagem *plano de vôo aprovado* e envia a mensagem *permissão de decolagem concedida*, para o Agente Aeronave, com o plano de vôo aprovado, executando a ação *Permitir Decolagem* que mostra na tela a mensagem *Decolagem Permitida*, e a mensagem *Pouso Autorizado* para o Agente Aeroporto, com o tipo de avião e a hora de chegada e executa a ação *Enviar Dados* que mostra na tela a mensagem Dados Enviados.

Cardinalidade mínima: 1.

Cardinalidade máxima: 1.

Tempo de vida: desde o pedido de autorização para decolar até o plano de vôo ser aprovado.

Inicialização: pedido de autorização para decolar.

Término: quando o plano de vôo for aprovado.

Mensagens recebidas:

Permissão para decolar: Aeronave para Controle.

Mensagens enviadas:

Permissão de decolagem concedida: Controle para Aeronave;

Pouso autorizado: Controle para Aeroporto;

Permissão de decolagem não concedida: Controle para Aeronave

Mensagens internas:

plano de vôo aprovado: Aprovar Plano de vôo para Enviar confirmação

Ações: Permitir Decolagem, Enviar Dados.

Objetivo: Evitar Colisões.

Papéis: Permissão de Decolagem, Comunicação de Pouso.

Protocolo: Comunicação.

Planos incluídos: Enviar confirmação.

Capacidades incluídas: Aprovar Plano de vôo.

A seguir são descritas as entidades contidas no Agente Controle conforme a Fig. 21:

Plano: Enviar confirmação.

Descrição: vai distribuir as informações para Aeronave e Aeroporto.

Ativador: plano de vôo aprovado: Aprovar Plano de vôo para Enviar confirmação (Mensagem).

Mensagens enviadas:

Permissão de decolagem concedida: Controle para Aeronave;

Pouso autorizado: Controle para Aeroporto.

Ações: Permitir Decolagem, Enviar Dados.

Procedimentos:

Vai ser ativado com a mensagem plano de vôo aprovado;

Vai enviar a mensagem Permissão de decolagem concedida com o Plano de Vôo para o Agente Aeronave;

Vai executar a ação Permitir decolagem mostrando na tela a mensagem Decolagem Concedida;

Vai enviar a mensagem Pouso autorizado com o tipo Avião e a Hora de Chegada.

Vai executar a ação Enviar Dados que mostra na tela a mensagem Dados Enviados.

Mensagem: plano de vôo aprovado.
Descrição: mensagem que aprova o plano de vôo.
Distribuição: Aprovar Plano de vôo para Enviar confirmação.
Finalidade: ativar Enviar Confirmação.
Informação contida: Aeroporto de Origem, Aeroporto de destino, Hora de saída, Hora de Chegada, Rota, Rumo Magnético, Tipo de Vôo, Tipo Avião.

Mensagem: Pouso autorizado.
Descrição: mensagem que comunica a hora e o avião que vai aterrissar.
Distribuição: Controle para Aeroporto.
Finalidade: avisar a hora que a aeronave irá pousar e o tipo Avião.
Informação contida: Hora de Chegada, Tipo Avião.

Capacidade Aprovar Plano de vôo:

Esta capacidade do Agente Controle tem por finalidade ver se a rota contida na mensagem *Permissão para decolar* está disponível, se sim, atualiza a base de dados de rotas ocupadas e aprova o plano de vôo, senão, não permite a decolagem enviando a mensagem *Permissão de decolagem não concedida*.

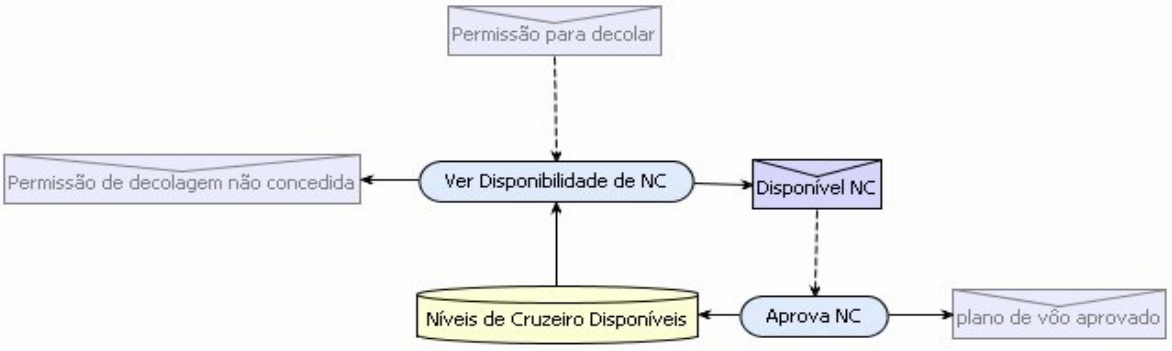


Figura 22 – Arquitetura interna da capacidade Aprovar Plano de vôo

A seguir é feita a descrição da capacidade Aprovar Plano de Vôo:

Objetivo: evitar colisões

Processos:

De acordo como mostra a figura 22 a mensagem P*ermissão para decolar* ativa o plano *Ver Disponibilidade de NC* que vai ver a rota contida na mensagem e verificar na base de dados *Níveis de Cruzeiro Disponíveis* se a rota está disponível no intervalo de tempo estipulado na rota, se sim, envia a mensagem *Disponível NC* com o plano de vôo para o plano *Aprova NC*, se não, envia a mensagem *Permissão de decolagem não concedida* para o Agente Aeronave.

Já o plano *Aprova NC* é ativado pela mensagem *Disponível NC* e atualiza a base de dados *Níveis de cruzeiros Disponíveis* com a rota aprovada contida na mensagem e envia a mensagem *plano de vôo aprovado* para o plano *Enviar confirmação*.

Mensagens recebidas:

Permissão para decolar: Aeronave para Controle.

Mensagens enviadas:

plano de vôo aprovado: Aprovar Plano de vôo para Enviar confirmação;

Permissão de decolagem não concedida: Controle para Aeronave.

Mensagens internas:

Disponível NC: Ver Disponibilidade de NC para Aprova NC.

Leitura de dados internos: Níveis de Cruzeiro Disponíveis.

Escrita de dados internos: Níveis de Cruzeiro Disponíveis.

Planos incluídos: Ver Disponibilidade de NC, Aprova NC.

Entidades contidas na capacidade Aprovar Plano de vôo de acordo com a Fig. 22 são descritas a seguir:

Plano: Ver Disponibilidade de NC.

Descrição: vai ver se os Níveis de cruzeiro da aerovia está disponível para trafegar no intervalo de tempo entre a hora de saída até a hora de chegada.

Ativador: Permissão para decolar: Aeronave para Controle (Mensagem).

Mensagens enviadas:

Disponível NC: Ver Disponibilidade de NC para Aprova NC;

Permissão de decolagem não concedida: Controle para Aeronave.

Procedimentos:

Ver a Rota;

Ver se os níveis de cruzeiro estão disponíveis na base de dados, Níveis de Cruzeiro Disponíveis, no intervalo de tempo que serão trafegados na aerovia determinada na Rota.

Se estão disponíveis envia a mensagem Disponível NC contendo o Plano de Vôo recebido.

Senão, ler na base de dados Níveis de cruzeiro disponíveis os níveis de cruzeiros disponíveis, no determinado intervalo de tempo contido no plano de vôo, na aerovia contida na rota e envia a mensagem Permissão de decolagem não concedida com os níveis de cruzeiro disponíveis no intervalo de tempo.

Plano: Aprova NC.

Descrição: vai Atualizar a Base de dados Níveis de Cruzeiro e aprovar o plano de vôo.

Ativador: Disponível NC: Ver Disponibilidade de NC para Aprova NC (Mensagem).

Mensagem enviada:

plano de vôo aprovado: Aprovar Plano de vôo para Enviar confirmação.

Procedimentos:

É ativado pela mensagem Disponível NC;

Coloca a Rota e o Tipo Avião na base de dados Níveis de Cruzeiro Disponíveis atualizando-o;

Envia a mensagem plano de vôo aprovado com o Plano de Vôo.

Mensagem: Disponível NC.

Descrição: mensagem aprovando a rota.

Distribuição: Ver Disponibilidade de NC para Aprova NC.

Finalidade: Ativar o plano Aprova NC.

Informação contida: Aeroporto de Origem, Aeroporto de destino, Hora de saída, Hora de Chegada, Rota, Rumo Magnético, Tipo de Vôo, Tipo Avião.

6.3.3 Agente Aeroporto

O agente Aeroporto possui a função de atualizar a base de dados que contém as aterrissagens feitas no aeroporto e comunicar que os dados foram atualizados indicando que será feita nova aterrissagem.

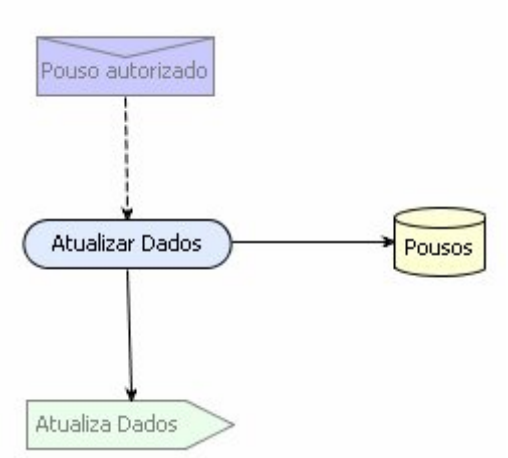


Figura 23 – Arquitetura interna do Agente Aeroporto

Descrição: de acordo com a Fig. 23 o Agente Aeroporto recebe a mensagem *Pouso autorizado* que ativa o plano *Atualizar Dados* que atualiza a base de dados *Pousos* e executa a ação *Atualiza Dados*.
Cardinalidade mínima: 1.
Cardinalidade mínima: 1.
Tempo de vida: desde o recebimento da mensagem que avisa a aterrissagem até executar a ação de dados atualizados.
Inicialização: com o recebimento da mensagem Pouso autorizado.
Término: com a ação Atualiza Dados.
Mensagem recebida: Pouso autorizado: Controle para Aeroporto.
Ação: Atualiza Dados.
Dados internos: Pousos.
Objetivo: Evitar Colisões.
Papel: Atualização da Base de Dados de Pouso.
Protocolo: Comunicação.
Plano incluído: Atualizar Dados.

A seguir é descrito o plano incluído no agente Aeroporto como mostra a Fig. 23:

Planos: Atualizar Dados.

Descrição: vai atualizar a Base de dados Pousos.

Ativador: Pouso autorizado: Controle para Aeroporto (Mensagem).

Ação: Atualiza Dados.

Procedimentos:

É ativado pela mensagem Pouso autorizado;

Ver a Hora de Chegada e Tipo Avião e colocar na base de Dados Pousos;

Executa a ação Atualiza Dados mostrando na tela a mensagem Dados Atualizados.

6.4 Crosscheck

Com o término da terceira fase da metodologia seguida e conseqüentemente a finalização da modelagem foi feita a verificação de inconsistências através do *cross checking* da ferramenta Prometheus Design Tool.

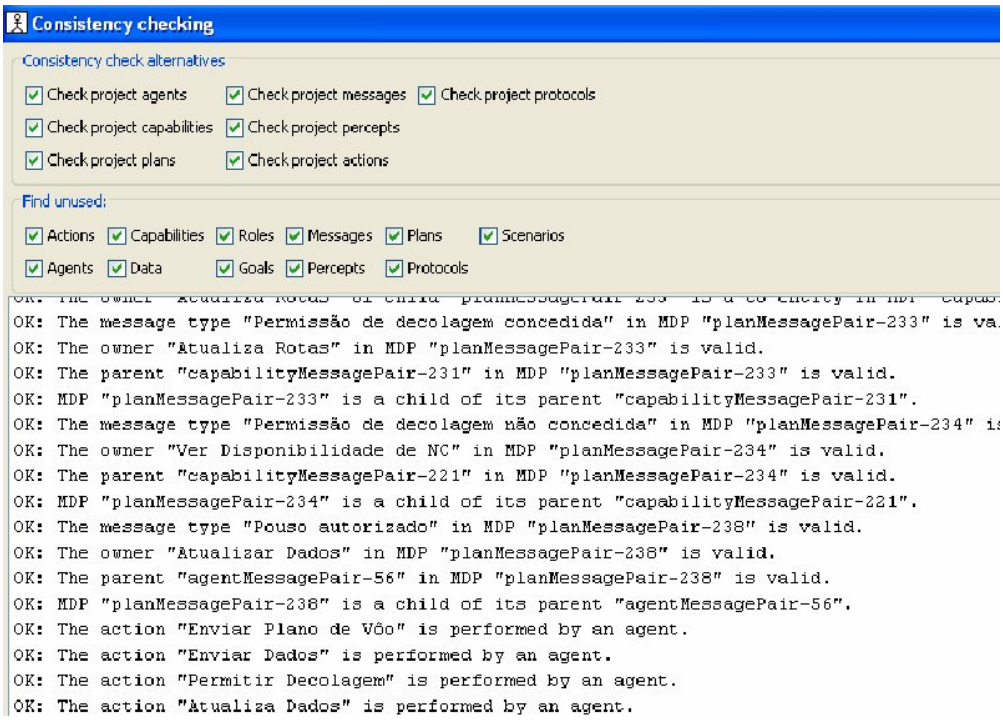


Figura 24 – CrossCheck.

Como demonstra a Fig. 24 não há inconsistência entre as entidades envolvidas, ou seja, não demonstrou nenhuma possível inconsistência no sistema modelado.

Com isso, Concluí-se a viabilidade de implementar as entidades modeladas do sistema. Demonstrando também a convergência entre a abordagem de sistemas multiagentes e o controle do espaço aéreo modelado seguindo a metodologia orientada a agentes Prometheus.

7 Conclusão

Este trabalho teve como proposta utilizar a abordagem de sistemas multiagentes, utilizando a arquitetura de agentes BDI e a ferramenta Prometheus Design Tool (PDT), que se baseia na metodologia orientada a agentes Prometheus, para modelar um sistema que resolva o problema do controle do espaço aéreo, baseado no controle de tráfego aéreo brasileiro que é do tipo centralizado.

Procurando demonstrar que sistemas baseados em agente onde existe uma rede complexa de protocolos de coordenação, exige novos modelos de abstração e ferramentas de conceitualização e implementação.

A técnica de sistemas multiagentes possibilita que se possa inferir características de interação, cooperação, negociação, conflitos e controle. Pois, baseando-se na arquitetura BDI que tem como característica o processo de decidir antes do momento de executar as ações adiantando as próprias metas, apesar de os agentes modelados possuírem características diferentes,foi garantido um comportamento global coerente dos agentes no que se refere a aquelas características inferidas.

Com isso, concluiu-se que existe uma forte convergência entre a abordagem de sistemas multiagentes e o problema do controle do espaço aéreo. Sendo a utilização de Sistemas Multiagentes uma técnica viável para resolver o problema evitando assim as colisões que é o objetivo principal deste sistema.

Este trabalho também apresenta o estudo de uma Metodologia Orientada a Agentes, tomando como base a análise (sem a implementação) de um sistema de controle do espaço aéreo, detalhando os passos da metodologia Prometheus necessários para a especificação do sistema proposto.

Prometheus demonstrou ser uma metodologia independente de linguagem de programação, tendo como objetivo principal direcionar o analista em todos os passos do ciclo de vida com ênfase no desenvolvimento detalhado dos agentes. O

projetista pode ir facilmente da especificação do sistema até a implementação do mesmo.

A metodologia Prometheus, que foi seguida para fazer a modelagem mostrou-se extremamente detalhada e de grande valia, pois engloba a utilização de modelos internos de agentes baseados na arquitetura BDI e, juntando, com a utilização da ferramenta Prometheus Design Tool que segue a metodologia Prometheus tornou a modelagem menos complicada apesar de bem completa.

Demonstrou-se que a ferramenta PDT facilita a composição dos diagramas de análise da metodologia Prometheus, indicando falhas, identificando e direcionando a construção de comunicação e a criação de agentes e a implementação da estrutura interna dos agentes, semi-automatizando a criação de sistemas multiagentes que correspondam as exigências especificadas.

O Prometheus Design Tool cumpre satisfatoriamente com a proposta de uma ferramenta que suporte e reforce a metodologia Prometheus, em um ambiente acionado através de menus e diagramas manipuláveis graficamente em janelas.

Cabe ressaltar que o código JACK gerado pelo PDT é específico e contém as informações necessárias facilitando a codificação para futuras implementações dos sistemas modelados com essa ferramenta.

Conclui-se que apesar da ferramenta PDT ainda estar em desenvolvimento a tarefa de desenvolver sistemas orientados a agentes usando uma metodologia também orientada a agentes cria novas camadas de abstração que permitem uma visão global do projeto, diminuindo a complexidade inerente a este tipo de sistema.

A modelagem dos agentes do sistema de controle do espaço aéreo foi efetuada sem dificuldades, pois a ferramenta PDT auxiliou na identificação dos agentes e na construção do diálogo entre eles bem como na construção dos diagramas e suas entidades.

A modelagem proposta de acordo com modelos pré-estabelecidos foi mostrada para especialistas na área de tráfego aéreo do Aeroporto Internacional de Pelotas para que fosse validado o modo de como foi feita a modelagem do sistema e concluiu-se que o objetivo principal proposto foi alcançado e que o modelo de sistema feito satisfaz as exigências estabelecidas para que não aconteçam colisões de acordo com os dados estabelecidos.

Ao final do trabalho, chegou-se a conclusão, que a partir do estudo realizado abre-se possibilidades de dar continuidade na pesquisa realizada por ser um

instrumento viável de concretização e grande valia para auxílio no controle do espaço aéreo.

7.1 Trabalhos futuros

Na modelagem do sistema do controle do espaço aéreo não foram consideradas a velocidade da aeronave, a prioridade da aeronave e a razão de decolagem, que é o ângulo de subida do avião até o nível de cruzeiro inicial. Considerando a velocidade constante, o ponto de partida sem a razão de subida e que todas aeronaves possuem a mesma prioridade. Diante disso futuramente pode-se:

- Simular, através da implementação, o sistema modelado de forma parcial e preliminar utilizando a técnica de sistemas multiagentes;
- Fazer a modelagem de forma completa, a partir do modelo feito considerando as informações abstraídas;
- Implementá-la levando em consideração as informações abstraídas a partir da pesquisa realizada.
- Concluir a modelagem tratando possíveis conflitos na hora da aterrissagem, considerando o número de pistas de pouso dos aeroportos.

REFERÊNCIAS

ARDANS, Ornar. A perspectiva etológica do estudo do comportamento. **Psicologia revista**. PUC-SP, Nº 2, maio de 1996.

ALVARES, L. O.; SICHMAN, J. S. Introdução aos Sistemas Multiagentes**. Publicado nas Jornadas de Atualização em Informática – XVI JAI XVII Congresso da SBC**, Brasília, 2-8 agosto 1997.

BASTOS, Alexandre L. D.; BAUM, Derick M. **Controle de tráfego aéreo**. Disponível em: < http://www2.ita.br >. Acessado em: 18 de maio 2007.

BOND, A.; GASSER, L. eds. **Readings in Distributed Artificial Intelligence**. San Mateo, Morgan Kaufmann, 1988.

BRENNER, W.; RÜDIGER, Z.; WITTIG, H. **Intelligent Software Agents. Foundations and applications**. Berlin: Springer-Verlag, 1998.

DEMAZEAU, Y. From interactions to collective behaviour agent-based systems. **In: Proc. of the 1st. European Conference on Cognitive Science**, St. Malo, France, 1995.

FERBER, J.; GASSER, L. Intelligence artificielle distribuée**. In: XI International Workshop on Expert Systems & their Applications**, Avignon, France, 1991, Cours n.9.

FININ, T.; LABROU, Y.; MAYFELD, J. KQML as an agent communication language. IN: BRADSHAW, Jeffrey (Ed.). **Software Agents**. Menlo Park: AAAI Press/The MITPress, 1997.

GASSER, L. Boundaries, **identity and aggregation: plurality issues in multiagent systems**. In: Werner, E. e Demazeau, Y. eds. Decentralized AI 3, Amsterdam, Elsevier, 1992.

HILL, J. C.; ARCHIBALD, J. K.; STIRLING, W.C.; FROST, R.L. **A Multi-Agent System Architecture for Distributed Air Traffic Control**, 2005. Disponível em: <http://www .ee.byu.edu/faculty/jka/papers/aiaa05.pdf>. Acessado em: 16 de Dezembro de 2006.

HÜBNER, Jomi Fred. **Um modelo de Reorganização de Sistemas Multiagentes**. São Paulo, 2003.

JONKER, Geert; MEYER, John-Jules; DIGNUM, Frank. Efficiency and Fairness in Air Traffic Control. This research is supported by the Technology Foundation STW, applied science division of NWO and the technology programme of the Ministry of Economic Affairs. **Project DIT5780: Distributed Model Based Diagnosis and Repair**. Utrecht University, 2005.

PADGHAM, Lin; WINIKOFF, Michael. **Prometheus: A Pragmatic Methodology for Engineering Intelligent Agents.** RMIT University, 2002. Disponível em: <http://www.cs.rmit. edu.au>. Acessado em: 20 de fevereiro de 2007.

PALLOTTINO, L., BICCHI, A., and PANCANTI, S., “Safety of a decentralized scheme for Free-Flight ATMS using Mixed Integer Linear Programming,” **American Control Conference**, 2002.

RUSSEL, Stuart J.; NORVIG, Peter. **Artificial Intelligence – A Modern Approach**. New Jersey: Prentice Hall, 1995.

SALES, Carlos Rogério. **Regulamento de Tráfego Aéreo**. Disponível em: <http://www.airandinas.com>. Acessado em: 15 de Abril de 2007.

SANTOS, D. R. dos; RIBEIRO, M. B.; BASTOS, R. M. **A Comparative Study of Multi-Agent Systems Development Methodologies.** Study developed by Intelligent Systems Engineering Group of the PUCRS, financed by Dell Computers of Brazil Ltd. with resources of Law 8.248/91, 2006.

SOCIOLOGIA. Disponível em: <http://tiosam.com/Sociologia>. Acessado em: 20 de maio de 2007.

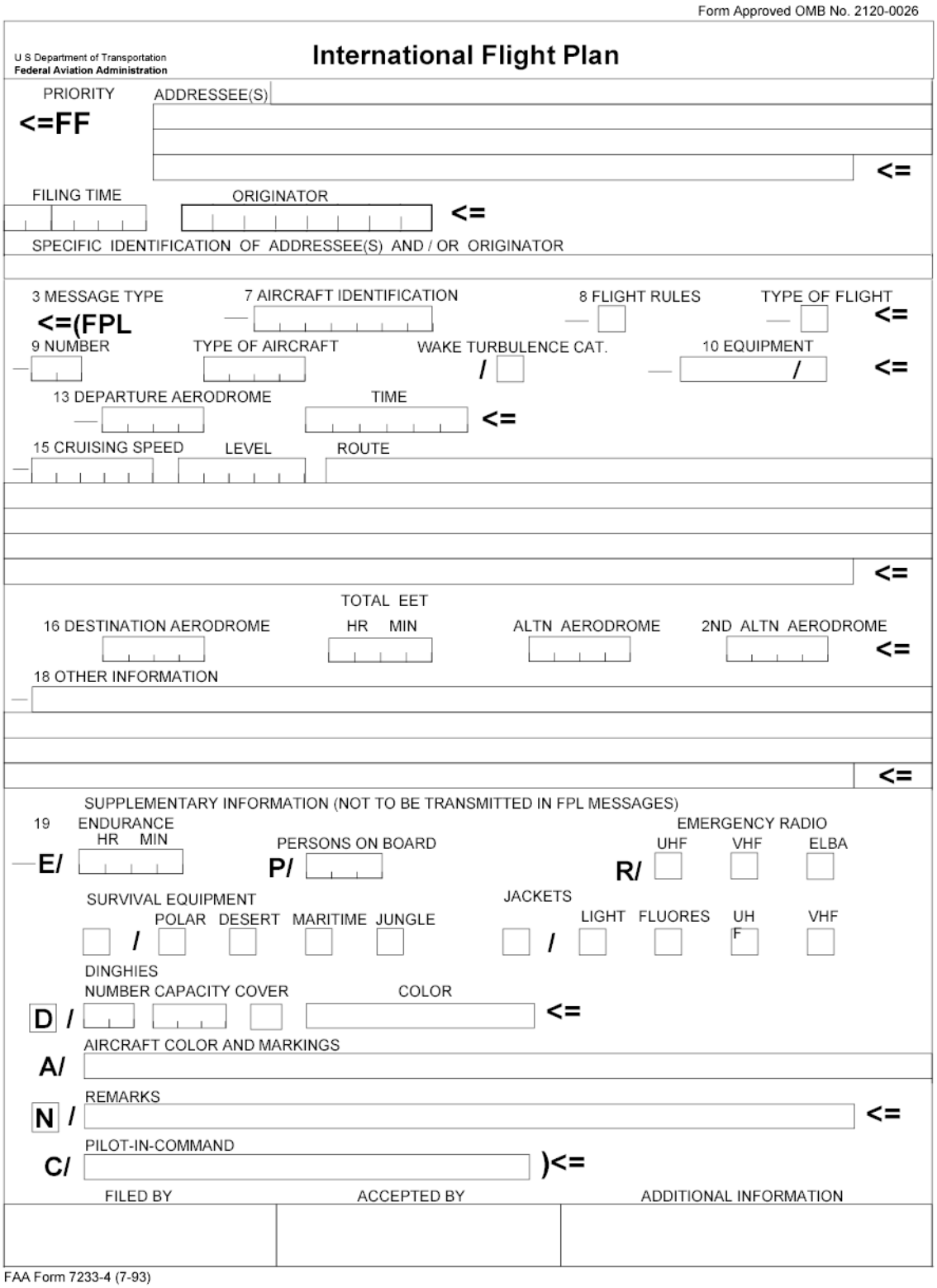
STONE, Peter; VELOSO, Manuela. **Multiagent Systems: A Survey from a Machine Learning Perspective**. Disponível em: <http://www.cs.cmu.edu/afs/cs/usr /pstone/public/papers/97MAS-survey/revised-survey.html>. Acessado em: 15 de dezembro de 2006.

WEISS, Gerhard. **Multiagent Systems: a modern approach to distributed artificial intelligence**. Massachusetts, 1999**.**

WOOLDRIDGE, M. & JENNINGS, N.R. **Intelligent Agents: Theory and Practice. Knowledge Engineering Review**, Vol. 10, No. 2. Cambridge: Cambridge University Press, 1995.

Anexos

Anexo A – Plano de Vôo



Form Approved OMB No. 2120-0026

U S Department of Transportation
Federal Aviation Administration

International Flight Plan

PRIORITY
<=FF

ADDRESSEE(S)

<=

FILING TIME

ORIGINATOR

<=

SPECIFIC IDENTIFICATION OF ADDRESSEE(S) AND / OR ORIGINATOR

3 MESSAGE TYPE
<=(FPL

7 AIRCRAFT IDENTIFICATION

8 FLIGHT RULES

TYPE OF FLIGHT

<=

9 NUMBER

TYPE OF AIRCRAFT

WAKE TURBULENCE CAT.
/

10 EQUIPMENT
/

<=

13 DEPARTURE AERODROME

TIME

<=

15 CRUISING SPEED

LEVEL

ROUTE

<=

TOTAL EET

16 DESTINATION AERODROME

HR MIN

ALTN AERODROME

2ND ALTN AERODROME

<=

18 OTHER INFORMATION

<=

SUPPLEMENTARY INFORMATION (NOT TO BE TRANSMITTED IN FPL MESSAGES)

19 ENDURANCE
HR MIN
E/

PERSONS ON BOARD
P/

EMERGENCY RADIO
UHF VHF ELBA
R/

SURVIVAL EQUIPMENT
/ POLAR DESERT MARITIME JUNGLE

JACKETS
/ LIGHT FLUORES UHF VHF

DINGHIES
NUMBER CAPACITY COVER COLOR
D /

<=

AIRCRAFT COLOR AND MARKINGS
A/

REMARKS
N /

<=

PILOT-IN-COMMAND
C/

)<=

FILED BY	ACCEPTED BY	ADDITIONAL INFORMATION

FAA Form 7233-4 (7-93)