

**UNIVERSIDADE FEDERAL DE SÃO CARLOS**

**CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA**

**PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO**

**ARQUITETURA HOLÔNICA DE CONTROLE PARA O  
DESPACHO MULTICRITÉRIOS DE AGVS**

**FLAVIO ALDROVANDI MONTORO**

**ORIENTADOR: PROF. DR. ORIDES MORANDIN JUNIOR**

São Carlos - SP

Março/2015

**UNIVERSIDADE FEDERAL DE SÃO CARLOS**

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**ARQUITETURA HOLÔNICA DE CONTROLE PARA O  
DESPACHO MULTICRITÉRIOS DE AGVS**

**FLAVIO ALDROVANDI MONTORO**

Tese apresentada ao Programa de Pós-Graduação em  
Ciência da Computação da Universidade Federal de  
São Carlos, como parte dos requisitos para a obtenção  
do título de Doutor em Ciência da Computação, área  
de concentração: Inteligência Artificial.

Orientador: Prof. Dr. Orides Morandin Junior

São Carlos - SP

Março/2015

Ficha catalográfica elaborada pelo DePT da Biblioteca Comunitária UFSCar  
Processamento Técnico  
com os dados fornecidos pelo(a) autor(a)

M798a Montoro, Flavio Aldrovandi  
Arquitetura holônica de controle para o despacho multicritérios de AGVs / Flavio Aldrovandi Montoro. - São Carlos : UFSCar, 2016. - 184 p.

Tese (Doutorado) -- Universidade Federal de São Carlos, 2015.

1. Arquitetura holônica. 2. Despacho multicritério. 3. Sistemas holônicos. 4. Sistema de controle. I. Título.



**UNIVERSIDADE FEDERAL DE SÃO CARLOS**  
Centro de Ciências Exatas e de Tecnologia  
Programa de Pós-Graduação em Ciência da Computação

---

**Folha de Aprovação**

---

Assinaturas dos membros da comissão examinadora que avaliou e aprovou a Defesa de Tese de Doutorado do candidato Flavio Aldrovandi Montoro, realizada em 02/04/2015:

---

Prof. Dr. Orides Morandin Junior  
UFSCar

---

Profa. Dra. Tatiana Taveira Pazelli  
UFSCar

---

Prof. Dr. Roberto Fernandes Tavares Neto  
UFSCar

---

Prof. Dr. Alex Sandro Roschildt Pinto  
UFSC

---

Prof. Dr. José Reinaldo Silva  
USP

*Dedico este trabalho aos meus pais Edson e Mirnei; à minha namorada Máisa; à minha irmã Liege e especialmente à minha avó Eunice que infelizmente não pode estar aqui para acompanhar toda essa jornada.*

# **AGRADECIMENTO**

Agradeço aos meus pais Edson e Mirnei, por todo o suporte que me deram durante todos esses anos e por terem acreditado, apoiado e investido na minha formação pessoal e profissional.

À minha irmã Liege, por todo amor, carinho e paciência que teve comigo durante esta jornada.

À minha namorada Maísa pelo apoio incondicional, pelas dicas e pela ajuda na realização deste trabalho, as vezes orientando, as vezes apenas ajudando no esclarecimento das ideias.

Ao meu orientador Orides, por todo o apoio, ensinamentos e suporte dado durante essa longa jornada, visando sempre o meu crescimento pessoal e profissional.

Aos grandes amigos do laboratório Tear: Vinícius, Carlos e Rogers, que sempre auxiliaram nas dúvidas que surgiram durante esse trabalho e também pelo apoio dado.

# RESUMO

Na manufatura, a busca por novas técnicas para melhorar a produção é sempre um desafio. Estas técnicas devem se adequar ao ambiente dinâmico da manufatura. Um ponto tratado neste âmbito é o sistema de transporte, que tem como um de seus subproblemas a atribuição de tarefas para os AGVs. Para a resolução deste problema, é comum utilizar técnicas de despacho de veículos, sendo que estas técnicas geralmente são multicritério. Apesar de serem multicritério, não é comum contemplarem critérios de diferentes setores da fábrica, o que poderia indicar estratégias conflitantes. Outro ponto que influencia no desempenho da manufatura é a arquitetura de controle. Tradicionalmente, a arquitetura é centralizada ou descentralizada, apresentando prós e contras. Existe uma busca sobre novas arquiteturas para atingir benefícios proveniente das duas tradicionais; uma opção muito usada é a holônica. Neste trabalho é apresentada uma arquitetura holônica de controle para o despacho de veículos usando regras multicritério, atendendo critérios de otimização de diferentes setores da fábrica e proporcionando um viés de otimização global. Foram realizadas análises para avaliar a aderência da arquitetura holônica proposta no ambiente de manufatura, atendendo à inclusão de novos elementos de maneira natural, analisando o comportamento diante de eventos não programados e possibilitar o atendimento de diferentes variáveis de otimização pertencentes à diferentes setores da fábrica.

**Palavras-chave:** AGV, Despacho Multicritério, Arquitetura Holônica, Sistemas Holônicos de Manufatura.

# ABSTRACT

In manufacturing, the search for new techniques to improve the production is always a challenge. These techniques must be adapted to the dynamic environment. A point very dealt in this area is the transportation system which has, as one of its sub problems, the assigning tasks to the AGVs. To solve this problem, it is common to use vehicle dispatching, and these techniques are usually multi-criteria. Although they are multi-criteria, is not common contemplate criteria of different sectors of the factory, which could indicate conflicting strategies. Another point that influences the performance of the manufacturing is the control architecture. Traditionally, this architecture is centralized or decentralized, and have pros and cons. To achieve benefits from both traditional architectures, new architectures have been pursued; a widely used option is the holonic. This research presented a holonic architecture control for the dispatch of vehicles using multiple criteria rules, given optimization criteria from different sectors of the factory and providing a global optimization bias. Analyzes were performed to assess the adhesion of the proposed holonic architecture in the manufacturing environment, achieving the inclusion of new elements naturally, analyzing the behavior when an unpredicted event occurs and enabling compliance with different optimization variables belonging to different sectors of the factory.

**Keywords:** AGV, Multicriteria Dispatching, Holonic Architecture, Manufacturing Holonic System.



# LISTA DE FIGURAS

|  |    |
|--|----|
| Figura 1.1. Apresentação da hipótese de pesquisa .....   | 23 |
| Figura 2.1. Blocos básicos de construção do HMS e suas relações (adaptado de Van Brussel et al (1998)). .....                          | 35 |
| Figura 2.2. Versão UML dos blocos de construção de um HMS e suas relações (adaptado de Van Brussel (1998))......                       | 36 |
| Figura 2.3. Agregação dos <i>holons</i> de recurso (adaptado de Van Brussel (1998)) .....  | 37 |
| Figura 2.4. Especialização do <i>Holon</i> de Recurso (adaptado de Van Brussel (1998))..   | 38 |
| Figura 2.5. Programador centralizado como <i>Staff Holon</i> (adaptado de van Brussel (1998)). .....                                   | 39 |
| Figura 2.6. Modelo funcional dos <i>holons</i> básicos representados por um diagrama de classes (adaptado de Van Brussel (1998))...... | 40 |
| Figura 2.7. Realizando um pedido no HMS (adaptado de Van Brussel (1998))......   | 41 |
| Figura 2.8. Modelo conceitual para um <i>Holon</i> ADACOR (adaptado de Leitão e Restivo (2006)) .....                                  | 44 |
| Figura 2.9. Característica Fractal na abordagem ADACOR (adaptado de Leitão e Restivo (2006)).....                                      | 45 |
| Figura 2.10. <i>Holons</i> ADACOR e suas interações (adaptado de Leitão e Restivo (2006)). .....                                       | 46 |
| Figura 2.11. Diagrama de Interação para a coordenação global (adaptado de Leitão, Colombo e Restivo (2005)). .....                     | 47 |
| Figura 2.12. Abordagem ADACOR para a reorganização dinâmica (adaptado de Leitão e Restivo (2006)). .....                               | 50 |
| Figura 2.13. Um agente e seu ambiente (adaptado de Wooldridge e Jennings (2002)). .....  | 52 |
| Figura 2.14. Diagrama UML do domínio do problema do metamodelo ASPECS (COSENTINO <i>et al.</i> , 2010a).....                           | 55 |
| Figura 2.15. Diagrama UML do Domínio dos agentes do metamodelo ASPECS (COSENTINO <i>et al.</i> , 2010a).....                           | 57 |
| Figura 2.16. Diagrama UML do Domínio de solução do metamodelo ASPECS (COSENTINO <i>et al.</i> , 2010a).....                            | 59 |
| Figura 2.17. Processo ASPECS (adaptado de COSENTINO <i>et al.</i> , 2010a) .....   | 60 |

|  |     |
|--|-----|
| Figura 2.18. Mapa mental da organização dos trabalhos correlatos.....                                  | 67  |
| Figura 3.1. Sistema <i>fuzzy</i> utilizado para o despacho de veículos .....                           | 87  |
| Figura 3.2. Variável <i>fuzzy</i> de entrada: Distância .....  | 87  |
| Figura 3.3. Variável <i>Fuzzy</i> de entrada: Quantidade de cruzamentos.....                           | 88  |
| Figura 3.4. Variável <i>fuzzy</i> de entrada: data devida .....  | 88  |
| Figura 3.5. Variável <i>fuzzy</i> de entrada: Buffer de saída.....                                     | 89  |
| Figura 3.6. Variável <i>fuzzy</i> de saída: Prioridade. ....   | 90  |
| Figura 3.7. Fase 1, atividade 1: descrição dos requisitos do domínio – caso de uso....                 | 95  |
| Figura 3.8. Fase 1, atividade 1: descrição dos requisitos do domínio – diagrama de caso<br>de uso..... | 96  |
| Figura 3.9. Fase 1, atividade 2: descrição da ontologia do problema.....                               | 97  |
| Figura 3.10. Fase 1, atividade 3: Identificação das organizações.....                                  | 98  |
| Figura 3.11. Fase 1, atividade 4: Identificação dos papéis e interações .....                          | 99  |
| Figura 3.12. Fase 1, atividade 5: Descrição de cenários.....   | 101 |
| Figura 3.13. Fase 1, atividade 6: Plano dos papéis.....  | 102 |
| Figura 3.14. Fase 1, atividade 7: Identificação das capacidades .....                                  | 103 |
| Figura 3.15. Fase 2, atividade 1: Descrição da ontologia de solução.....                               | 104 |
| Figura 3.16. Fase 2, atividade 2: Identificação dos agentes .....                                      | 105 |
| Figura 3.17. Fase 2, atividade 3: Descrição da arquitetura dos agentes .....                           | 106 |
| Figura 3.18. Fase 2, atividade 4: Descrição da ontologia de comunicação. ....                          | 107 |
| Figura 3.19. Fase 2, atividade 5: Descrição do comportamento do papel.....                             | 108 |
| Figura 3.20. Fase 2, atividade 7: Descrição das dependências da organização.....                       | 109 |
| Figura 3.21. Fase 2, atividade 8: Identificação das restrições dos papéis. ....                        | 110 |
| Figura 3.22. Fase 2, atividade 9: Descrição do plano de agente.....                                    | 111 |
| Figura 3.23. Fase 2, atividade 10: Projeto da holarquia. ....  | 112 |
| Figura 4.3. Fase 3, atividade 1: Definição da arquitetura do <i>holon</i> .....                        | 117 |
| Figura 4.4. Código da organização "controle produção" .....  | 118 |
| Figura 4.5. Código do papel AGV.....   | 119 |
| Figura 4.6. Código de teste unitário do papel Rota. ....   | 120 |
| Figura 4.7. Código do <i>holon</i> Rota. ....  | 121 |
| Figura 4.8. Teste unitário do <i>holon</i> Pedidos.....  | 122 |
| Figura 5.1. Descrição da ontologia do problema de roteamento. ....                                     | 124 |
| Figura 5.2. Detalhamento dos papeis da organização Rota.....   | 125 |
| Figura 5.3. Interação entre os papéis da organização Rota. ....  | 126 |

|   |     |
|---|-----|
| Figura 5.4. Descrição das atividades dos papéis de roteamento.....                            | 127 |
| Figura 5.5. Descrição da ontologia da solução de roteamento. ....                             | 128 |
| Figura 5.6. Descrição do comportamento dos papéis para o roteamento. ....                     | 129 |
| Figura 5.7. Descrição das dependências da organização para acrescentar o roteamento.<br>..... | 130 |
| Figura 5.8. Holarquia detalhando o controle do roteamento. ....                               | 131 |
| Figura 5.9. Super <i>holon</i> Produto.....   | 133 |
| Figura 5.10. Holarquia responsável por distribuir os produtos aos <i>holons</i> .....         | 134 |

## LISTA DE TABELAS

|  |    |
|--|----|
| Tabela 2.1 Conceitos chaves do domínio do problema.....                                | 56 |
| Tabela 2.2. Conceitos chaves do domínio dos agentes.....                               | 57 |
| Tabela 2.3. Conceitos chaves do domínio de solução. ....                               | 59 |
| Tabela 3.1. Base de regras do sistema fuzzy .....                                      | 91 |
| Tabela 3.2. Comparação ASPECS e ANEMONA (adaptada de Cossentino et al. 2010a)<br>..... | 94 |

## LISTA DE ABREVIATURAS E SIGLAS

|               |  |
|---------------|--|
| <b>AAD</b>    | <i>Agent Architecture Description</i>                                  |
| <b>AD-FMS</b> | <i>Autonomous Decentralized Flexible Manufacturing Systems</i>         |
| <b>AGV</b>    | <i>Automated Guided Vehicle</i>  |
| <b>AI</b>     | <i>Agent Identification</i>  |
| <b>ANP</b>    | <i>Analytical Network Process</i>                                      |
| <b>APD</b>    | <i>Agent Plan Description</i>  |
| <b>APH</b>    | <i>Adjunct Product Holon</i>   |
| <b>ASPECS</b> | <i>Agent-oriented Software Process for Engineering Complex Systems</i> |
| <b>AUML</b>   | <i>Agent Unified Modeling Language</i>                                 |
| <b>BDI</b>    | <i>Beliefs, Desires, Itention</i>                                      |
| <b>CGA</b>    | <i>Canonical Algorithm Genetic</i>                                     |
| <b>CNP</b>    | <i>Contract Net Protocol</i>   |
| <b>COD</b>    | <i>Communication Ontological Description</i>                           |
| <b>ComC</b>   | <i>Comunication Component</i>  |
| <b>CI</b>     | <i>Capacity Identification</i>   |
| <b>CP</b>     | <i>Constraint Programming</i>  |
| <b>CPH</b>    | <i>Code Production of Holons</i>                                       |
| <b>CPN</b>    | <i>Colaborative Petri Net</i>  |
| <b>CPOR</b>   | <i>Code Production of Organizations and Roles</i>                      |
| <b>CR</b>     | <i>Code Reuse</i>  |
| <b>CRIO</b>   | <i>Capacity, Role, Interaction e Organization</i>                      |
| <b>DC</b>     | <i>Deployment Configuration</i>  |
| <b>DeC</b>    | <i>Decision Component</i>  |
| <b>DRD</b>    | <i>Domain Requirements Description</i>                                 |
| <b>FIPA</b>   | <i>Foundation for Intelligent Physical Agents</i>                      |
| <b>FMS</b>    | <i>Flexible Manufacturing System</i>                                   |
| <b>GUI</b>    | <i>Graphical User Interface</i>  |
| <b>HAD</b>    | <i>Holon Architecture Description</i>                                  |
| <b>HD</b>     | <i>Holarchy Design</i>   |
| <b>HMAS</b>   | <i>Holonic Mult Agent Systems</i>                                      |

|              |  |
|--------------|--|
| <b>HMES</b>  | <i>Holonic Manufacturing Execution System</i>                      |
| <b>HMS</b>   | <i>Holonic Manufacturing System</i>                                |
| <b>HS</b>    | <i>Holonic System</i>  |
| <b>HUT</b>   | <i>Holon Unit Test</i>   |
| <b>IMS</b>   | <i>Intelligent Manufacturing Systems</i>                           |
| <b>IRI</b>   | <i>Interactions and Role Identification</i>                        |
| <b>IT</b>    | <i>Integration Test</i>  |
| <b>JIT</b>   | <i>Just In Time</i>  |
| <b>LCD</b>   | <i>Logical Control Device</i>                                      |
| <b>MAS</b>   | <i>Multi-Agent System</i>  |
| <b>MÊS</b>   | <i>Manufacturing Execution System</i>                              |
| <b>MNP</b>   | <i>Minimum Number of Processing</i>                                |
| <b>MOORA</b> | <i>Multi-Objective Optimization on the basis of Ratio Analysis</i> |
| <b>ODD</b>   | <i>Organisation Dependencies Description</i>                       |
| <b>OH</b>    | <i>Order Holon</i>   |
| <b>OID</b>   | <i>Organisation Identification</i>                                 |
| <b>PD</b>    | <i>Protocol description</i>  |
| <b>PH</b>    | <i>Product Holon</i>   |
| <b>PIC</b>   | <i>Physical Interface Component</i>                                |
| <b>POD</b>   | <i>Problem Ontology Description</i>                                |
| <b>PU</b>    | <i>Processo Unificado</i>  |
| <b>QoS</b>   | <i>Quality of Service</i>  |
| <b>RBD</b>   | <i>Role Behaviour Description</i>                                  |
| <b>RCI</b>   | <i>Role Constraints Identification</i>                             |
| <b>RDF</b>   | <i>Resource Description Framework</i>                              |
| <b>RH</b>    | <i>Resource Holon</i>  |
| <b>RMS</b>   | <i>Reconfigurable Manufacturing Systems</i>                        |
| <b>RP</b>    | <i>Role Plan</i>   |
| <b>RSAM</b>  | <i>Resource Service Allocation Model</i>                           |
| <b>SD</b>    | <i>Scenario Description</i>  |
| <b>SH</b>    | <i>Staff Holon</i>   |
| <b>SOD</b>   | <i>Solution Ontology Description</i>                               |
| <b>SPEM</b>  | <i>Software Process Engineering Metamodel Process</i>              |

|            |                                  |
|------------|----------------------------------|
| <b>TA</b>  | <i>Test Activities</i>           |
| <b>TH</b>  | <i>Task Holon</i>                |
| <b>UML</b> | <i>Unified Modeling Language</i> |
| <b>WIP</b> | <i>Work In Process</i>           |

# SUMÁRIO

|  |           |
|--|-----------|
| <b>CAPÍTULO 1 .....</b>  | <b>16</b> |
| 1 INTRODUÇÃO .....   | 16        |
| 1.1 Contextualização.....  | 16        |
| 1.2 Motivação .....  | 20        |
| 1.3 JUSTIFICATIVA .....  | 22        |
| 1.4 Hipótese do trabalho.....  | 22        |
| 1.5 Objetivos .....  | 24        |
| 1.6 Delimitações do trabalho .....   | 25        |
| 1.7 Método de pesquisa e desenvolvimento .....   | 26        |
| 1.8 Organização do trabalho .....  | 27        |
| <b>CAPÍTULO 2 .....</b>  | <b>29</b> |
| 2 ESTADO DA ARTE .....   | 29        |
| 2.1 Considerações iniciais.....  | 29        |
| 2.2 Fundamentação teórica .....  | 30        |
| 2.2.1 Sistemas Holônicos de Manufatura .....   | 30        |
| 2.2.2 Sistemas Multiagentes.....   | 51        |
| 2.2.3 Sistemas multiagente holônicos .....   | 54        |
| 2.3 Artigos correlatos.....  | 67        |
| 2.3.1 Despacho de veículos .....   | 67        |
| 2.3.2 Sistemas holônicos de manufatura.....  | 72        |
| 2.4 Considerações finais.....  | 80        |
| <b>CAPÍTULO 3 .....</b>  | <b>81</b> |
| 3 PROPOSTA DO TRABALHO .....   | 81        |
| 3.1 Considerações iniciais.....  | 81        |
| 3.2 Proposta de uma arquitetura holônica para o controle do despacho multicritério de<br>veículos..... | 82        |
| 3.2.1 Descrição do ambiente do problema .....  | 83        |
| 3.2.2 Proposta para o despacho multicritério .....   | 85        |
| 3.2.3 Proposta para o controle holônico de despacho .....  | 92        |
| 3.2.4 Modelagem do ambiente de controle do despacho multicritério.....                                 | 95        |
| 3.2.5 Modelagem do sistema holônico de controle do despacho multicritério .....                        | 103       |



|                          |   |            |
|--------------------------|---|------------|
| 3.3                      | <i>Considerações finais</i> .....   | 114        |
| <b>CAPÍTULO 4</b> .....  |   | <b>116</b> |
| 4                        | DESENVOLVIMENTO DO TRABALHO.....  | 116        |
| 4.1                      | <i>Considerações iniciais</i> .....   | 116        |
| 4.2                      | <i>Implementação arquitetura holônica de controle</i> .....   | 116        |
| 4.3                      | <i>Considerações finais</i> .....   | 122        |
| <b>CAPÍTULO 5</b> .....  |   | <b>123</b> |
| 5                        | ANÁLISE DA ARQUITETURA.....   | 123        |
| 5.1                      | <i>Considerações iniciais</i> .....   | 123        |
| 5.2                      | <i>Análise qualitativa da generalidade da arquitetura</i> .....   | 123        |
| 5.2.1                    | Análise qualitativa da adaptação da arquitetura diante de eventos não programados.....                                      | 132        |
| 5.3                      | <i>Considerações finais</i> .....   | 134        |
| <b>CAPÍTULO 6</b> .....  |   | <b>136</b> |
| 6                        | CONCLUSÃO .....   | 136        |
| 6.1                      | <i>Síntese do contexto da proposta e dos objetivos</i> .....  | 136        |
| 6.2                      | <i>Contribuições e delimitações</i> .....   | 137        |
| 6.3                      | <i>Trabalhos futuros</i> .....  | 138        |
|                          | REFERÊNCIAS.....  | 140        |
| <b>APÊNDICE A*</b> ..... |   | <b>149</b> |
|                          | MODELAGEM DA ARQUITETURA HOLÔNICA DE DESPACHO.....  | 149        |
| A.1                      | <i>Requisitos do sistema</i> .....  | 149        |
| A.1.1                    | Descrição dos requisitos do domínio (DRD - <i>Domain Requirements Description</i> ).....                                    | 149        |
| A.1.2                    | Descrição da ontologia do problema (POD - <i>Problem Ontology Description</i> ) .....                                       | 156        |
| A.1.3                    | Identificação das organizações (OID - <i>Organisation Identification</i> ) .....  | 157        |
| A.1.4                    | Identificação dos papéis e interações (IRI - <i>Interactions and Role Identification</i> ) .....                            | 158        |
| A.1.4                    | Descrição de cenários (SD - <i>Scenario Description</i> ) .....   | 158        |
| A.1.6                    | Plano dos papéis (RP - <i>Role Plan</i> ) .....   | 163        |
| A.1.7                    | Identificação das capacidades (CP – <i>Capacity Identification</i> ) .....  | 167        |
| A.2                      | <i>Sociedade de agentes</i> .....   | 168        |
| A.2.1                    | Descrição da ontologia de solução (SOD – <i>Solution Ontology Description</i> ) .....                                       | 168        |
| A.2.3                    | Descrição da arquitetura de agentes (AAD – <i>Agent Architecture Description</i> ) .....                                    | 170        |
| A.2.4                    | (Projeto de organizações) Descrição da Ontologia de Comunicação (COD – <i>Communication Ontological Description</i> ) ..... | 172        |
| A.2.8                    | Identificação das restrições dos papéis (RCI – <i>Role Constraints Identification</i> ) .....                               | 181        |

# Capítulo 1

## INTRODUÇÃO

---

### 1.1 Contextualização

As indústrias usam cada vez mais a automação para se adequar ao mercado consumidor e se manterem competitivas. Com isso, nos últimos anos, há uma intensa busca por técnicas e métodos aplicados ao ambiente de manufatura visando aumento na eficiência da produção.

Leitão (2009) afirma que os sistemas de manufatura geralmente são de difícil controle, além de sofrerem constantes pressões provindas da demanda de mercado por produtos customizados com um curto prazo de entrega, o que exige uma resposta rápida e uma adaptação eficiente às mudanças.

Adam *et al.* (2011) classificam as perturbações que ocorrem no processo de manufatura em dois tipos: externa e interna. As perturbações externas são ocasionadas, por exemplo, pelo aumento demanda de mercado, conforme já apontadas por Leitão (2009), por solicitação de alteração dos prazos, entre outros; as perturbações internas são caracterizadas por quebra de máquinas, falta de matéria prima, absenteísmo, entre outros. Estes fatores aumentam a dinâmica do ambiente de manufatura tornando-o ainda mais complexo.

Para contornar os problemas citados anteriormente, existem vários esforços em direção ao sistema automático de manufatura (AMS – *Automated manufacturing systems*), que segundo Groover (2007), é geralmente composto por um sistema de controle e por elementos tais quais: sistemas de transporte, máquinas de produção e recursos humanos.

O sistema de transporte é responsável por realizar a integração do processo, pois transporta os materiais em processo e os produtos de um ponto ao outro da fábrica, permitindo

assim a existência de um fluxo de produção. Um ponto importante destacado por Groover (2007) é que o custo de transportar os materiais pela fábrica tem grande impacto no custo total do produto.

Ainda segundo Groover (2007), um dos possíveis elementos do sistema de transporte é o AGV, que executa o manuseio de material, independente de operador humano e circula na fábrica por caminhos bem definidos. Seu uso é apropriado quando diferentes materiais são transportados entre vários pontos de carga e descarga dentro da fábrica. O autor ainda afirma que os AGVs são adequados tanto em produções de lotes quanto em manufatura de produtos discretos.

Le-Anh e De Koster (2006) e Naso e Turchiano (2005) apresentam o estudo do AGV como sendo uma tarefa complexa, pois para obter uma otimização é necessário avaliar vários aspectos, sendo alguns deles: atribuição de tarefas, resolução de conflitos, arranjo físico da planta, tamanho da frota de veículos, entre outros subproblemas. Os autores afirmam que isolar um subproblema para estudo, apesar de diminuir a complexidade, afeta o desempenho geral, pois dessa forma deixa-se de avaliar a interação existente entre os subproblemas.

Apesar do conhecimento da interdependência dos subproblemas e do ganho, caso analisados em conjunto, é comum encontrar estudos na literatura que os tratam de maneira isolada devido à alta complexidade de avaliá-los de maneira integrada.

O subproblema da atribuição de tarefas determina a próxima tarefa a ser executada dentre um grupo de tarefas esperando para serem atendidas (KIZIL, ÖZBAYRAK & PAPADOPOULOU, 2006). De acordo com algum critério, um AGV é escolhido para transportar o item de um ponto a outro dentro da fábrica.

Apesar de parecer uma tarefa trivial (a escolha de um AGV para atender a requisição feita por uma máquina), esta tarefa se torna mais complexa quando adicionamos várias máquinas ou vários AGVs ao processo de decisão, que segundo Lin, *et al.* (2006) é um problema combinatorial.

Uma forma muito encontrada na literatura para resolver o problema apontado anteriormente é com uso de regras de despacho, sendo este tipo de problema bem similar ao problema de despacho de veículos que foi apresentado primeiramente por Dantzing e Ramser (1959).

Os autores Egbelu e Tanchoco (1984) classificaram o problema do despacho de veículos em duas categorias, sendo elas: I) Seleção de um veículo dentre um grupo de veículos ociosos para atender uma tarefa; no qual envolve uma única máquina que faz uma requisição e vários AGVs a serem escolhidos para atender o pedido. II) Seleção de uma máquina a ser atendida, dentre um grupo de máquinas que solicitam o serviço de um AGV, esta decisão envolve um único AGV e várias máquinas.

Para definir a regra de despacho é necessário avaliar um ou mais critérios e definir se o despacho será iniciado pela estação de trabalho ou pelo AGV. No primeiro caso, são avaliados os AGVs para atender a estação de trabalho em questão, já no outro caso, é avaliada qual estação de trabalho tem maior prioridade para o AGV em questão.

Após definir se a atribuição de tarefa é iniciada pela estação de trabalho ou pelo AGV, outro ponto necessário é a definição da regra de despacho a ser usada. Antigamente, as regras eram definidas por apenas um único critério, porém não apresentavam um desempenho razoável diante de diferentes configurações de produção.

O desempenho das regras de despacho é dependente das configurações de produção (configuração específica da fábrica, levando em conta o arranjo físico, variáveis de otimização a serem atendidas, etc.) sendo que, geralmente, uma regra simples de despacho pode não garantir um desempenho satisfatório para diferentes condições de operação da fábrica (NASO & TURCHIANO, 2005). Para possibilitar uma melhora de desempenho, tendo em vista o ambiente dinâmico e os vários objetivos a serem alcançados, atualmente é comum usar regras de despacho multicritérios (GUAN & DAÍ, 2009).

Com o uso de regras de despacho multicritério é possível atender um maior número de configurações de produção, suavizando assim, o impacto das mudanças no ambiente de manufatura. Porém as regras ainda são muito limitadas a critérios do chão de fábrica (como por exemplo o *makespan*), sem levar em conta critérios de outros setores (como por exemplo a data devida).

Em busca de uma regra de despacho que sofra menos influência do ambiente dinâmico da manufatura e sabendo do impacto que todas as áreas da fábrica causam no setor de produção, se torna importante avaliar não só indicadores do chão de fábrica, mas também os indicadores provenientes de outros setores (MORANDIN *et al.*, 2011).

Com isso é possível realizar uma produção alinhada às estratégias competitivas, tirando proveito das regras de negócios definidas pela empresa. Um exemplo seria contemplar estratégias competitivas na regra de despacho, como por exemplo, atender melhor aos prazos e consequentemente apresentar um possível diferencial no mercado.

Para executar as regras de despacho, sendo o AGV um veículo independente de controlador humano, se torna necessário o uso de uma arquitetura de controle para realizar o despacho de maneira apropriada e integrar a produção. Segundo Le-Ahn e De Koster (2006), é comum encontrar duas abordagens principais para a arquitetura de controle: centralizada (hierárquico) e descentralizada (heterárquico), cada uma tendo suas vantagens e desvantagens.

Nas arquiteturas de controle centralizadas, há um controlador central monitorando e gerenciando todo o movimento dos AGVs. Para atribuir a tarefa aos AGVs, o controlador usa os dados referentes a todos os AGVs (armazenados em sua memória) para executar controle do despacho (LE-AHN & DE KOSTER, 2006). Segundo os autores Herrero-Pérez e Martínez-Barberá (2008), a vantagem de uma arquitetura desse tipo, é que para a atribuição de tarefa realizado pelo controle do despacho é considerada a informação de todo o sistema, permitindo chegar a uma solução ótima global. Por outro lado, apresenta alta complexidade e a estrutura cresce exponencialmente em relação à quantidade de AGVs no sistema, o que dificulta a implementação em tempo real.

Nas arquiteturas de controle descentralizadas, o controle é distribuído entre as unidades, mantendo a flexibilidade em aplicações mais complexas e diminuindo a complexidade computacional. Os autores Berman e Edan (2002), apontam a vantagem de o sistema ser mais tolerante a falhas e apresentar melhor escalabilidade. Já Herrero-Pérez e Martínez-Barberá (2008) afirmam que este tipo de arquitetura soluciona problemas de coordenação quando surge algum conflito, produzindo soluções ótimas locais.

Tendo em vista que cada uma das arquiteturas tradicionais possui pontos fortes e fracos, e que nenhuma delas por si só é capaz de tratar os problemas de otimização e os problemas gerados pelo ambiente dinâmico em conjunto, optou-se por buscar novas abordagens que pudessem trazer os benefícios necessários para atender os requisitos da manufatura.

Segundo os pesquisadores Wang e Choi (2014) as abordagens que apresentam as características fortes de ambas as arquiteturas tradicionais, centralizada e descentralizada, tem

se tornado cada vez mais populares. Uma abordagem bastante utilizada em vários setores, além da área de manufatura, é a holônica.

Os sistemas holônicos foram investigados para serem aplicados à área de manufatura, devido as suas características de conseguir atender as vantagens das arquiteturas centralizadas e descentralizadas ao mesmo tempo, permitindo uma configuração dinâmica para a arquitetura.

Visando uma padronização da aplicação de sistemas holônicos na manufatura, foram estabelecidas diretrizes para adaptar o conceito ao controle da manufatura. A partir dessas diretrizes surgiu o PROSA (VAN BRUSSEL, 1995) como sendo uma arquitetura de referência para os sistemas holônicos de manufatura (HMS – Holonic Manufacturing System).

## 1.2 Motivação

O PROSA surgiu como uma referência de HMS apresentando as ideias holônicas aplicadas ao controle da manufatura, porém deixou a cargo de cada arquitetura a definição de funcionamento de cada holon. Com isso os autores Leitão, Colombo e Restivo (2005) propuseram a arquitetura ADACOR com base na arquitetura de referência PROSA.

O ADACOR não é apenas uma arquitetura de referência, ele apresenta funções bem definidas para cada holon, além de apresentar um conceito de dinamicidade para a reorganização da arquitetura a partir das demandas do ambiente de aplicação.

Apesar do PROSA e do ADACOR serem fortemente referenciados na literatura, eles não apresentam um método bem definido para se criar um sistema holônico. Focando neste ponto, os autores Botti e Giret (2008) propuseram o método ANEMONA, que se baseia fortemente na arquitetura de referência PROSA e na teoria de sistemas multiagentes para a criação de uma arquitetura holônica voltada para a manufatura.

Assim como o ANEMONA, os autores Cossentino, *et al.* (2010a) propuseram o método ASPECS, também voltado para a criação de sistemas holônicos, porém não específicos para o ambiente de manufatura.

Estes métodos (ASPECS e ANEMONA) surgiram com a ideia de padronizar a criação de um sistema holônico, sejam eles de manufatura ou não, indicando os passos para a modelagem ou até mesmo a parte da codificação do sistema.

Apesar desses métodos permitirem a criação de um HMS, nenhum deles detalham o controle do despacho, sendo que os controles de despacho detalhados encontrados na literatura (NASO & TURCHIANO, 2005; GUAN & DAÍ, 2009; MORANDIN *et al.*, 2011), não seguem os conceitos holônicos.

Detalhando um sistema de despacho que siga os preceitos holônicos se torna possível definir o despacho que considere variáveis locais e não locais (globais). Um exemplo de uma variável local que poderia ser analisada para definir o despacho, é o makespan, que poderia indicar um possível aumento na produção. Porém essa análise local poderia ser errônea caso analisada isoladamente, pois poderia atrasar a entrega dos produtos (variável global) em detrimento de um aumento na produção.

Com essa visão holônica sobre o sistema de despacho seria possível definir, de maneira mais eficiente, a produção na fábrica em relação ao atendimento aos prazos de entrega (incluindo possíveis aumentos de demanda do mercado), tornando essa análise para o despacho mais completa.

Outro aspecto de motivação para este trabalho é que ele faz parte da pesquisa realizada no laboratório Tear buscando abordar o controle da manufatura de outra forma, visando melhorias já identificadas em outros trabalhos. Estas melhorias estão relacionadas à inserção de novos elementos de controle, quão fácil é adaptar o modelo e estendê-lo; conseguir distribuir o controle, permitindo novas abordagens para lidar com o ambiente dinâmico da manufatura (visto que os trabalhos anteriores tratavam o controle de maneira centralizada).

Alguns dos trabalhos já realizados que serviram de inspiração foram: Morandin, *et al.* (2011); Sanches, *et al.* (2009); Morandin, *et al.* (2008); Morandin, *et al.* (2006).

### 1.3 JUSTIFICATIVA

Existem vários trabalhos usando sistemas holônicos na manufatura como por exemplo o trabalho de Barbosa *et al.* (2015) que aplica uma arquitetura holônica para o controle de um sistema flexível de manufatura focando na organização dos elementos diante distúrbios (falha de máquinas). Já Jovanovic (2014) aplica um controle holônico em um ambiente simulado para o processo produtivo de uma fábrica de pneus, para diminuir o tempo de produção e sofrer menos impactos com as quebras de máquina. Rey, *et al.* (2013) propuseram um framework holônico para o controle de um sistema flexível de manufatura de pneus focando no problema de atribuição de tarefas para a programação da produção. Jana, *et al.* (2013) aplicaram as ideologias holônicas para realizar a alocação dinâmica de tarefas para realizar a programação da produção.

Existem vários outros trabalhos que utilizaram os sistemas holônicos para a manufatura, conforme apresentado na seção 2.3.2, porém nenhum deles detalhou o controle dos AGVs utilizando um sistema de controle holônico.

Além do ambiente de manufatura, os sistemas holônicos também estão sendo aplicados, como no caso do trabalho de Philips *et al.* (2013) que apresentam um sistema holônico para a coordenação de semáforos para o controle de tráfego. Demange (2012) apresenta um modelo holônico para a modelagem comportamental de passageiros em um terminal de aeroporto. Versteegh, Salido e Giret (2010) apresentam uma arquitetura holônica voltado para o controle do sistema de transporte viário. No trabalho de Galland *et al.* (2009) são apresentados dois estudos de caso de um sistema multiagente holônico sendo um para o controle de multidões em um evento e o outro para simulação de pedestres em área industrial de uma cidade.

Os sistemas holônicos tem sido usado em diferentes ambientes além da manufatura, se mostrando uma opção viável para o controle de ambientes complexos.

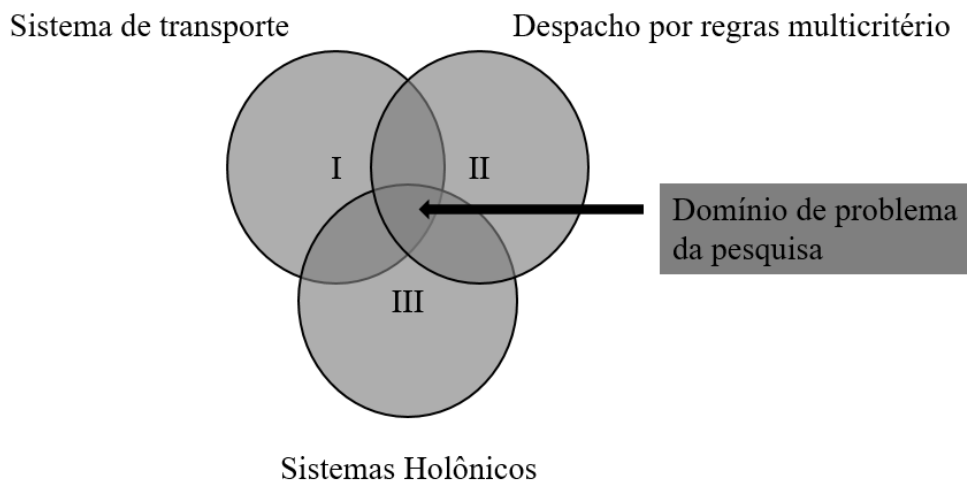
### 1.4 Hipótese do trabalho

Para estabelecer a hipótese de trabalho, foram avaliados três pontos principais, ou domínios de problemas, sendo eles: I) o sistema de transporte como um elemento importante



dentro do sistema de manufatura, e o seu controle exercendo papel fundamental para a otimização; II) o despacho por meio de regras multicritério para permitir a avaliação conjunta de variáveis pertencentes a diferentes setores da fábrica, e com isso, sofrer menos impacto do dinamismo das configurações do ambiente e atender melhor as regras de negócio; III) uma arquitetura holônica para realizar o controle e com isso obter as vantagens inerentes a esse tipo de abordagem (conjuntos ilustrados na Figura 1.1).

Os pontos apresentados anteriormente, já foram abordados em outros trabalhos, porém de maneira isolada. Sendo o ponto I, nos trabalhos: Hsueh (2010), Ho e Liu (2009), Kesen e Bayokoç (2007), Naso e Turchiano (2005), entre outros. O ponto II, nos trabalhos: Morandin *et al.* (2011), Chiba, Arai e Ota (2010), Kesen e Bayokoç (2007), Le-Anh e De Koster (2006), Naso e Turchiano (2005), entre outros. E por último o ponto III, nos trabalhos: Van Brussel (1998), Hsieh (2004), Leitão e Restivo (2006), Zhao, Zhang e Wang (2008), Iwamura *et al.* (2009), Leitão (2011), Wang e Choi (2014), Jovanovic, *et al.* (2014) entre outros.



**Figura 1.1. Apresentação da hipótese de pesquisa**

Buscando construir uma análise dos três pontos em conjunto o domínio de problema da pesquisa fica delimitado à intersecção dos mesmos (Figura 1.1) e com isso se estabelece a pergunta de pesquisa para este trabalho:

É possível detalhar uma arquitetura de controle que realize o controle do despacho e que use o despacho multicritério, sendo esses critérios provenientes tanto do chão de fábrica quanto das regras de negócio, e que respeite às características de um sistema holônico?

A partir da pergunta de pesquisa, estabeleceu-se a hipótese de trabalho sendo:

*Detalhar uma arquitetura holônica de controle do despacho que usa a holonicidade para definir os critérios para o despacho, sendo eles provenientes de diferentes setores da fábrica.*

## 1.5 Objetivos

O objetivo geral deste trabalho é detalhar uma arquitetura holônica de controle para o despacho de veículos, visando tornar o sistema mais robusto<sup>1</sup> em ambientes de manufatura. Esta arquitetura de controle será responsável pelo gerenciamento dos AGVs por meio de regras de despacho multicritério, proporcionando a ponderação de diferentes requisitos para realizar o atendimento, independentemente do nível organizacional ao qual eles pertençam, contribuindo assim para um refinamento na escolha do AGV.

Como objetivo específico do trabalho propõe-se uma técnica para realizar o despacho multicritério que consiga contemplar tanto critérios referentes ao chão de fábrica (makespan), quanto da estratégia de negócio (tardiness), inserindo assim uma visão holônica ao despacho.

Também foi definido como objetivo específico a escolha por um método de desenvolvimento de HMS (ASPECS) que contemplasse desde o levantamento de requisitos, até a etapa de implementação do sistema e que apresentasse etapas bem definidas identificando o problema e a solução dele.

Outro objetivo específico é analisar um ambiente holônico de manufatura para avaliar a modelagem da arquitetura, aplicando distúrbios como, por exemplo, quebra de máquina, falta de matéria prima, adição de pedidos na programação, quebra de AGVs, entre outros.

---

<sup>1</sup> Neste trabalho, robusto tem o sentido de suportar distúrbios no sistema produtivo – como por exemplo: novos pedidos, quebra de elementos, e assim por diante – e ainda assim garantir o seu funcionamento.

## 1.6 Delimitações do trabalho

Vários fatores impactam na tarefa de transporte de materiais, alguns relacionados ao AGV, outros relacionados ao arranjo físico da fábrica, outros vinculados a outros setores que não o chão de fábrica.

Por se tratar de um problema complexo, foi delimitado o escopo apenas para tratar o despacho de AGV por meio de regras multicritérios (makespan e tardiness), sem avaliar o impacto de outros elementos como por exemplo, definição da melhor rota, gerenciamento de bateria dos AGVs, melhor arranjo físico para a fábrica, prevenção de *deadlock*, ciclo de manutenção dos AGVs, tamanho da frota de AGVs, entre outros.

Outro ponto que está fora do escopo deste trabalho é a busca pela otimização. Devido à complexidade do tema, optou-se por definir a estrutura de maneira que, em outra ocasião, fosse possível realizar um estudo de técnicas e métodos para compor a regra de despacho, exclusivamente com o intuito de otimizar o sistema, utilizando a arquitetura aqui proposta.

Conforme já abordado em alguns trabalhos de despacho de veículos do laboratório Tear, é de conhecimento que as regras de despacho são míopes, porque não consideram estados futuros da fábrica. Como apresentado no trabalho Morandin *et al.* (2011), considerando o encadeamento de tarefas, é possível obter melhores resultados no despacho de veículos.

Porém neste trabalho o foco não é definir a melhor regra de despacho para obter um sistema otimizado, e sim apenas demonstrar que é possível detalhar uma arquitetura holônica de despacho que consiga um resultado aceitável dentro dos parâmetros da indústria.

Outro ponto de delimitação do escopo foi a opção de ter como entrada do sistema de controle uma programação da produção já definida. Esta escolha se deu pelo fato de que a definição da mesma traria outros problemas, tornando o trabalho mais complexo e desviaria do objetivo principal que é a arquitetura de controle do despacho de veículos.

## 1.7 Método de pesquisa e desenvolvimento

O método de pesquisa a ser utilizado é o indutivo, que segundo Marconi e Lakatus (2003), partindo-se de premissas verdadeiras obtém-se conclusões prováveis, não podendo garantir sua veracidade. Já Silva e Menezes (2005) definem o método indutivo como sendo um método que obtém a generalização a partir de casos da realidade concreta.

Orientado pelo tipo de pesquisa, foi definida uma série de atividades para desenvolver o trabalho, iniciado por um levantamento bibliográfico das áreas em questão (despacho de veículos, sistemas holônicos de manufatura, sistemas multiagente), para identificar técnicas e teorias com potencial de compor o trabalho.

Com base no levantamento bibliográfico e do histórico de trabalhos realizados no laboratório Tear, foram levantadas as técnicas e métodos para realizar o despacho de veículos, identificando as variáveis a serem analisadas e a importância de considerar variáveis relativas às estratégias competitivas, visto que a competitividade da indústria também é guiada por suas estratégias.

Após a escolha da forma que será realizado o despacho, assim como as variáveis consideradas, foram avaliadas as opções de modelagem holônica presentes na literatura e também os métodos de desenvolvimento de sistemas holônicos existentes. Foram identificadas e avaliadas as arquiteturas holônicas (PROSA, proposto por Van Brussel *et al.* (1998) e ADACOR, proposto por Leitão e Restivo (2006)) e os métodos de desenvolvimento de sistemas holônicos (ASPECS, proposto por Cossentino *et al.* (2010a) e ANEMONA, proposto por Botti e Giret (2008)).

Identificando os prós e os contras para cada uma das opções de arquiteturas holônicas de manufatura e dos métodos, foi analisado qual delas apresentaria maior aderência para modelar a arquitetura de controle do despacho, seguindo as diretrizes já escolhidas anteriormente.

Seguindo o modelo escolhido, foi modelada a arquitetura para o controle do despacho usando regras multicritério.

Após a modelagem do sistema, foram realizadas análises qualitativas, sendo elas relacionadas à análise comportamental do modelo. Foram realizadas duas análises qualitativas,

a primeira avaliando o comportamento do modelo diante de perturbações e a segunda diante da adição de um novo elemento, sendo ele o controle do roteamento de veículos.

Para avaliar qualitativamente esta característica da arquitetura, foi analisado a adição de um novo módulo de controle para realizar o roteamento do veículo, a adição foi realizada apenas atualizando os modelos já gerados pelo método ASPECS.

## 1.8 Organização do trabalho

Para clareza e melhor compreensão deste trabalho, a seguir é apresentada e justificada a sua divisão. O “capítulo 2 – Estado da arte” apresenta os conceitos fundamentais necessários para o entendimento da proposta, assim como o estado da arte das áreas de conhecimentos envolvidas. Para isso, este capítulo está estruturado em duas seções, sendo a primeira delas para apresentar os conceitos da fundamentação teórica de sistemas holônicos de manufatura e de sistemas multiagentes. A segunda seção apresenta os trabalhos correlatos importantes para o desenvolvimento da tese, sendo eles divididos em áreas de: I) despacho de veículos, apresentando técnicas e métodos usados; II) Sistemas holônicos, apresentando abordagens holônicas no âmbito da manufatura, incluindo sistemas holônicos multiagentes.

Vale ressaltar que o “capítulo 2 – Estado da arte” apresenta apenas os conceitos envolvidos na proposta, e não apresenta a fundamentação teórica completa de sistemas holônicos de manufatura e sistemas multiagentes. Outro ponto importante a observar é que os conceitos relacionados ao despacho de veículo se encontram ao longo do presente capítulo (Capítulo 1 – Introdução) e da subseção “artigos correlatos” pertencente ao “capítulo 2 – Estado da arte”, optou-se por essa abordagem por ser um tema bastante abordado em outros trabalhos realizados no TEAR, conforme apresentado nos trabalhos: Benicasa, Morandin e Kato (2003), Morandin, *et al.* (2007) e Morandin, *et al.* (2011).

A seguir, é apresentado o “capítulo 3 – Proposta do trabalho” no qual são discutidos e apresentados detalhadamente todos os pontos da proposta. Já no “capítulo 4 – Desenvolvimento do trabalho” é apresentado o desenvolvimento da proposta, detalhando como foi desenvolvido até se obter o produto final o qual foi avaliado e analisado no capítulo a seguir.

No “capítulo 5 – Análise da arquitetura” são apresentadas análises da arquitetura proposta, avaliando as características do modelo proposto, para que no “capítulo 6 – Conclusão” seja apresentada uma análise completa dos resultados atingidos, considerando as contribuições e as delimitações encontradas e possíveis trabalhos futuros.

Com o intuito de diminuir a densidade de material apresentado na tese, foi adicionado o apêndice “Modelagem da arquitetura de controle utilizando o método ASPECS”. Este apêndice foi adicionado com o intuito de contemplar todos os artefatos gerados pelo método de desenvolvimento ASPECS. Por se tratar de vários modelos em UML, optou-se em apenas apresentar um exemplo de cada atividade do ASPECS no corpo da tese para servir de base, e o processo completo descrito no apêndice.

## Capítulo 2

# ESTADO DA ARTE

---

### 2.1 Considerações iniciais

Este capítulo está dividido em duas seções: I) “2.2 Fundamentação teórica” e II) “2.3 Artigos correlatos”. A primeira seção apresenta os conceitos básicos para a compreensão do trabalho, que incluem os conceitos de sistemas holônicos, sistemas multiagentes e sistemas multiagente holônicos. A segunda seção apresenta os artigos referentes à área de despacho de veículos e de sistemas holônicos de produção.

Na subseção “2.2.1 Sistemas Holônicos de Manufatura” é apresentado o conceito fundamental para a compreensão da teoria de HMS, incluindo a apresentação de uma referência de arquitetura e uma arquitetura, sendo elas: PROSA (VAN BRUSSEL *et al.*, 1998) e ADACOR (LEITÃO e RESTIVO, 2006).

O PROSA, proposto por Van Brussel *et al.* (1998), é uma importante referência, apresentando diretrizes para a criação de uma arquitetura holônica de manufatura. O ADACOR, proposto por Leitão e Restivo (2006), é baseado no PROSA e apresenta uma arquitetura já definida focando na adaptação em ambientes dinâmicos. Apesar da importância de ambas as contribuições sobre arquiteturas holônicas de manufatura, nenhuma delas apresenta um método a ser seguido para a construção de um sistema de controle.

Na subseção “2.2.2 Sistemas Multiagentes” é apresentada a teoria básica sobre agentes, arquitetura de agentes e de sistemas multiagentes. Estes conceitos se tornam importantes pelo fato de que a grande maioria dos HMS são implementados usando sistemas multiagentes (MAS – *MultAgent System*).

Na subseção “2.2.3 Sistemas multiagente holônicos” é apresentado o conceito básico sobre sistemas holônicos de manufatura (HMAS – *Holonc Multiagent System*), a diferença entre agentes e *holons*; e um método de desenvolvimento para HMAS, o ASPECS, proposto por Cossentino *et al.* (2010a).

A apresentação do método ASPECS se dá principalmente por dois motivos: I) dificuldade em implementar uma arquitetura holônica usando o PROSA ou o ADACOR, por não apresentarem um método detalhado. II) por possuir um *framework* de desenvolvimento de HMAS criado exclusivamente para o método.

Na seção “2.3 Artigos correlatos” são apresentados os artigos que corroboram com este trabalho, sendo divididos em dois temas principais: I) despacho de veículos, apresentando diferentes abordagens de como realizar o despacho e II) sistemas holônicos de manufatura, apresentando arquiteturas holônicas, abordando diferentes aspectos da manufatura.

## 2.2 Fundamentação teórica

Nesta seção será apresentada estritamente a teoria referente à parte da área de conhecimento que serviu de apoio para a elaboração do trabalho, não tem como objetivo ser uma seção que ilustre todo o conceito relativo ao tema abordado.

### 2.2.1 Sistemas Holônicos de Manufatura

Os conceitos de sistemas holônicos foram propostos pelo filósofo húngaro Arthur Koestler, apresentados no livro “O Fantasma da máquina” (título original em inglês – “*The ghost in the machine*”). Koestler postulou um conjunto de princípios para explicar a tendência auto organizacional dos sistemas biológicos e sociais dos organismos vivos.

O termo *holon* foi proposto para descrever a unidade básica deste tipo de sistema. O termo foi cunhado a partir da combinação das palavras gregas “*holos*”, que significa todo, com o sufixo “*on*” que significa parte. Este termo reflete a tendência dos organismos observados por Koestler que apresentavam uma autonomia, ao mesmo tempo em que trabalhavam em conjunto com outros organismos para alcançarem um objetivo em comum. Eles se aglomeravam em um subsistema apresentando assim uma auto-organização de hierarquias.



Para a definição do termo *holon*, Koestler explicitou dois aspectos importantes dos sistemas observados, sendo eles: I) os sistemas complexos irão evoluir de maneira muito mais rápida se existirem formas intermediárias estáveis, que neste caso apresentam a configuração hierárquica. II) Embora seja fácil observar o todo e a parte, eles não existem de maneira isolada. Esta observação levou Koestler a propor a palavra *holon* para descrever a natureza híbrida do todo/parte em sistemas reais. Os *holons* são simultaneamente conjuntos independentes para as suas partes subordinadas e partes dependentes quando visto a partir da direção inversa. (VAN BRUSSEL *et al.*, 1998).

Com isso o aspecto principal de um *holon* se dá pela natureza híbrida de ser ao mesmo tempo um sistema autônomo e uma parte subordinada de um outro *holon*, dependendo da ótica com que é analisado, este conceito é conhecido como “a face de Janus”<sup>2</sup> (Koestler, 1969).

Um exemplo amplamente usado para explicar esse conceito do *holon* (parte/todo) é o corpo humano. O corpo humano é um sistema complexo composto de vários outros subsistemas (órgãos) que por sua vez é composto por células, que são compostos por moléculas e assim por diante. Fazendo uma analogia com sistemas holônicos, cada célula do corpo humano seria um *holon* sendo cada uma delas autônoma, um conjunto de células formam um órgão (que apresenta um objetivo específico) até chegar no conjunto de todo os órgãos que compõem o corpo humano.

No início da década de 90, a área da manufatura detectou a necessidade de se atualizar para lidar com os novos desafios gerados devido a mudança na demanda de mercado. Para atender tal expectativa, foi necessário mudar a forma com que a manufatura era controlada e com isso foi criado um consórcio com pesquisadores e grandes empresas de vários países, e o intitularam de IMS<sup>3</sup> (*Intelligent Manufacturing Systems*).

O IMS iniciou várias pesquisas para que fosse possível que o controle da manufatura permitisse gerenciar as mudanças na produção e os distúrbios ocorridos no ambiente, de maneira eficiente e efetiva. Uma dessas pesquisas foi intitulada de sistemas holônicos de manufatura (HMS – *Holonic Manufacturing Systems*), que segundo os pesquisadores Giret e

---

<sup>2</sup> Termo usado por Koestler para descrever a dualidade do holon, este termo foi escolhido por conta do deus romano Janus que possuía duas faces, geralmente associadas ao passado e futuro. Janus é associado a Deus das transições.

<sup>3</sup> Site do IMS: <http://www.ims.org/>

Botti (2009) e Babiceanu e Chen (2006), consistiu basicamente em aplicar os conceitos de sistemas holônicos propostos por Koestler no ambiente de manufatura.

Com a finalização do projeto, o consórcio elaborou uma lista de definições para auxiliar o entendimento e ao mesmo tempo servir de guia para adaptar o conceito holônico dentro do ambiente de manufatura.

De acordo com Van Brussel, *et al.* (1995) as definições elaboradas pelo IMS foram:

- Holon: uma unidade autônoma e cooperativa de um sistema de manufatura para transformar, transportar, armazenar e / ou validar informações ou objetos físicos. O *holon* pode representar uma parte lógica do processamento e também às vezes uma parte física. O *holon* pode fazer parte de outro *holon*.
- Autonomia: a capacidade de uma entidade de criar e controlar a execução de seus próprios planos e estratégias.
- Cooperação: um processo pelo qual um conjunto de entidades desenvolve planos mutuamente aceitáveis e executa esses planos.
- Holarquia: um sistema de *holons* que coopera para atingir um objetivo ou uma meta. A holarquia define regras básicas para a cooperação entre os *holons*, definindo assim sua autonomia.
- Sistemas holônicos de manufatura: uma holarquia que integra todas as atividades de fabricação desde o pedido, através do planejamento, produção e marketing para tornar a empresa de manufatura ágil.

Apesar da forte aderência de sistemas holônicos no ambiente de manufatura, não é uma tarefa trivial implementar um HMS, pois devido à complexidade do problema a ser representado se torna difícil definir as holarquias e os *holons*.

Para auxiliar no desenvolvimento de um HMS ou adaptação de uma arquitetura tradicional em uma arquitetura holônica, foi proposta uma arquitetura de referência chamada PROSA. Esta arquitetura auxilia a modelagem indicando tipos de *holons* específicos para compor a arquitetura, mais detalhes dessa arquitetura serão apresentados na subseção 2.2.1.1.

De acordo com Leitão (2009), devido às características de holarquia e da dualidade do *holon* (parte/todo) é possível um *holon* em vários outros *holons*, permitindo assim a redução da complexidade do problema. Outro ponto que também se beneficia por estas características é a adição de novos elementos no sistema.

Van Brussel (2014) afirma que os conceitos do HMS combinam as melhores características das organizações hierárquicas e heterárquicas, pois a holarquia preserva a estabilidade (hierárquica) enquanto permite uma flexibilidade e dinâmica (heterárquica) à arquitetura.

Por combinar as características das organizações hierárquicas e heterárquicas ao mesmo tempo, a arquitetura holônica vem sendo bastante utilizada para o controle da manufatura, conforme apresentado nos trabalhos de Wang e Choi (2014), Jovanovic, *et al.* (2014), Bal e Hashemipour (2011), Leitão (2009), entre muitos outros.

### 2.2.1.1 Prosa

Esta subseção foi baseada nos seguintes trabalhos: Van Brussel *et al.* (1998); Bongaerts (1998); Wyns (1999); Bongaerts *et al.* (1995).

Para sobreviver ao cenário industrial, foi necessário adaptar os sistemas de manufatura, incorporando novas tecnologias, produtos, estruturas organizacionais, entre outras características. Motivados por este fato, na universidade católica de Leuven na Bélgica, foi desenvolvido o projeto GOA/HMS o qual foi vinculado ao projeto do IMS sobre sistemas holônicos de manufatura.

Após uma análise sobre as características da manufatura, os autores identificaram alguns aspectos importantes, destacados tanto pela academia quanto pelas indústrias, sendo eles: I) aspecto sobre os recursos, como por exemplo, maximizar capacidade. II) aspectos tecnológicos relativos a produtos e processos, como por exemplo, quais as operações necessárias para garantir qualidade do produto. III) aspectos logísticos, como por exemplo, data de entrega.

A partir dessa análise, os autores da PROSA definiram a estrutura para a referência da arquitetura holônica de manufatura sendo composta por três *holons* básicos: *holon* de pedido (OH – *Order Holon*), *holon* de produto (PH – *Product Holon*) e *holon* de recurso (RH – *Resource Holon*). Cada um deles sendo responsável por um aspecto específico. Os *holons* foram estruturados usando os conceitos de orientação a objetos, como agregação e especialização, permitindo representar de maneira bem integrada todos os elementos da manufatura.

Foi definido também um quarto *holon*, chamado de “*Staff Holon*” (SH), responsável por auxiliar os outros 3 *holons* básicos com conhecimento especialista sobre a produção. Além

disso, permite um comportamento centralizado, quando possível, e também facilita o acoplamento de outros sistemas, caso necessário.

Detalhando cada um dos *holons*, de acordo com Van Brussel (1998), temos:

- Holon de recurso: representa a parte física, um recurso da produção ou ainda a informação sobre o processo de uma peça. Responsável por fornecer informações sobre a capacidade de produção e as funcionalidades para os *holons* vizinhos. Por meio de métodos pré-definidos, é possível acessar o conhecimento para usar, organizar, alocar e controlar os recursos de produção. O *holon* de recurso pode ser a abstração da fábrica, do chão de fábrica, esteiras, fornalhas, *pallets*, ferramentas, estoque de material, entre outros.

Em contraste com a maioria das outras arquiteturas de controle, o HMS se diferencia por não separar o sistema de produção de seu controle, pois incorpora a parte física da produção dentro deste *holon*.

- Holon de produto: representa o processo e o conhecimento sobre o produto, para assegurar o correto processo de produção e garantir a qualidade. Este *holon* contém informações consistentes e atualizadas sobre o ciclo de vida do produto, requerimentos dos clientes, design, planos de processos, contas de materiais, processos para assegurar a qualidade, entre outros. Representa o modelo do produto e não o estado no qual um produto específico, que está sendo produzido, se encontra.

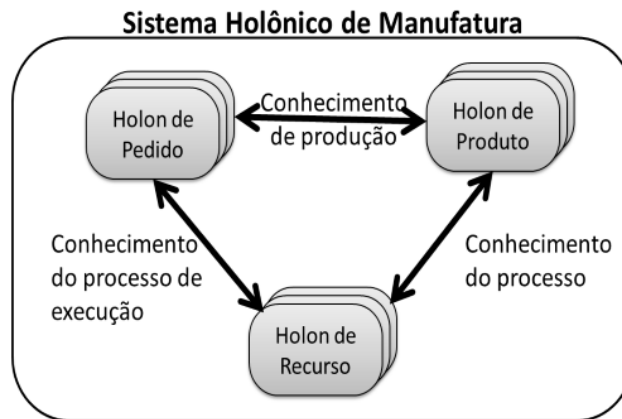
Para os outros *holons*, exerce o papel de um centro de informações sobre como produzir determinado produto, oferecendo métodos que abrangem o design, os planos de processos e a garantia de qualidade.

- Holon de pedido: representa uma tarefa no sistema de manufatura. É responsável por realizar a atribuição de tarefa de maneira correta no tempo certo. Gerencia todos os produtos que estão em produção, verificando: o estado do produto em relação ao seu modelo e a informação sobre o processo logístico relacionado a cada tarefa. Este *holon* pode ser representado pelos pedidos dos clientes, pedidos realizados para estocagem, pedidos de protótipos, manutenção e reparo de recursos, entre outros.

Fazendo uma comparação, este *holon* pode ser representado pela peça e seu comportamento durante sua produção ao longo da fábrica, realizando negociações com outras peças ou recursos para ser produzida. Como exemplo de um *holon* de pedido,

temos o bloco de programação de curto prazo, o despacho, a monitoria de progresso de uma peça, entre outros.

Na Figura 2.1 são apresentados os três tipos básicos de *holons*, propostos pela PROSA, assim como o conhecimento trocado sobre o sistema de produção. Esta comunicação é necessária para que todos os elementos do sistema sejam executados em harmonia.



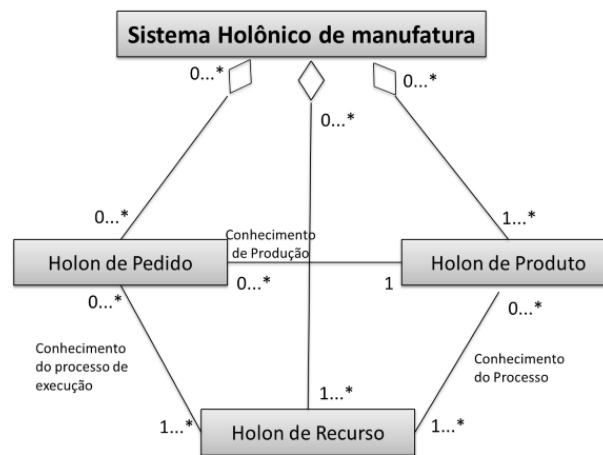
**Figura 2.1. Blocos básicos de construção do HMS e suas relações (adaptado de Van Brussel et al (1998)).**

Detalhando cada um dos conhecimentos trocados entre os *holons*, de acordo com Van Brussel (1998), temos:

- O conhecimento do processo, comunicação exercida pelos *holons* de produto e de recurso, contém informações e métodos de como realizar certos processos sobre determinados recursos. Este conhecimento representa a capacidade dos recursos, quais processos podem ser executados sobre eles, os parâmetros relevantes para o processo, entre outros.
- O conhecimento de produção, comunicação existente entre os *holons* de pedido e o de produto, representa a informação e os métodos responsáveis por informar como produzir determinado produto, levando em consideração os recursos. É o conhecimento sobre possíveis sequências de processos que podem ser executadas nos recursos, sobre métodos de como acessar o plano de processos, entre outros.
- O conhecimento do processo de execução, comunicação entre os *holons* de pedido e de recurso, contém informações e métodos relacionados ao progresso da execução dos processos nos recursos. É o conhecimento de como iniciar o pedido de execução de um

processo, realizar a reserva de recursos, como monitorar o progresso da execução, entre outros.

Para formalizar a modelagem dos *holons* e tornar esse modelo mais próximo de uma modelagem computacional, os autores optaram por usar a UML, baseados no fato de que as características que a modelagem por orientação a objetos proporciona, vão ao encontro das características dos sistemas holônicos de manufatura. Traduzindo a Figura 2.1 em um diagrama de classes, foi definido o diagrama apresentado na Figura 2.2.



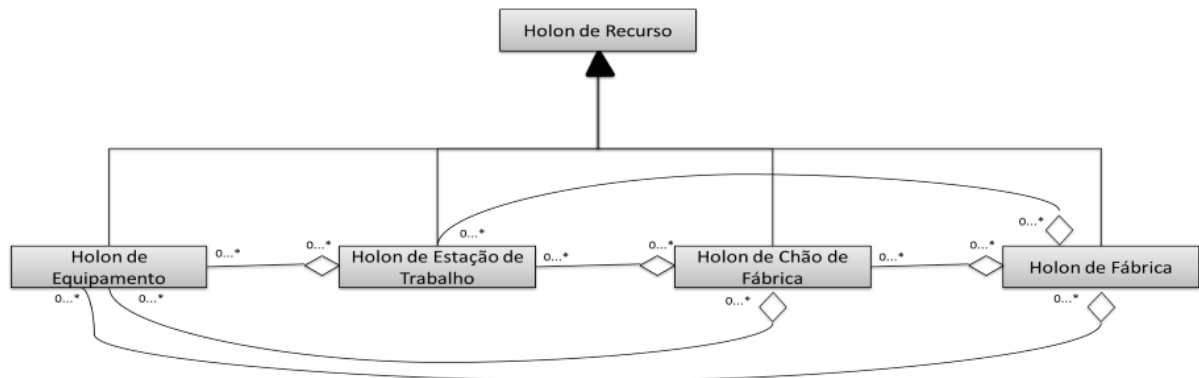
**Figura 2.2.** Versão UML dos blocos de construção de um HMS e suas relações (adaptado de Van Brussel (1998)).

O diagrama (Figura 2.2) apresenta cada entidade, representada pelos retângulos, sendo um tipo de *holon* do sistema. A relação entre as entidades, representadas pelas linhas, representa a relação entre eles. Além disso, com esse diagrama é possível diferenciar o tipo de dependência, por meio dos diferentes tipos de linhas, além de apresentar sua cardinalidade.

Existem três diferentes tipos de linhas, sendo elas: a seta com o diamante na ponta que caracteriza uma relação do tipo “tem um”, representando o conceito de agregação. A linha normal que caracteriza uma relação simples, representando o conceito de associação. Ainda existe a linha com a seta na ponta (não apresentado na Figura 2.2) que caracteriza uma relação do tipo “é um”, representando o conceito de especialização.

Com os conceitos de agregação, especialização e associação, é possível representar cada *holon* presente no sistema de manufatura, assim como suas relações e quantidade de *holons* existentes.

A associação nos permite caracterizar os conhecimentos compartilhados entre cada *holon*, enquanto a agregação auxilia na diminuição da complexidade do modelo, permitindo representar por um único *holon*, mais genérico, vários outros *holons*. Esse arranjo hierárquico pode ser analisado tanto por uma abordagem “*botton up*” quanto uma “*top-down*”, dependendo do aspecto que for necessário trabalhar, sendo os *holons* divididos em subconjuntos ou tratados como um único, conforme exemplo apresentado na Figura 2.3.



**Figura 2.3. Agregação dos *holons* de recurso (adaptado de Van Brussel (1998))**

Um ponto importante destacado por Van Brussel (1998) é que a hierarquia da agregação não é necessariamente como uma árvore, cada *holon* pode pertencer a múltiplas agregações. A agregação não é um conjunto estático de *holons*, mas sim dinâmico, permitindo mudar este conjunto de acordo com as necessidades do sistema de produção. Tal feito é possível graças à auto-organização dos *holons*.

A especialização permite separar os *holons* de acordo com suas características. A PROSA já usou o conceito de especialização logo no início para separar os *holons* de produto, pedido e recurso. Tais *holons*, em uma arquitetura específica, podem ainda ser muito abstratos para se modelar, para isso, a especialização permite distinguir os diferentes tipos de um *holon* básico, conforme exemplo apresentado na Figura 2.4.



Figura 2.4. Especialização do *Holon de Recurso* (adaptado de Van Brussel (1998))

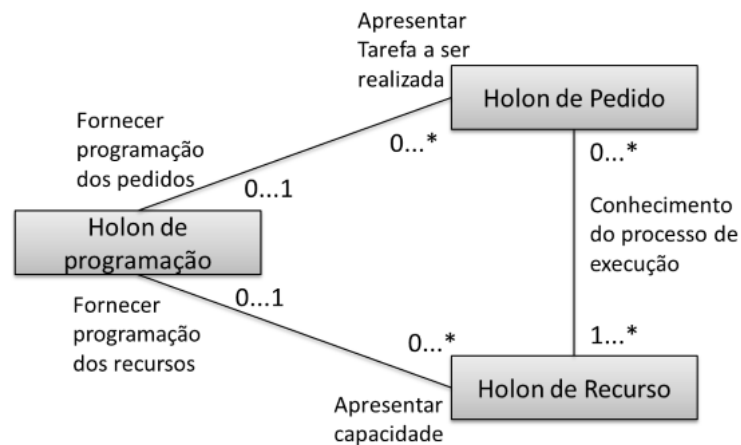
Conforme mencionado anteriormente, a arquitetura prevê a possibilidade de um quarto *holon*, o *Staff Holon*, para dar suporte aos *holons* básicos em suas tarefas. Ele leva em consideração alguns aspectos dos problemas dos *holons* básicos e provê informação suficiente para que os *holons* tomem a decisão correta na solução de problemas.

Mesmo com o *Staff Holon*, os *holons* básicos é que continuam sendo responsáveis por tomar a decisão, pois o quarto *holon* apenas aconselha, como o papel de um especialista externo à fábrica. Esses conselhos exercem um papel fundamental ao sistema, pois com ele, é possível definir um caráter centralizado ao sistema, fornecendo todas as vantagens de uma arquitetura desse tipo.

Essas vantagens podem ser desde o auxílio à solução de problemas, que seriam difíceis serem solucionados de maneira descentralizada, até auxiliar na migração de um sistema hierárquico para um sistema holônico. Mesmo com o uso deste *holon*, o sistema não se torna rígido como o hierárquico pelo fato de que a decisão continua sendo do *holon* básico.

Como exemplo para o uso do *Staff Holon*, Van Brussel (1998), representou um *Staff Holon* como sendo um *holon* responsável pela programação, que possui uma visão geral de todos os recursos e todos os pedidos. Este *holon* gera uma programação otimizada e fornece como conselho para os *holons* de pedidos e de recursos, conforme apresentado na Figura 2.5.

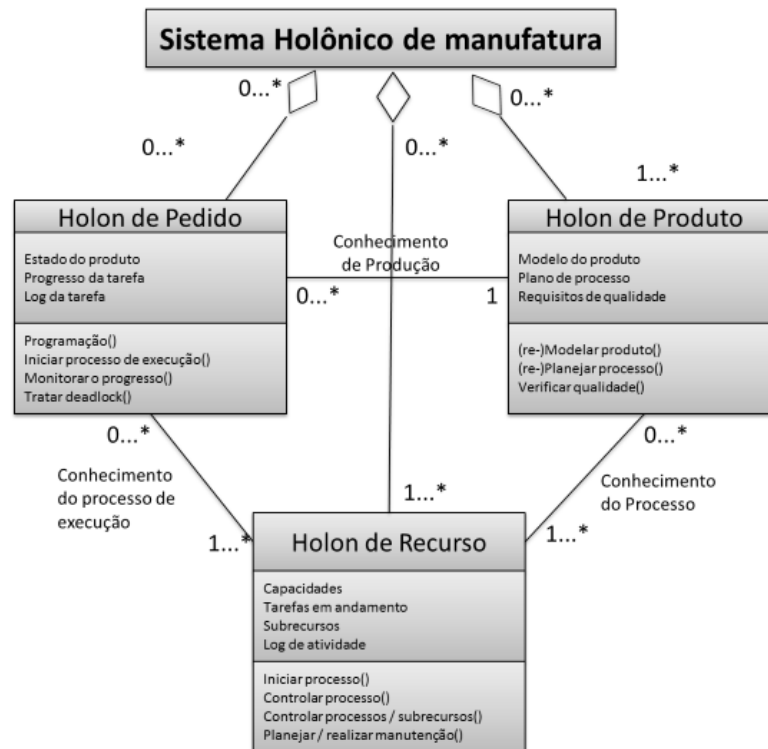




**Figura 2.5. Programador centralizado como *Staff Holon* (adaptado de van Brussel (1998)).**

Com a adição deste *holon*, o sistema pode apresentar um comportamento hierárquico, para resolver problemas de otimização, como por exemplo, o controle para um gargalo de recursos para aumentar o desempenho, ou apresentar um comportamento heterárquico para, por exemplo, resolver problemas relacionados a eventos não programados, como uma quebra de máquina forçando uma eventual mudança na programação. Para obter o comportamento heterárquico no sistema, basta os *holons* não aceitarem o conselho do *Staff Holon*.

Para detalhar melhor o funcionamento de cada *holon*, os autores optaram por usar o diagrama de classes, chamando-o de modelo funcional, pois não é simplesmente um meio de apresentar a interface de funções do *holon*, como em um diagrama de classes simples, mas sim descrever as funcionalidades que são abrangidas pelos *holons*. A Figura 2.6 apresenta as funcionalidades abrangidas pelos *holons* básicos por meio do diagrama de classes.



**Figura 2.6. Modelo funcional dos *holons* básicos representados por um diagrama de classes (adaptado de Van Brussel (1998)).**

Conforme apresentado na Figura 2.6, o *holon* de recurso está apto a iniciar o processo. Isso implica, de maneira implícita, que ele poderá aceitar ou rejeitar a tarefa que lhe foi atribuída. Também é responsável por executar o processo (suspender, reiniciar ou abortar), gerenciar os sub-recursos, controlar o processo, entre outros.

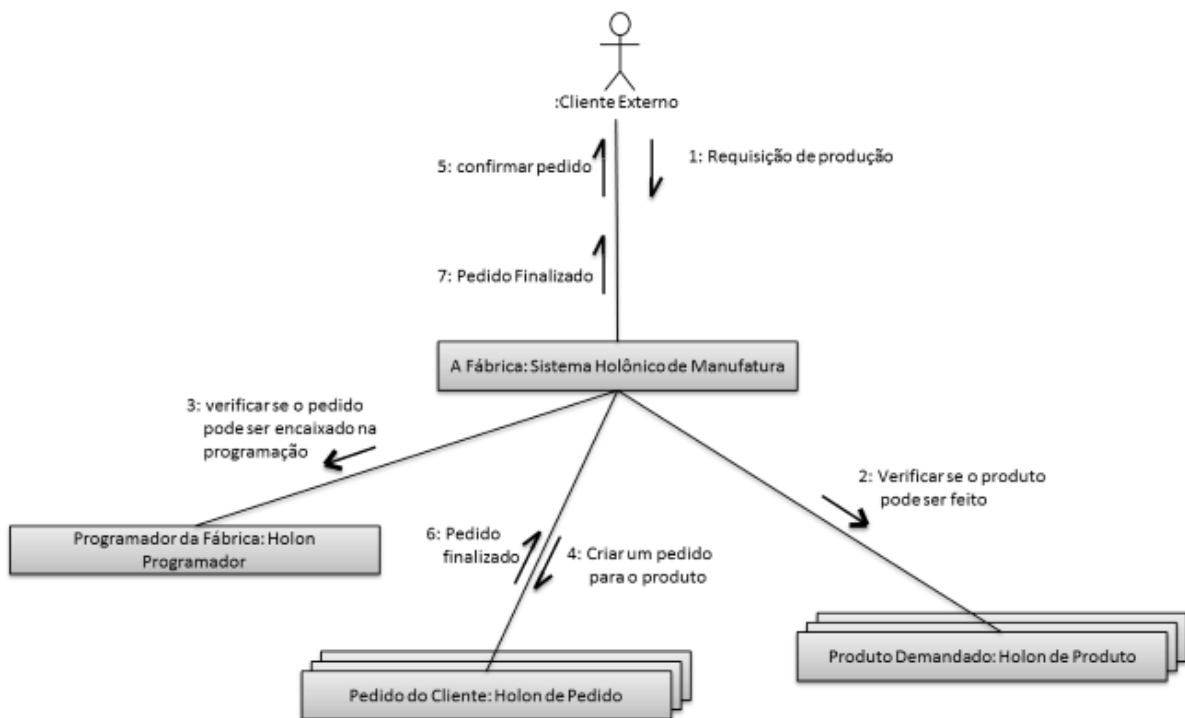
O *holon* de produto abrange as tarefas de (re)modelar, (re)planejar o processo e verificar a qualidade baseando-se no resultado das tarefas. O *holon* de pedido abrange a função de programação, conforme detalhado em Bongaerts *et al.* (1995), tratamento de *deadlocks*, monitoramento do progresso, além de desencadear o processo sobre um determinado recurso (iniciar, suspender, resumir, parar).

Cada *holon* possui um comportamento específico, além de interagir com os outros *holons* para atingir as metas individuais e as globais. Para modelar esse comportamento, assim como a interação entre os *holons*, Van Brussel (1998) propôs o uso de diagramas de colaboração (diagrama pertencente à UML).

Esse diagrama é composto de retângulos, que apresentam o nome e o tipo de cada *holon*, se tiver vários retângulos, significa que existe mais de um *holon* com o mesmo comportamento.

A ligação entre os *holons* é representada por uma linha, enquanto as setas representam as interações entre esses *holons*. A numeração presente em cada seta indica a sequência em que as interações irão ocorrer.

Ainda é possível indicar o tipo de interação por meio de uma seta comum ou pela metade, sendo a primeira referente à interação simples, em que o *holon* que iniciou espera o fim da execução para prosseguir, enquanto a segunda se refere a uma atividade assíncrona, na qual o *holon* que iniciou não espera o fim da execução para prosseguir. Também é possível representar transferência de dados entre os *holons*, para tal, usa-se um círculo na base da seta. A Figura 2.7, apresentada em Van Brussel (1998), indica um exemplo de modelagem do comportamento e interação de alguns *holons* para realizar o lançamento de um pedido por meio de um diagrama de colaboração.



**Figura 2.7. Realizando um pedido no HMS (adaptado de Van Brussel (1998)).**

A Figura 2.7 é a modelagem do comportamento de um HMS quando um cliente solicita a produção de um produto (seta 1) para “a fábrica”. O HMS verifica se é possível produzir esse tipo de produto (seta 2) e se é possível encaixar esse produto na programação (seta 3). Após esses procedimentos, é criado um *holon* de pedido específico (seta 4) e confirmado a aceitação da produção para o cliente (seta 5). Quando o *holon* de pedido finaliza a tarefa (seta 6), o HMS informa ao cliente (seta 7).

A partir dessa modelagem formal, assim como a definição dos *holons* existentes é possível iniciar uma arquitetura holônica, sendo necessário acrescentar os pontos específicos do HMS em questão, sejam esses pontos: novos *holons*, novas funcionalidades ou novos comportamentos.

### 2.2.1.2 ADACOR

Esta subseção foi baseada nos seguintes trabalhos: Leitão, Colombo e Restivo (2005); Leitão e Restivo (2006); Leitão e Restivo (2008); Barbosa *et al.* (2011); Leitão (2011).

A arquitetura holônica ADACOR, desenvolvida em conjunto pela universidade do Porto e pelo instituto politécnico de Bragança, foi proposta para melhorar o desempenho dos sistemas de controle de manufatura, aumentando a agilidade e a flexibilidade da empresa diante do ambiente dinâmico, caracterizado por frequentes distúrbios. O foco da ADACOR é no âmbito do chão de fábrica, especialmente em sistemas flexíveis, que possuem processos concorrentes e assíncronos com operações não preemptivas e rotas alternativas.

Diante do problema enfrentado no ambiente dinâmico, os autores propuseram uma arquitetura que fosse descentralizada quando possível e centralizada quando necessário, objetivando a centralização para a otimização e a descentralização na presença de eventos não programados e/ou modificações.

A proposta do ADACOR baseia-se no paradigma de sistemas holônicos de manufatura e segue como principais fundamentos: sistemas descentralizados, entidades supervisoras e auto-organização. As funções de controle da manufatura são executadas pela comunidade de *holons* autônomos e cooperativos, atingindo características como modularidade, descentralização, agilidade, flexibilidade, robustez e escalabilidade.

Com o papel do *holon* supervisor, elemento proposto na arquitetura, é possível estabelecer hierarquias nos sistemas descentralizados, permitindo a otimização global da produção. Já a capacidade de auto-organização permite uma evolução dinâmica e reconfigurável da estrutura de controle, unindo a otimização global da produção com a reação ágil a eventos não programados.

Para a modelagem da arquitetura, Leitão e Restivo (2006), tiveram como referência algumas arquiteturas, dentre elas a arquitetura PROSA. Também toda a proposta foi baseada nos conceitos de HMS propostas pelo IMS. Foram definidos quatro diferentes tipos de *holons*,

sendo eles: produto (PH), tarefa (TH), operacional (OH) e supervisor (SH), cada um com suas funções e objetivos.

Os *holons* de produto, tarefa e operacional são similares aos *holons* básicos da PROSA (produto, pedido e recurso), já o *holon* supervisor, adota características distintas dos *holons* supervisores da PROSA, pois na ADACOR, este *holon* introduz a coordenação e a otimização global no controle descentralizado, além de ser o responsável por formar e coordenar os grupos de *holons*.

Cada produto, passível de ser produzido na fábrica, é representado por um *holon* de produto, que contém toda a informação sobre o produto e é o responsável pelo planejamento de processo de curto prazo. O *holon* de produto (PH) atua como uma ponte entre o chão de fábrica e o nível de gerenciamento, realizando a integração de todas as funções de controle da fábrica.

Cada pedido de produção expedido para o chão de fábrica (para executar um produto ou um subproduto) é representado por um *holon* de tarefa (TH), que contém a informação atualizada do pedido de produção, assim como de sua execução.

O *holon* operacional (OH) representa a parte física dos recursos disponíveis no chão de fábrica, como por exemplo, operadores, esteiras, robôs, AGVs, entre outros. Também é responsável por gerenciar seus comportamentos de acordo com os objetivos e habilidades de cada elemento. Este *holon* também é responsável por gerenciar a programação de cada recurso, como por exemplo, a lista de tarefas a ser seguida para executar cada pedido ao longo do tempo.

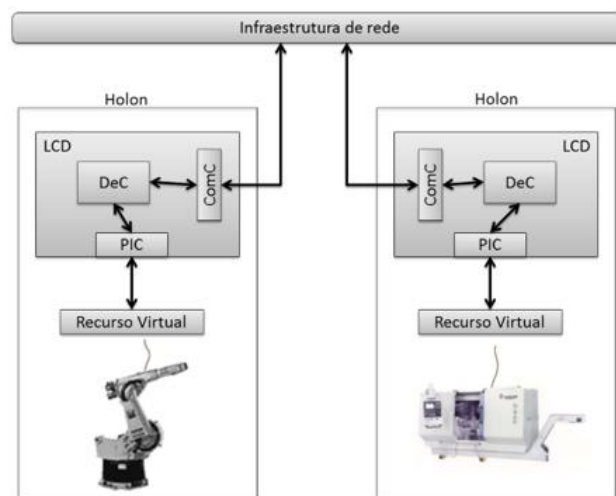
A otimização global é geralmente associada a uma hierarquia estável. Devido a essa hierarquia, é comum chegar à ideia da necessidade de um coordenador para realizar a sinergia, agregar as funcionalidades de cada membro do grupo e oferecer os serviços combinados para outras entidades do sistema de manufatura. Na arquitetura ADACOR, esse papel é realizado pelo *holon* supervisor (SH), que tem como sua principal função a elaboração de um planejamento de produção otimizado para os *holons* operacionais que estão sob sua coordenação. Essa programação otimizada é elaborada em um intervalo de tempo, definido pelo *clock* interno do *holon* ou pela ocorrência de um distúrbio.

O SH é responsável por coordenar grupos de *holons* e a evolução dinâmica que ocorre nesses grupos de acordo com o contexto do ambiente. A decisão de criar um grupo é definida pela necessidade de realizar a sinergia, para otimizar a produção, ou pela existência de uma restrição. À medida que um novo membro é adicionado ao grupo, o SH agrega as informações

relativas a esse novo membro, listando as funcionalidades que ele poderá contribuir para o grupo.

Um grupo pode representar, por exemplo, uma máquina equipada com um conjunto de ferramentas, uma célula de manufatura ou um chão de fábrica completo, com o SH assumindo o papel de coordenador.

Para facilitar a criação e a adição de novos *holons* no modelo, permitindo assim uma alta flexibilidade na adaptação do sistema e a possibilidade de reconfiguração, Leitão e Restivo (2006), propuseram um modelo genérico de *holon*, chamado de *holon* ADACOR. Este conceito genérico do modelo apresenta um dispositivo de controle lógico (LCD – *logical control device*) e um recurso físico capaz de realizar a tarefa de manufatura, caso exista (Figura 2.8). O LCD é organizado em três componentes principais: I) comunicação (ComC – *Communication Component*); II) decisão (DeC – *Decision Component*); III) interface física (PIC – *Physical Interface Component*).



**Figura 2.8.** Modelo conceitual para um *Holon* ADACOR (adaptado de Leitão e Restivo (2006))

O componente de comunicação, ComC, é responsável pela interação interholon, dando suporte à troca local de conhecimento distribuídos pelos *holons*. Para a interoperabilidade é necessário o uso de um vocabulário comum ou uma ontologia apropriada para a manufatura.

O componente de decisão, DeC, regula o comportamento do *holon*, isto é, realiza o controle das funções de manufatura, como por exemplo, planejamento do processo, programação, execução do planejamento (incluindo o despacho, monitoramento ou a

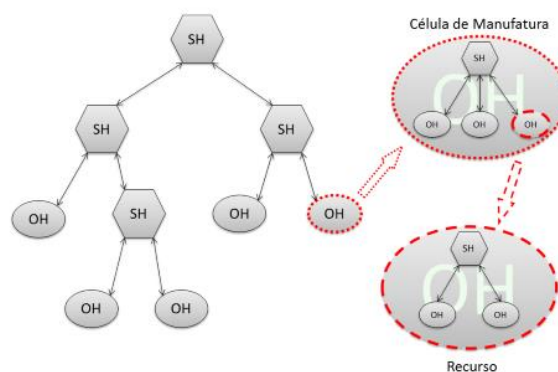
reorganização dinâmica). O modelo de tomada de decisão é compreendido basicamente como uma repetição sob as fases de sentir, decidir, agir e aprender.

Essas quatro fases compõem o comportamento de todos *holons*, pois cada um possui a capacidade de perceber o ambiente e responder rapidamente a mudanças e reagir a estímulos gerados pelo ambiente. Apesar desse comportamento de reação a estímulos, o *holon* também é capaz de tomar a iniciativa para agir no ambiente, como por exemplo, realizar uma predição de distúrbios futuros.

O *holon* permite a tomada de decisão contínua, de acordo com o conhecimento existente e da técnica de decisão implantada. O conhecimento é adquirido sensoriando o ambiente e por meio da comunicação com outros *holons*. Após a tomada de decisão, são enviadas mensagens para os outros *holons* informando-os sobre a execução que será realizada. Os resultados de tal execução são avaliados e com isso são gerados novos conhecimentos.

O componente de interface física, PIC, é necessário para criar um protocolo de comunicação com o componente físico, mas para isso se torna fundamental o papel do recurso virtual. Atua como a abstração de uma máquina, que representa as funcionalidades reais de tal equipamento, permitindo assim, invocar os métodos pelo lado do cliente.

Na ADACOR um OH pode ser composto por um conjunto de vários OH, como sendo partes físicas, e um SH agindo como componente lógico. Essa característica fractal permite o desenvolvimento de uma aplicação para a estrutura de controle da manufatura, por meio de encapsulamento das funções (Figura 2.9).

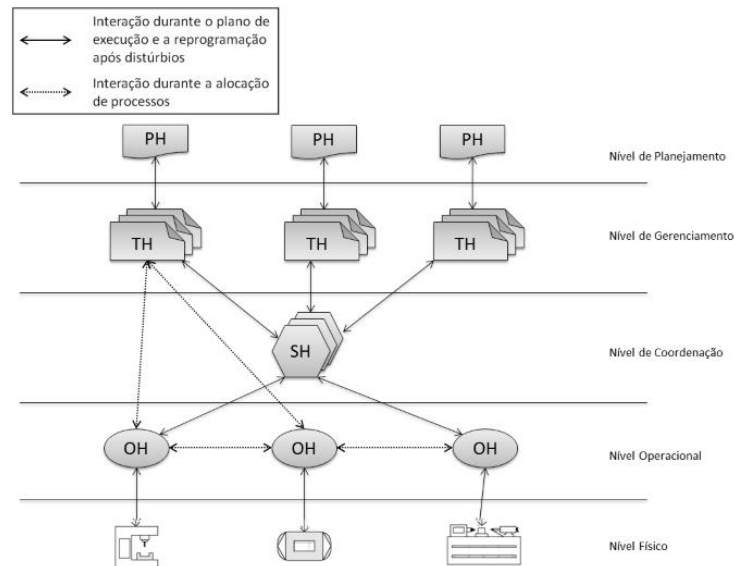


**Figura 2.9. Característica Fractal<sup>4</sup> na abordagem ADACOR (adaptado de Leitão e Restivo (2006))**

<sup>4</sup> Fractal é uma característica de auto semelhança na qual a parte se assemelha ao todo.

Como exemplo, apresentado na Figura 2.9, a célula de manufatura pode ser representada por um OH que é um conjunto de vários OH, sendo que cada um deles é a representação de um recurso, e um SH que exerce o papel de controlador da célula de manufatura. Outro OH, por exemplo, pode ser a representação de um estojo de ferramentas composto por outros OH que poderiam representar tais ferramentas.

No ambiente distribuído de manufatura, cada *holon* é autônomo e possui um conhecimento parcial do problema. O controle de manufatura como um todo, se dá a partir da interação dos *holons*, sendo que cada um deles contribui com sua parte de conhecimento. Na arquitetura ADACOR, durante o ciclo de vida de um pedido, existem diferentes tipos de interações entre os *holons* ADACOR, determinados de acordo com sua interdependência entre os tipos de *holons*. (Figura 2.10)



**Figura 2.10. Holons ADACOR e suas interações (adaptado de Leitão e Restivo (2006)).**

Na Figura 2.10, os *holons* de produto, contidos no nível de planejamento, interagem com os *holons* de tarefa, situados no nível de gerenciamento, para a troca de informações sobre produtos e planejamento de processos. De maneira indireta, os *holons* de produto interagem com os *holons* operacionais e supervisores durante a elaboração de planos de processos alternativos, pois existe a necessidade de verificar quais *holons* operacionais estão ativos na planta.

O *holon* de tarefa interage com os *holons* operacionais, situados no nível operacional, com o objetivo de executar o planejamento, monitorar e elaborar a reprogramação dinâmica e

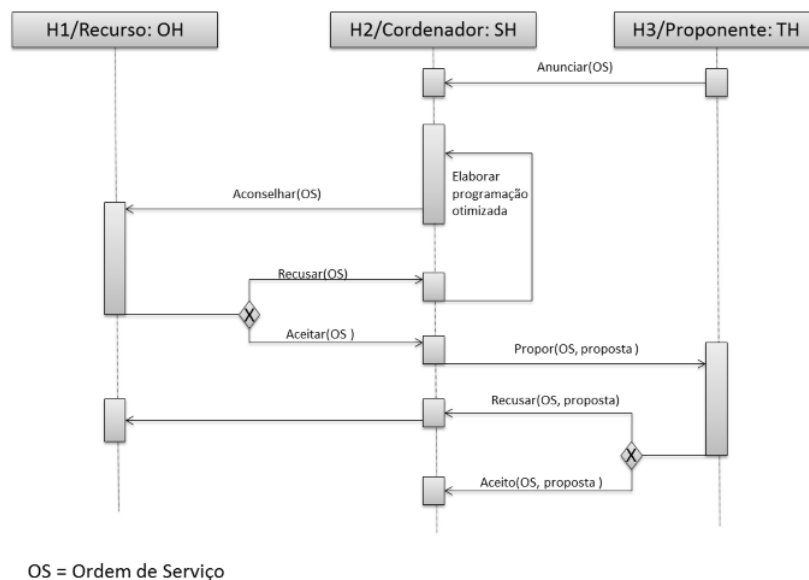


distribuída. Ainda para uma coordenação global e otimizada, os *holons* de tarefa, interagem com os *holons* supervisores, situados no nível de coordenação.

Nesses processos de coordenação, os *holons* supervisores propõem ao *holon* de tarefa a execução das operações de uma maneira otimizada, trocando informações relacionadas à alocação de recursos e monitoramento da execução dessas operações. A presença do nível de coordenação é opcional e só existe em ambientes estáveis, com o sistema rodando conforme o planejado e sem a ocorrência de eventos não programados.

Os *holons* operacionais interagem permitindo uma sincronização de atividades. Durante o tratamento de distúrbios, os *holons* operacionais interagem entre si e com o *holon* supervisor para habilitar a reorganização do controle da estrutura, e também interage com o *holon* de tarefa, para executar um plano alternativo para minimizar o impacto do evento não planejado.

Para a modelagem das interações existentes entre os *holons*, os autores optaram por usar o diagrama de interação da AUML, que é uma extensão da UML específica para a modelagem de sistemas multiagentes. Como exemplo, é apresentado na Figura 2.11 o diagrama de interação para o processo de coordenação global usando a estrutura de controle descentralizada.



**Figura 2.11. Diagrama de Interação para a coordenação global (adaptado de Leitão, Colombo e Restivo (2005)).**

Na Figura 2.11, o TH interage com o SH para anunciar novas ordens de serviço. Periodicamente o SH elabora a otimização da programação para coordenar os recursos, decompondo essa atividade em ordem de serviços individuais e a propõem aos OH. Os *holons*

operacionais recebem o conselho sobre a proposta de programação e dependendo da autonomia do *holon*, essa proposta pode ser aceita ou rejeitada. Caso o *holon* não aceite este conselho, o SH reprograma o plano de produção, procurando alternativas. Quando o OH aceita o conselho de ordem de serviço, o SH retorna ao TH toda a programação, sendo que este *holon* pode ou não aprovar essa proposta.

O mecanismo de adaptação do ADACOR inicia em uma abordagem “*bottom-up*”, iniciada a partir da auto-organização individual de cada *holon*. A adaptação dinâmica de cada *holon* a eventos não programados contribui para a adaptação ágil do sistema como um todo em caso de emergência.

O mecanismo de auto-organização requer forças dirigidas locais e globais para auxiliar na adaptação. Na arquitetura ADACOR, as forças dirigidas locais são o fator de autonomia de cada *holon* e a sua capacidade de aprendizado, características inerentes de cada *holon* ADACOR. A auto-organização global do sistema é atingida por meio da interação local dos *holons*, propagando a emergência e a necessidade de reorganização.

O fator de autonomia,  $\alpha$ , associado a cada *holon* operacional, é um parâmetro que reflete no grau de autonomia de cada *holon*. Esta variável pode ser contínua ou discreta, definida por um mecanismo de decisão que evolui dinamicamente com o objetivo de adaptar-se de acordo com o comportamento dos *holons* e as mudanças no ambiente em que eles se encontram.

Normalmente o OH possui um baixo fator de autonomia, aceitando todos os conselhos do SH coordenador. Com isso, o *holon* coordenador introduz a otimização global no sistema, pois ele possui uma visão mais ampla do que a dos OH que estão sob sua coordenação.

No caso de uma emergência, normalmente causada por um evento não planejado, é iniciado o comportamento adaptativo para determinar o fator de autonomia associado a cada OH, de acordo com um conjunto de regras previamente definidas. Este novo valor do fator de autonomia é uma função do seu valor atual, do tempo de reestabelecimento, do tempo estimado para recuperar-se do distúrbio e do parâmetro de feromônio, sendo este um indicador do nível de impacto do distúrbio.

Para a propagação da emergência, os autores optaram por usar o mecanismo semelhante ao do feromônio para realizar a indicação no âmbito global. No caso de um distúrbio no sistema, a necessidade de reorganização é propagada por meio da deposição de certa quantidade de

feromônio nos SH vizinhos, essa quantidade é estimada proporcionalmente ao tempo de reestabelecimento, de acordo com o tipo do distúrbio previsto e de acordo com o histórico.

Os *holons* associados a cada SH recebem o pedido de reorganização sentindo o feromônio propagado e o repassando para os *holons* vizinhos. A intensidade do odor do feromônio vai diminuindo à medida que essa variável vai subindo nos níveis dos SH, similar à distância na técnica original do feromônio.

A vantagem do uso desta técnica, destacada pelos autores, é a sua aderência à característica dinâmica e contínua de adaptação a sistemas que apresentam distúrbios, dão suporte a auto-organização global, reduzem a sobrecarga de mensagem e melhoram a reação diante dos distúrbios.

A arquitetura de controle é a chave para atingir o bom desempenho do sistema de controle da manufatura, exercendo um papel fundamental para que o sistema consiga obter bom desempenho em relação à resposta a mudanças e a capacidade de aprendizado.

Com as estruturas heterárquicas, é possível obter uma boa reação a distúrbios ocorridos no ambiente, porém perde-se em termos de otimização global. Por outro lado, as estruturas hierárquicas apresentam uma boa otimização global, mas perdem em termos de reação aos distúrbios. A arquitetura ADACOR implementa o conceito de auto-organização, que se torna a chave para balancear o sistema, alterando sua estrutura de controle, ora para hierárquica, ora para heterárquica, atingindo assim uma abordagem adaptativa de controle que combina as características de otimização global e agilidade diante de distúrbios.

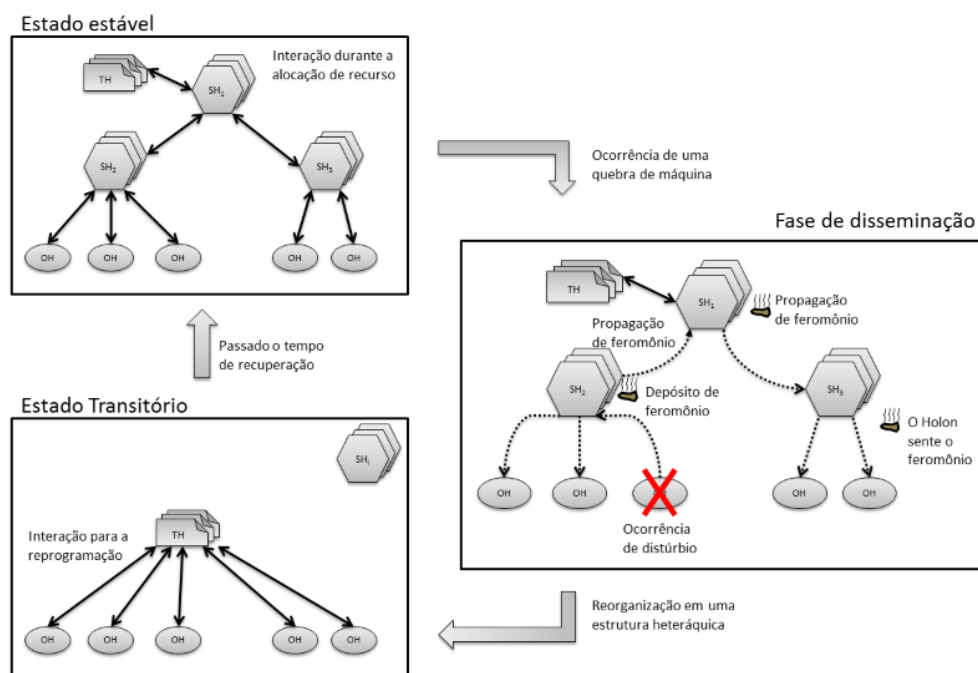
Para atingir a auto-organização, é usada a evolução adaptativa e dinâmica do fator de autonomia de cada *holon* ADACOR. A partir desse parâmetro, o controle do sistema pode evoluir para dois estados distintos, sendo eles:

- Estado estável, no qual o sistema de controle se pauta nos níveis de coordenação e no papel do supervisor para atingir a otimização global no processo de produção.
- Estado transitório, iniciado a partir da ocorrência de distúrbios, aproximando ao comportamento heterárquico em termos de agilidade e adaptabilidade.

Na Figura 2.12 é apresentada a evolução dinâmica da arquitetura de controle para um cenário que contém a fábrica, controlada pelo SH<sub>1</sub>, com duas células de manufatura, controlada

pelos *holons* supervisores 2 e 3, sendo essas células compostas, uma por três máquinas e a outra por duas máquinas (cada uma delas controladas por um *holon* operacional).

No estado estável os *holons* são organizados em uma estrutura hierárquica, com os *holons* supervisores representando controladores da célula e do chão de fábrica, interagindo diretamente com o *holon* de tarefa durante o processo de alocação de recursos. O *holon* supervisor elabora um plano de produção otimizado e propõe as tarefas aos *holons* operacionais que estão sob sua coordenação. Neste estado cada *holon* operacional possui um fator de autonomia baixo, o que leva a aceitação desse conselho enviado pelo supervisor.



**Figura 2.12. Abordagem ADACOR para a reorganização dinâmica (adaptado de Leitão e Restivo (2006)).**

Após as alocações das operações de manufatura, os *holons* de tarefa interagem diretamente com os *holons* operacionais durante a execução das operações. Quando um *holon* operacional rejeita uma ou mais tarefas, cabe ao *holon* supervisor replanejar a produção e enviar novamente como conselho aos *holons* operacionais.

Se por algum motivo o plano é desviado do previsto, por exemplo, devido a uma quebra de máquina, o sistema de controle entra em um estado transitório. Esse estado é caracterizado pela reorganização dos *holons* em uma arquitetura heterárquica, o que permite uma reação ágil da estrutura de controle frente ao distúrbio.

Para atingir essa mudança de estado, no caso de uma quebra de máquina, o próprio *holon* tenta se recuperar analisando os sintomas e realizando um autodiagnóstico, caso isso não seja suficiente, o fator de autonomia é aumentado e o *holon* propaga a necessidade de reorganização por meio do feromônio.

Os autores realizaram a validação da arquitetura em uma plataforma híbrida, parte virtual (simulado) e a outra parte física. E por meio de extensivos testes, simulando situações reais de um sistema flexível de controle, conseguiram observar com sucesso os ganhos obtidos por essa arquitetura.

### 2.2.2 Sistemas Multiagentes

Os sistemas multiagentes vêm se tornando cada vez mais populares sendo alvo de várias pesquisas devido as suas características, que segundo Wooldridge e Jennings (1995), viabiliza a resolução de problemas complexos de maneira não tradicional e de maneira descentralizada.

Por se tratar de uma área de conhecimento multidisciplinar, é possível achar várias definições de agentes que na sua grande maioria, aborda características específicas da área em questão. Porém uma definição bastante utilizada é a apresentada em Wooldridge e Jennings (2002), que define um agente sendo um sistema de computador que fica situado em um ambiente e que é capaz de executar ações autônomas para alcançar seus objetivos.

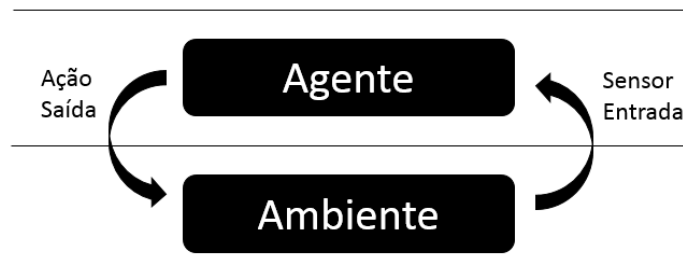
Russell e Norvig (2009) definem um agente como um sistema capaz de perceber por meio de sensores e agir em um ambiente específico através de atuadores.

Estas definições são apoiadas por um conjunto de propriedades que são amplamente atribuídas aos agentes, sendo elas:

- Autonomia: capacidade de definir a ação a ser tomada baseado na própria experiência. Estas ações não sofrem intervenção direta de humanos ou outros agentes;
- Habilidade social: capacidade de interagir com outros agentes ou com o ambiente por meio de um protocolo de comunicação;
- Reatividade: capacidade de reagir aos estímulos fornecidos pelo ambiente;
- Mobilidade: capacidade de migrar para outro ambiente;

- Pró atividade: capacidade de agir de acordo com seus objetivos e não apenas com os estímulos do ambiente.

Na Figura 2.13 é ilustrada uma visão abstrata de um agente e como ele afeta o seu ambiente.



**Figura 2.13.** Um agente e seu ambiente (adaptado de Wooldridge e Jennings (2002)).

Um agente é capaz de monitorar o ambiente em que ele se encontra, e é capaz de interferir no mesmo direcionado a seus objetivos. De acordo com Wooldridge e Jennings (2002), na maioria dos domínios de complexidade razoável, um agente não terá o controle completo sobre seu ambiente, sendo que o máximo que ele possui é um controle parcial. Esta observação se dá pelo fato de que o agente pode se encontrar em um ambiente não determinístico e que ao executar duas vezes a mesma ação, pode obter resultados completamente distintos. Conhecer o ambiente em que irá inserir o agente se torna de crucial importância para o sucesso de sua execução e para o entendimento de seu comportamento.

Para que os agentes possam interferir no ambiente, eles devem ter um repertório de ações sendo que cada ação pode e deve estar vinculada a uma pré-condição. Essas pré-condições em conjunto com seus objetivos, permitem ao agente distinguir quando executar uma ação em detrimento de outra, sendo que essa escolha é controlada pela arquitetura do agente.

Existem vários tipos de arquiteturas de agentes, sendo a arquitetura BDI (*Beliefs, Desires, Intention*) uma arquitetura bem difundida na literatura (WOOLDRIDGE, 2002). Esta arquitetura é construída a partir de estados mentais, permitindo assim produzir comportamentos adaptativos para os agentes.

A arquitetura BDI é definida por 3 conceitos: crenças (*Beliefs*), desejos (*Desires*) e intenções (*Intention*). As crenças definem a percepção que o agente tem do ambiente em um determinado momento, descrevendo assim o estado do mundo em que o agente está inserido. Os desejos estão relacionados ao que o agente deseja obter, porém a forma de alcançar esses

desejos pode não estar bem definida ainda. As intenções estão relacionadas às ações que o agente tem que executar para atingir seus objetivos.

Segundo Wooldridge (2002), no BDI os agentes são equipados com uma biblioteca de planos pré-compilados, definidos pelo programador do agente. Esses planos são definidos pelos seguintes componentes:

- Objetivo: Uma póscondição do plano;
- Contexto: Uma précondição do plano;
- Corpo: A parte das ações a serem tomadas pelo plano.

De acordo com Poslad (2007), sistemas multiagentes ou MAS (*Multi Agent System*) são sistemas distribuídos compostos por entidades de software autônomas chamadas de agentes, e é caracterizado pelo seu comportamento interno e interação externa entre os agentes. Poslad (2007) ainda define as propriedades para caracterizar o comportamento interno dos agentes como sendo: o tipo de cognição e a medida de desempenho utilizada para escolher a ação (reativa, baseada em modelo, baseada em objetivo, entre outras); quão adaptativos eles são; caracterização do ambiente em que eles estão situados (incluindo a infraestrutura computacional e o ambiente social gerado com outros agentes); e o grau de autonomia de suas ações.

Poslad (2007) afirma que o MAS é um modelo poderoso para solucionar problemas de computação distribuída, inclusive é capaz de adaptar suas operações para lidar com ambientes abertos e dinâmicos. Por causa dessas características, vem sendo aplicado no ambiente industrial há bastante tempo, conforme afirmam Jennings e Wooldridge (1998), que apontam a primeira solução MAS aplicado na indústria em meados de 1987.

Com o crescente aumento de soluções baseadas em sistemas multiagentes, foi criado em 1996 uma fundação para padronizar e permitir a interoperabilidade dos agentes. Esta fundação foi batizada de FIPA<sup>5</sup> (*Foundation for Intelligent Physical Agents*), e desde de 2005 faz parte do comitê de padronização da IEEE *Computer Society*.

A FIPA gerou um conjunto de especificações relativas a diferentes categorias relacionadas ao MAS, sendo elas: comunicação, transporte, gerenciamento, arquitetura e aplicação. A comunicação representa a categoria mais importante para a FIPA, sendo

---

<sup>5</sup> <http://www.fipa.org/>

responsável por estabelecer os padrões para comunicação entre agentes (linguagem usada, padrões de mensagens, protocolos de interação, entre outros).

### 2.2.3 Sistemas multiagente holônicos

Os MAS inicialmente apresentavam seus agentes como sendo um elemento atômico, não levando em conta um possível detalhamento desses agentes em termos de novos agentes.

À medida que as pesquisas foram aprofundando, surgiram questões relativas às organizações de agentes, questões como: como representar essa organização de agentes que possui um comportamento específico? Usando um nível de abstração, perceberam que essa organização se comporta como um único agente. A partir dessa observação apresentada por Gasser (1992), apesar da natureza do agente ser atômica, surgiu a ideia de agentes recursivos, também encontrado na literatura com o nome de agentes compostos.

O sistema multiagente holônico (HMAS – *Holonomic Mult Agent Systems*), é um sistema multiagente compostos por agentes holônicos, que segundo Rodriguez (2005) nada mais são que agentes compostos por outros agentes. Segundo Gerber *et al.* (1999), os agentes holônicos, ou simplesmente *holons*, subordinados a outros *holons*, são chamados de sub *holons*, enquanto o *holon* que agrega os outros *holons* é chamado de super *holon*.

Na literatura existem vários trabalhos que apontam a diferença entre *holons* e agentes (apesar da grande semelhança, possuem conceitos diferentes). Giret e Botti (2004) apresentam uma análise teórica detalhada sobre a diferença deles.

Para não desviar do escopo do trabalho e apresentar apenas os pontos necessários para a compreensão desta tese, consideraremos a diferença entre os dois sendo apenas o fato de que o agente, por natureza, é atômico e que o *holon* pode ser visto como o conjunto de outros *holons*, ou seja, abordaremos a diferença conforme apresentado em Rodriguez (2005) e Cossentino, *et al.* (2010a).

#### 2.2.3.1 ASPECS

Toda esta subseção foi baseada nos trabalhos: Cossentino *et al.* (2010a) e Cossentino *et al.* (2010b) e no site do projeto ASPECS<sup>6</sup>.

---

<sup>6</sup> <http://www.aspecs.org/Home>



O ASPECS, proposto por Cossentino *et al.* (2010a), é um método de software orientado a agentes, focado no desenvolvimento de sistemas complexos. Este método é baseado em um metamodelo organizacional holônico e provê diretrizes que abrangem desde o levantamento de requisitos até a parte de implementação e implantação do sistema, permitindo a modelagem de um sistema em diferentes níveis de detalhamento.

O método ASPECS é baseado em um ciclo de vida iterativo e incremental, sendo sua estrutura composta por fases, atividades e tarefas. Cada fase propõe uma coleção de metamodelos, compostos por um ou mais documentos de diferentes tipos. Estes documentos são gerados pelas atividades que compõem as fases, sendo que, para cada atividade são propostas tarefas para criar ou compor o modelo em questão. É importante ressaltar que, no final do método, o resultado obtido pela modelagem serão vários artefatos, e não apenas um. Cada um dos artefatos gerados fornece visões distintas da modelagem, focando em características específicas.

Cada fase no método é referente à um domínio diferente, sendo eles: 1º fase referente ao domínio do problema, 2º fase ao domínio dos agentes e a 3º fase ao domínio de solução.

O domínio do problema (1º fase) fornece uma descrição organizacional do problema de maneira independente da especificação da solução. Os conceitos introduzidos neste domínio são usados principalmente na fase de análise e no início da fase de projeto (Figura 2.14).

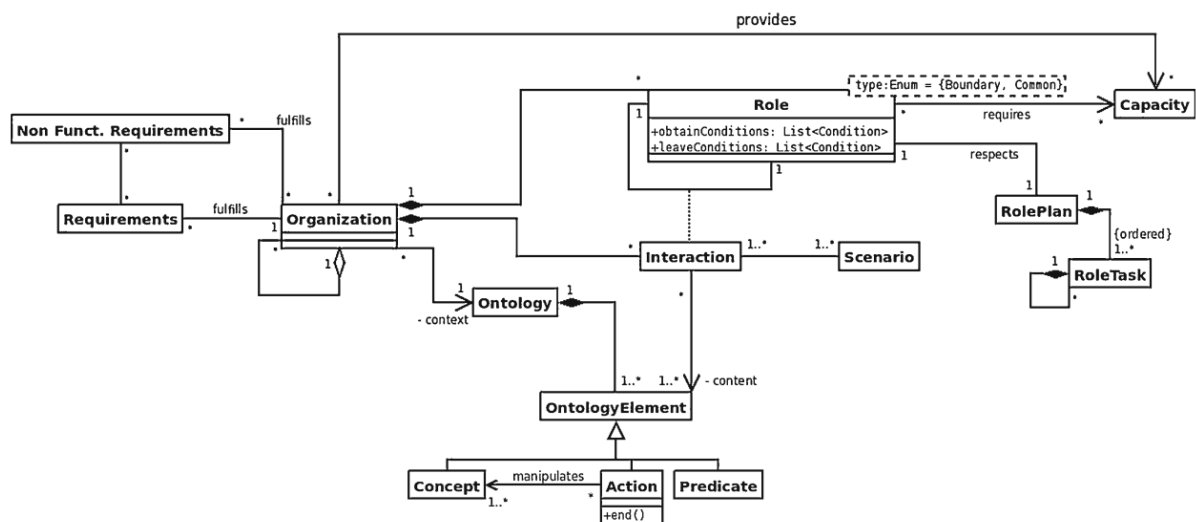


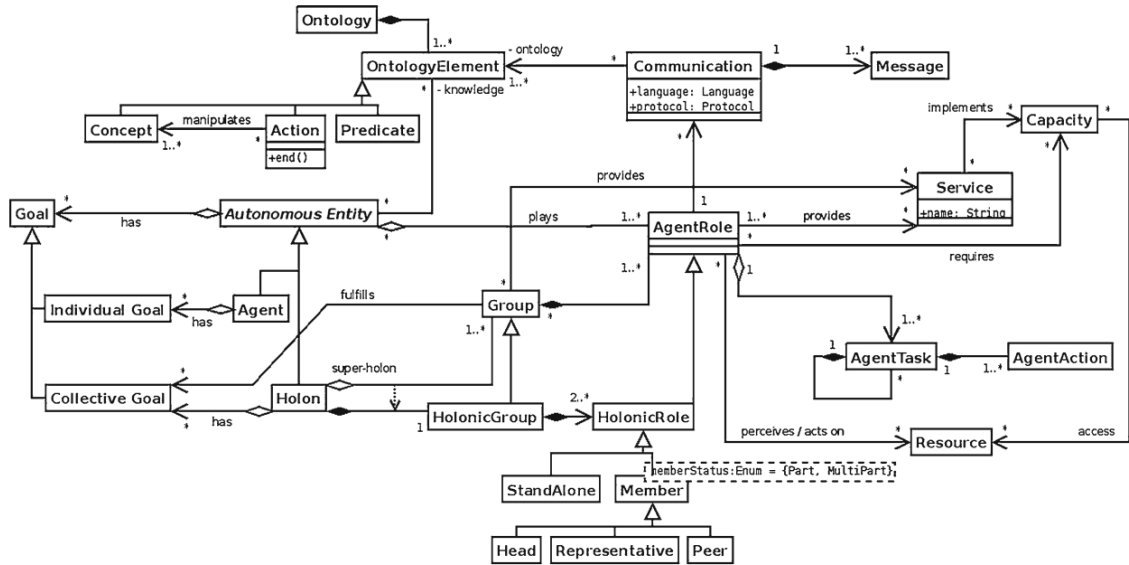
Figura 2.14. Diagrama UML do domínio do problema do metamodelo ASPECS (COSENTINO *et al.*, 2010a)

Para a compreensão da Figura 2.14, cada um de seus elementos está definido na tabela a seguir (Tabela 2.1 Conceitos chaves do domínio do problema).

**Tabela 2.1 Conceitos chaves do domínio do problema**

| <b>Termo</b>           | <b>Definição</b>   |
|------------------------|--|
| <b>Ontologia</b>       | Uma explicitação do conhecimento do domínio. Fornece uma visão distinta do domínio, incluindo a parte física da fábrica para a aplicação proposta no trabalho.   |
| <b>Organização</b>     | Uma organização é definida por um conjunto de requisitos que interagem entre si em um mesmo contexto. Este contexto consiste no conhecimento compartilhado e regras / normas do ambiente, e é definido de acordo com a ontologia. O objetivo de uma organização é representar certos requisitos do sistema.  |
| <b>Papel</b>           | Atribui um título (nome do papel) para um comportamento esperado e fornece uma ideia do que ele deverá fazer dentro do contexto da organização. Um papel pode ser especificado, ou como um "Papel comum", ou como um "Papel de fronteira". Um "Papel comum" é um papel localizado no interior do sistema e que interage com outros "Papéis comuns" ou com "Papéis de fronteira". Um "Papel de fronteira" é um papel localizado na fronteira entre o sistema e o seu exterior e é responsável por interações que acontecem nesta fronteira (ou seja, GUI, banco de dados, etc.) |
| <b>Interação</b>       | Uma dinâmica ou a forma com que um papel interage com outro papel ou com entidades externas ao sistema. Os papéis podem reagir a eventos de acordo com seus comportamentos.  |
| <b>Capacidade</b>      | É uma especificação de pós ou pré-condições de uma ação.   |
| <b>Tarefa de papel</b> | Uma tarefa detalha ações específicas relacionadas no plano de papel. A "Tarefa de papel" pode ser uma ação atômica ou ações compostas subordinadas e coordenadas. As definições destas "Tarefas de papel" podem ser baseadas em capacidades exigidas pelos papéis.   |
| <b>Plano de papel</b>  | Define o comportamento macro de um papel especificando ordens de execução de "Tarefas de papel" e suas interações para atender uma exigência.  |
| <b>Cenário</b>         | Descreve uma sequência de interações de papéis que atendem um requisito.   |

O domínio dos agentes (2º fase) introduz os conceitos relacionados aos agentes e fornece a descrição da solução holônica multiagente resultante de um refinamento dos elementos do domínio do problema (Figura 2.15).



**Figura 2.15. Diagrama UML do Domínio dos agentes do metamodelo ASPECS (COSSENTINO et al., 2010a)**

Na Tabela 2.2 são apresentadas as definições e relações dos elementos que compõem o domínio dos agentes (Figura 2.15).

**Tabela 2.2. Conceitos chaves do domínio dos agentes**

| Termo                   | Definição  |
|-------------------------|--|
| <b>Comunicação</b>      | Uma interação entre dois ou mais “Papéis” no qual o conteúdo (linguagem, ontologia e codificação) e a sequência de “atos de comunicação” (protocolos) são explicitamente detalhados. Uma comunicação é composta de mensagens expressando “atos comunicativos”. Em uma comunicação os participantes são os “Papéis” dos agentes e o conhecimento trocado entre eles é explicitamente representado por um conjunto de elementos da “Ontologia”. Um “Protocolo” define uma sequência de mensagens esperadas e representa um padrão comum de comunicação, sendo uma estratégia de alto nível que rege a troca de informação entre os “Papéis” dos agentes. |
| <b>Grupo</b>            | Uma instância de uma organização (definida no domínio do problema). Usado para realizar uma agregação de papéis de agentes executados pelos <i>holons</i> .  |
| <b>Papel de agente</b>  | Uma instância de um papel (definido no domínio do problema). Assim como na definição de “papel”, o papel do agente é o título dado a um conjunto de tarefas de agentes, que tem direitos e obrigações no contexto de um grupo específico. Os papéis de agentes interagem entre si usando comunicação dentro do contexto do grupo ao qual ele pertence. Geralmente os papéis de agentes são agrupados nas entidades autônomas que os executam. Ele também pode ser responsável por prover um ou mais serviços para o restante da comunidade.  |
| <b>Grupo holônico</b>   | É um grupo que contém exclusivamente papéis holônicos e é responsável pelo processo de decisão interna do <i>holon</i> (governança dos <i>holons</i> compostos). São usados papéis holônicos para representar em uma maneira organizacional a noção de moderação do grupo. Eles descrevem o nível de autoridade de um membro dentro da comunidade de membros e o grau de comprometimento deste membro com o <i>super holon</i> .   |
| <b>Tarefa de agente</b> | É um refinamento da tarefa de papel (definida no domínio do problema). É uma porção de um comportamento de um papel e pode ser composto por outras tarefas de agentes ou por ações atômicas. Pode contribuir para prover o serviço do papel de agente.   |
| <b>Ação de agente</b>   | Uma unidade atômica que compõe o um comportamento. Uma ação considera um conjunto de entradas e converte em um conjunto de saídas, embora um ou ambos os conjuntos   |

|                            |   |
|----------------------------|---|
|                            | possam estar vazios. Um exemplo da mais básica ação de agente consiste em invocar uma capacidade ou um serviço que requer a mesma saída.  |
| <b>Entidade autônoma</b>   | É uma entidade racional abstrata que adota uma decisão para satisfazer um ou mais de seus objetivos. Uma entidade autônoma pode executar um conjunto de papéis de agentes dentro de vários grupos. Estes papéis interagem entre si em um contexto provido pela própria entidade, sendo este contexto gerado pelo conhecimento, pelas capacidades possuídas pela entidade. Papéis compartilham o mesmo contexto pelo simples fato de fazerem parte da mesma entidade.  |
| <b>Agente</b>              | Uma entidade autônoma que especifica os objetivos individuais e a habilidade intrínseca para executar certas capacidades.   |
| <b>Objetivo</b>            | A descrição de uma meta perseguida e representa uma abstração de um estado que se pretende atingir.   |
| <b>Objetivo individual</b> | Uma meta individual perseguida por um agente que está relacionada com seus próprios desejos e intenções. Este agente considerará um plano ou estratégia própria para atingir seus objetivos.  |
| <b>Objetivo coletivo</b>   | Uma meta perseguida por uma comunidade de indivíduos, que possuem um compromisso (ou uma parte dele) com a comunidade. Geralmente os membros empenham objetivos coletivos, pois muitas vezes esses objetivos auxiliam na execução de objetivos individuais.   |
| <b>Serviço</b>             | Provê o resultado da execução de uma capacidade que cumpre um conjunto de funcionalidades de seu proprietário (um papel, um grupo, um agente ou um <i>holon</i> ). Essas funcionalidades podem ser efetivamente consideradas como uma implementação concreta de várias capacidades. Um papel pode publicar algumas de suas capacidades e outros membros do grupo podem obter ganho com troca de serviços executando-as. Assim como grupos podem prover serviços a outros grupos. A capacidade é um aspecto interno de uma organização ou de um agente, enquanto o serviço é designado para ser dividido entre várias organizações ou entidades. Para publicar a capacidade e permitir que outras entidades se beneficiem com isso, um serviço é criado. |
| <b>Recurso</b>             | Abstração de uma entidade do ambiente. Pode ser manipulado pelos papéis por meio de capacidades específicas.  |

O Domínio de solução (3º fase) é relacionado à implementação da solução em uma plataforma específica. Este domínio é, portanto, dependente de uma determinada aplicação e plataforma de implementação (Figura 2.16). Neste caso, esta parte do processo é baseada na plataforma JANUS<sup>7</sup> que foi projetada especificamente para facilitar a implementação holônica organizacional dos modelos.

<sup>7</sup> <http://www.janus-project.org/Home>

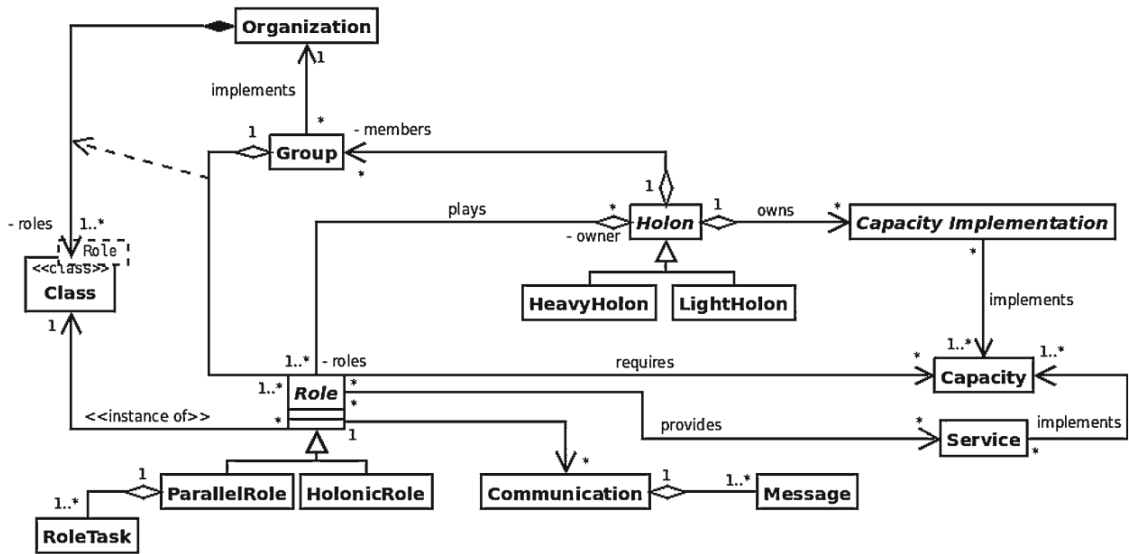


Figura 2.16. Diagrama UML do Domínio de solução do metamodelo ASPECS (COSENTINO et al., 2010a)

Como o método ASPECS aconselha o uso do framework JANUS para o domínio da solução (Figura 2.16), as definições e suas relações são apresentadas na tabela a seguir (Tabela 2.3).

Tabela 2.3. Conceitos chaves do domínio de solução.

| Termo              | Definição  |
|--------------------|--|
| <b>Organização</b> | Implementada como uma classe, possui um conjunto de classes de papéis. Uma organização pode ser instanciada na forma de grupos, contendo um grupo de instâncias de classes de papéis associadas às organizações as quais elas pertencem.   |
| <b>Papéis</b>      | Implementada por uma classe que apresenta um comportamento a ser executado. Cada papel é independente da entidade que o executa.   |
| <b>Holon</b>       | Possui dois tipos de <i>holons</i> , <i>heavy</i> e <i>Light holon</i> . O <i>heavy holon</i> possui seus próprios mecanismos de execução, o que o torna independente. Enquanto o <i>light holon</i> é associado com mecanismos de execuções síncronos o que o torna muito útil em algumas simulações baseadas em agentes. |
| <b>Capacidade</b>  | Permite a representação das competências dos <i>holons</i> . Funciona como uma interface entre o papel e o <i>holon</i> que o executa.   |

O processo definido pelo ASPECS está apresentado na Figura 2.17. Processo ASPECS, por meio de suas fases e atividades.

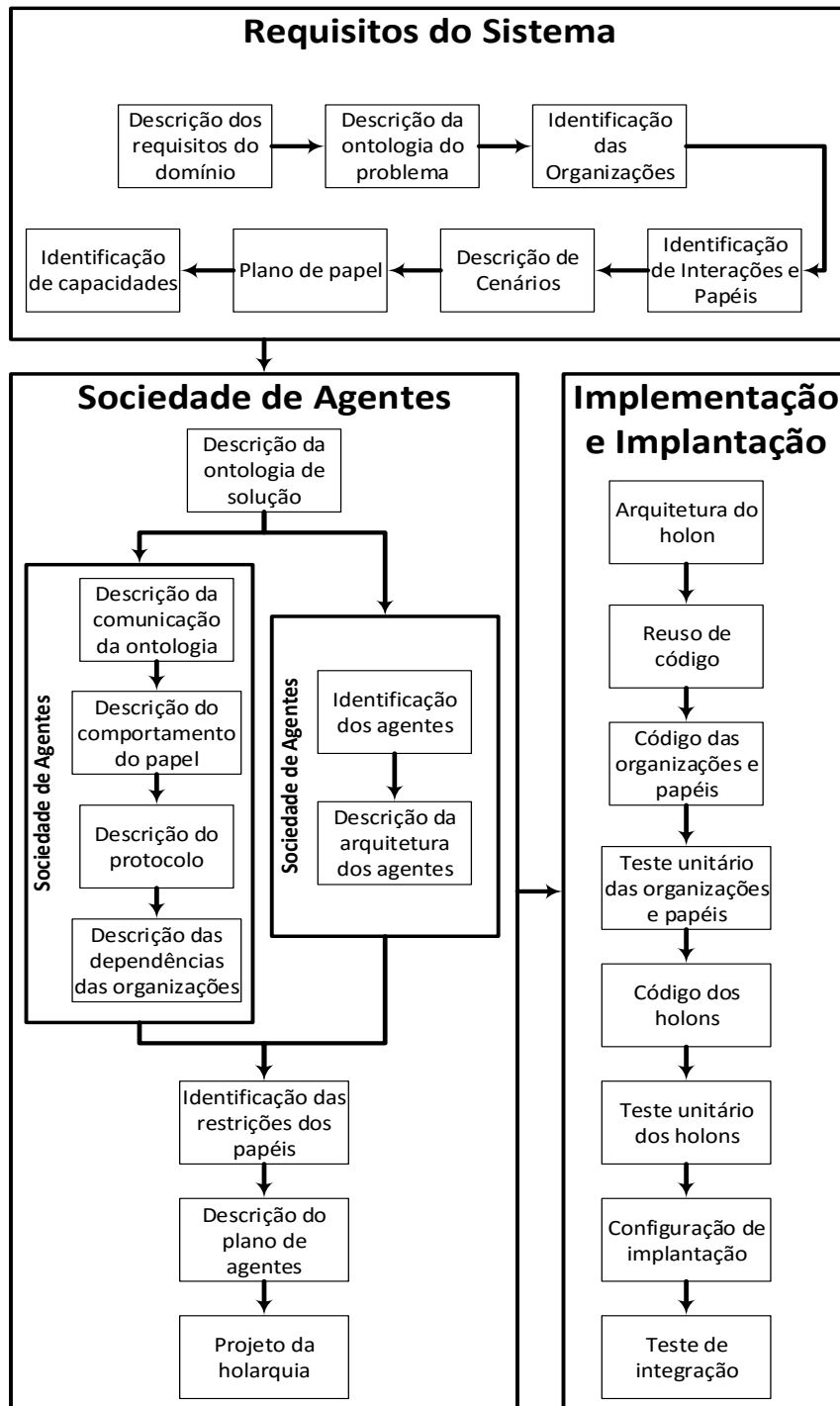


Figura 2.17. Processo ASPECS (adaptado de COSENTINO *et al.*, 2010a)

Apesar da Figura 2.17 apresentar um processo que pode parecer contínuo, o ciclo de vida do processo ASPECS é iterativo e incremental.

A primeira fase (requisitos do sistema) tem como objetivo a identificação de uma hierarquia de organizações, da qual o comportamento global deverá atender todos os requisitos do sistema. Ela é composta por 7 atividades, sendo elas:

1. Descrição dos requisitos do domínio (DRD - *Domain Requirements Description*):

O objetivo desta etapa é coletar as necessidades e as expectativas dos envolvidos e fornecer uma descrição completa do comportamento da aplicação (por meio de funções) a ser desenvolvida. Utiliza o caso de uso assim como seu diagrama como base.

Os casos de uso (assim como suas descrições) formalizam os requisitos levantados e que irão ser contemplados no modelo, além de apresentar uma descrição detalhada de seus funcionamentos.

2. Descrição da ontologia do problema (POD - *Problem Ontology Description*)

O objetivo desta etapa é fornecer uma visão geral do domínio do problema, permitindo assim descrever os elementos existentes. Para criar a ontologia é utilizado o diagrama de classe.

Neste diagrama são representados os elementos lógicos, e para o domínio de manufatura representado neste trabalho, também foram representados os elementos físicos da fábrica, fornecendo uma outra visão do domínio do problema, não só a visão funcional (como a apresentada no DRD).

3. Identificação das organizações (OID - *Organisation Identification*)

O objetivo desta etapa é vincular cada um dos requisitos (definidos no DRD) a um comportamento global pertencente a uma organização. O comportamento global pode ser identificado de duas formas, estrutural ou funcional. Na estrutural, é identificado estruturas similares para formar a organização; já na funcional, são agrupadas funções que possuem uma certa similaridade e com isso é criado uma organização. Para esta atividade é usado o diagrama de caso de uso.

Para o domínio de aplicação específico, identificou-se a necessidade de criar uma organização que represente a fábrica fisicamente, composta pelos elementos físicos identificados anteriormente.

4. Identificação dos papéis e interações (IRI - *Interactions and Role Identification*)

O objetivo desta etapa é decompor o comportamento global pertencente a uma organização, em pequenos “comportamentos de interação”. Cada um destes

comportamentos será representado por um papel. Para esta atividade é usado o diagrama de classe.

5. Descrição de cenários (SD - *Scenario Description*)

O objetivo desta etapa é descrever a sequência de interações que ocorrem entre os papéis envolvidos em cada cenário. Para tal atividade, usa-se o diagrama de sequência.

6. Plano dos papéis (RP - *Role Plan*)

O objetivo desta etapa é contribuir para o cumprimento de todo, ou parte dos requisitos de uma organização conforme foi definida nas etapas anteriores. O comportamento de um papel é descrito por um plano de papel. Usa-se o diagrama de atividade.

7. Identificação das capacidades (CP - *Capacity Identification*)

O objetivo desta etapa é definir comportamento genérico para um papel, identificando quais são as pré e pós condições exigido por um papel para ser executado por um elemento.

A segunda fase (sociedade de agentes) tem como objetivo definir a sociedade de agentes cujo comportamento irá fornecer uma solução para o problema modelado na fase anterior. É composta por 10 atividades, sendo que depois da primeira atividade há uma bifurcação no processo, dividindo-o em projeto de agentes (composto por 2 atividades) e projeto de organização (composto por 4 atividades).

1. Descrição da ontologia de solução (SOD – *Solution Ontology Description*)

O objetivo desta atividade consiste em refinar a ontologia do problema descrita na atividade POD, adicionando novos conceitos relacionados a solução baseada em agentes e refinando os elementos existentes.

2. (Projeto de agentes) Identificação de agentes (AI – *Agent Identification*)

Consiste em identificar os agentes que irão compor o mais baixo nível da hierarquia, assim como suas responsabilidades. As responsabilidades são modeladas usando a noção de objetivos individuais e será a base para identificar a arquitetura de agentes



da próxima atividade. Esta atividade é representada pelo diagrama de objetivos do TROPOS (GIUNCHIGLIA; MYLOPOULOS; PERINI, 2002).

3. (Projeto de agentes) Descrição da arquitetura de agentes (AAD – *Agent Architecture Description*)

O objetivo desta atividade é prover indicações precisas das arquiteturas adotadas pelos agentes. A arquitetura de agentes é definida pelo menos por um conjunto de papéis executados pelos agentes e um conjunto mínimo de serviços que implementam as capacidades requisitados pelos papéis. Neste diagrama é apresentado os agentes e suas capacidades de execução. Usa-se o diagrama de classes da UML.

4. (Projeto de organizações) Descrição da Ontologia de Comunicação (COD – *Communication Ontological Description*)

O objetivo desta atividade é descrever as comunicações existentes entre os papéis. A comunicação é definida por uma interação entre dois ou mais papéis, da qual seu conteúdo (linguagem, ontologia e codificação) e a sequência de atos de comunicação (protocolo) são explicitadas. É aconselhado a usar os protocolos padronizados da FIPA<sup>8</sup>. Para esta atividade é usado o diagrama de classes da UML.

5. (Projeto de organizações) Descrição do Comportamento do Papel (RBD – *Role Behaviour Description*)

O objetivo desta atividade é definir o ciclo de vida completo de um papel. Esta atividade é um refinamento do RP (1º fase). O comportamento de cada papel é descrito usando o diagrama de atividades da UML.

6. (Projeto de organizações) Descrição do Protocolo (PD – *Protocol description*)

O objetivo desta atividade é definir protocolos específicos, criados para atender a atividade COD, caso os protocolos padronizados pela FIPA não atenderem algum caso. Esta tarefa é opcional visto que é possível usar os protocolos FIPA.

---

<sup>8</sup> <http://www.fipa.org/specs/fipa00037/SC00037J.html>

7. (Projeto de organizações) Descrição das Dependências da organização (ODD – *Organisation Dependencies Description*)

Define as relações entre: I) capacidades requisitadas pelos papéis e organizações e II) serviços que as realizam. Também é possível identificar os recursos existentes (acessados pelos papéis de fronteira).

Os recursos são abstrações das entidades do ambiente que são acessados pelos papéis de fronteira. Para acessar os recursos, os papéis necessitam de capacidades específicas que serão introduzidas especificamente para esse motivo e serão executadas por serviços, caso seja necessário. Com isso, são explicitadas as dependências que as organizações possuem com o mundo externo.

8. Identificação das restrições dos papéis (RCI – *Role Constraints Identification*)

Esta atividade tem como objetivo identificar as restrições entre os papéis, por exemplo, papéis que tem que ser executados simultaneamente, prioridades de execução, exclusão mútua, dependências, etc. Para tal atividade, usa-se o diagrama de classes.

9. Descrição do plano de agente (APD – *Agent Plan Description*)

Esta atividade tem como objetivo finalizar o projeto da arquitetura interna dos agentes. Cada um dos agentes do sistema é associado a um conjunto de papéis que eles têm que executar de acordo com suas capacidades. Usa-se um diagrama de atividades da UML.

10. Projeto da holarquia (HD – *Holarchy Design*)

Neste ponto do método, as organizações, papéis e comunicação já foram identificados e especificados. Esta atividade tem como objetivo transformar os diagramas anteriores em uma holarquia, descrevendo toda a estrutura e identificando as regras que governam sua dinâmica. Esta atividade é composta em 4 tarefas:

- a. Tarefa de holonificação: Mapear a hierarquia de organizações em uma holarquia. Este mapeamento é realizado na associação dos *holons* que compõem a holarquia com um conjunto de papéis que eles têm que executar,

definidos na hierarquia de organizações, sendo elas instanciadas na forma de grupos.

- b. Tarefa de definição do modelo de governança do holon: Identificar o tipo de governança dos *holons* criados. Descreve as várias regras usadas para a tomada de decisão nos super *holons*. Dois aspectos de tomada de decisão devem ser analisados: I) quem está no controle das decisões e como isso acontece (voto, um único que manda, etc.); II) quem é contatado por um *holon* externo que quer entrar no super *holon* ou requisitar um serviço e também como esse processo é iniciado.
- c. Tarefa de definição da holarquia: Descrever os elementos anteriores que combinam para obter um conjunto completo de *holons* (compostos ou não) envolvidos em uma solução. O Resultado desta tarefa é descrito em uma extensão do “*cheese-board diagram*” (proposto por Ferber, 1999).
- d. Tarefa de definição do mecanismo de auto-organização do holon: Especifica a criação do mecanismo de auto-organização do *holon* para complementar a tarefa anterior. Esse mecanismo pode ser a criação de um novo membro integrante do super *holon*, política de agendamento dos papéis, etc. Esta atividade auxilia o mecanismo de evolução da holarquia.

A terceira fase (implementação e implantação) tem como objetivo a implementação e a implantação da solução orientada a agentes projetada na segunda fase, adaptando a plataforma de implementação escolhida (neste caso, JANUS). Esta fase é composta por 8 atividades.

1. Definição da arquitetura do *holon* (HAD – *Holon Architecture Description*)

Esta tarefa tem como objetivo definir a arquitetura de cada *holon* envolvido na implementação da atividade anterior (HD). Cada organização em conjunto com seus conjuntos de papéis e tarefas associadas devem ser descritas. Cada *holon* é associado a um conjunto de papéis que ele deve executar e um conjunto de capacidades e serviços que ele possui. Para esta atividade é usado o diagrama de classe da UML.

2. Reuso de código (CR – *Code Reuse*)

Esta atividade tem como objetivo integrar os códigos dos padrões (usados nas atividades OID, IRI e na tarefa de identificação da governança do *holon*) presentes na aplicação.

3. Produção do código das organizações e papéis (CPOR – *Code Production of Organizations and Roles*)

Esta atividade tem como objetivo a codificação das organizações e papéis identificado nas atividades anteriores, sendo elas os blocos mais elementares do JANUS.

4. Atividades de teste (TA – *Test Activities*)

A abordagem usada para testes no ASPECS consiste em testes unitários para cada papel e organização em todos os contextos aos quais eles pertencem.

5. Produção do código dos *holons* (CPH – *Code Production of Holons*)

Esta atividade tem como objetivo a codificação dos *holons* modelados nas etapas anteriores.

6. Teste unitário dos *holons* (HUT – *Holon Unit Test*)

A abordagem usada para testes nesta atividade consiste em realizar testes unitários em todos os *holons* codificados.

7. Configuração da implantação (DC – *Deployment Configuration*)

Esta atividade tem como objetivo detalhar como o desenvolvimento anterior será implementado. Inclui detalhes relacionados aos aspectos de distribuição, locais dos *holons*, relação com dispositivos externos e recursos.

8. Teste de integração (IT – *Integration Test*)

Esta atividade tem como objetivo realizar o teste de integração de todo o sistema, confirmando a perfeita execução de todos os *holons*, papéis, organizações e capacidades.

## 2.3 Artigos correlatos

Esta seção foi dividida em 2 subseções, conforme o mapa mental apresentado representado na Figura 2.18



**Figura 2.18.** Mapa mental da organização dos trabalhos correlatos.

Cada uma das ramificações da Figura 2.18 apresenta uma série de trabalhos correlatos que serviram para corroborar com a tese, sendo que estes trabalhos estão divididos por áreas do conhecimento, seguindo a seguinte divisão: I) Despacho de veículos, que apresenta trabalhos que abordam técnicas e métodos para realizar o despacho, desde despacho por regras de um único critério, até despachos mais elaborados que apresentam mais de um critério. II) Sistemas holônicos de manufatura, apresentando propostas de maneira ampla, abordando problemas de arquiteturas, programação da produção, sistemas de execução de manufatura, entre outros. Também apresenta trabalhos que optaram por usar MAS para implementar o HMS.

### 2.3.1 Despacho de veículos

As primeiras regras de despacho propostas na literatura levavam em consideração apenas um atributo, como apresentado por Egbelu e Tanchoco (1984), que avaliaram o desempenho de várias regras dentre elas: veículo mais próximo, veículo menos utilizado, menor tempo de viagem; Egbelu (1987), apresentou uma regra de despacho tendo como objetivo atender as estações de trabalho ociosas; Bartholdi III e Platzman (1989), propuseram uma

estratégia que entrega a carga ao primeiro veículo que aparecer; Han e McGinnis (1989), apresentaram uma regra para atender o mais significativo, para isso foi calculado o índice de prioridade de cada estação de trabalho; já Taghaboni (1989), baseou-se no *buffer* de saída das máquinas, definindo um índice crítico e um índice de prioridade para tentar diminuir o bloqueio de máquina.

Porém, devido à alta complexidade e a dinamicidade do ambiente de manufatura, notou-se que atender apenas um critério não era suficiente para atingir um bom resultado na maioria dos casos. Para contornar esse problema, os pesquisadores começaram a avaliar mais de um critério para realizar a tomada de decisão.

Para comprovar a melhor eficiência das regras multicritérios, Kim, Tanchoco e Koo (1999), apresentaram um estudo comparativo entre regras multicritérios e de um único critério. Para tal avaliação, foi medido por meio de simulação o desempenho em relação às variáveis: tempo de produção de uma peça, tempo de viagem do AGV, tamanho da fila do *buffer* de saída e tempo de espera por transporte. Observou-se que as regras multicritérios obtiveram melhores resultados.

Outras abordagens multicritérios foram levantadas e avaliadas, como por exemplo, em Kim e Hwang (1999), que propuseram uma regra de despacho que mescla além do tempo de viagem do AGV, o inventário em processo (WIP – *work in process*) no *buffer* das estações de trabalho envolvidas na tarefa (origem e destino). Também propuseram um algoritmo adaptativo que atualiza os parâmetros de cada estação, ajudando a prevenir que máquinas que estiverem com um dos *buffers* cheios requisitem tarefas. Foi comparado por meio de simulações com outras regras de despacho já existentes na literatura e foi detectada a eficiência da regra proposta assim como da abordagem adaptativa.

O trabalho de Tan e Tang (2000) apresentou uma abordagem híbrida *Fuzzy*-Taguchi, sendo o *Fuzzy* usado para a modelagem da regra multicritério e o Taguchi para realizar a sintonia fina nas regras *Fuzzy* e ainda permitir uma análise de sensibilidade em cada uma delas. A modelagem foi composta por duas variáveis *Fuzzy*, uma baseada na demanda e outra na fonte (tarefa puxada e empurrada). Foi realizada uma bateria de testes avaliando a taxa de produção, tempo decorrido para produzir um número fixo de peças, tamanho da frota e o número de níveis. Após os testes foi validado estatisticamente e comparando com algumas soluções apresentadas na literatura.

Benicasa, Morandin e Kato (2003), propuseram um sistema *Fuzzy* para realizar o despacho de veículos usando uma regra multicritérios que considera a distância entre o AGV e a estação de trabalho, o número de nós no percurso e o espaço restante no *buffer* de saída da estação de trabalho. A escolha por essas variáveis se deu após um estudo da literatura e de verificar que a maior parte dos trabalhos levantados usa a distância entre o AGV e a estação de trabalho e o número de espaços do *buffer* de saída. Já o número de nós foi considerado, pois a modelagem foi dividida em zonas nas quais apenas um AGV pode entrar de cada vez para evitar conflitos. O trabalho foi validado comparando-se o seu desempenho com o da regra FIFO, mostrando-se superior para o caso aplicado.

O trabalho proposto por Kizil, Ozbayrak e Papadopoulou (2006), abordou a preocupação com a paralisação da produção nos sistemas composto por células de manufatura, quando há um bloqueio de cargas entre as intercélulas e as intracélulas, causadas pela sobrecarga de peças. Para solucionar esse problema, os autores propuseram uma regra baseada no número de operações para a produção de um produto (MNP – *Minimum Number of Processing*). Os mesmos são ordenados em uma fila, sendo que os produtos que ocuparem menos estações de trabalhos serão produzidos primeiro, diminuindo assim a carga nas estações de trabalho. Por meio de simulação, foi comparado o desempenho da regra proposta com outras encontradas na literatura e ela obteve melhores resultados em relação ao tempo de produção e a taxa de produção e ainda apresentou um menor número de bloqueio do sistema.

Umashankar e Karthik (2006) apresentaram uma abordagem inteligente para o despacho de AGVs baseado em um controlador *Fuzzy* multicritério. O controlador foi composto por duas fases, sendo a primeira para a definição de qual AGV irá atender a máquina por meio do índice da programação, e a segunda, para resolução de conflitos, caso dois AGVs tenham sido ranqueados igualmente, se baseará no índice de resolução de conflito. Para a segunda fase, foi proposto um controlador *Fuzzy* que avalia a utilização dos AGVs, a fila do *buffer* de saída da estação de trabalho, a distância a ser percorrida e a taxa de ociosidade de cada um dos AGVs. Foi realizada a validação com o apoio da ferramenta MATLAB na qual foi simulado um cenário e avaliado o seu desempenho, sendo que segundo os autores, o sistema teve 100% de acerto na escolha do melhor AGV para atender uma determinada tarefa.

Morandin *et al.* (2007), baseado no trabalho citado anteriormente de Benicasa (2003), propôs a construção da base de regras por meio de um algoritmo genético. Para a validação,

foram comparados os dois trabalhos, sendo que a geração da base de regras por meio de um algoritmo genético se mostrou mais eficiente.

Os autores Kesen e Baykoç (2007), apresentaram a simulação de um sistema de produção *job-shop* agregado à estratégia JIT (do inglês, *Just In Time*) para verificar o desempenho por meio de simulação neste tipo de ambiente. A opção por simulação foi feita pelo fato de se tratar de um sistema com muitas variáveis, o que seria mais complicado usar a abordagem de modelos analíticos, pois seria necessário fazer várias suposições tornando o modelo menos próximo do real. Já a escolha pela estratégia JIT, mesmo não sendo muito usado em sistemas *job-shop*, é amplamente reconhecida por sua eficiência. A partir dessa suposição, os autores analisaram como o JIT se comporta no ambiente proposto, considerando o comportamento dos AGVs.

No trabalho foi definido um modelo de simulação, por meio do software ARENA, com algumas suposições iniciais para caracterizar o ambiente, porém, não foi considerada a quebra de máquina ou mesmo dos AGVs. Foi proposto também um arranjo físico composto de 8 máquinas com *buffers* na frente para servir tanto para entrada quanto para saída, sendo que foi determinada 5 tarefas a serem realizadas simultaneamente com um mix pré-determinado. Foi definido um modelo de roteamento de veículos levando em consideração as fichas do kanban presente em cada. Após essas definições, foram realizados vários testes verificando o fator de impacto de 4 variáveis, sendo elas: (I) número de veículos, (II) regra de despacho de veículos, (III) número de fichas do kanban e (IV) taxa de chegada de demanda. Após análise por métodos estatísticos, verificou-se por meio de 4 critérios de desempenho (tempo médio das tarefas no sistema, tamanho médio das filas de espera, taxa média de saída do sistema e o tempo da tarefa entre duas máquinas) mostrando que o (I) é extremamente importante para o desempenho do sistema segundo esses critérios; o (II) não afetou a performance avaliada; o (III) apresentou influencia apenas nos critérios de tempo médio das tarefas e tamanho médio das filas; já o (VI) influenciou apenas o tempo médio das filas e a taxa de saída do sistema.

Já Chiba, Arai e Ota (2010) basearam-se na afirmação de que os problemas relacionados ao ambiente de veículos auto guiados, mais especificamente os problemas de modelagem do caminho do AGV e do roteamento de AGVs estão ligados, sendo que a decisão em um impacta no outro. Para apresentar uma solução integrada, levando em consideração a modelagem do caminho e do problema de roteamento dos AGVs, foi apresentada uma abordagem coevolutiva cooperativa, que considera por meio de um algoritmo genético a evolução dos dois problemas



em conjunto. Foram aplicados quatro métodos em uma fábrica automotiva, sendo eles: (I) Cooperativo coevolutivo, (II) Algoritmo genético canônico (CGA – do inglês, *Canonical Algorithm Genetic*), (III) Apenas a modelagem do caminho e (IV) Apenas o roteamento. O primeiro desempenho medido foi o tempo de resposta sendo que o (IV) obteve o melhor resultado, porém quando levado em consideração os parâmetros de *fitness*, quantidade de veículos e a complexidade da rede criada, o (I) obteve o melhor resultado. Foi concluído que o método apresentado foi efetivo nos testes com dados reais.

Outro ponto interessante sobre as regras de despacho, que foi levantado na literatura em pesquisas recentes, é o fato de que as regras de despacho estão focadas para tomar uma decisão levando em consideração apenas informações naquele dado momento. Porém, em alguns em alguns trabalhos é apresentada uma melhora nesse quesito, abordando a tomada de decisão, antecipando possibilidades futuras de tarefas ou até mesmo de ações dos AGVs.

Como exemplo dessa antecipação, o trabalho proposto por Naso e Turchiano (2005) além de adotar uma estratégia de decisão multicritério por meio de um sistema *Fuzzy* hierárquico e realizar a sintonia fina das regras com a ajuda de um algoritmo genético em cada um dos níveis da hierarquia, foi criada uma heurística e incorporada ao sistema *Fuzzy*, na qual é realizado um encadeamento das tarefas, sendo que, a maior cadeia de tarefas é escolhida para diminuir o tempo de deslocamento sem cargas dos AGVs. Com essa diminuição, há um aumento na eficiência do transporte, pois existirá menos movimento sem aproveitá-lo para o transporte de materiais. Foi realizada uma comparação com outras técnicas sendo a abordagem tratada a que obteve melhores resultados mesmo em ambientes com distúrbios, como por exemplo, quebra de máquinas.

Já Hidehiko (2008) trabalha com o despacho de veículos em sistemas de manufatura autônomos, descentralizados e flexíveis (AD-FMS – *Autonomous Decentralized Flexible Manufacturing Systems*) por meio de simulação de hipóteses. Para definir a próxima tarefa do AGV, foi definido um algoritmo de raciocínio para a antecipação do futuro. Esse algoritmo prevê as possíveis próximas ações para o AGV e por meio de uma simulação das hipóteses e verificação de inconsistências, verifica a prioridade de cada uma delas definindo a próxima ação a ser tomada. Para aumentar a eficiência dessa verificação de hipóteses, foi proposta uma avaliação baseada em casos passados, possibilitando uma escolha baseada na experiência. Para a validação da proposta, foi realizada uma simulação em diferentes arranjos físicos de fábrica e verificado o desempenho em cada uma delas.

Smolick-Rocak *et al.* (2010) apresentaram um método dinâmico para o controle de múltiplos AGVs, visando o roteamento de cada um deles. O objetivo almejado pelos autores foi aumentar a eficiência da fábrica, em relação à taxa de produção, e também diminuir o número de *deadlocks*. Para tal feito, foi proposta por meio da técnica de janelas de tempo, a previsão do caminho a ser traçado por um AGV, e verificado se há a possibilidade de ocorrer *deadlocks* por essa trajetória. Para conseguir obter essa resposta, o arranjo físico da fábrica foi dividido em zonas, sendo que cada zona pode conter apenas um AGV. Para a validação, usou-se de *softwares* de simulação e após aprovado, foi implantado em seis fábricas.

Morandin *et al.* (2011) apresentaram um sistema *Fuzzy*-genético para realizar o despacho de veículos considerando múltiplas variáveis, e sendo uma delas em destaque, referente a previsão de tarefas o que permite o sistema realizar a predição e o encadeamento de atividades. As outras variáveis consideradas no trabalho foram divididas em duas categorias, sendo elas para o atendimento do *makespan* e para o atendimento do *tardiness*. Para o *makespan* foi considerada a distância, o número de nós, tamanho do buffer de entrada e de saída. Para o *tardiness* foi considerada: a distância, o número de nós, o encadeamento de tarefas e a data devida dos produtos. Para a validação, foi comparado em termos de *makespan* com o trabalho apresentado por Benicasa, Morandin e Kato (2003), apresentando um melhor resultado e o critério de desempenho *tardiness* foi avaliado realizando um tratamento estatístico o que mostrou pouco atraso das entregas na abordagem de produção por demanda.

### 2.3.2 Sistemas holônicos de manufatura

Em Fischer (1999), foi proposta uma nova abordagem para o projeto de uma arquitetura de uma manufatura integrada por computador. A proposta é baseada em uma estrutura hierárquica de planejamento e controle de um FMS e apresenta a arquitetura de agentes InteRRaP para a base do FMS projetado de acordo com a ideia de sistemas holônicos.

O autor apresentou pontos de ligação entre conceitos holônicos com os conceitos de agentes, afirmando que um meio de implementar os sistemas holônicos em sistemas computacionais é com sistemas multiagentes. A arquitetura proposta, é constituída de 5 camadas: (I) camada de planejamento e controle da produção; (II) controle do *shop floor*; (III) células flexíveis; (IV) sistemas autônomos; (V) máquinas. Sendo que as camadas (III), (IV), (V) foram projetadas sendo um sistema holônico (HS – *Holonic System*). Já para a camada (I) foi proposta a modelagem por meio da arquitetura de agentes InteRRaP. Essa proposta foi

implementada em 2 cenários distintos, sendo que os resultados obtidos no primeiro apresentaram a possibilidade de sistemas físicos por meio dessa arquitetura; já no segundo, foi demonstrado que a combinação do planejamento centralizado em conjunto com o descentralizado obteve bons resultados.

Liu *et al.* (2000) apresentaram um HMS para o controle distribuído de AGVs, com o intuito de superar algumas deficiências da arquitetura centralizada frente ao ambiente dinâmico da manufatura. Para isso estabelecerem que cada AGV é um *holon*, suprimindo assim a necessidade de um controlador, pois o processamento para atribuição de tarefa se dá no próprio AGV e em conjunto com os outros. Foi usada como base a arquitetura PROSA, adaptando os *holons* propostos nela para a realidade do AGV. Para a coordenação dos AGVs foi usado o protocolo de contrato de rede (*Contract Net Protocol*) no qual é criada uma oferta de serviço para todos os AGVs e a partir do processamento de cada um, é respondido ao elemento que fez a oferta com uma prioridade de cada AGV, com isso o elemento determina qual a qual AGV será atribuída a tarefa.

Para a validação foi simulado, com o auxílio de uma ferramenta de simulação 3-D, um ambiente de manufatura e aplicado a abordagem proposta e uma arquitetura centralizada e realizado o teste nas duas arquiteturas com diferentes quantidades de AGVs e alterando também suas velocidades. A partir da análise dos resultados, os autores identificaram um melhor desempenho da arquitetura holônica, assim como a maior flexibilidade e robustez do sistema.

Cheng, Chang e Wu (2004) propuseram uma abordagem sistemática para o desenvolvimento de sistema holônico de execução da manufatura (HMES – *holonic manufacturing execution system*) para a indústria de semicondutores. Foram abordados os conceitos de *holon* e holarquia para atribuir ao *holon* funcional do HMES as propriedades de inteligência, autonomia, cooperação, reconfigurabilidade e extensibilidade. Com a inclusão do conceito holônico na abordagem, os autores agregaram fundamentais características ao sistema de execução de manufatura (MES – *manufacturing execution system*) na indústria de semicondutores, que são: capacidade de recuperação diante de uma falha e certificação de segurança.

O processo de desenvolvimento do HMES se dá em quatro estágios: I) análise do sistema; II) projeto da holarquia; III) construção da aplicação; IV) integração e teste do sistema. Sendo estas etapas divididas em subetapas para garantir a identificação de funções genéricas para o desenvolvimento de um *holon* básico e suas especialidades. Foi realizada uma comparação

com outros tipos de MES, dentre eles *frameworks* MES e MES legados, avaliando diferentes características como: modularidade, interoperabilidade, recuperação de falhas, entre outros. Na comparação, o HMES se mostrou como uma boa opção para a nova geração de MES.

No trabalho proposto por Hsieh (2004), foi apresentado um framework para a modelagem e o controle de sistemas holônicos de manufatura (HMS – *Holonic Manufacturing Systems*) baseado na fusão de redes de Petri e sistemas multiagentes (MAS – *Multi Agent Systems*). Para modelar o controle descentralizado dos sistemas holônicos e possibilitar a capacidade de análise que as redes de Petri proporcionam, o autor apresentou uma rede de Petri colaborativa (CPN – *Colaborative Petri Net*). A CPN passa ao MAS toda a movimentação dos pedidos existentes e os que chegam de última hora, que por sua vez, por meio da técnica de protocolos de contrato de rede, realiza a interação entre os recursos, pedidos e produtos ligando aos *holons* modelados para representar o ambiente. Foi apresentada a análise do HMS por meio de teoremas fornecidos pela rede de Petri, favorecendo a simulação e a percepção do funcionamento do sistema.

Giret e Botti (2005) apresentaram um método multiagente para o projeto e a análise de um sistema de controle holônico de manufatura, baseando-se na arquitetura de referência PROSA e na INGENIAS (método de desenvolvimento de sistemas multiagentes complexos). A partir da observação da falta de um método para o apoio do desenvolvimento de sistemas de controle holônicos de manufatura, os autores propuseram um que apresenta diretrizes claras e não ambíguas para realizar o projeto e a análise. O método tem como base o INGENIAS, com uma extensão para o tratamento de modelos abstratos de agentes e para estabelecer propriedades para modelar o comportamento de tempo real. No artigo é apresentado como exemplo da proposta, a modelagem de uma fábrica de cerâmica, apresentando todos os passos desde a análise de requisitos até a construção da arquitetura do sistema.

Babiceanu e Chen (2006) publicaram um *survey* sobre HMS, apresentando os principais problemas no desenvolvimento e na aplicação dos HS na área de manufatura. Foi realizado um levantamento das primeiras aplicações dos HMS, apresentando a evolução das pesquisas nessa área. Após esse levantamento foram apresentadas as mais importantes abordagens sobre o tema e discutidas algumas linhas para pesquisas futuras, dentre elas: a interação entre HMS e MAS, aprofundando mais sobre as técnicas do MAS é possível obter melhoras no controle dos HMS; Na área de sistemas críticos também foi apontado um valioso uso dos HMS; também foi

levantada a questão da adaptação dos HMS em sistemas com distúrbios frequentes, e direcionado o foco para realização do controle desses ambientes.

No trabalho apresentado por Liu, Liu e Huang (2008) foram destacados alguns pontos dos sistemas holônicos de manufatura, sendo alguns como vantagens, estabilidades do sistema hierárquico e flexibilidade dinâmica do heterárquico, porém a complexidade deste sistema se torna alta se for levar em conta a interação entre esses *holons*, principalmente se a quantidade de *holons* for alta, pelo fato de que essa interação não é algo estático e sim dinâmico, pois a agregação dos *holons* se dá a partir da necessidade.

Para auxiliar a agregação dos *holons*, os autores propuseram o uso de técnicas de clusterização. Optaram pelo *Fuzzy C-means* para identificar os grupos pela característica da técnica de lidar com elementos não binários e sim definir graus de pertinência de cada elemento para cada grupo. Também propuseram o uso de um algoritmo genético para definir a quantidade de grupos iniciais para o algoritmo de clusterização. Para a validação foi apresentado um exemplo contendo diversos elementos como: matéria prima, equipamentos, estações de trabalho e demandas. A partir desses conjuntos de *holons*, e das demandas criadas, foi escolhido os grupos de *holons*.

Zhao, Zhang e Wang (2008) propuseram um método de alocação de tarefas e reconfiguração dinâmica em um HMS baseado em MAS. Para a negociação entre agentes e os *holons* foi proposta a abordagem de protocolo de contrato de rede de uma maneira dinâmica possibilitando essa interação em ambientes que estão em constante mudança. Foram propostos dois algoritmos, um para a alocação das tarefas e outro para a reconfiguração, sendo a modelagem de cada um deles proposta por meio de diagramas. A aplicação do método se mostrou muito eficiente após simulação de vários cenários.

Já, Iwamura *et al.* (2009) propuseram um método para programação em tempo real de um HMS. Foi modelado o sistema holônico por meio de 3 *holons* (*holons* de tarefa, *holons* de recurso e *holons* de coordenação). Foi proposto um aprendizado por meio de MAS para refinar a avaliação dos *holons* de tarefa e de recurso na hora de selecionar um candidato a ser produzido, essa avaliação é baseada no status atual da fábrica. Foram realizados alguns estudos de caso para garantir a efetividade da proposta, sendo apresentada a melhora por meio do aprendizado. Essa avaliação foi comprovada estatisticamente.

Abid e Labib (2011) apresentaram uma arquitetura holônica para sistemas de manufatura reconfiguráveis (RMS – *reconfigurable manufacturing systems*) capaz de adaptar-se diante de mudanças não previstas na demanda. O papel do RMS é produzir vários produtos agrupados em uma mesma família em um curto prazo de tempo e a baixo custo. Para tornar a arquitetura de controle dinâmica e robusta frente aos eventos não programados, os autores propuseram a criação de um sistema holônico para a abordagem. Neste HMS os *holons* representam as características para se estabelecer um RMS que são interligados a um modelo de uma rede analítica de processo (ANP – *analytical network process*) com uma abordagem multicritério para avaliar o desempenho do sistema. Foi criado um modelo genérico para garantir a flexibilidade para os *holons* e para facilitar a avaliação de desempenho do RMS considerando aspectos econômicos e operacionais como principais objetivos de desempenho.

Para a validação os autores apresentaram um estudo de caso em uma empresa fornecedora de módulos e componentes para o motor de veículos. Esta fábrica produz cerca de 2000 produtos similares para diferentes indústrias automobilísticas o que exige um alto desempenho na troca da configuração das máquinas para atender a demanda. Com a arquitetura proposta, foi possível obter um desempenho aceitável diante de eventos não programados, avaliando o impacto de variáveis críticas do sistema como tempo de reconfiguração e reuso.

Bal e Hashemipour (2011) apontam que o sucesso de uma linha de manufatura depende da efetividade da produção e do controle, e ainda levantam a falta de flexibilidade para lidar com interrupções e quebra de recursos, o que ocasiona uma drástica queda de desempenho. Para contornar esse problema, foi proposta uma arquitetura holônica de manufatura para controlar e programar uma linha de produção, sendo este HMS, baseado na arquitetura ADACOR, proposta por Leitão e Restivo (2006). Nesta arquitetura, foi criado o “*holo-flow*”, nome dado aos *holons* deste sistema. Foram definidos 4 tipos de *holons*, todos eles derivando características genéricas dos “*holo-flow*”, além de acrescentar o comportamento específico de cada.

Com a cooperação e a organização dos *holons*, os autores definiram uma hierarquia para realizar o controle e a programação da linha de produção. Para a validação da proposta, os autores modelaram uma fábrica real de peças de alumínio, usando um software de simulação e aplicaram 2 abordagens de controle, uma centralizada e a outra holônica. Depois de baterias de vários testes, simulando ambientes estáveis e instáveis, chegaram a conclusão do melhor desempenho do sistema holônico, em ambos ambientes, diante de 4 variáveis, sendo elas: *tardiness*, *lead-time*, rendimento (*throughput*) e pela utilização de recurso.

Jana, *et al.* (2013) propuseram um algoritmo de alocação dinâmica de tarefas para realizar a programação da produção baseando-se na manufatura multiagente holônica. A priorização do produto é realizada por meio de um sistema fuzzy multicritério que leva em conta vários aspectos da fábrica, dentro deles, geração de receitas, urgência, complexidade, entre outros. Para o sistema holônico foram propostos 3 *holons* (*Product Holon*, *Process Holon* e *Process Plan Holon*) e um sistema supervisor para coordenar as ações e para modelar a forma de operação foi usado a UML.

A fórmula de priorização foi dividida em duas etapas, a programação do produto e os protocolos de comunicação para a alocação da tarefa. Para cada tarefa, existe a contratação de um recurso específico, o que pode gerar conflitos com interesses de outras tarefas, para resolver essa situação, os autores usaram o MOORA (*Multi-Objective Optimization on the basis of Ratio Analysis*) sob um sistema *fuzzy* de tomada de decisão. Para o protocolo de comunicação foi usado a abordagem de contratação de recursos (CNP – *Contract Net Protocol*), usando tecnologias XML e J2EE para a realização de acordo com os padrões FIPA. O algoritmo proposto apresentou resultados otimizados diante de vários cenários, incluindo cenários com eventos não programados.

Rey *et al.* (2013) apresentam um estudo do impacto da tomada de decisão míope nos sistemas heterárquicos, que geralmente atingem ótimos locais ao invés de ótimos globais. Os autores identificaram esses comportamentos míopes, sob diferentes óticas e áreas, e propuseram mecanismos para controlar este comportamento e suavizar seu impacto. Para tal tarefa, foi proposto um framework holônico em conjunto com uma organização holônica que integra os mecanismos de controle temporais e da miopia social. Para o framework holônico foi proposto um *holon* genérico, baseado no proposto por Christensen (1994). Para a organização holônica, os autores se basearam nas diretrizes do PROSA, e propuseram 4 *holons*, sendo eles: *Order Holon* (OH), *Product Holon* (PH), *Adjunct Product Holon* (APH) e *Resource Holon* (RH). Sendo apenas o APH (*holon* orientado a tarefa) diferente dos do PROSA, sendo responsável por criar uma ponte entre os PH e os RH, trabalhando entre seus pares APH para definir uma sequência de alocação mais apropriada e reagindo a eventos internos disparados pelo monitoramento de algum recurso.

Para a validação, foi apresentado um estudo de caso simulado no sistema flexível de manufatura da AIP-PRIMECA *Center*. Foram realizados 3 experimentos, sendo todos eles para

provar a miopia temporal e social dos mecanismos de controle, confirmando a hipótese dos autores e apresentando uma abordagem de estruturação do problema.

Jovanovic, *et al.* (2014) aplicaram um controle holônico em um ambiente virtual simulando uma fábrica de pneus, para que, depois de estável e otimizado este controle, fosse aplicado em uma fábrica real. Eles descrevem a produção dos pneus em 5 etapas: I) mistura das matérias primas para a manufatura dos componentes dos pneus, II) preparação dos componentes, III) “manufatura de pneus verde” (ecologicamente correto), IV) vulcanização e V) desenvolvimento incremental, sendo que foi escolhido o foco na terceira etapa. Os autores afirmaram que a maioria dos sistemas de controle nessa área são centralizados, o que traz perdas em adaptação e rendimento diante do ambiente dinâmico apresentado por este ambiente. Foi criado um controlador holônico seguindo as diretrizes da estrutura de controle do ADACOR, modelados usando a UML e implementados utilizando a plataforma JADE (*Java Agent Development Environment*), depois foi realizado a comunicação com o ambiente virtual.

A estrutura de controle proposta é composta por 4 *holons*: *Product Holon*, *Planning Holon*, *Control Holon* e *Operational Holon*. Para a validação foi realizada uma simulação com o viés de otimização do tempo de processamento do processo de “pneus verdes” em um ambiente estável, e comparado com outras arquiteturas de controle. Também foi realizada uma análise em um ambiente instável, comparando o impacto da abordagem holônica com outras estruturas de controle convencionais. Os testes apresentados, apontam a abordagem holônica para a estrutura de controle como sendo uma boa alternativa, atingindo tanto a otimização em ambientes estáveis, quanto a adaptação diante do ambiente dinâmico.

Borangiu *et al* (2015) propuseram uma abordagem semi heterárquica para o controle da produção, mesclando planejamento e programação, em um sistema flexível de manufatura, sendo esta abordagem baseada em manufatura holônica e automação baseada em produto. A abordagem semi heterárquica muda de centralizada para descentralizada (e vice-versa) de acordo com a necessidade, visando otimização a curto e longo prazo (minimização de makespan e máxima utilização de recursos) e lidando com eventos não programados (quebra e recuperação de recursos). A arquitetura proposta é baseada no PROSA e no ADACOR, apresentando extensões como: I) troca automática de estratégia, realizada por um *Staff-Holon* multifuncional; II) orientação de serviço para a programação de operações e para alocação de recursos, por meio de agregação das informações de qualidade de serviço (QoS – *Quality of Service*) sobre os recursos, realizada por uma agregação de entidades (agentes) chamadas



RSAM (Resource Service Allocation Model), que é replicada em várias máquinas de trabalho para evitar pontos de falha. A arquitetura proposta foi organizada em duas camadas de decisões fracamente acopladas e uma camada de controle de recursos.

Para a implementação da arquitetura proposta, foi utilizado o framework JADE de acordo com os padrões FIPA. Para a validação, foi aplicada numa plataforma industrial piloto existente no *Robotics & CIM Laboratory* (Universidade politécnica de Bucareste), simulando 3 diferentes cenários. No primeiro deles foi realizada uma análise comparativa em 3 tipos de estratégias de controle na presença de falha de recursos produzindo bateladas de 200 pedidos. No segundo cenário, foi realizada uma comparação do tempo de processamento com uma abordagem usando CP (*Constraint Programming*), usando 100 diferentes pedidos de complexidade média. Por último, foi realizada uma comparação entre uma abordagem hierárquica com a abordagem semi heterárquica, diante de falha e recuperação de recursos. Em todos os cenários apresentados a abordagem proposta obteve bons resultados diante dos comparados.

Barbosa *et al.* (2015), apresentaram uma arquitetura multiagente holônica, baseada nos conceitos do ADACOR, focada no conceito de auto-organização. Os autores afirmam que apesar dos benefícios das abordagens holônicas, principalmente o PROSA e o ADACOR, eles não realizam uma auto-organização de fato (permitindo seus elementos se adaptarem diante de distúrbios). Apontam que essa “deficiência” é devida ao fato da alta complexidade de implementação e controle de grupo de entidades heterogêneas.

O trabalho apresentado é uma evolução do ADACOR, os autores apontam uma limitação em sua adaptação, permitindo apenas duas formas pré-configuradas. Diante disso, propuseram o ADACOR<sup>2</sup>, permitindo à arquitetura holônica descobrir novas configurações que melhor respondem as mudanças apresentadas no ambiente. Eles separaram a arquitetura em dois módulos, um arquitetural e outro comportamental, sendo o primeiro responsável pelas relações estruturais dos *holons* no sistema, e o segundo pela mudança dinâmica comportamental interna dos *holons*, e apresentaram por meio de um diagrama de “*Colleman's boat*” as inter-relações entre o nível macro e micro. Propuseram um módulo responsável por analisar o ambiente e propor mudanças ao comportamento dos *holons* (que possuem uma lista de comportamento), sendo que podem ser aprendidos novos comportamentos por meio de um módulo de aprendizado.

Foram realizados testes de simulação usando o sistema flexível de manufatura AIP-PRIMECA, comparando 4 abordagens de controle (hierárquica, heterárquica, ADACOR, ADACOR<sup>2</sup>) em dois cenários, uma estável e outro instável. No primeiro cenário, como esperado, a abordagem hierárquica apresentou um melhor resultado, já na segunda o ADACOR<sup>2</sup> foi o que se adaptou melhor, apresentando melhores resultados; levando a abordagem proposta a ser a melhor opção em um ambiente real.

## 2.4 Considerações finais

Neste capítulo foram apresentados os conceitos chave para o entendimento e desenvolvimento deste trabalho. Os artigos da literatura corroboram a escolha de usar regras multicritérios ao invés de um único critério. Outro ponto observado nos artigos levantados, foi a falta de vínculo entre as variáveis abordadas para a criação do despacho, gerando assim uma visão holônica, ou seja, global e não apenas local. É comum avaliarem apenas variáveis mais próximas do chão de fábrica, desconsiderando variáveis mais comuns em outros setores que possam ter um impacto significativo na escolha do despacho.

Conforme apresentado nas seções “2.2.1” e na “2.3.2”, os sistemas holônicos apresentam vantagens para a criação de arquiteturas de controle no ambiente de manufatura. Outro ponto levantado é o uso de sistemas multiagentes para a implementação dos HMS.

Para a construção da arquitetura holônica de manufatura, foram apresentadas diferentes abordagens, sendo que o PROSA é uma referência para a construção de uma arquitetura holônica, o ADACOR é uma arquitetura holônica baseada no PROSA que apresenta um tratamento específico do ambiente dinâmico da manufatura. Por sua vez o ASPECS que é um método para a construção de sistemas multiagentes holônicos não específico para a manufatura.

Outro ponto e de crucial importância para o trabalho é o fato de nenhum outro trabalho ter vinculado o detalhamento de um controle holônico para o despacho de AGVs.

# PROPOSTA DO TRABALHO

---

### 3.1 Considerações iniciais

Neste capítulo serão abordados os elementos que compõem a proposta e serão apresentados os detalhes de cada deles assim como suas justificativas de escolhas.

De acordo com a hipótese do trabalho, motivação e justificativa apresentados anteriormente, optou-se por: detalhar o despacho multicritério, utilizando um sistema *Fuzzy*; utilizar o HMS para modelar a arquitetura de controle; apoiar-se nos MAS para implementar a arquitetura proposta; sendo que todo o processo, desde o seu início até o fim, seguiu o método ASPECS tendo o framework JANUS para o desenvolvimento.

Nas seções 3.2.4 e 3.2.5 são apresentadas respectivamente as fases 1 e 2 do ASPECS, porém pela grande quantidade de modelos gerados por essas fases, optou-se por colocar uma amostra de cada modelo gerado (quando é gerado mais de um modelo por atividade) para a compreensão do uso de cada uma das atividades aplicada à proposta do trabalho, sendo que o modelo completo, assim como a explanação de todas as atividades de todas as fases será apresentado no apêndice A.

### **3.2 Proposta de uma arquitetura holônica para o controle do despacho multicritério de veículos**

Como apresentado anteriormente, as indústrias buscam por técnicas e métodos aplicados ao seu processo para se manterem competitivas e continuar no mercado e por conta das características de seu ambiente, o sistema se torna de difícil controle.

Dentro do AMS, o sistema de transporte possui um papel importante, pois ele é o responsável por transportar todo o material em processo de um ponto a outro, permitindo assim o fluxo da produção. Esse fluxo tem grande impacto no custo total do produto, que caso não realizado de uma maneira eficiente, pode causar grandes prejuízos para a indústria.

Uma opção usada para realizar o transporte dentro da fábrica são os AGVs. Dentro da área de AGVs, é possível encontrar vários estudos, um deles é sobre a atribuição de tarefas, responsável por definir a melhor tarefa a ser executada em um determinado momento.

Uma forma usada para determinar a melhor tarefa é por meio de regras de despacho, pois avaliam a configuração atual no momento em que a decisão se torna necessária. Porém é torna comum avaliar apenas variáveis do chão de fábrica, descartando variáveis pertencentes a outros níveis da fábrica que podem ter um impacto significativo dependendo da regra de negócio adotada pela fábrica.

Para realizar o despacho dos veículos é necessário definir uma arquitetura de controle, sendo que a opção pelo tipo de arquitetura pode causar grande impacto na sua execução.

Diante desse cenário, busca-se por uma arquitetura de controle capaz de executar o sistema de maneira que os possíveis impactos negativos causados pelo ambiente, não tornem a produção um processo inviável. Para isso, a proposta seguida no presente trabalho é:

*Detalhar uma arquitetura holônica de controle do despacho que usa a holonicidade para definir os critérios para o despacho, sendo eles provenientes de diferentes setores da fábrica.*

### 3.2.1 Descrição do ambiente do problema

O ambiente tratado neste trabalho é de uma fábrica, sendo composta por: máquinas (estações de trabalho), AGVs, rotas entre as máquinas e um *mix* de produtos que podem ser produzidos nessa fábrica.

Para a produção dos produtos identificados no *mix*, se tem mapeados os roteiros de produção (podendo ter mais de um possível, nesse caso identificados pelo nome de “roteiro alternativo de produção”). A partir dos pedidos realizados pelos clientes, é criado um plano para atender (produzir) os pedidos gerando a programação da produção (planejada para o horizonte de uma semana), definindo qual máquina será usada para processar qual produto em qual momento.

Para definir a programação da produção, é necessário seguir alguns critérios de otimização, neste caso são considerados o *makespan* e o *tardiness*. Tal programação pode ser alterada ao longo da semana para atender os eventos não planejados que porventura possam ocorrer (alta frequência de casos), ou para se adaptar melhor aos eventos do sistema de transporte.

Para o transporte das matérias primas e dos produtos (finalizados e em processamento) são usados os veículos autoguiados (AGVs). Estes veículos são despachados por meio de uma avaliação de atributos (distância entre a máquina e os AGVs, quantidade de cruzamentos na rota, quantidade de espaço livre no *buffer* de saída e data devida do produto) gerando as regras multivariáveis para que assim, no momento em que uma tarefa de transporte for requisitada, seja definido qual o melhor AGV para transportar o produto por uma rota também definida no exato momento.

O ambiente tratado neste trabalho, conforme já apresentado anteriormente, apresenta alta complexidade, tornando as escolhas para a construção de uma arquitetura de controle um trabalho não trivial.

A arquitetura de controle necessita ser flexível o suficiente para atender as demandas do mercado, inserindo novos pedidos no sistema de produção, de uma maneira que não cause um impacto muito negativo na produção dos pedidos que já estavam programados.

Outro ponto também almejado é conseguir atender variáveis de otimização de diferentes níveis organizacionais da fábrica, mesmo que elas sejam antagônicas entre elas (atender uma

pode impactar no não cumprimento da outra). Conforme apresentado anteriormente, atender variáveis de otimização de diferentes setores da fábrica, permite a definição de uma regra de despacho que possibilite um melhor atendimento do sistema de produção.

Para a arquitetura de controle, buscou-se uma opção que:

- Possibilite alcançar a otimização do sistema;
- Seja flexível para funcionar em um ambiente com eventos não planejados;
- Seja genérica o suficiente para permitir a entrada de novos elementos ao sistema;
- Possibilite a adição ou melhoria de novos módulos de controle sem que tenha alterações significativas na arquitetura.

Levando em consideração as características desejadas, citadas anteriormente, se torna necessário definir a forma com que a arquitetura proposta irá interagir com o ambiente de produção. Foram levantadas 4 possíveis formas de funcionamento, sendo elas: I) Ter a programação da produção e fazer com que o despacho de veículos tente ao máximo cumprir essa programação; II) Ter a programação da produção e dar maior autonomia ao despacho, permitindo assim a alteração da programação caso o despacho julgue necessário; III) Não ter uma programação da produção já definida e realizar o “despacho de produtos” e um despacho de veículos sem nenhum acoplamento entre eles; IV) Não ter uma programação da produção já definida e realizar o “despacho de produtos” e um despacho de veículos com acoplamento (entende-se por acoplamento a dependência do processo).

Esta decisão é essencial para a descrição do domínio do problema, pois interfere diretamente na arquitetura escolhida. Optou-se pela II (Ter a programação da produção e dar maior autonomia ao despacho, permitindo assim a alteração da programação caso o despacho julgue necessário); por se julgar um cenário comum dentro de uma indústria: seguir uma programação da produção definida, e pelo fato de que o despacho de veículos realiza a escolha do produto a ser transportado (alterando a ordem da produção) com base na configuração atual do momento em que a decisão se torna necessária.

Após a definição pela abordagem II, foi necessário definir a arquitetura de controle, se ela seguiria uma abordagem tradicional (centralizada ou descentralizada) ou seguiria uma nova abordagem. Após o estudo das possibilidades, identificou-se que a abordagem holônica apresenta uma melhor aderência ao cenário da manufatura descrito anteriormente.

Com a arquitetura holônica é possível atender as características fortes da arquitetura centralizada (otimização global) e da descentralizada (flexibilidade diante do ambiente dinâmico e mais tolerante a falhas), além de permitir uma melhor adaptação caso necessite a inserção de novos elementos de controle na arquitetura por apresentar a característica básica da dualidade do *holon*.

### 3.2.2 Proposta para o despacho multicritério

Para o despacho de veículos, optou-se por realizar um despacho multicritério ao invés de um mono critério, pois como apontado anteriormente pelos artigos correlatos, o desempenho do despacho depende da configuração do ambiente em que ele está sendo executado, ou seja, para um contexto específico a regra pode ser adequada, mas para outros ela pode não ser a melhor opção. Para tentar minimizar essa variação, o despacho pode ser usado, abordando mais de um critério e aumentando a cobertura para os contextos tratados.

Outro ponto importante para agregar na escolha do despacho é definir quais critérios de otimização serão atendidos, e que também aborde critérios referentes a diferentes níveis organizacionais da fábrica, não só ao chão de fábrica. Para este trabalho optou-se em contemplar o *makespan* e o *tardiness*. Sendo o primeiro responsável por indicar o tempo total de produção (comumente tratado pelas estratégias do chão de fábrica) e o segundo a quantidade de produtos que extrapolam a data de entrega (comumente tratado pelas estratégias de negócio).

A opção destes dois critérios de otimização se deu ao fato de serem amplamente usadas na literatura e ao mesmo tempo serem antagônicas (atender uma pode prejudicar a qualidade da outra) e por basear-se no trabalho de Morandin *et al.* (2011). Ao atender o menor tempo de produção, ou seja, encaixar os produtos a serem produzidos da melhor forma possível, procurando minimizar o tempo de produção e ter uma melhor utilização dos recursos, geralmente não é verificado se a data de entrega do produto vai ser atendida em 100% dos casos. Já o contrário, caso queira atender os prazos estabelecidos pelo setor comercial, é comum não se preocupar em otimização dos tempos de produção e com o melhor uso dos recursos.

Para realizar o despacho multicritério, será usado um sistema *fuzzy* similar ao proposto por Morandin *et al.* (2011), pois analisando as características do problema em questão e as características da técnica de sistemas *fuzzy*, percebeu-se uma alta aderência, além de várias

indicações presentes na literatura reforçando a escolha por esta técnica (MORANDIN *et al*, 2011; MORANDIN *et al*, 2007; NASO e TURCHIANO, 2005; BENICASA, 2003, *entre outros*).

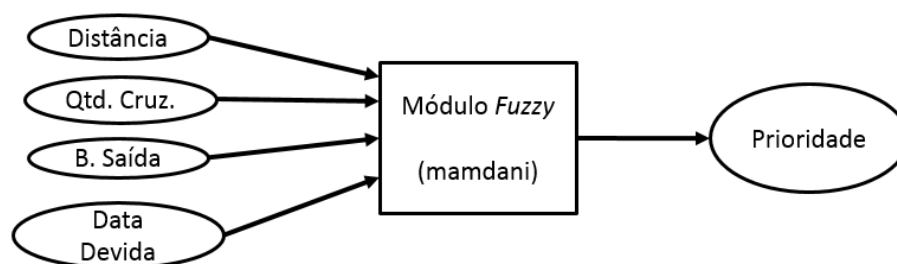
Algumas vantagens que o uso de *fuzzy* poderá proporcionar à solução é o fato de que as regras de despacho podem ser diretamente traduzidas em regras *fuzzy*, pela semelhança entre elas. Também permite uma fácil avaliação de múltiplas regras com o auxílio da inferência *fuzzy*, além de permitir avaliar a melhor condição por meio de vários critérios.

Outro ponto em que a técnica irá apoiar é na dificuldade em mapear e ponderar a relação entre as regras de despacho, pois nem sempre é possível conhecer toda a influência de uma variável ao sistema. Com o sistema *fuzzy* é possível realizar cálculos por meio da inferência sem ser necessário explicitar o comportamento e a influência de cada regra.

Para a elaboração das regras de despacho, é comum usar termos incertos e termos linguísticos, que podem ser modelados facilmente pelo sistema *fuzzy*, devido à aproximação ao raciocínio humano e da linguagem usada pelos operadores e especialistas da área.

Outra característica que será inserida no sistema *fuzzy* para ponderar as regras de despacho é uma sintonia do sistema *fuzzy* em relação à saída da produção e as regras de negócios adotadas pela fábrica, em outras palavras, o sistema *fuzzy* irá ponderar as variáveis de entrada de acordo com o momento, dando mais peso para uma entrada beneficiando assim uma regra específica ao invés de outra.

O módulo *fuzzy* (Figura 3.1) foi projetado usando o MATLAB<sup>9</sup>, tendo como variáveis de entrada: distância, quantidade de cruzamentos, data devida e buffer de saída. Estas variáveis de entrada foram escolhidas para atender dois critérios de otimização específicos, sendo eles, *tardiness* e *makespan*, conforme justificado anteriormente. A variável de saída escolhida para o sistema foi a prioridade e para o motor de inferência o Mamdani.



<sup>9</sup> MATLAB: <http://www.mathworks.com/products/matlab>

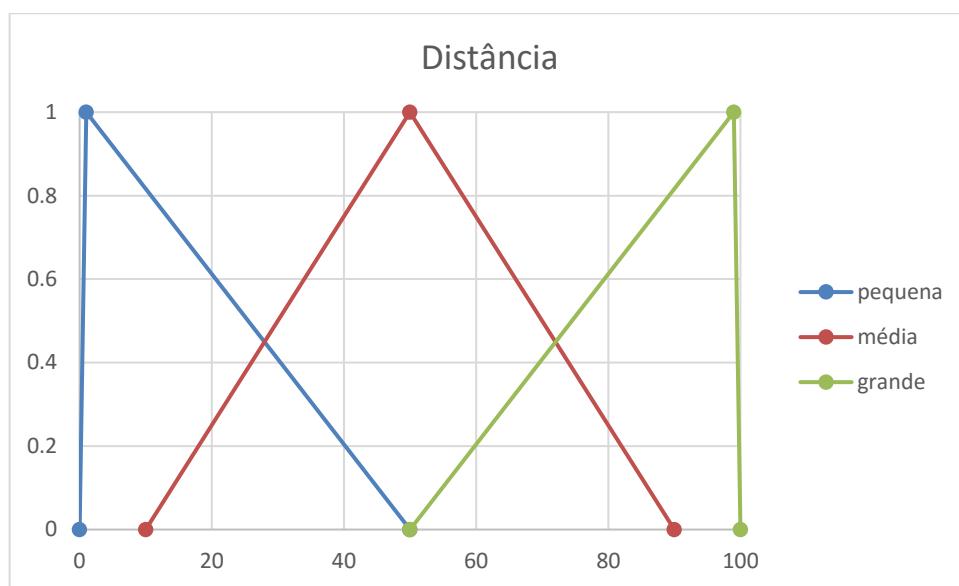


**Figura 3.1. Sistema *fuzzy* utilizado para o despacho de veículos**

Cada uma das variáveis (entrada e saída) foi modelada de maneira arbitrária conforme apresentado a seguir:

- Distância (Figura 3.2)

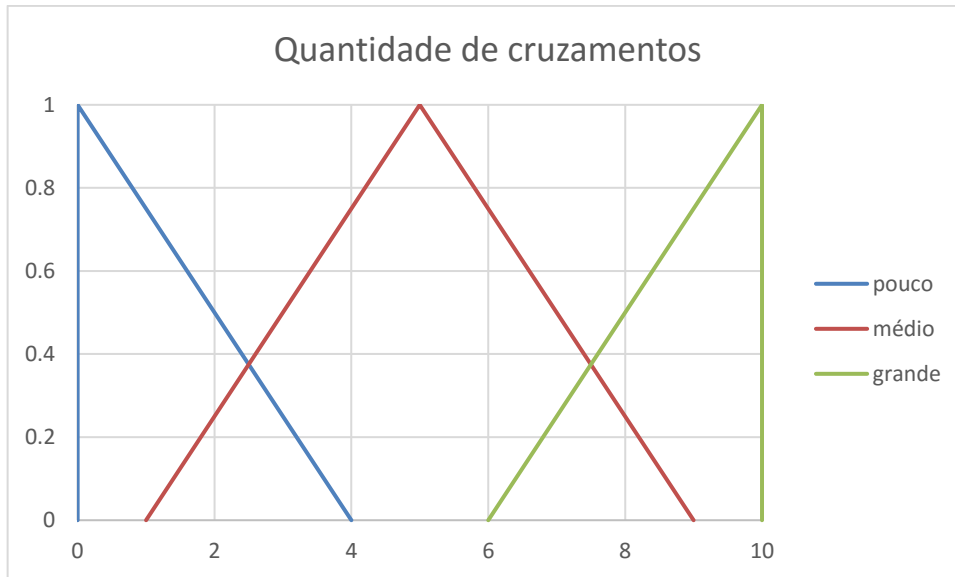
A distância está ligada ao *makespan* e ao *tardiness* e indica a distância que o AGV irá percorrer até a máquina que o solicitou, impactando diretamente no tempo de produção e consequentemente no tempo em que o produto ficará pronto para a entrega.

**Figura 3.2. Variável *fuzzy* de entrada: Distância**

A variável foi modelada por meio de 3 conjuntos *fuzzy* triangulares; com variáveis linguísticas: pequena (*p*), média (*m*) e grande (*g*); e o domínio de 0 a 100. Sendo que o *p* tem como pontos [0, 1, 50], o *m* [10, 50, 90] e o *g* [50, 99, 100].

- Quantidade de cruzamentos (Figura 3.3)

A quantidade de cruzamentos está relacionada ao *makespan* e ao *tardiness* e indica a quantidade de cruzamentos que existem no caminho que o AGV optou para transportar o produto, impactando diretamente no tempo em que a peça ficará pronta.

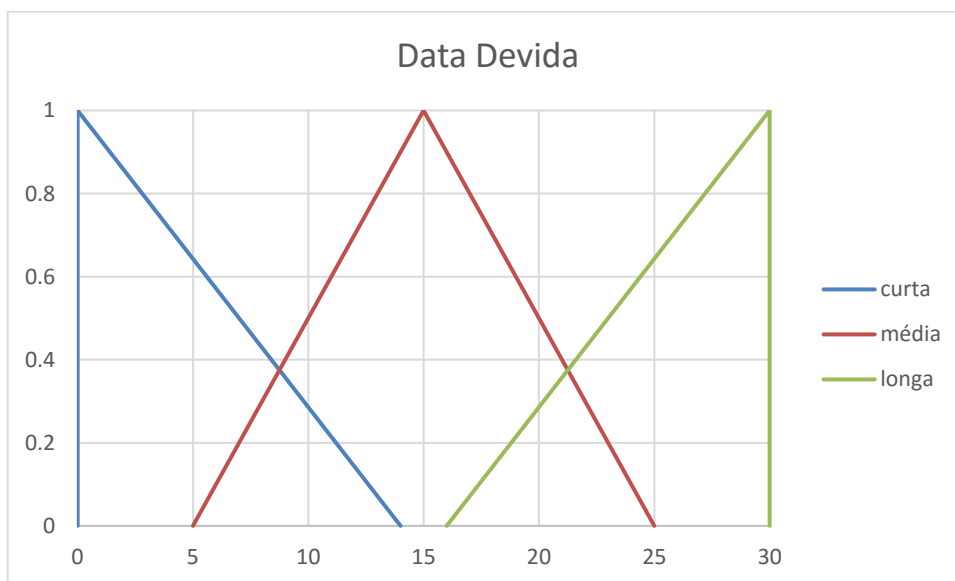


**Figura 3.3. Variável Fuzzy de entrada: Quantidade de cruzamentos**

A variável foi modelada por meio de 3 conjuntos *fuzzy* triangulares; com variáveis linguísticas: pouco (**p**), médio (**m**) e grande (**g**); e o domínio de 0 a 10. Sendo que o **p** tem como pontos [0, 0, 4], o **m** [1, 5, 9] e o **g** [6, 10, 10].

- Data devida (Figura 3.4)

A data devida é uma variável diretamente ligada ao *tardiness*, indicando a quantidade de dias restante para que o produto em questão seja entregue ao cliente.

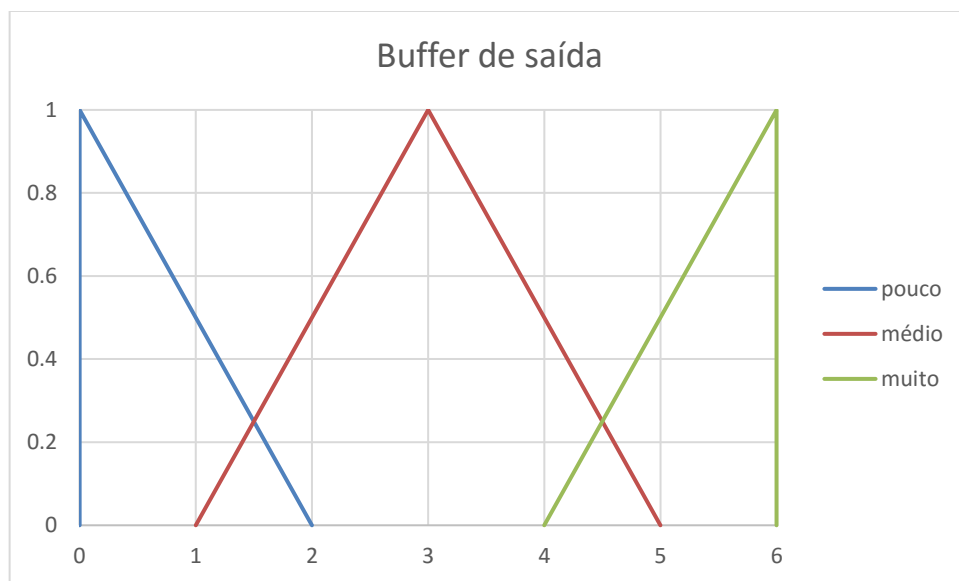


**Figura 3.4. Variável fuzzy de entrada: data devida**

A variável foi modelada por meio de 3 conjuntos *fuzzy* triangulares; com variáveis linguísticas: curta (C), média (m) e longa (L); e o domínio de 0 a 30. Sendo que o C tem como pontos [0, 0, 14], o m [5, 15, 25] e o L [16, 30, 30].

- Buffer de saída (Figura 3.5)

O buffer de saída está relacionado tanto ao *makespan* quanto ao *tardiness* e indica a quantidade de espaço restante no buffer de saída da máquina que requisitou a tarefa, tendo a importância de prevenir que máquinas fiquem com o buffer de saída cheio e parem o processamento.

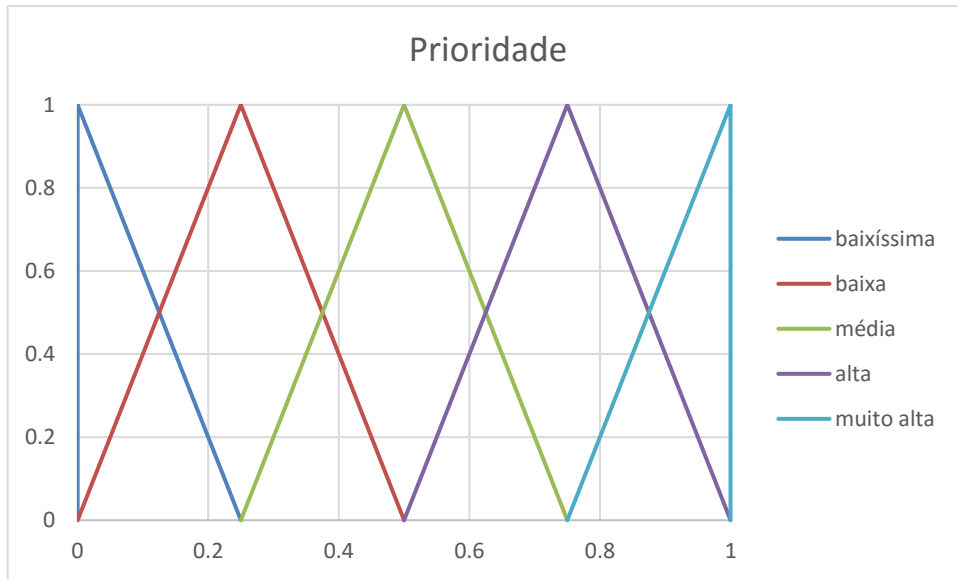


**Figura 3.5. Variável *fuzzy* de entrada: Buffer de saída**

A variável foi modelada por meio de 3 conjuntos *fuzzy* triangulares; com variáveis linguísticas: pouco (p), médio (me) e muito (mu); e domínio de 0 a 6. Sendo que o p tem como pontos [0, 0, 2], o me [1, 3, 5] e o mu [4, 6, 6].

- Prioridade (Figura 3.6):

A prioridade é usada para definir a ordem dos AGVs para cumprir tarefas; O AGV com maior prioridade será designado para atender a tarefa.



**Figura 3.6. Variável *fuzzy* de saída: Prioridade.**

A variável foi modelada por meio de 5 conjuntos *fuzzy* triangulares; com variáveis linguísticas: baixíssima (**bb**), baixa (**b**), média (**m**), alta (**a**) e muito alta (**aa**); e domínio de 0 a 1. Sendo que o **bb** tem como pontos [0, 0, 0.25], o **b** [0.25, 0.5, 0.75], o **m** [0.25, 0.5, 0.75], o **a** [0.5, 0.75, 1] e o **aa** [0.75, 1, 1].

Para a construção da base de regras, não se aplicou nenhuma técnica de otimização de geração de base, sendo assim, a base de regras foi criada de maneira arbitrária e é composta por 12 regras, conforme apresentado na Tabela 3.1

Tabela 3.1. Base de regras do sistema fuzzy

| Regras    |                  |             |                 |            |
|-----------|------------------|-------------|-----------------|------------|
| Distância | Qtd. Cruzamentos | Data devida | Buffer de saída | Prioridade |
| p         | p                | c           | p               | aa         |
| m         | m                | c           | me              | aa         |
| g         | g                | c           | mu              | a          |
| p         | p                | m           | p               | a          |
| m         | m                | m           | me              | m          |
| g         | g                | m           | mu              | m          |
| p         | p                | l           | p               | m          |
| m         | m                | l           | me              | b          |
| g         | g                | l           | mu              | bb         |
| p         | p                | m           | me              | a          |
| g         | g                | m           | me              | a          |
| p         | p                | l           | me              | a          |

Foi elaborada uma forma de que o sistema fuzzy se adaptasse ao ambiente e reagisse conforme a produção. Para isso, a saída da produção é analisada e caso tenha produtos sendo produzidos com muita antecedência, é imposto uma penalidade à variável “data devida” (vinculada diretamente ao *tardiness*) para que o sistema reforce o cálculo para o *makespan*. Caso contrário, ou seja, com muitos produtos atrasados, é imposta uma penalidade nas variáveis que estão vinculadas ao *makespan* (distância, quantidade de cruzamentos e buffer de saída), reforçando assim o cálculo para o *tardiness*.

Esse ajuste dinâmico automático leva em consideração as estratégias de negócio da empresa, assim como as regras *fuzzy* geradas, partições *fuzzy* e as variáveis de entrada. Para este caso, a estratégia de negócio é não permitir uma quantidade muito alta de produtos atrasados aos clientes, mas quando os produtos são finalizados com muita antecedência, é desejável otimizar os recursos utilizados, melhorando assim o ritmo da produção.

A regra de auto ajuste segue o modelo a seguir:

*Caso 50% dos produtos atrasem*

→ (*distância x 0.5; qtd. cruz x 0.5; b. saida x 0.5; data. devida x 1*)

*Caso 50% dos produtos antecipem*

→ (*distância x 1; qtd. cruz x 1; b. saida x 1; data. devida x 0.5*)

*Caso contrário*

→ (*distância x 1; qtd. cruz x 1; b. saida x 1; data. devida x 1*)

Estas regras ainda devem atender os valores do domínio para cada uma das variáveis, caso o fator multiplicativo extrapole o valor do domínio (tanto para cima quanto para baixo), considerará que o valor está saturado, ou seja, ficará com o valor máximo ou mínimo do domínio.

Vale ressaltar que as variáveis de otimização consideradas, assim como as estratégias de auto ajuste podem variar dependendo das regras de negócio da empresa.

### 3.2.3 Proposta para o controle holônico de despacho

Para realizar o controle holônico é necessário definir a arquitetura de controle, sendo que a opção pela arquitetura irá influenciar no rendimento e no funcionamento do sistema. Como estamos lidando com um ambiente dinâmico, a flexibilidade da arquitetura é fundamental para diminuir o impacto negativo no sistema causado pelas constantes mudanças do ambiente. Apesar da importância da flexibilidade, também é importante conseguir buscar otimizações para aproveitar ao máximo o sistema, visto que um sistema pouco produtivo pode ter um impacto muito negativo na empresa.

Tendo em vista essas características, uma opção viável é a arquitetura holônica. Pelas características desta arquitetura, é possível adicionar novos componentes (ou até mesmo detalhar melhor um componente já existente, tornando-o mais eficiente) ao mesmo tempo em que possibilita a otimização global por meio de sua holarquia.

Apesar de essa arquitetura ser bem difundida na literatura, nenhuma delas trata de maneira detalhada o despacho de veículos por meio de regras multicritério. Para criar uma arquitetura holônica específica para o despacho de veículos por regras multicritério, foram levantadas as possíveis abordagens para o projeto da arquitetura.

A primeira referência estudada foi o PROSA, que é uma importante referência para a criação de arquiteturas holônicas de manufatura, apresentando diretrizes para a criação da mesma. A segunda, o ADACOR, que é baseado no PROSA, apresenta uma arquitetura focando na adaptação em ambientes dinâmicos. O ADACOR apresenta algumas modificações em relação ao PROSA, apresentando uma arquitetura já definida, e não apenas referências como no PROSA, e algumas modificações em relação aos *holons* criados, sendo estas orientadas a

característica de melhor adaptação ao ambiente dinâmico. Apesar da importância de ambas arquiteturas holônicas de manufatura, nenhuma delas apresenta um método a ser seguido para a construção do mesmo.

Pelo fato de não possuírem um método bem definido, o que dificulta a adaptação para a criação da arquitetura de controle para o despacho multicritério de veículos, buscou-se por métodos que seguissem os conceitos holônicos e foram avaliados 2, sendo eles: ANEMONA e ASPECS.

A ANEMONA, proposto por Botti e Giret (2008), é um método que segue as diretrizes do PROSA e da FIPA para a construção de sistemas holônicos de manufatura, abrangendo desde a análise até a codificação e baseia-se no framework JADE<sup>10</sup> para a criação do sistema multiagente.

Já o ASPECS, proposto por Cossentino, *et al.* (2010a) é um método para o desenvolvimento de sistemas holônicos que abrangem desde a análise até a codificação e implantação. Baseado no método CRIO<sup>11</sup>, permite criar um modelo organizacional, identificando papéis, organizações e capacidades para a execução do sistema. Um ponto forte do ASPECS é que para a codificação, os autores criaram um framework de desenvolvimento que permite uma fácil codificação do modelo gerado no ASPECS, dando todo o suporte para a criação dos *holons*.

Um ponto negativo do ASPECS é o fato de ser um método genérico, o que leva a necessidade de adaptação de alguns de seus conceitos e palavras chaves para a realidade do domínio da manufatura. Alguns de seus conceitos foram adaptados para este trabalho, como no caso do conceito do termo “papel”, que se entende como sendo o título dado ao comportamento almejado, e não o comportamento em si, porém para não interferir no método e mudar o escopo do projeto, optou-se por manter os termos originais e adaptar o significado de cada para o trabalho em questão (conforme apresentado na seção 2.2.3.1).

A tabela a seguir (Tabela 3.2) apresenta um comparativo entre o ASPECS e o ANEMONA sob algumas óticas.

---

<sup>10</sup> <http://jade.tilab.com/>

<sup>11</sup> Método baseada em capacidades – papéis – interações – organizações (CRIO – Capacity, Role, Interaction e Organization).

Tabela 3.2. Comparação ASPECS e ANEMONA (adaptada de Cossentino et al. 2010a)

|                              | ANEMONA                                 | ASPECS   |
|------------------------------|---|--|
| Ciclo de vida                | Processo unificado (PU)                 | Iterativo e incremental                              |
| Cobertura do ciclo de vida   | Análise, projeto e implementação        | Análise, projeto, implementação e <b>implantação</b> |
| Domínio de aplicação         | <b>Sistemas holônicos de manufatura</b> | Independente   |
| Integração com padronizações | PU, SPEM, FIPA, UML                     | SPEM, UML, FIPA                                      |
| Plataformas e bibliotecas    | JADE e PROSA                            | <b>JANUS</b>   |
| Sistema Aberto               | Não                                     | <b>Sim</b>   |
| Estrutura dinâmica           | Não                                     | <b>Sim</b>   |

Os elementos em vermelho da tabela apresentam características desejadas para o trabalho proposto, apontando uma vantagem para o ASPECS em relação à ANEMONA. Em relação ao ciclo de vida, nenhuma das características apresentam uma vantagem direta; Em relação a cobertura do ciclo de vida, o fato do ASPECS detalhar a implantação, permitindo a identificação dos locais dos *holons*, relação com os dispositivos externos, entre outros; Em relação ao domínio de aplicação, a ANEMONA apresenta uma vantagem por ser específico para a área de manufatura enquanto o ASPECS é genérico; Em relação à integração com as padronizações também não há diferença significativa; Em relação as plataformas e bibliotecas, enquanto o ASPECS possui um framework próprio que dá suporte para a criação dos *holons*, a ANEMONA segue as diretrizes do PROSA e do JADE para a codificação, o que gera a necessidade da criação da estrutura do *holon* (recursão de agentes). Em relação ao sistema aberto, o ASPECS permite a adição e remoção de agentes e organizações, enquanto isso não é possível na ANEMONA; em relação a estrutura dinâmica, o ASPECS permite uma auto organização, reconfiguração dinâmica do sistema, enquanto a ANEMONA não permite.

A escolha pelo uso do ASPECS se deu principalmente por dois motivos:

I) Dificuldade em implementar uma arquitetura holônica usando o PROSA ou o ADACOR, por não apresentarem um método detalhado.

II) Por ter um framework de desenvolvimento de HMAS criado exclusivamente para o método, facilitando a criação dos *holons*.

O ASPECS apresenta passos bem definidos para a criação da arquitetura, atendendo todos os conceitos de sistemas holônicos. Vale ressaltar que apesar de não ser um método exclusivo para sistemas de manufatura, ele atende todos os seus conceitos, permitindo assim sua aplicação a este ambiente específico.



### 3.2.4 Modelagem do ambiente de controle do despacho multicritério

Apoiado no método ASPECS, a primeira fase é a modelagem do problema. Essa fase é composta por 7 atividades, sendo que em cada atividade é gerado um ou mais artefatos UML para modelar o problema (conforme já apresentado em detalhes na subseção 2.2.3.1).

Na primeira atividade, “descrição dos requisitos do domínio” – DRD, foi modelado o ambiente proposto na subseção 0 por meio de casos de uso (Figura 3.7) e um diagrama de caso de uso, conforme apresentado na Figura 3.8.

- **Identificador:** RF01.
- **Nome:** Despachar produto.
- **Tipo:** Abstrato
- **Descrição:** É selecionada a máquina de destino para processar o produto de acordo com a programação da produção (caso não seja possível é verificado um roteiro alternativo); Em seguida é selecionado um AGV para transportar este produto até a máquina definida, seguindo uma rota também definida neste momento.
- **Atores:** A01, A02.
- **Pré-condições:** Chegada de um produto inacabado no buffer de saída ou a existência de um pedido de produção de um produto.
- **Pós-condições:** Finalizada uma etapa do processo de produção de um produto; Produto pronto para ser transportado.
- **Fluxo normal:**
  - 1. Verificar produto a ser produzido e etapa de produção atual (RF02);
  - 2. Verificar a máquina planejada na programação da produção (RF03.1);
    - 2.a. Caso haja máquina, designá-la como destino;
    - 2.b. Caso contrário esperar até que haja máquina disponível.
  - 3. Selecionar AGV (RF04.1);
  - 4. Selecionar a rota utilizada (RF5);
  - 5. Transportar o produto (RF06).
- **Variações/Fluxo alternativo:** Inexistente.
- **Não funcional:** inexistente.
- **Autor e Datas:** Flavio Montoro – 12/12/2014.

Figura 3.7. Fase 1, atividade 1: descrição dos requisitos do domínio – caso de uso.

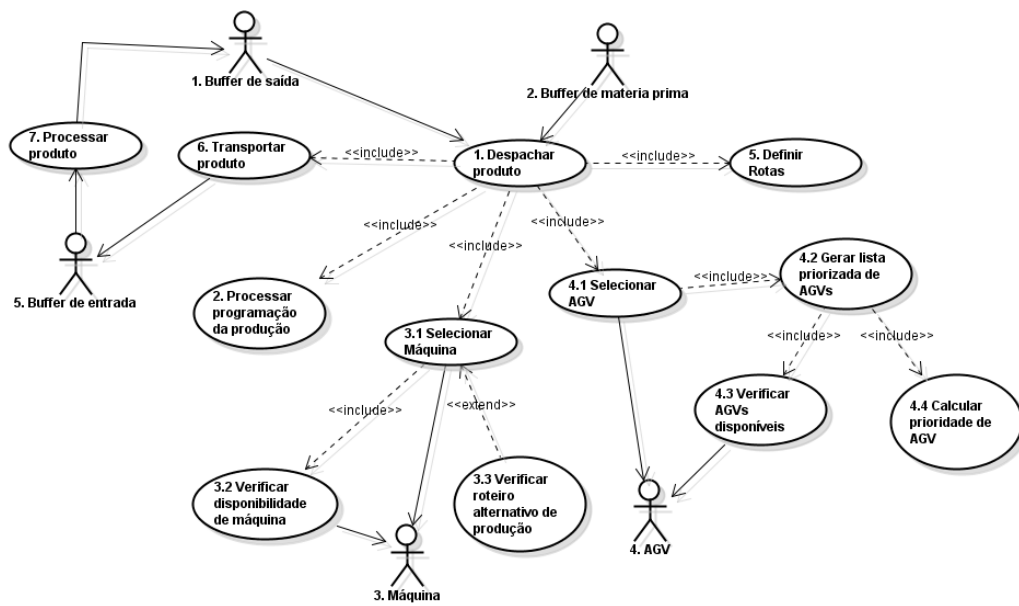


Figura 3.8. Fase 1, atividade 1: descrição dos requisitos do domínio – diagrama de caso de uso.

O diagrama de caso de uso (Figura 3.8) é composto por 12 casos de uso e 5 atores, sendo que todos eles possuem um documento de caso de uso, similar ao apresentado na Figura 3.7. (todos os casos de uso estão apresentados no apêndice A).

No diagrama de caso de uso, o caso de uso “1. Despachar produto” é ativado pelos atores “buffer de matéria prima” ou “buffer de saída”, em seguida o caso de uso “2. Processar programação da produção” é ativado, identificando as tarefas a serem realizadas para a produção do produto. Após a identificação da tarefa, é selecionado a máquina (3.1 Selecionar máquina) caso ela esteja disponível (3.2 Verificar disponibilidade de máquina), caso contrário procura-se por uma outra máquina disponível, desde que esse produto tenha um roteiro alternativo (3.3 Verificar roteiro alternativo de produção).

Com a máquina já selecionada, se torna necessário definir o AGV que irá transportar o produto (4.1 Selecionar AGV), para isso, é necessário identificar quais AGVs estão disponíveis (4.3 Verificar AGVs disponíveis), calcular a prioridade de cada um deles (4.4 Calcular prioridade de AGV) e gerar uma lista priorizada dos AGVs (4.2 Gerar lista priorizada de AGVs).

Após o AGV selecionado (o de maior prioridade), escolhe-se a rota que ele irá usar. A partir daí o produto é transportado (6. Transportar produto) para o “buffer de entrada da máquina” que por sua vez é processado e enviado ao “buffer de saída”.

Na segunda atividade, “descrição da ontologia do problema” – POD, foi criada a ontologia que descreve o cenário tratado (Figura 3.9).

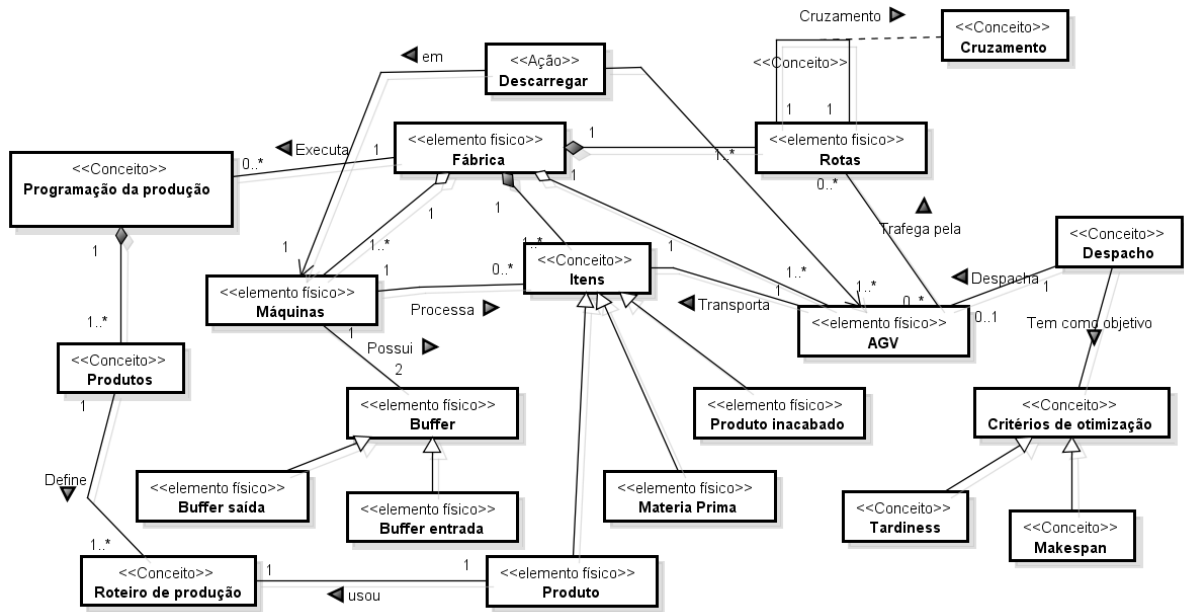


Figura 3.9. Fase 1, atividade 2: descrição da ontologia do problema

A ontologia apresentada na Figura 3.9 é composta por 3 tipos de elementos: elemento físico, conceito e ação. O primeiro refere-se a um elemento físico existente na fábrica; o segundo é referente a um elemento lógico tratado na fábrica; o terceiro representa uma ação realizada entre os elementos (lógicos ou físicos) existentes no ambiente.

A ontologia é representada pelo elemento físico “fábrica”, que é composto pelos elementos físicos: “rotas”, “AGV”, “itens” e “máquinas” e executa o conceito “programação da produção”. O elemento físico “rotas”, possui uma relação com outras rotas, gerando o conceito de “cruzamento”. O elemento físico “AGV” possui um relacionamento “trafega pela” com “rotas”, um relacionamento “transporta” com o elemento físico “itens” e um relacionamento “despacha” com o conceito “despacho”, que por sua vez “tem como objetivo” “critérios de otimização” que podem ser do tipo “tardiness” ou “makespan”.

O elemento físico “itens” pode ser do tipo “produto inacabado”, “matéria prima” e “produto”, todos do tipo elemento físico. O elemento físico “produto” possui uma relação “usou” com o conceito “roteiro de produção” que por sua vez possui um relacionamento “define” com o conceito “produto”. Nota-se que existem dois elementos “produto” um relacionado ao elemento físico (referente ao produto que foi produzido) e outro relacionado ao

conceito (conhecimento de como produzir o produto). O conjunto do conceito produto compõe o conceito “programação da produção”.

O elemento físico “máquina” é composto pelo elemento físico “buffer” que pode ser do tipo “buffer de saída” ou “buffer de entrada” e o elemento físico “AGV” realiza uma ação de “descarregar” no elemento físico “máquina”.

Na terceira atividade, “Identificação das organizações” – OID, agrupa-se por similaridade dos requisitos apresentados no DRD, e a partir desses grupos é gerada uma organização, que tem como comportamento global o atendimento do conjunto de requisitos. Esta tarefa está modelada na Figura 3.10.

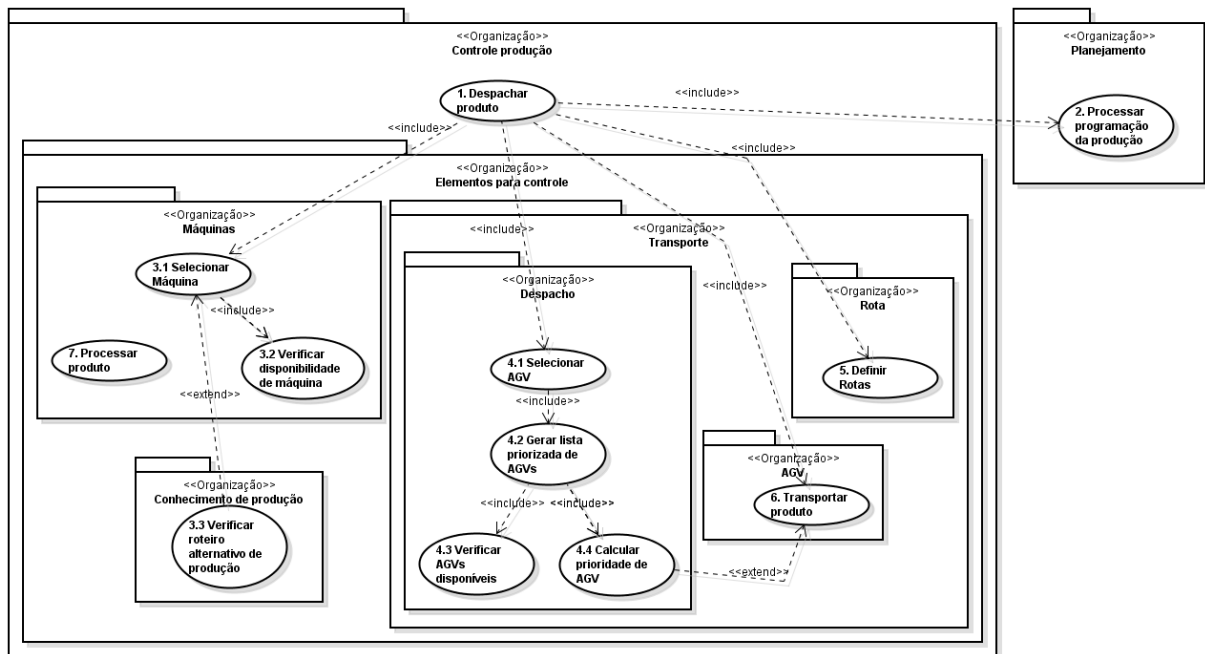


Figura 3.10. Fase 1, atividade 3: Identificação das organizações.

Foram identificadas 9 organizações, sendo elas: Controle produção, Planejamento, Elementos para controle, Máquinas, Conhecimento de produção, Transporte, Despacho, AGV e Rota.

A organização Despacho ficou responsável por representar o comportamento dos casos de uso: 4.1 Selecionar AGV, 4.2 Gerar lista priorizada de AGVs, 4.3 Verificar AGVs disponíveis e 4.4 Calcular prioridade de AGV. A organização AGV ficou responsável pelo comportamento do requisito: 6. Transportar produto. A organização rota ficou responsável pelo requisito 5. Definir rotas. Essas três organizações (Rota, AGV e Despacho) compõem a organização Transporte.

A organização elementos para controle é composta pelas organizações: Transporte, Máquinas e Conhecimento de produção. A organização Máquinas é responsável pelos casos de uso: 3.1 Selecionar máquina, 3.2 Verificar disponibilidade de máquina e 7. Processar produto. Já a organização Conhecimento de produção é responsável pelo caso de uso 3.3 verificar roteiro alternativo de produção.

A organização Controle produção além de englobar a organização Elementos para controle, também é responsável pelo caso de uso 1. Despachar produto. A organização Planejamento ficou responsável pelo caso de uso 2. Processar programação da produção.

Na quarta atividade, “Identificação dos papéis e interações” – IRI, são criados papéis que em conjunto com suas interações, representam o comportamento global (ou parte dele) da organização ao qual ele pertence. Para isso, foi gerado o artefato representado na Figura 3.11, sendo os papéis representados pelos elementos em cinza.

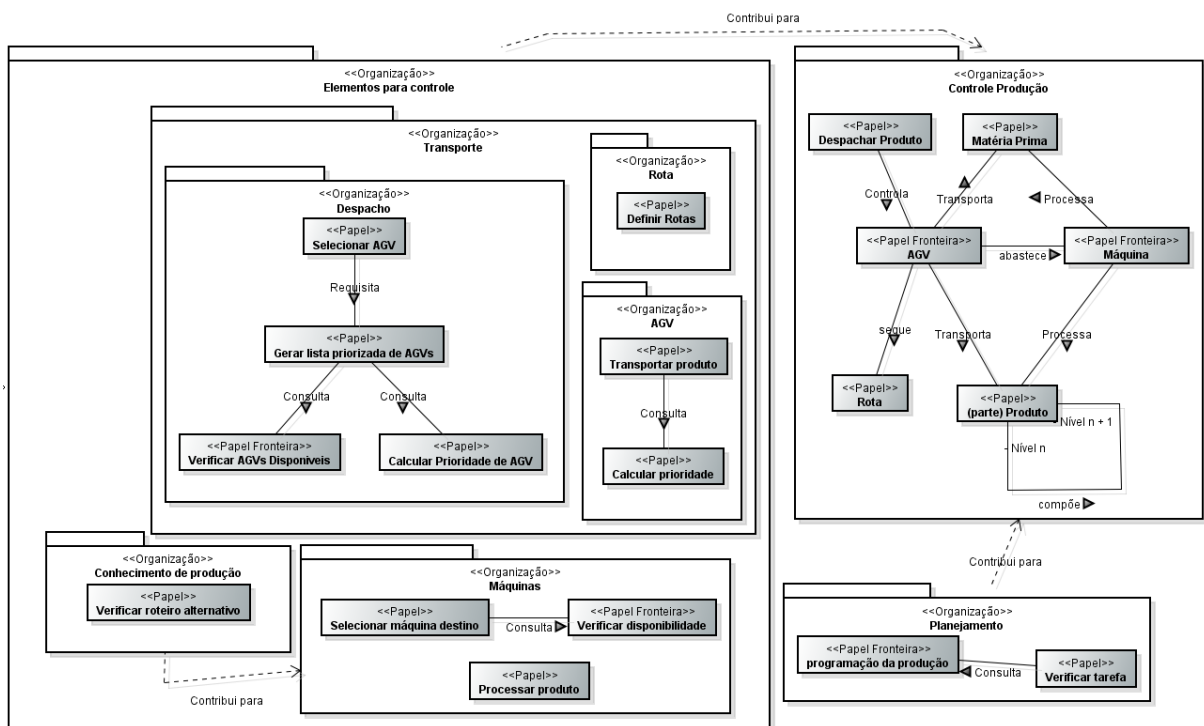


Figura 3.11. Fase 1, atividade 4: Identificação dos papéis e interações

Existe uma relação “contribui para” entre as organizações que indica quando uma organização fornece alguma informação / conhecimento para outra.

A organização Planejamento é composta por dois papéis, sendo eles: verificar tarefa e programação da produção. Este último é responsável por consultar e carregar a programação

da produção, enquanto o papel verificar tarefa é responsável por separar toda a programação da produção em tarefas e gerenciar a ordem em que elas são passadas para a organização Controle produção. Esses dois papéis cumprem o comportamento global da organização planejamento que é identificar as tarefas que compõem a programação da produção.

A organização Elementos para controle é composta por 3 outras organizações, sendo elas: Máquinas, Conhecimento de produção e Transporte. A organização conhecimento de produção possui um único papel, verificar roteiro alternativo, que é responsável por cumprir o comportamento global da organização que é servir de base de conhecimento sobre a forma de produção dos produtos. Esta organização contribui para a organização Máquinas, fornecendo o conhecimento necessário para identificar os roteiros de produção.

A organização Máquinas possui 3 papéis, sendo eles, Selecionar máquina destino e Verificar disponibilidade que interagem entre si para definir a máquina que será o destino do produto (caso ela esteja disponível) e o terceiro papel, Processar produto. Estes papéis cumprem o comportamento global da organização que é responsável por processar os produtos e definir quais máquinas estão disponíveis e podem processar o produto.

A organização Transporte por sua vez, é composta por 3 organizações, sendo elas: Despacho, Rota e AGV. A organização AGV, possui os papéis transportar produto e calcular prioridade, que são responsáveis respectivamente por realizar o transporte do produto e caso necessário, definir uma nota (prioridade) para a tarefa em questão, compondo assim o comportamento global da organização que é responsável por gerenciar o transporte dos produtos e permitir uma autonomia ao AGV caso ele não obtenha contato com o módulo de despacho.

A organização Rota é composta por um único papel, Definir rotas, responsável por escolher uma possível rota para o AGV percorrer, atendendo assim o comportamento global da organização. A organização Despacho é composta por 4 papéis, sendo eles: selecionar AGV, gerar lista priorizada de AGV, verificar AGVs disponíveis e calcular prioridade de AGV. Que são responsáveis por cumprir com o comportamento global da organização, que é verificar os AGVs disponíveis e selecionar o de maior prioridade para uma determinada tarefa.

A organização Controle produção é composta por papéis, sendo eles: despachar produto, matéria prima, AGV, maquina, rota e (parte)Produto. Estes papéis são responsáveis

por representar o comportamento global da organização que simulam o comportamento da fábrica por meio de seus elementos físicos.

Com as organizações e os papéis (assim como suas interações) identificados, a próxima atividade (5ª atividade – Descrição de cenários – SD) é responsável por apresentar a sequência de interações que ocorrem entre os papéis para cada uma das organizações.

Para isso é gerado um diagrama de sequência da UML para representar as interações, como apresentado na Figura 3.12.

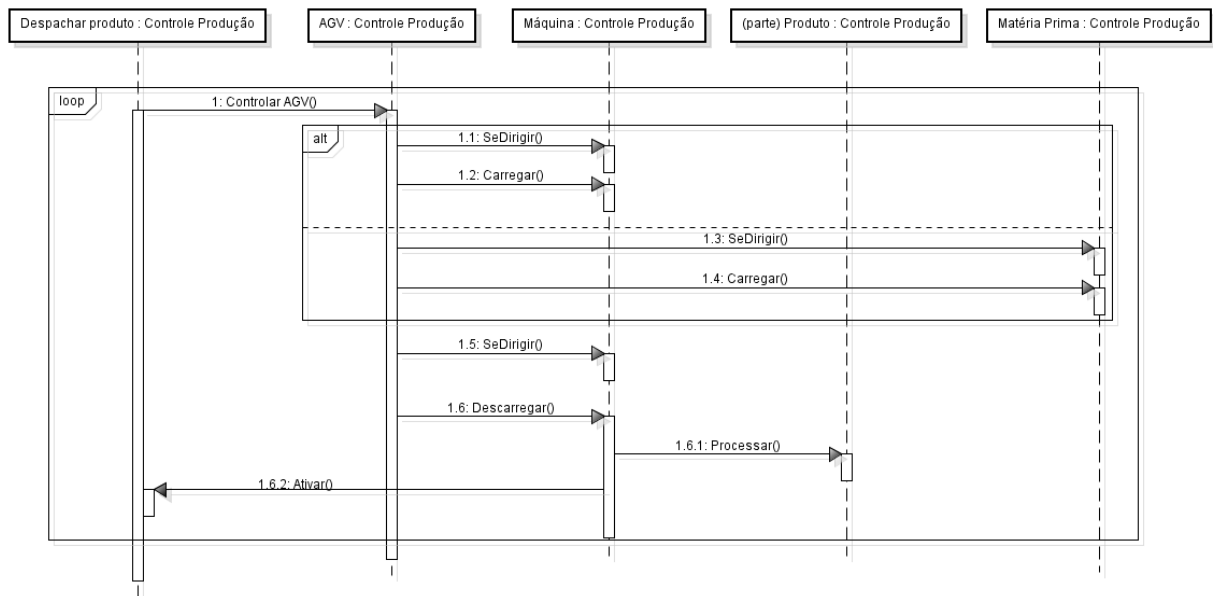


Figura 3.12. Fase 1, atividade 5: Descrição de cenários

Na Figura 3.12 é apresentada a sequência de interações dos papéis que compõem a organização Controle produção. A execução começa quando o papel AGV recebe uma mensagem externa da organização Controle produção, esta mensagem (**controlar()**) faz com que o papel AGV execute e mande uma mensagem (**se dirigir()**) e depois uma outra mensagem (**carregar()**) para o papel Máquina ou para o papel Matéria prima, dependendo da situação. Depois o papel AGV envia mensagens para o papel Máquina (neste caso é a máquina destino), sendo a primeira **seDirigir()** e a segunda **Descarregar()**. Após essas mensagens o papel máquina envia uma mensagem (**processar()**) ao papel (parte)Produto e por fim envia uma mensagem (**ativar()**) externa para a outra parte da organização Controle produção.

Este diagrama de sequência é apenas um dos SD gerados para modelar o problema, os outros diagramas podem ser encontrados no apêndice A.

Na sexta atividade, “plano dos papéis” – RP, o papel é descrito por meio de tarefas que compõem o plano de papel. Cada um dos papéis é apresentado por um plano de papel na forma de um diagrama de atividades da UML. A Figura 3.13 apresenta uma parte do “plano de papel” dos papéis que compõem a organização Controle produção.

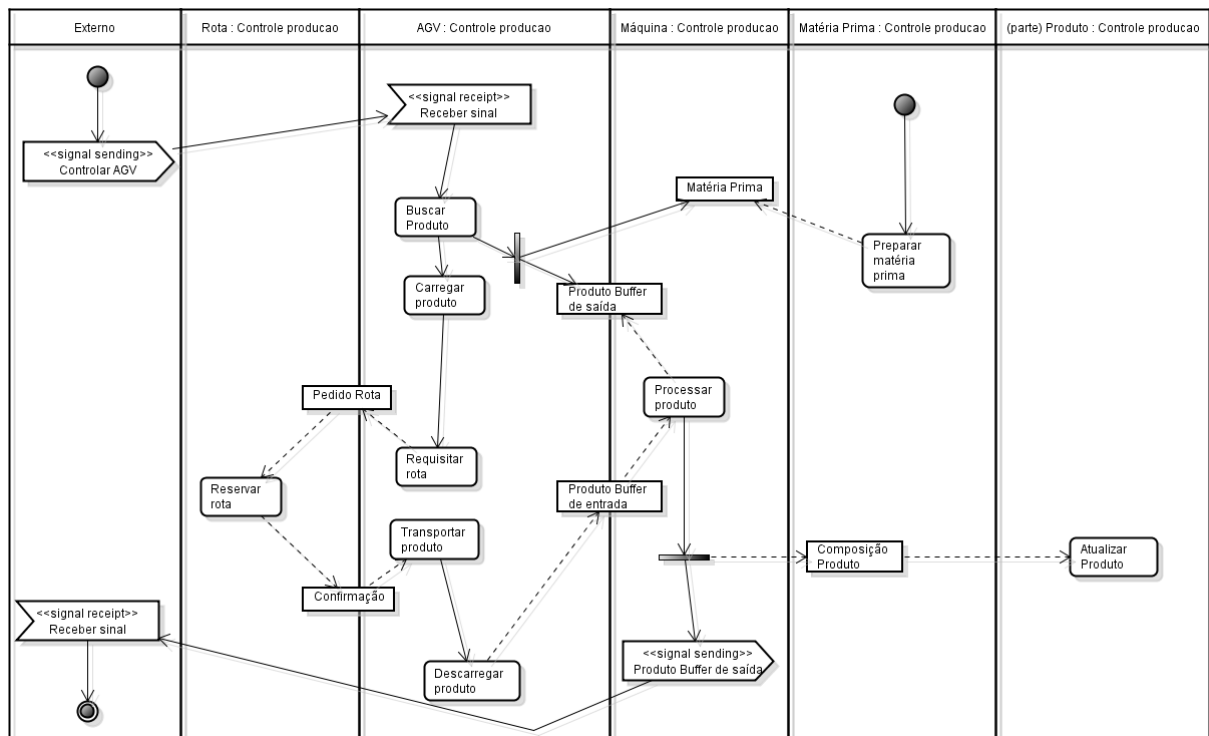


Figura 3.13. Fase 1, atividade 6: Plano dos papéis

Na Figura 3.13 cada uma das raias representam um papel da organização Controle produção, com exceção da primeira que representa alguma entidade externa não pertencente à organização. O papel AGV é ativado por meio de uma entidade externa e executa a tarefa Receber sinal, quando recebido, executa a Buscar por produtos, que pode ser tanto no *buffer* de saída de uma máquina quanto no *buffer* de matéria prima. Para o caso de ser no *buffer* de matéria prima, o papel Matéria prima executa a tarefa de Preparar a matéria prima, para o caso de ser no *buffer* de saída o papel Máquina executa o Processa o produto.

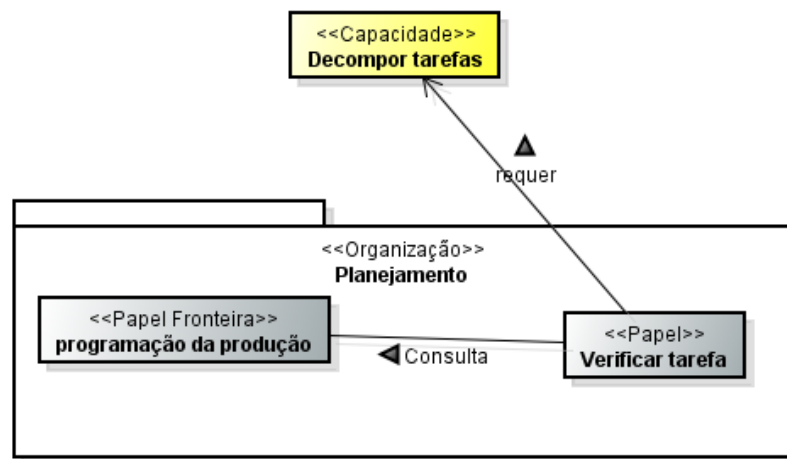
Após buscar o produto o papel AGV carrega o produto e requisita a rota para o papel Rota. Este por sua vez, reserva uma rota e envia a rota reservada para o papel AGV transportar o produto. Quando o produto chega em seu destino, é descarregado pelo papel AGV e é



processado pelo papel Máquina, que por sua vez envia uma mensagem externa à organização para indicar o fim do processamento e também envia uma mensagem para o papel (parte)Produto para atualizar o status do produto.

Os outros diagramas do RP estão descritos no apêndice A.

Na sétima e última atividade da primeira fase, “identificação das capacidades” – CP, são identificadas as capacidades que um elemento tem que possuir para executar o papel. Na Figura 3.14 é apresentado um recorte do diagrama gerado, apresentando uma capacidade necessária para o papel verificar tarefa.



**Figura 3.14. Fase 1, atividade 7: Identificação das capacidades**

Na Figura 3.14 é apresentada a organização Planejamento composta por seus dois papéis (Programação da produção e Verificar tarefas). Para algum elemento executar o papel Verificar tarefas, é necessário que esse elemento tenha a capacidade (elemento em amarelo) de Decompor tarefas.

O diagrama completo é apresentado no apêndice A, informando todas as capacidades levantadas para os papéis identificados no IRI.

### 3.2.5 Modelagem do sistema holônico de controle do despacho multicritério

A segunda fase do ASPECS é referente à solução do problema já com um viés de sistemas multiagentes (já direcionando para o software de sistemas multiagentes que será desenvolvido na terceira fase). Essa fase é composta por 10 atividades, sendo que em cada

atividade é gerado um ou mais artefatos para modelar a solução do problema (conforme já apresentado em detalhes na subseção 2.2.3.1).

Na primeira atividade, “descrição da ontologia de solução” – SOD, foi refinada a ontologia gerada na atividade POD (primeira fase), para representar os conceitos relacionados à solução já baseada em agentes. Na Figura 3.15 são apresentados esses refinamentos (elementos em cinza).

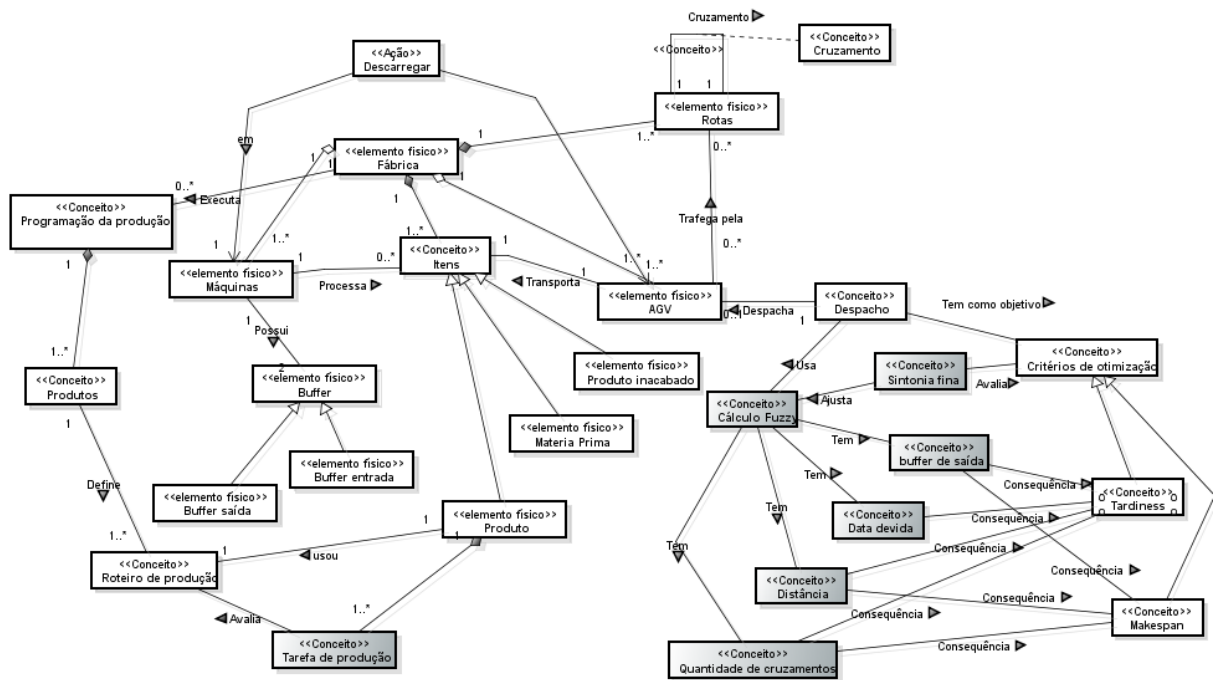


Figura 3.15. Fase 2, atividade 1: Descrição da ontologia de solução

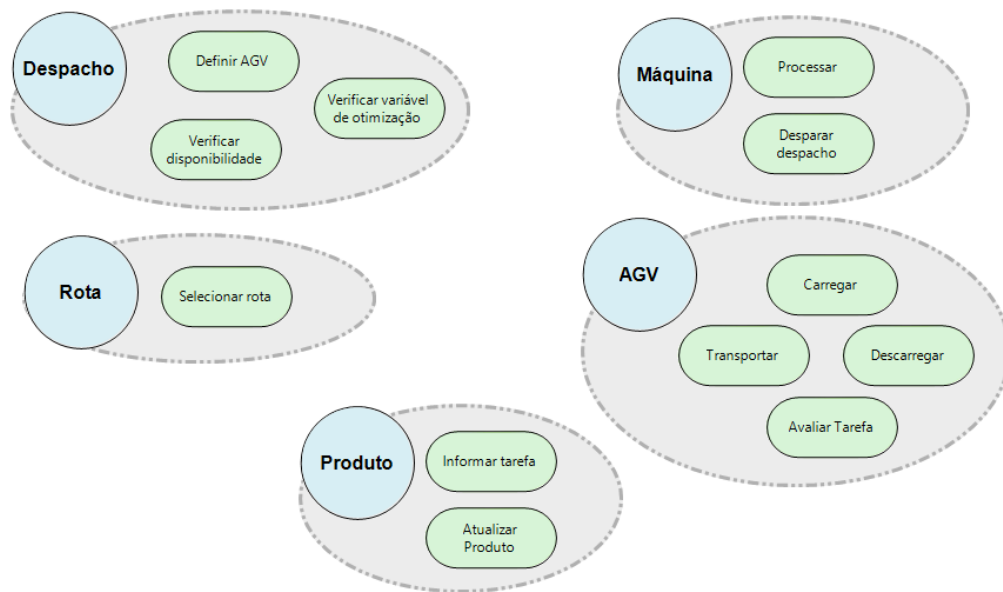
Foi inserido o conceito de Tarefa de produção que está relacionado com o conceito Roteiro de produção pela relação “avalia” e compõe o elemento físico Produto. Este conceito está relacionado a solução do problema pelo fato de decompor os roteiros de produção em uma granularidade menor, tarefas, para que os produtos possam ser produzidos.

Outro conceito inserido foi o Cálculo *fuzzy*, que é usado pelo conceito Despacho e se relaciona pelo relacionamento “tem” com os conceitos: Quantidade de cruzamentos, distância, data devida e *buffer* de saída, sendo que cada um desses conceitos se relaciona ou com o conceito *tardiness*, com o *makespan* ou com os dois.

Por fim, o último conceito relacionado à solução adicionado foi a Sintonia fina. Este conceito avalia os critérios de otimização e ajusta o Cálculo *fuzzy*.

Neste ponto, o método é dividido em duas ramificações (projeto de agentes e projeto de organizações), sendo cada uma delas responsável por modelar a solução sob uma perspectiva diferente.

No projeto de agentes na atividade 2 (numeração geral da fase), “identificação dos agentes” – AI, são identificados os agentes que irão compor o baixo nível da hierarquia. Nesta atividade, é usado o diagrama de objetivo TROPOS, sendo que para esta modelagem, obteve-se o seguinte diagrama (Figura 3.16).



**Figura 3.16. Fase 2, atividade 2: Identificação dos agentes**

Foram identificados 5 agentes, sendo eles: Despacho, Máquina, Rota, AGV e Produto. Sendo que cada um deles possui seus objetivos individuais, como por exemplo o agente AGV que possui os seguintes objetivos individuais: carregar, descarregar, transportar e avaliar tarefa. Os objetivos são as elipses verdes representadas dentro da área de cada agente.

Na atividade 3, “descrição da arquitetura de agentes” – AAD, ainda dentro do projeto de agentes, são apresentados: o agente e as capacidades que ele pode executar por meio das implementações de capacidades, sendo que na Figura 3.17 as capacidades estão representadas em amarelo, os agentes em branco e as implementações de capacidade em laranja.

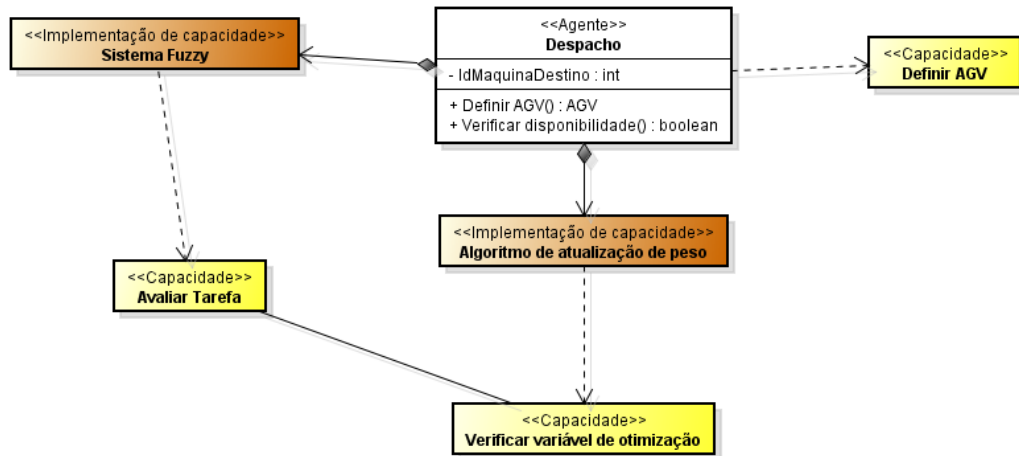


Figura 3.17. Fase 2, atividade 3: Descrição da arquitetura dos agentes

Neste exemplo, o agente Despacho é apresentado e relacionado a 3 capacidades: definir AGV, avaliar tarefa e verificar variável de otimização, sendo estas duas últimas estão atreladas as implementações sistema *fuzzy* e algoritmo de atualização de peso, respectivamente.

A arquitetura dos outros agentes está descrita no apêndice A.

Finalizadas as atividades da ramificação relacionada ao projeto de agentes, dá-se início ao projeto de organizações, com a quarta atividade, “descrição da ontologia de comunicação” – COD.

Nesta atividade são descritas as comunicações existentes entre os papéis, sendo seu conteúdo (linguagem, ontologia e codificação) e sequência de atos de comunicação (protocolos) explicitados, conforme apresentado na Figura 3.18.

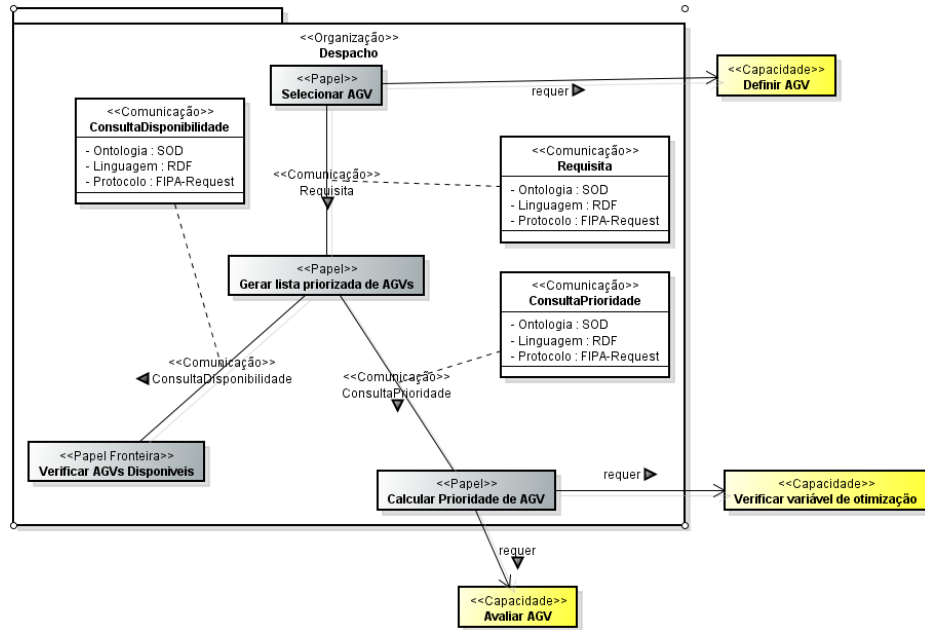


Figura 3.18. Fase 2, atividade 4: Descrição da ontologia de comunicação.

Na Figura 3.18, a comunicação é descrita pelos elementos em branco, sendo que ela ocorre sempre que há uma interação entre papéis, nesse caso, entre os papéis: Selecionar AGV e Gerar lista priorizada de AGVs; Verificar AGVs disponíveis e Gerar lista priorizada de AGVs; e por fim Calcular prioridade de AGV e Gerar lista priorizada de AGVs. Nos três casos é seguido a ontologia SOD, a linguagem é a RDF<sup>12</sup> e o protocolo seguido é o FIPA-Request (protocolo padrões FIPA).

As demais comunicações para as interações entre os outros papéis existentes no modelo estão descritas no apêndice A.

Na atividade 5, “descrição do comportamento do papel” – RBD, ainda dentro do projeto de organizações, foi realizado um refinamento da atividade RP (1º fase), descrevendo o comportamento de cada um dos papéis. Na Figura 3.19 é apresentado o refinamento dos papéis da organização Fábrica.

<sup>12</sup> RDF é a sigla para *Resource Description Framework* que é um padrão de troca de dados criado pela w3c (<http://www.w3.org/2001/sw/#rdf>)

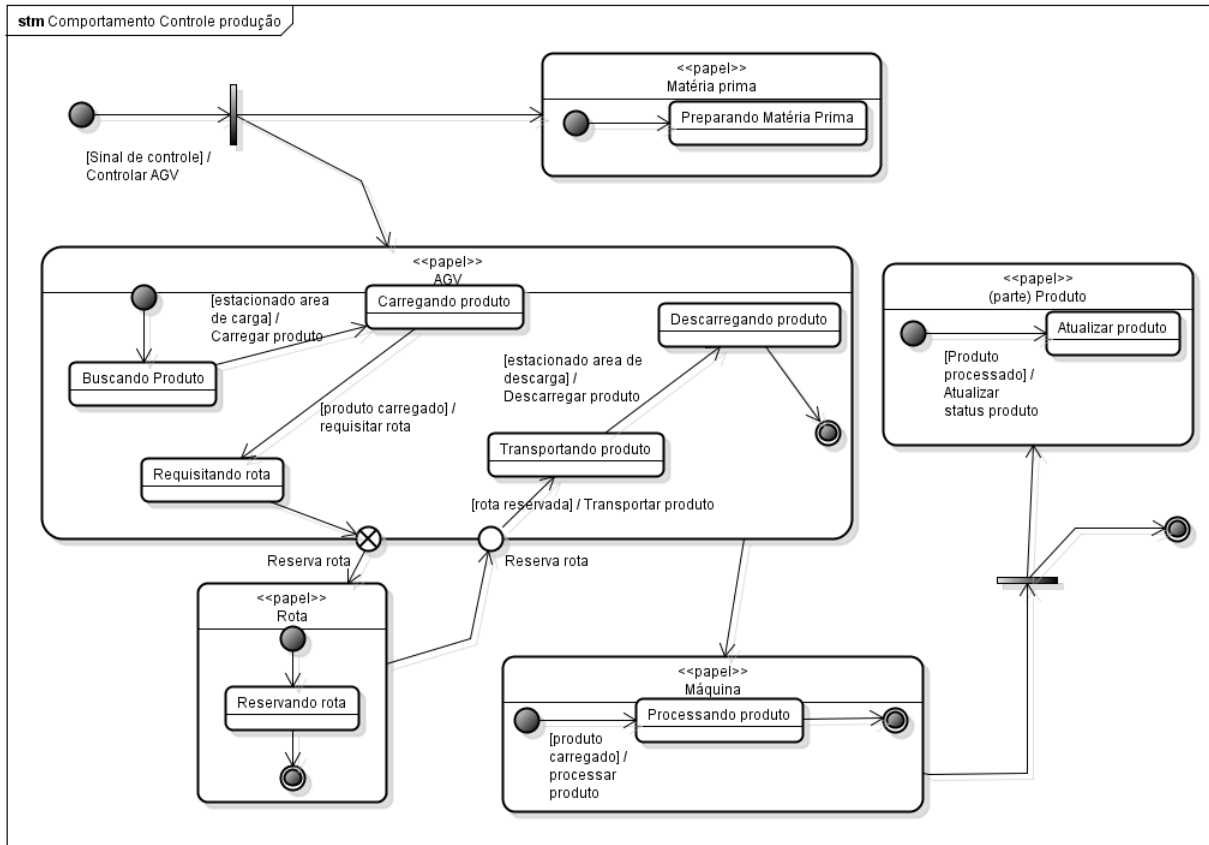


Figura 3.19. Fase 2, atividade 5: Descrição do comportamento do papel.

O papel “matéria prima” é composto por uma única atividade “preparando matéria prima”, ativada toda vez que um sinal de controle chega na organização. O papel AGV também é ativado pelo mesmo sinal de controle, porém ele apresenta mais atividades para descrever seu comportamento, que funciona da seguinte forma: quando ativado, o AGV vai buscar um produto, quando estaciona na área de carga dá máquina, o produto é carregado e é requisitada uma rota. Após a resposta do papel Rota, o produto é transportado até a área de descarga da máquina de destino e o produto é descarregado.

O papel Rota apresenta apenas a atividade “reservando rota” e é ativado quando o papel AGV faz uma requisição por uma rota. O papel Máquina é descrito por apenas uma atividade de processamento e é ativado assim que o papel AGV descarrega um produto para ser processado. Ao fim desta atividade, o papel Máquina informa o fim do processamento ao papel (parte)Produto que atualiza o status do produto e ao mesmo tempo envia um sinal para fora da organização.

As descrições do comportamento dos outros papéis podem ser encontradas no apêndice A.

A atividade 6, “descrição do protocolo” – PD, ainda no projeto de organização, é uma atividade facultativa. Como não foi utilizado nenhum protocolo fora os estabelecidos pela FIPA, esta atividade não foi desenvolvida.

Na atividade 7, “descrição das dependências da organização” – ODD, ainda no projeto de organização, foram descritas as relações entre as capacidades requisitadas pelos papéis e organizações, assim como os serviços que as realizam. Na Figura 3.20 são apresentados os serviços oferecidos pelos papéis (elementos em verdes), os recursos utilizados (elementos em marrom) e suas relações.

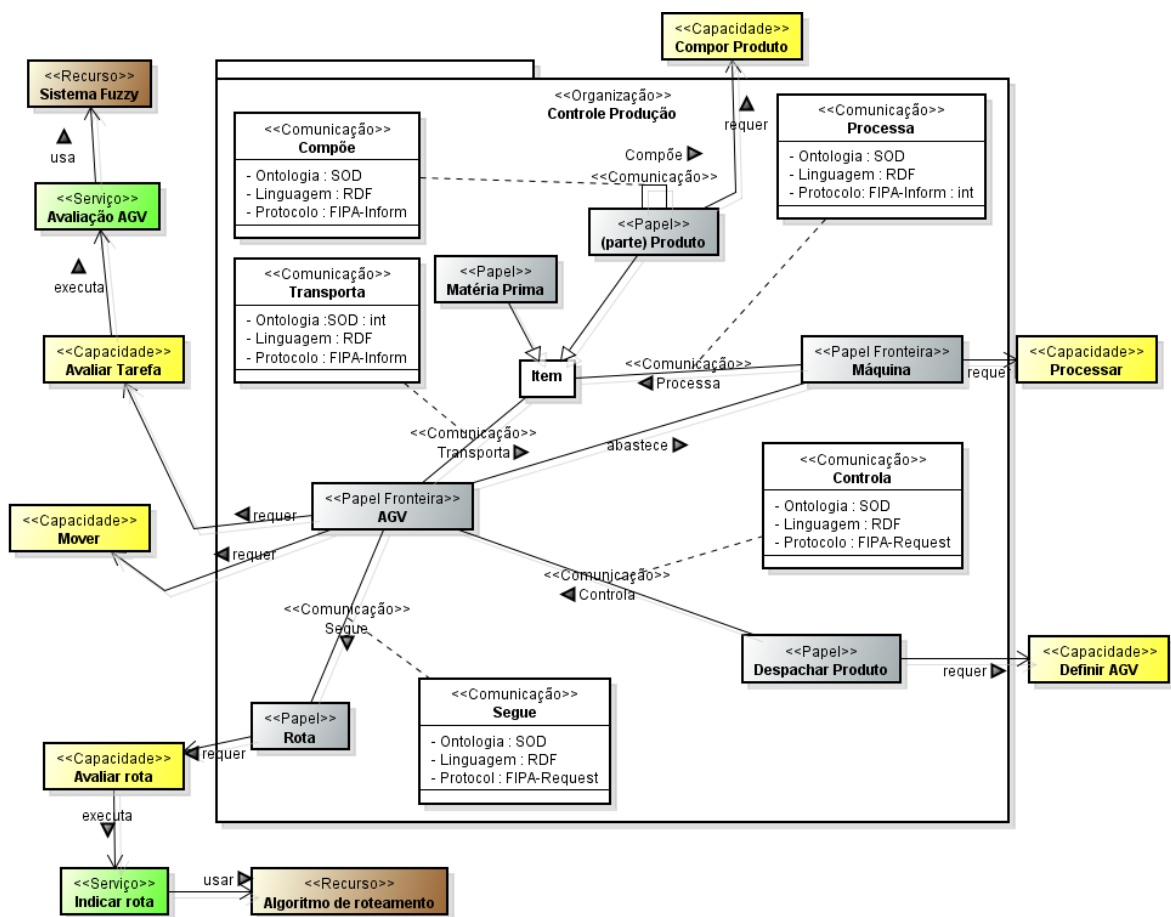


Figura 3.20. Fase 2, atividade 7: Descrição das dependências da organização.

O papel AGV requisita a capacidade Avaliar tarefa, de quem o for executar. Neste caso em específico, a capacidade Avaliar tarefa é executada pelo serviço Avaliação de AGV que utiliza o Sistema *fuzzy* como recurso.

O papel Rota exige a capacidade Avaliar rota que é executada pelo serviço Indicar rota que usa algoritmo de roteamento como recurso.

Neste ponto do método, as duas ramificações (projeto de organizações e de agentes) se unem novamente em uma única atividade, “identificação das restrições dos papéis” – RCI, na qual foi realizada a explicitação das restrições dos papéis (execução simultânea, exclusão mútua, dependência, etc...). Na Figura 3.21 é apresentada uma das restrições (relação em vermelho) existentes entre os papéis.

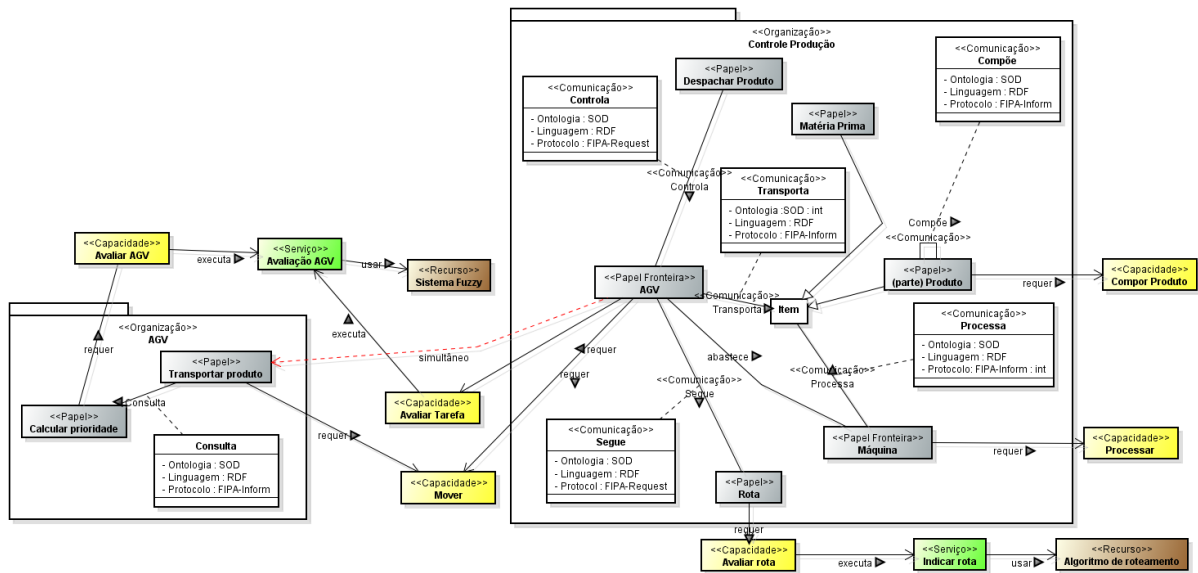


Figura 3.21. Fase 2, atividade 8: Identificação das restrições dos papéis.

No diagrama da Figura 3.21 é apresentada a relação simultânea (em vermelho) que indica a execução do papel Transportar produto e do papel AGV ao mesmo tempo e executada pelo mesmo elemento. As outras restrições são apresentadas no apêndice A.

Na atividade 9, “descrição do plano de agente” – APD, é apresentado a arquitetura interna dos agentes por meio dos papéis que ele executa, de acordo com suas capacidades. Na Figura 3.22 é apresentado o plano do agente AGV.



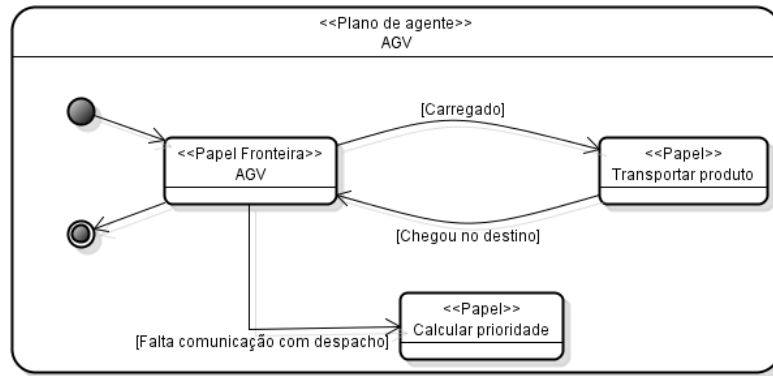


Figura 3.22. Fase 2, atividade 9: Descrição do plano de agente.

No plano de agente apresentado na Figura 3.22, é indicado que o agente AGV irá executar o papel AGV; quando ele estiver carregado, irá executar o papel Transportar produto e quando chegar no destino voltará a executar o papel AGV. No caso de uma eventual falta de comunicação com o despacho, este agente irá executar o papel Calcular prioridade.

Com todos os diagramas criados nas atividades anteriores, já se tem toda a definição / especificação dos agentes, papéis, capacidades e organizações. A partir destas especificações, na atividade 10, “projeto da holarquia” – HD, é realizada uma adaptação da hierarquia proposta em um holarquia, apresentando também as regras que governam a dinâmica da estrutura.

Na primeira tarefa do HD, “tarefa de identificação dos *holons*”, foi mapeada a hierarquia das organizações em uma holarquia, conforme apresentado na Figura 3.23.

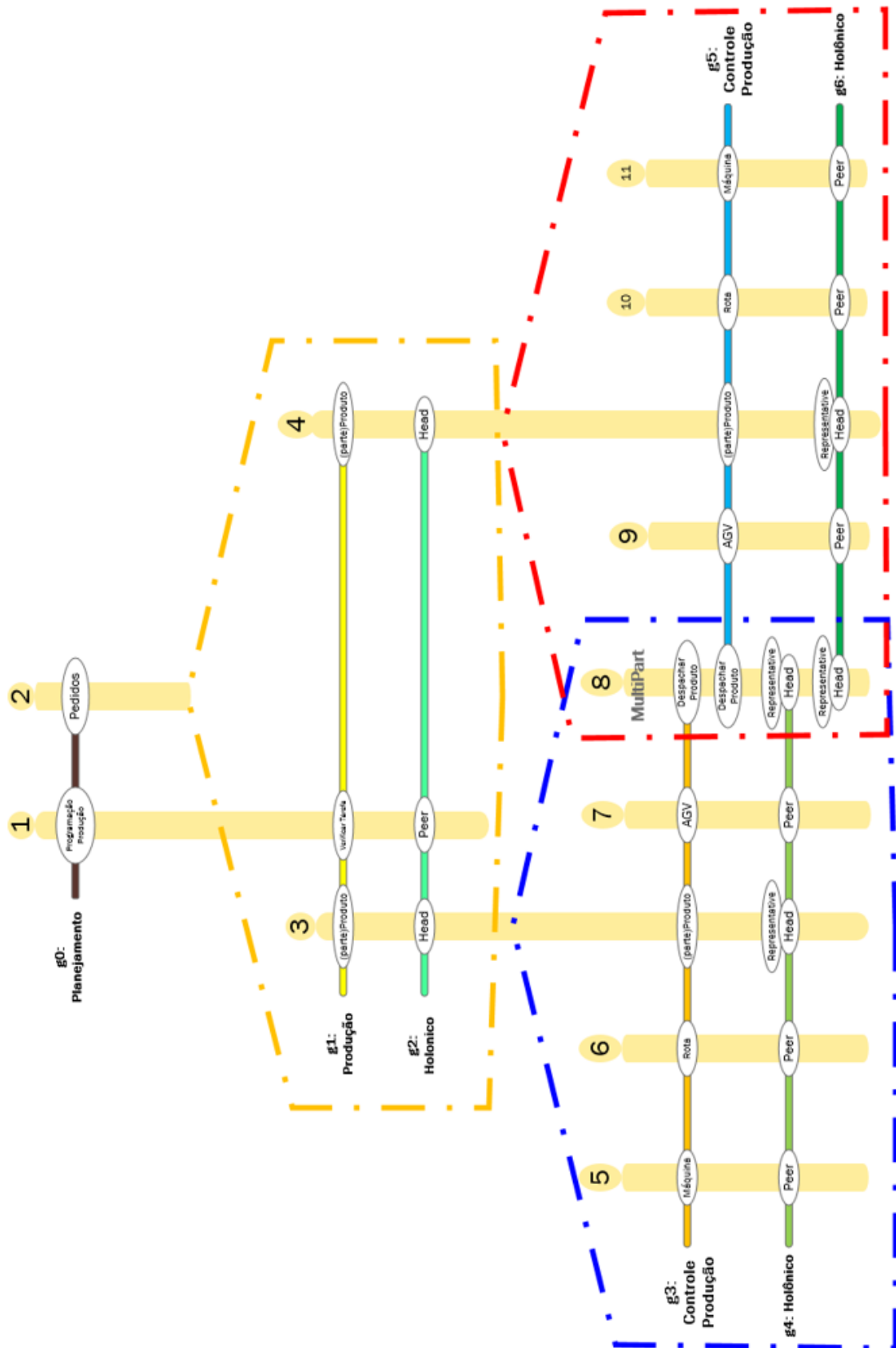


Figura 3.23. Fase 2, atividade 10: Projeto da holarquia.

Na holarquia apresentada na Figura 3.23, o primeiro nível é composto por dois *holons* (1 e 2) que participam do mesmo grupo (g0: planejamento). O primeiro *holon* executa o papel Programação da produção enquanto o segundo *holon* representa todos os pedidos recebidos pela fábrica.

No segundo nível, são apresentados 3 *holons*, sendo que os 3 estão no contexto do *holon* 2. Estes 3 *holons* participam dos grupos produção (g1) e holônico (g2). Nota-se que um destes *holons* (o *holon* número 1) participa em dois níveis diferentes, executando papéis diferentes em diferentes grupos. Os *holons* 3 e 4, representam os produtos e executam os papéis (parte)Produto no grupo g1. No grupo g2 estes *holons* são *heads* que significa que são eles que tomam as decisões dentro do grupo. Essas decisões estão relacionadas a como formar um subgrupo (sub *holon*) com relação a consulta realizada no outro *holon* que executa o papel Verificar tarefa.

No terceiro nível, no contexto interno de um *holon* (parte)Produto, estão os *holons* responsáveis por processar o produto que ele está representando. Este conjunto é formado de maneira dinâmica, sendo formado dos recursos necessários para cada uma das tarefas de produção do produto. Este grupo é formado por *holons* que representam os recursos: máquinas, AGVs e rotas; pelo próprio (parte)Produto e por um *holon* que participa de vários grupos ao mesmo tempo (*multipart*) que representa o despacho. Cada um desses *holons* executa o papel especificado para ele contribuindo assim para a execução da tarefa indicada pelo *holon* (parte)Produto, que é o *head* desse grupo.

Pode-se observar que existem dois *heads* no terceiro nível que pertencem ao mesmo grupo. Isso se deve ao fato das diferentes decisões que devem ser tomadas dentro do grupo. A primeira delas é relativa a tarefa que deverá ser executada, gerenciado pelo *holon* 3; a segunda é relativa a quais recursos deverão entrar no grupo para que a tarefa seja executada, gerenciado pelo *holon* 8. Os outros *holons* por serem *peer*, não participam das decisões dentro do grupo.

Essa holarquia apresenta apenas 3 níveis de *holons*, a holarquia completa (4 níveis) pode ser consultada no apêndice A.

### 3.3 Considerações finais

Neste capítulo foi abordado o despacho multicritério de veículos, conforme apontado nos artigos correlatos, foi escolhido o multicritério pela melhor eficiência diante do ambiente dinâmico e por sofrer menos impacto perante as mudanças do ambiente de manufatura.

Também se optou por atender as variáveis de otimização *makespan* e *tardiness*, mesmo sabendo que é mais comum serem utilizadas por diferentes níveis organizacionais, uma mais voltada ao chão de fábrica e outro às estratégias de negócio.

Para controlar os AGVs é necessária uma arquitetura de controle. Para este trabalho, com base nos artigos correlatos e na fundamentação teórica, optou-se por usar uma abordagem holônica para conseguir atingir vantagens inerentes das arquiteturas centralizadas e descentralizadas ao mesmo tempo e também permitir a adição de novos elementos à arquitetura sem muitas adaptações.

Outra opção adotada com base na literatura, foi a adoção de sistemas multiagentes para realizar a implementação do HMS.

Para realizar essa implementação, optou-se por seguir as diretrizes do ASPECS, por ser um método que dá suporte desde a fase de levantamento de requisitos até a fase de implementação e implantação do sistema.

Após um estudo e análise detalhada sobre qual arquitetura holônica usar como base, chegou-se à conclusão que por mais que o ANEMONA siga as diretrizes do PROSA (referência muito difundida na literatura), ela se torna rígida quando surge a necessidade de adicionar novos elementos que não se enquadram aos *holons* básicos propostos por ela. Essa rigidez é solucionada no ASPECS graças aos conceitos de papel, capacidade, interação e organização herdadas do método RIO. Outro fator determinante para a escolha do ASPECS foi a existência de um framework multiagente holônico (JANUS) pronto para usar, sendo que no ANEMONA, o framework de desenvolvimento (JADE) não tem suporte nativo para holônicos.

Neste capítulo também foi apresentado o sistema fuzzy em detalhes para priorizar os AGVs em função de uma determinada tarefa de transporte. Este sistema fuzzy apresenta um auto ajuste das variáveis de entrada dependendo da saída da produção, para permitir uma melhor adaptação do sistema ao ambiente dinâmico.

Também foi apresentada a modelagem do problema utilizando a primeira fase do ASPECS, assim como a modelagem da solução já com o viés de sistemas multiagentes. Essa modelagem é composta por vários diagramas (a maioria diagramas UML), sendo que foram apresentados apenas uma amostra dos diagramas gerados, sendo que os diagramas completos estão apresentados no apêndice A.

# DESENVOLVIMENTO DO TRABALHO

---

### 4.1 Considerações iniciais

Neste capítulo será apresentado o desenvolvimento da proposta, que consiste na realização da terceira fase do ASPECS (implementação e implantação).

Como o desenvolvimento do trabalho está relacionado aos códigos gerados para a implementação da arquitetura, optou-se por apresentar neste capítulo apenas uma amostra do código para servir de exemplo das atividades do ASPECS, vinculando a modelagem ao código.

### 4.2 Implementação arquitetura holônica de controle

Apoiado no método do ASPECS, a terceira fase é a implementação e implantação. Essa fase é composta por 8 atividades, sendo que para cada atividade serão gerados códigos de implementação em Java (usando a biblioteca do JANUS), com exceção da primeira atividade que possui como saída um diagrama de classes e da penúltima atividade que possui como saída um diagrama de implantação.

Na primeira atividade, “Definição da arquitetura do *holon*” – HAD, foi gerado o diagrama da arquitetura do *holon*, associando cada *holon* à um conjunto de papéis, à um conjunto de capacidades e a um conjunto de serviços. O diagrama gerado para esta atividade é apresentado na Figura 4.1.

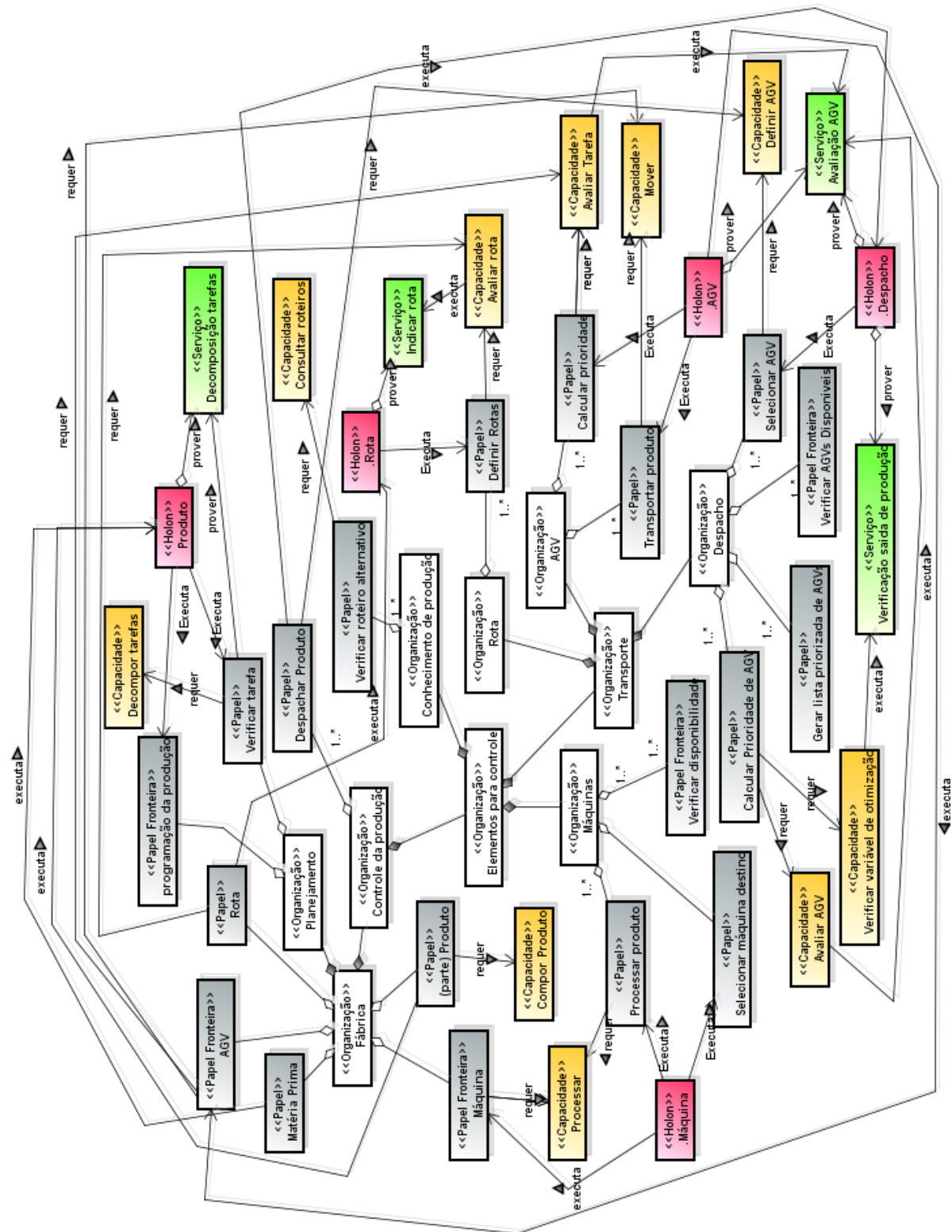


Figura 4.1. Fase 3, atividade 1: Definição da arquitetura do holon.

Neste diagrama, as organizações são representadas pelas classes em branco; os papéis pelas classes em cinza; as capacidades pelas classes em amarelo; os serviços pelas classes verdes e os holons pelas classes em vermelho.

O *holon* AGV executa os papéis “transportar produto”, “calcular prioridade” e o papel de fronteira “AGV”. O *holon* despacho executa os papéis “selecionar AGV” e “despachar produto”. O *holon* máquina executa os papéis “processar produto”, “selecionar máquina destino” e o papel fronteira “máquina”. O *holon* rota executa os papéis “rota” e “definir rota”. Por fim, o *holon* produto executa os papéis “verificar tarefa”, “matéria prima”, “(parte)Produto” e o papel fronteira “programação da produção”.

Após definir a arquitetura do *holon*, foi realizado o “reuso de código” – CR (segunda atividade da terceira fase). Nesta atividade não foi gerado nenhum código ou diagrama por não ter nenhum código de *template* para ser utilizado.

Na terceira atividade, “produção do código das organizações e papéis” – CPOR, foi gerado o código para cada uma das organizações e papéis existentes na modelagem. Para isso usou-se a ferramenta Eclipse<sup>13</sup> e a biblioteca do JANUS<sup>14</sup>. Na Figura 4.2 é apresentada a codificação da organização “controle produção” e na Figura 4.3 a codificação do papel AGV.

```

1 package tear.holonico.arquitetura.organizacoes;
2
3 import org.janusproject.kernel.crio.core.CRIOContext;
4 import org.janusproject.kernel.crio.core.Organization;
5
6 import tear.holonicos.arquitetura.papeis.controleProducao.AGV_papel;
7 import tear.holonicos.arquitetura.papeis.controleProducao.DespacharProduto_papel;
8 import tear.holonicos.arquitetura.papeis.controleProducao.Maquina_papel;
9 import tear.holonicos.arquitetura.papeis.controleProducao.MateriaPrima_papel;
10 import tear.holonicos.arquitetura.papeis.controleProducao.ParteProduto_papel;
11 import tear.holonicos.arquitetura.papeis.controleProducao.Rota_papel;
12
13
14 public class ControleProducao_org extends Organization {
15
16     public ControleProducao_org(CRIOContext context){
17         super(context);
18
19         //PAPEIS ADICIONADOS À ORGANIZAÇÃO
20         addRole(AGV_papel.class);
21         addRole(DespacharProduto_papel.class);
22         addRole(Maquina_papel.class);
23         addRole(MateriaPrima_papel.class);
24         addRole(ParteProduto_papel.class);
25         addRole(Rota_papel.class);
26
27     }
28
29
30
31 }
32

```

Figura 4.2. Código da organização "controle produção"

<sup>13</sup> <https://eclipse.org/>

<sup>14</sup> <http://www.janus-project.org/Home>



```

15 @RoleActivationPrototype(fixedParameters = {})
16 public class DefinirRotas_papel extends Role {
17
18     /* Variáveis internas do papel
19     *****/
20     // variável para definir a máquina de estados do papel
21     private State state;
22
23     // id da rota escolhida
24     int rotaEscolhida;
25
26     String maquinaFonte; // variavel de teste;
27
28     String maquinaDestino; // variavel de teste;
29
30
31     /* Método do ciclo de vida do papel: Inicializar
32     *****/
33     public Status activate(Object... parameters) {
34         this.state = State.WAIT_SINAL_ROTA;
35
36         addObtainCondition(new HasAllRequiredCapacitiesCondition(
37             AvaliarRota_cap.class));
38
39         return StatusFactory.ok(this);
40     }
41
42
43     /* Método do ciclo de vida do papel: Executar
44     *****/
45     @Override
46     public Status live() {
47         this.state = Run();
48
49         return StatusFactory.ok(this);
50     }
51
52
53     /* Método do ciclo de vida do papel: Finalizar
54     *****/
55
56
57
58
59     public Status end() {
60
61         return StatusFactory.ok(this);
62     }
63
64

```

Figura 4.3. Código do papel AGV

Para a codificação da organização Controle da produção, foi gerado uma classe em Java com o nome `ControleProducao_org`, sendo que essa estende a classe `Organization`, classe da API (*Application Programming Interface*) do Janus, que fornece todos os métodos relativos as organizações. Conforme apresentado um exemplo nas linhas 20 à 25 da Figura 4.2, é usado o método `addRole()`, o qual vincula um papel à uma organização, conforme modelado no IRI (quarta atividade da primeira fase).

Para a codificação dos papéis, se torna necessário estender à classe `Role` (API Janus), herdando assim os métodos do ciclo de vida já implementados nesta. Com esses métodos se torna possível simular o comportamento do papel descrito na modelagem; para isso, é necessário codificar o conteúdo dos métodos `activate()` (34), `live()` (47) e `end()` (59).

Foi gerado uma classe para cada organização e para cada papel descritos na modelagem, codificando seus comportamentos seguindo os diferentes diagramas (OID, IRI, SD, RP, CI, COD, RBD, ODD, RCI, APD e HD).

Na atividade 4, “atividade de teste” – TA, foi criado testes unitários para cada um dos papéis e das organizações criadas na atividade anterior. Na

```

1 package tear.holonico.teste.rota;
2
3+ import org.janusproject.kernel.Kernel;[]
6
7 public class TestarRota {
8-     public static void main(String[] args) {
9
10         Kernel k = Kernels.get(false);
11         Agent a1 = new AgentTestDefinirRotas();
12         Agent a2 = new AgentTestDespacharProduto();
13         k.launchLightAgent(a2);
14         k.launchLightAgent(a1);
15         k.kill();
16
17     }
18 }

```

Figura 4.4 é apresentado o código de teste do papel Rota.

```

1 package tear.holonico.teste.rota;
2
3+ import org.janusproject.kernel.Kernel;[]
6
7 public class TestarRota {
8-     public static void main(String[] args) {
9
10         Kernel k = Kernels.get(false);
11         Agent a1 = new AgentTestDefinirRotas();
12         Agent a2 = new AgentTestDespacharProduto();
13         k.launchLightAgent(a2);
14         k.launchLightAgent(a1);
15         k.kill();
16
17     }
18 }

```

**Figura 4.4. Código de teste unitário do papel Rota.**

Esta classe codificada em Java realiza o teste unitário do papel Rota, criando e executando agentes que irão executar o papel Rota e verificar suas funcionalidades.

Na atividade 5, “produção do código dos *holons*” – CPH, foram gerados os códigos de todos os *holons* levantados na modelagem. Na Figura 4.5 é apresentado o código do *holon* Rota.

```

1 package tear.holonico.arquitetura.agentes;
2
3 import org.arakhne.afc.vmutil.locale.Locale;
12
13 public class Rota_agente extends Agent {
14
15     * Variáveis
16     *****/
17     private static final long serialVersionUID = 441288287106959345L;
18
19
20     * Metodo de ativação - INICIALIZACAO
21     *****/
22     public Status activate(Object... parameters) {
23
24         print(Locale.getString(Rota_agente.class, "ROTA AGENTE: INICIALIZA", getAddress()));
25
26         //////////////////////////////////////////////////GRUPO DE PRODUCAO*****\////////////////////////////////////
27         //-----ROTA_papel-----\////////////////////////////////////
28         // recebe o grupo que irá atuar por parametro
29         GroupAddress gProducao = (GroupAddress) parameters[0];
30
31         if (requestRole(Rota_papel.class, gProducao) == null) {
32             print(Locale.getString(Rota_agente.class, "ERRO NA EXECUÇÃO DO PAPEL Rota", getAddress())); //$NON-NLS-1$
33             StatusFactory.cancel(this);
34         }
35         //////////////////////////////////////////////////CRIAR O GRUPO DE ROTAS*****\////////////////////////////////////
36         //-----DefinirRotas_papel-----\////////////////////////////////////
37         GroupAddress gRota = getOrCreateGroup(Rota_org.class);
38         if (requestRole(DefinirRotas_papel.class, gRota) == null) {
39             print(Locale.getString(Rota_agente.class, "ERRO NA EXECUÇÃO DO PAPEL DefinirRotas", getAddress())); //$NON-NLS-1$
40             StatusFactory.cancel(this);
41         }
42
43         return StatusFactory.ok(this); }
44
45     * Metodo de executar - EXECUCAO
46     *****/
47     // public Status live() {
48     // }
49
50
51     * Metodo de finalizar - FINALIZACAO
52     *****/
53     @Override
54     public Status end() {
55         print(Locale.getString(Rota_agente.class, "ROTA AGENTE MORRE", getAddress())); //$NON-NLS-1$
56         return StatusFactory.ok(this);
57     }
58
59 }

```

Figura 4.5. Código do *holon* Rota.

Para codificar o *holon* rota, é necessário estender a classe Agent (API JANUS), e também definir seus métodos do ciclo de vida, `activate()` (22), `live()` (47) e `end()` (54).

Esta classe é capaz de criar grupos para realizar papéis referentes as organizações especificadas na modelagem, por meio do comando `getOrCreateGroup()` (37) e executar papéis nesses grupos específicos, por meio do comando `requestRole()` (38).

Na atividade 6, “teste unitário dos *holons*” – HUT, foi realizado testes unitários em todos os *holons* codificados. Na Figura 4.6 é apresentado o código do teste do *holon* Pedidos.

```
1 package tear.holonico.teste.producao;
2
3 import org.janusproject.kernel.Kernel;
11
12 public class ExecutaPedidos {
13
14     public static void main(String[] args) {
15
16         Kernel k = Kernels.get(false);
17
18         //Criação dos agentes
19         Agent a1 = new Pedidos_agente();
20
21
22         // launch Pedidos
23         k.launchHeavyAgent(a1, "Pedidos");
24
25     }
26
27
28 }
```

Figura 4.6. Teste unitário do *holon* Pedidos.

Esta classe codificada em JAVA realiza o teste o unitário no *holon* referente ao pedido, criando-o no sistema para testar seu comportamento (modelado pelos papéis). Nas linhas 19 e 23, podemos ver os comandos para a criação e execução do *holon*, respectivamente.

Na atividade 7, “configuração de implantação” (DC – *Deployment Configuration*), foi definido onde e como a implantação foi realizada, identificando locais dos *holons*, relações com os dispositivos externos, com os recursos, etc. Pelo fato de que foi realizado apenas uma máquina e que o sistema não tem comunicação com sistemas externos, não foi gerado o diagrama de componentes.

Por fim, na atividade 8, “teste de integração” – IT, foi realizado o teste de integração para confirmar a correta execução do sistema, conforme a modelagem.

### 4.3 Considerações finais

Neste capítulo apresentou-se o desenvolvimento da proposta. Este desenvolvimento foi demonstrando por meio de exemplos de como implementar o sistema holônico de controle (3º fase do ASPECS), vinculando a modelagem criada nas fases anteriores do ASPECS com a terceira fase relacionada à codificação.

# ANÁLISE DA ARQUITETURA

---

### 5.1 Considerações iniciais

Neste capítulo serão apresentadas as duas análises qualitativas realizadas na arquitetura, que tem o intuito de: I) ilustrar e analisar a generalidade da arquitetura, adicionando um novo elemento; II) ilustrar e analisar o comportamento do processo de adaptação da arquitetura diante de eventos não programados.

A primeira análise está ligada diretamente ao objetivo da tese, que é detalhar uma arquitetura holônica especializada para o despacho de AGV. Esta arquitetura deverá atender os conceitos holônicos, permitindo assim adicionar um novo elemento ou até mesmo detalhar um elemento já existente, sem grandes alterações na arquitetura proposta. A segunda análise permite avaliar o comportamento da arquitetura diante do ambiente dinâmico da manufatura, ilustrando sua forma de decisão diante de perturbações internas e externas (quebra de máquina e chegada de novos pedidos, respectivamente).

### 5.2 Análise qualitativa da generalidade da arquitetura

Um dos objetivos almejados deste trabalho é a especialização de uma arquitetura holônica para realizar o despacho e que seja genérica suficiente, permitindo a inclusão de novos elementos de controle sem muita alteração.

Para atingir esse objetivo, optou-se pela modelagem holônica que por natureza permite a inclusão e ou detalhamento de elementos de maneira natural. Para realizar a análise sob essa ótica, foi adicionado um elemento de controle do roteamento de veículos no modelo proposto.

Para este caso, o elemento de roteamento de veículos já existe na modelagem, porém é um sistema bem simples e não usa nenhuma técnica de roteamento mais refinada para a realização do mesmo. Para ilustrar a inclusão (ou detalhamento) do elemento de controle do roteamento de veículos, foi seguido o método, adicionando os elementos necessários em cada etapa a partir da modelagem já existente. Para a atividade da POD (fase 1, atividade 2) foi acrescentado um elemento para representar essa inclusão (Figura 5.1).

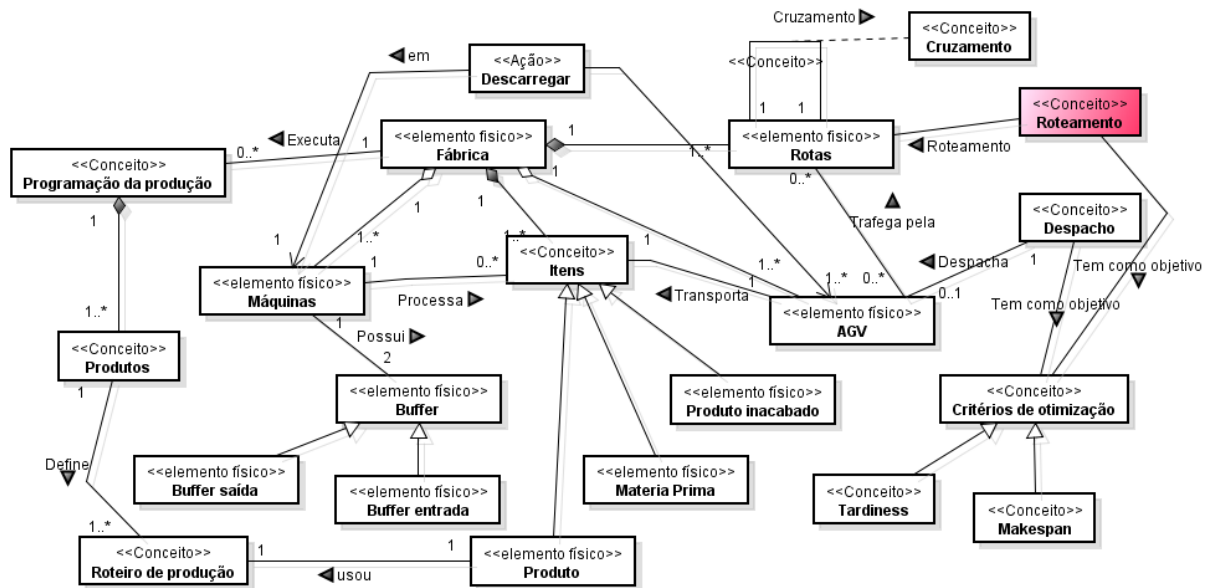


Figura 5.1. Descrição da ontologia do problema de roteamento.

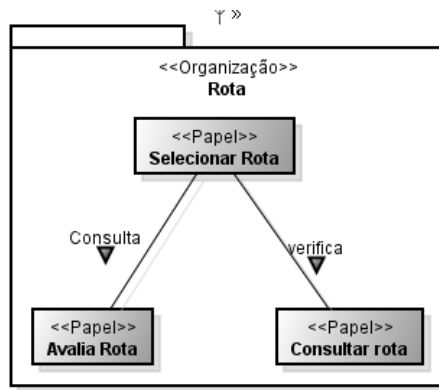
A partir da ontologia já definida, foi incluindo um único elemento, o Roteamento (em vermelho), acrescido de suas relações com o elemento físico Rotas e com o Critérios de otimização vinculando-o indiretamente com o *tardiness* e o *makespan*.

Após a identificação dos elementos ontológicos para modelar o roteamento de veículos, foram seguidas as atividades para se definir os papéis, organizações e capacidades relacionadas a esse elemento de controle.

Para a terceira atividade (OI), como é apenas um detalhamento de um elemento que já existe na modelagem, não se fez necessária a modificação do diagrama gerado na proposta.

Caso fosse a inclusão de um novo elemento que pertencesse a uma organização ainda não descrita, seria necessário acrescentá-la ao modelo.

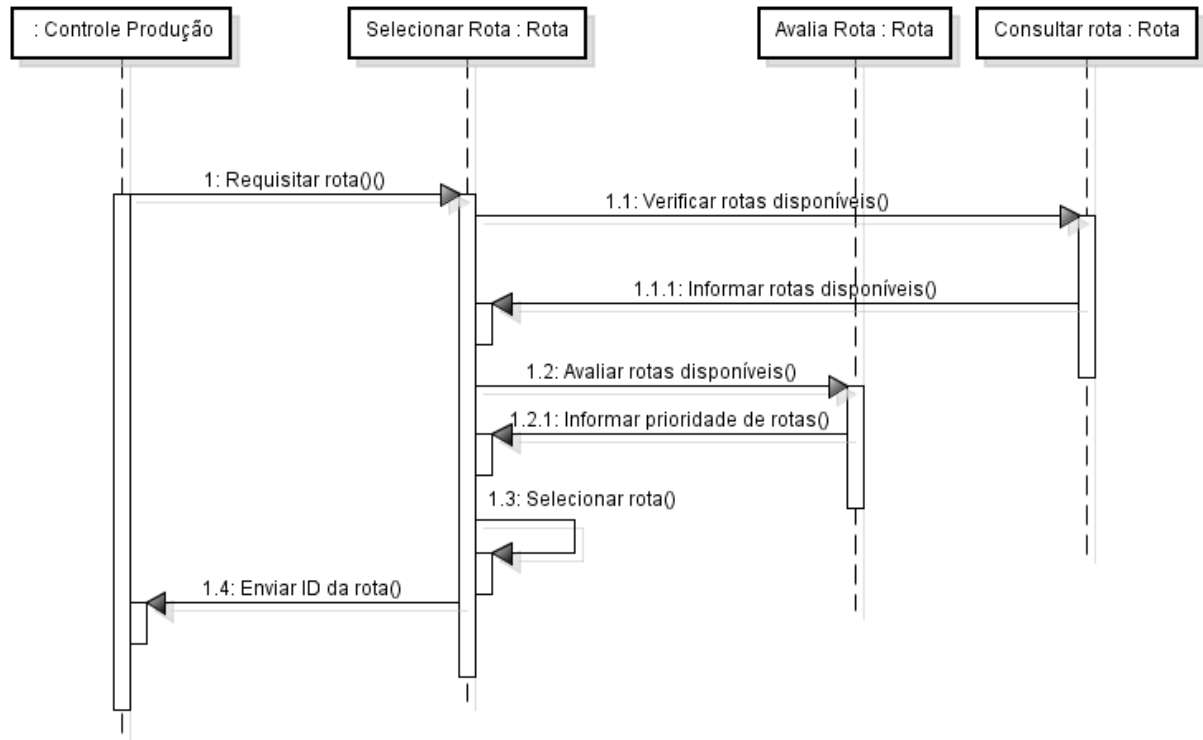
Para a quarta atividade (IRI), o papel antes definido (Definir Rotas) foi substituído por outros 3 papéis (Figura 5.2) para detalhar melhor a atividade do roteamento: Selecionar rota, Avalia Rota e Consultar Rota.



**Figura 5.2. Detalhamento dos papéis da organização Rota**

Neste caso, o papel Selecionar Rota é responsável por requisitar todas as possíveis rotas para o papel Consultar rota e depois encaminhar para serem avaliadas pelo papel Avalia Rota, para somente depois poder selecionar a melhor rota.

Após a identificação dos novos papéis, foi adicionado ao diagrama da atividade SD (atividade 5, fase 1) relacionada a organização Rota, apresentando os novos papéis e as interações entre eles (Figura 5.3)



**Figura 5.3. Interação entre os papéis da organização Rota.**

Neste diagrama, é apresentada a interação da organização Rota com a organização Controle Produção por meio da mensagem de requisição de uma rota. Esta requisição é recebida pelo papel Selecionar rota que verifica todas as rotas disponíveis por meio de mensagem com o papel Consultar rota, que informa todas as rotas disponíveis para aquela tarefa. Após a identificação de todas as rotas disponíveis, se torna necessário avaliá-las enviando uma mensagem para o papel Avalia rota que retorna a avaliação de todas as rotas identificadas. Com a avaliação, o papel Selecionar rota, escolhe a melhor rota e envia sua ID para a organização Controle Produção.

Após a descrição das interações dos papéis da organização Rota, foi necessário adicionar ao diagrama gerado pela atividade RP (atividade 6, fase 1), para descrever as novas tarefas dos papéis identificados. Foi gerado um novo diagrama de sequência conforme apresentado na Figura 5.4.



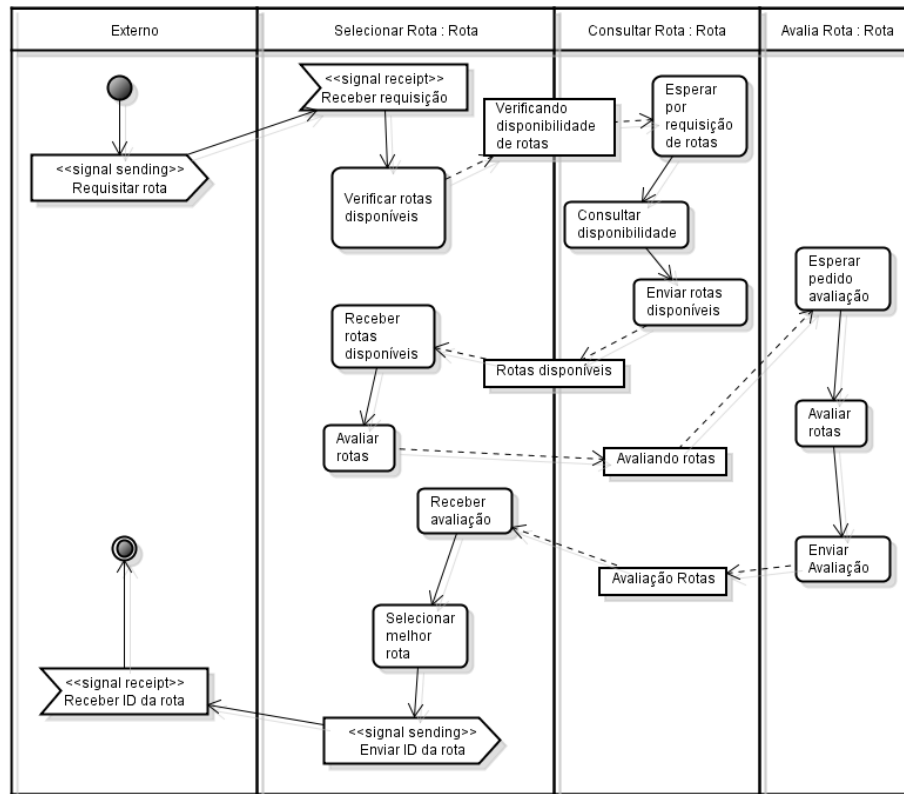


Figura 5.4. Descrição das atividades dos papéis de roteamento.

Neste diagrama é apresentado de maneira mais detalhada a execução dos papéis, por meio de tarefas, e suas interações. Nesta atividade, foi necessário gerar um novo diagrama pois o diagrama gerado na proposta representava apenas um papel mais genérico da organização Rota. Na última atividade da primeira fase, foram apenas atualizados os novos papéis, porém não foram identificadas novas capacidades, permanecendo com o diagrama já proposto anteriormente.

Já para a segunda fase do ASPECS, na SOD (atividade 1, fase 2), foi acrescentado um novo conceito conforme apresentado na Figura 5.5

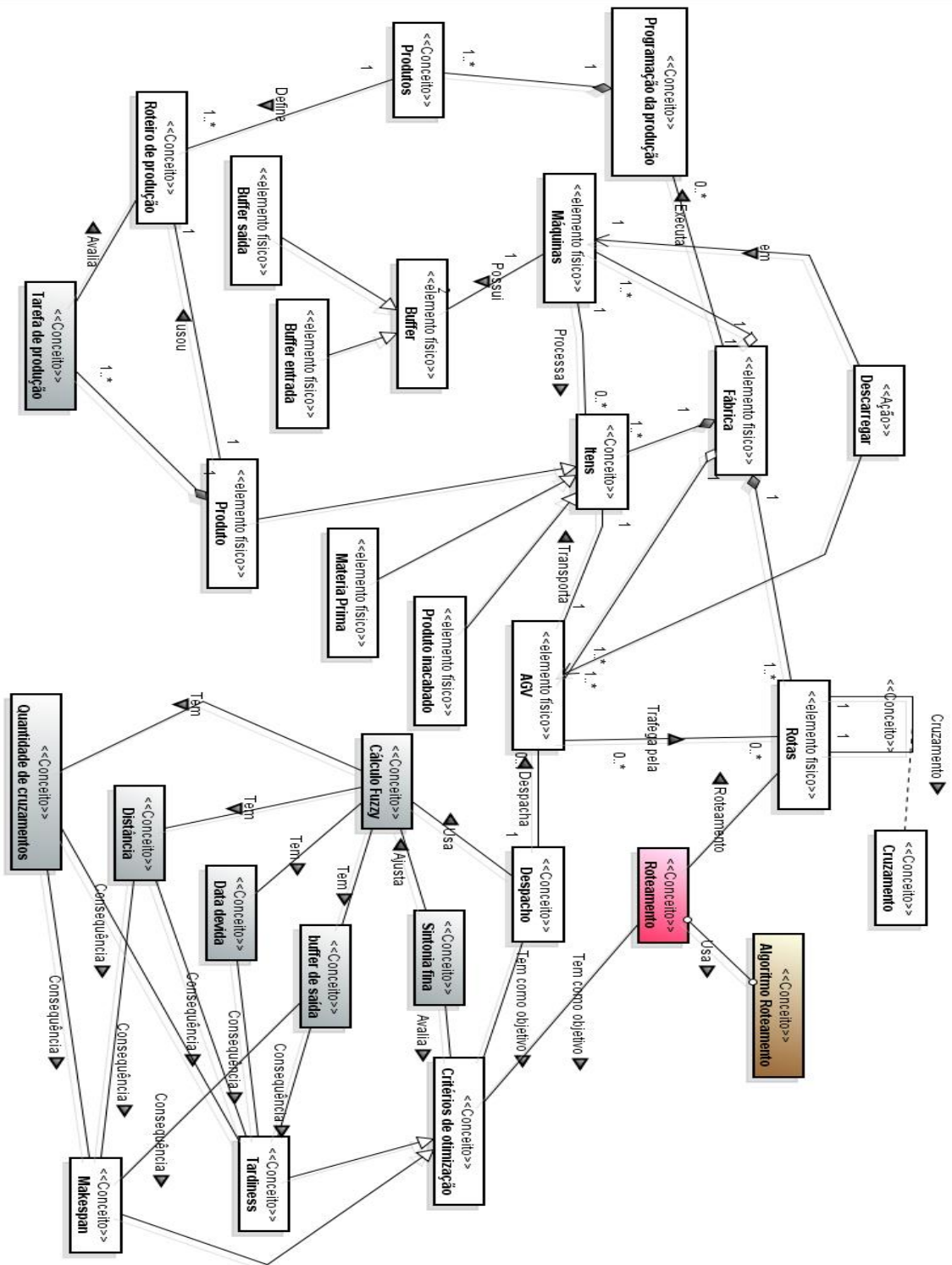


Figura 5.5. Descrição da ontologia da solução de roteamento.

Em marrom o conceito de Algoritmo Roteamento, foi acrescentado ao modelo para indicar que o Roteamento irá usar um algoritmo de roteamento específico.

Após essa atividade, foi avaliado o projeto de agentes (atividades 2 e 3 da fase 2) e não foi realizada nenhuma alteração por se tratar de apenas um detalhamento do funcionamento da organização Rota.

Para as atividades que compõem o projeto de organizações, foram identificadas as comunicações entre os novos papéis e foi adicionado ao diagrama seguindo a mesma lógica do diagrama da proposta (atividade 4, fase 2). Depois foi detalhado o comportamento dos papéis por meio do diagrama de estado de máquinas (atividade 5, fase 2) conforme apresentado na Figura 5.6.

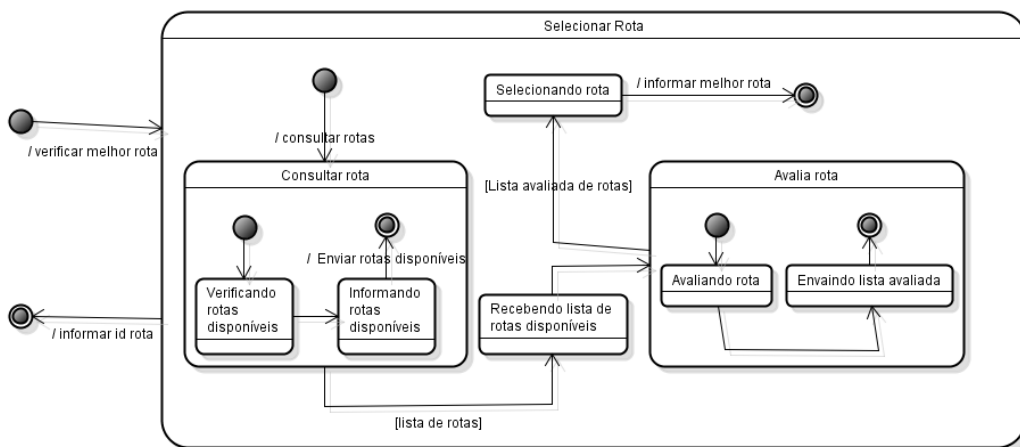


Figura 5.6. Descrição do comportamento dos papéis para o roteamento.

Neste diagrama é detalhado o comportamento de todos os papéis inseridos para representar o controle do roteamento. Para a atividade PD (atividade 6, fase 2) não foi inserido nenhum novo tipo de comunicação além dos padrões FIPA, portanto, assim como na modelagem proposta, não foi gerado nenhum diagrama.

Na ODD (atividade 7, fase 2), foram acrescentadas as dependências relacionadas ao papel Avalia Rota (Figura 5.7).

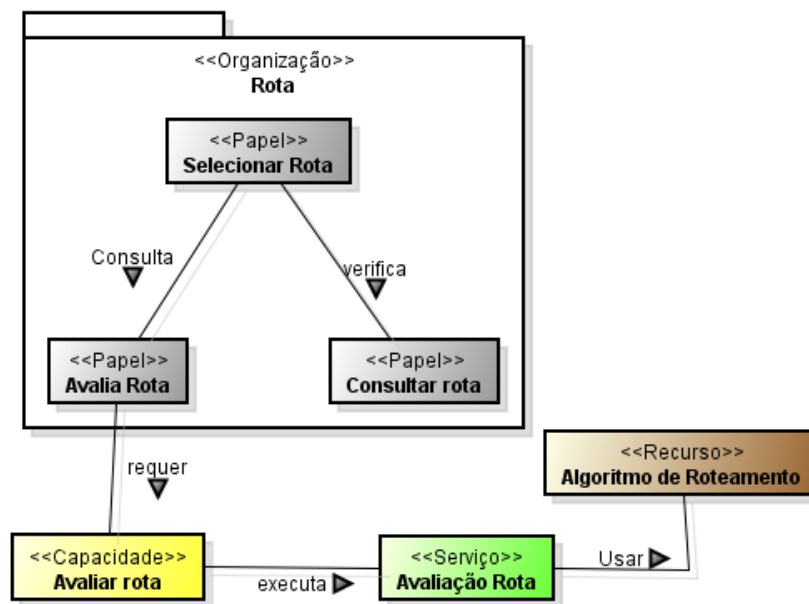


Figura 5.7. Descrição das dependências da organização para acrescentar o roteamento.

Nesta atividade foi identificado um novo serviço Avalia Rota (em verde) e um novo recurso Algoritmo de Roteamento (em marrom). Ambos estão relacionados à especificação da forma com que será realizado o roteamento.

A próxima atividade, a RCI (atividade 8, fase 2) identifica as dependências entre os papéis, porém não houve nenhuma nova dependência além das identificadas na modelagem da proposta.

Já na atividade 9, é necessário detalhar o comportamento do agente, apresentando sua arquitetura interna, que neste caso foram acrescentados (detalhados) os papéis referentes ao controle do roteamento, indicando o vínculo entre cada um deles, análogo ao realizado na proposta, diferindo apenas na atualização dos papéis relacionados à organização Rota.

Para a HD (atividade 10, fase 2), foi necessário atualizar a holarquia, apresentando os *sub holons* que compõem o *holon* que realiza o papel de roteamento.

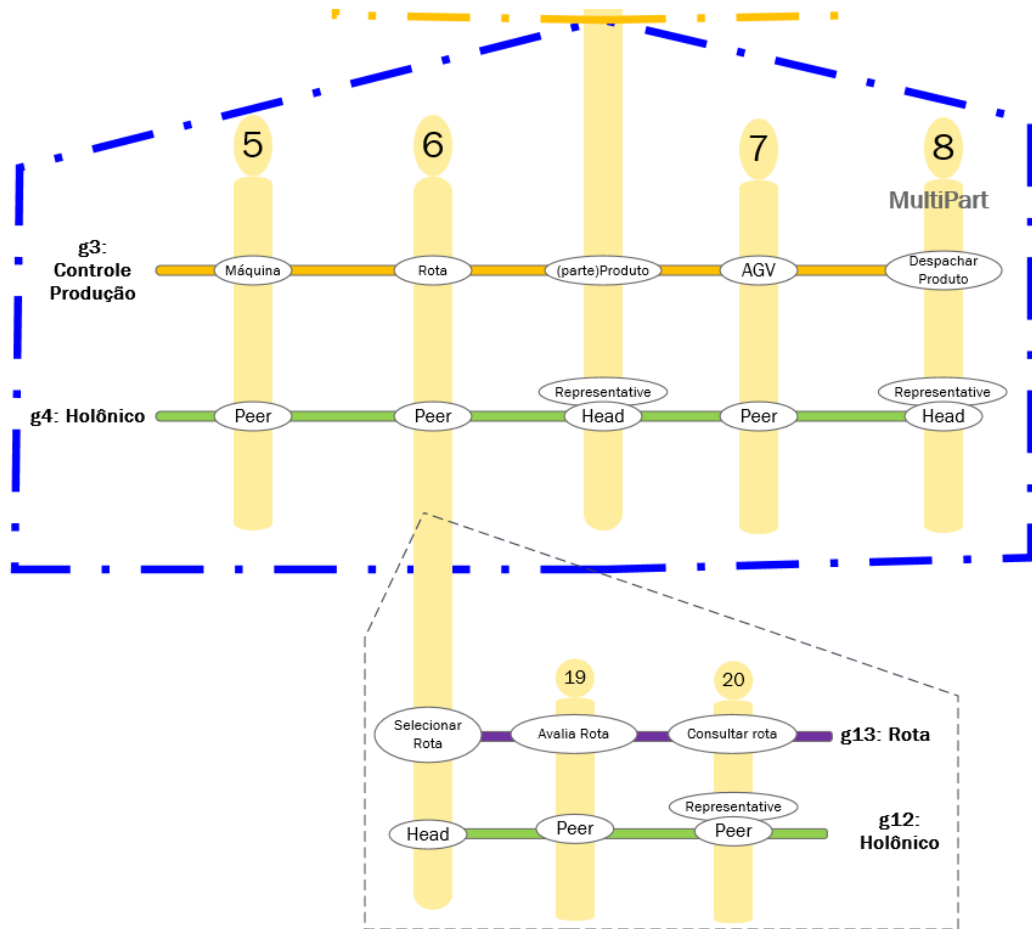


Figura 5.8. Holarquia detalhando o controle do roteamento.

Nesta atividade, foi detalhado o sub *holon* responsável pelo controle do roteamento, apresentando a governança entre eles e quais os papéis eles devem executar. Diferentemente do diagrama apresentado na proposta que apresentava o *holon* 6 como sendo único, este diagrama descreve-o como sendo um super *holon*, composto por outros *holons* responsáveis por realizar o controle do roteamento.

A partir dos novos modelos gerados, é possível gerar o código para implementar esse controle (de maneira análoga a apresentada na seção “4.2. Implementação arquitetura holônica de controle”) seguindo as atividades apresentadas pela terceira fase do ASPECS.

A inserção / detalhamento de um novo elemento na arquitetura proposta, ilustrou a possibilidade de adicionar novos elementos à arquitetura, sem que se tenha um grande retrabalho para adaptá-la aos novos elementos introduzidos. A escolha pela abordagem holônica em conjunto com o método ASPECS, apresentou-se uma opção viável para garantir a generalidade da arquitetura e assim atingir um dos objetivos almejados.

### 5.2.1 Análise qualitativa da adaptação da arquitetura diante de eventos não programados

A análise qualitativa da adaptação da arquitetura diante de eventos não programados foi dividida em dois momentos, o primeiro apresentando uma perturbação interna (quebra de máquina) e a segunda uma perturbação externa (chegada de um novo pedido). Ambas as avaliações permitem verificar o comportamento da arquitetura no ambiente dinâmico da manufatura, confirmando assim se está de acordo com os princípios almejados pela proposta.

Devido ao fato do ambiente ser dinâmico, há uma grande quantidade de eventos não programados durante a execução da produção o que exige que a arquitetura trate isso sem muito impacto negativo ao sistema.

Foram levantados dois tipos de eventos não programados, os ocasionados por problemas internos, como quebra de máquinas, falta de matéria prima, falha no AGV; e os ocasionados por problemas externos, como por exemplo, um novo pedido a ser encaixado na programação.

Para os ocasionados por problemas internos, a arquitetura deverá selecionar, caso exista, um recurso alternativo para que a programação continue sem nenhuma influência negativa. Para isto, a arquitetura deverá ser capaz de gerenciar os recursos, verificando disponibilidade e ao mesmo tempo verificando roteiros / recursos alternativos aos previamente programados.

A arquitetura proposta designa o recurso em tempo real, ou seja, só no momento em que ele for requisitado por uma tarefa. No momento em que for necessário, sua disponibilidade será consultada e caso esteja indisponível a arquitetura irá consultar se existe uma alternativa para ele. Caso exista, o novo recurso será designado, caso contrário, a tarefa entrará em espera até que o recurso indisponível seja liberado (ou volte a ficar ativo).

Esta abordagem se torna possível, pois a arquitetura holônica permite a inclusão de novos elementos de maneira dinâmica, fornecendo assim ao *holon* responsável por produzir o produto, a capacidade de “contratar” novos *holons* que irão contribuir para a produção, conforme apresentado na holarquia da Figura 5.9.

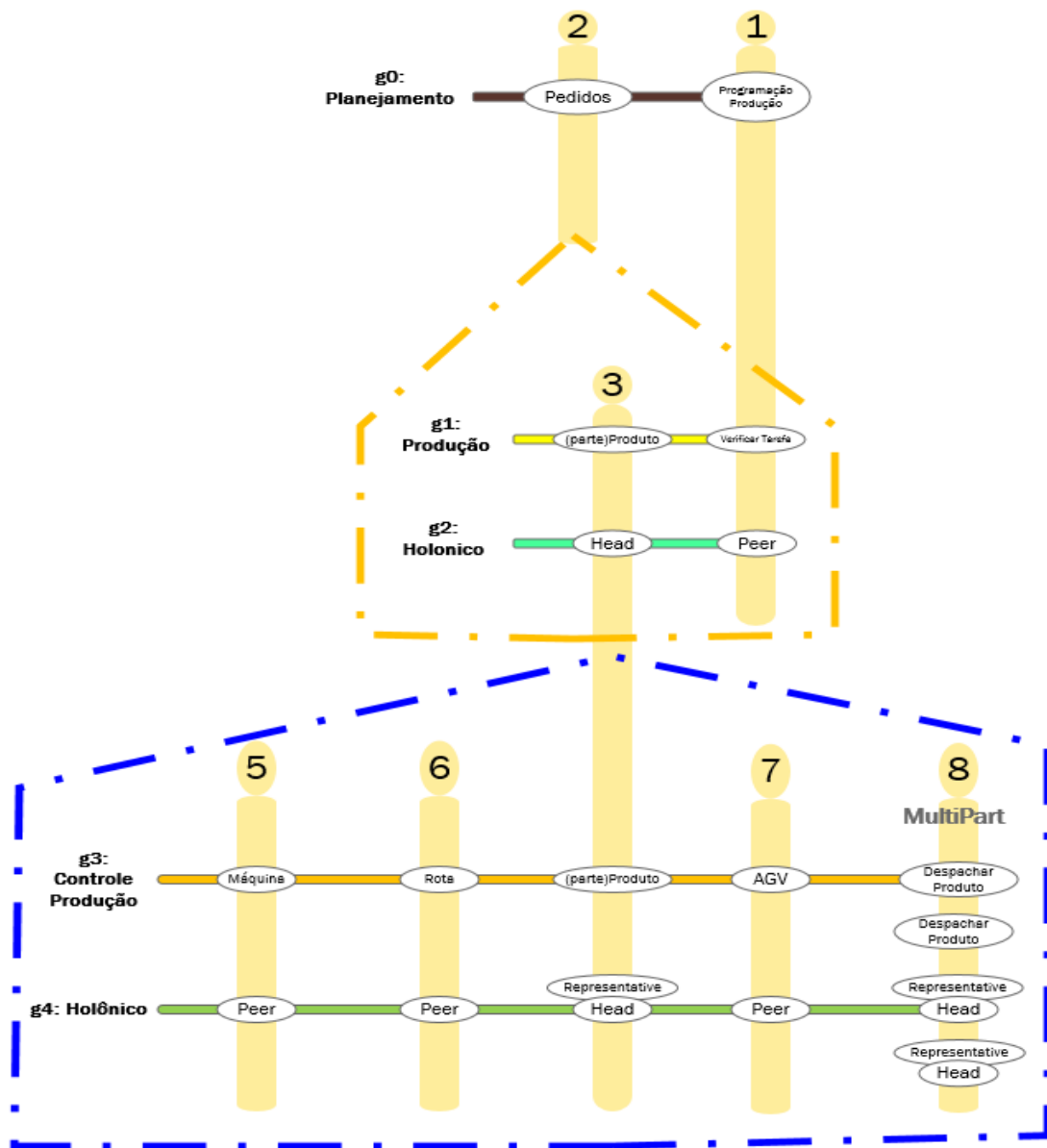
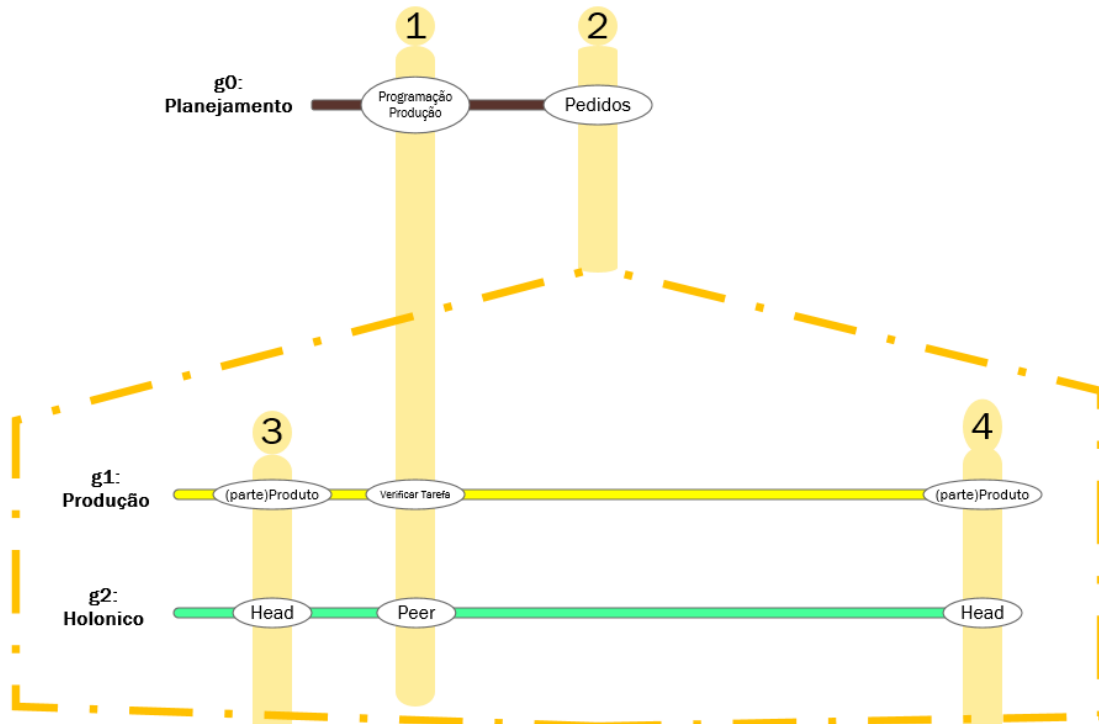


Figura 5.9. Super holon Produto

O holon 3 (super holon relacionado ao produto) é composto pelos holons 3, 5, 6, 7 e 8, responsáveis respectivamente pelo produto, pela máquina, pela rota, pelo AGV e pelo despacho. Este grupo interno ao contexto do super holon (holon 3) é gerado de maneira dinâmica e de acordo com a tarefa a ser executada no determinado momento. Caso algum dos recursos (representados pelos holons) esteja indisponível, os holons responsáveis por formar o grupo (holon 3 e 8) irão verificar algum holon alternativo que esteja disponível para adicionar ao grupo para assim, executar a tarefa.

Para os eventos não programados ocasionados por problemas externos, a arquitetura irá encaixar a tarefa na fila de produtos a serem produzidos. Pelo fato da arquitetura ser holônica,

ela permite a criação dinâmica de *holons*, designando um novo *holon* para a produção deste novo produto. Este *holon* irá disputar os recursos com os outros *holons* (responsáveis pelos outros produtos em produção) e essa disputa será guiada, neste caso, pelo *makespan* e pelo *tardiness*. A Figura 5.10 apresenta a holarquia responsável por atribuir um *holon* a um produto.



**Figura 5.10. Holarquia responsável por distribuir os produtos aos *holons***

Esta holarquia representa a designação dos produtos aos *holons* responsáveis por sua produção. Neste caso, quando surgir um novo pedido, serão criados *holons* que irão executar o papel (parte)Produto (igual ao *holon* 3) que por sua vez irão criar novos grupos para o processamento do produto, análogo ao apresentado na Figura 5.9.

### 5.3 Considerações finais

Neste capítulo foram apresentadas as análises realizadas para verificar o comportamento da arquitetura. Foram realizadas 2 análises qualitativas, sendo elas, diretamente vinculadas às características almejadas pela arquitetura.



A primeira delas é referente à generalidade da arquitetura, que ilustra o comportamento diante da inserção (ou detalhamento) de novos elementos à arquitetura. A segunda, referente à ocorrência de eventos não programados, demonstra o funcionamento da arquitetura diante de distúrbios conhecidos que ocorrem no ambiente.

## Capítulo 6

# CONCLUSÃO

---

### 6.1 Síntese do contexto da proposta e dos objetivos

No âmbito da manufatura, é comum pesquisas por métodos e técnicas para a melhoria da qualidade e aumento da produção. Um problema encontrado é que estas técnicas e métodos devem ser adaptados ao ambiente da manufatura que geralmente são de difícil controle.

Esta dificuldade no controle está associada às constantes perturbações que ocorrem no ambiente, tanto de conotação externa quanto interna. As externas estão relacionadas à pressão que o mercado exerce, como por exemplo, a demanda de mercado; as internas estão mais relacionadas às falhas de recursos, como por exemplo, quebra de máquina. Estes fatores tornam o ambiente de produção ainda mais complexo.

Dentro do ambiente de manufatura, os AMS possuem um sistema de transporte que é responsável por realizar a integração da produção, transportando os produtos de um ponto a outro da fábrica; este fluxo de produção, segundo Groover (2007), tem uma grande influência no custo total de produção.

O sistema de transporte automatizado pode ser composto por AGVs, sendo eles responsáveis por inúmeros benefícios ao sistema de produção, como por exemplo, maior confiabilidade no transporte, fácil integração com outros sistemas automatizados, entre outros. Porém, por serem independentes, necessitam de um sistema de controle para coordená-los.

Para realizar o controle é necessária uma arquitetura de controle, que tradicionalmente pode ser centralizada ou descentralizada. Existem outros tipos de arquitetura que trazem benefícios das duas tradicionais, como por exemplo, a arquitetura holônica.

A arquitetura Holônica apresenta a característica de cada um de seus elementos poderem ser um sistema completo ou uma parte de um sistema, gerando benefícios de facilidade de inclusão de novos elementos, além de permitir uma variação de controle que apresenta os benefícios dos controles centralizados e descentralizados ao mesmo tempo.

Para controlar o sistema de transporte é comum utilizar regras de despacho, podendo ser de um único critério ou multicritério. O despacho depende da configuração de operação da fábrica, fazendo com que uma regra funcione bem para um modo de operação específico e para outro nem tanto. Para diminuir o impacto desta dependência, é comum usar despacho multicritério para conseguir atender um maior número de cenários de maneira mais adequada.

A partir desse ambiente, foi definida a proposta do trabalho como: detalhar uma arquitetura holônica para realizar o despacho de veículos por meio de regras multicritério, contemplando critérios comumente usados em diferentes setores da fábrica (mesmo indicando estratégias conflitantes) e proporcionando um viés de análise global.

## 6.2 Contribuições e delimitações

No capítulo de análise da arquitetura foi apresentado o atendimento do objetivo geral de detalhar uma arquitetura holônica de controle para o despacho de veículos usando regras multicritério, ponderando diferentes requisitos para realizar o atendimento aos critérios que comumente pertencem a diferentes setores da fábrica (mesmo sendo conflitantes)

Para o atendimento do objetivo geral foram tratados alguns objetivos específicos, sendo eles:

- I) Adicionar à arquitetura uma estrutura para contemplar uma técnica que realize o ajuste dinâmico no despacho multicritério, para ponderar o atendimento ao *tardiness* e ao *makespan* ao mesmo tempo em decorrência da saída da produção. Este objetivo foi alcançado e para sua análise foi definida uma regra simples de atualização dos pesos de entrada do sistema *fuzzy* de despacho, o que se mostrou suficiente a ponderar entre o *makespan* e o *tardiness*. Foi observado que esta estrutura permite a troca da forma de atualização da regra de auto ajuste por uma regra detalhada que permita ponderar entre os critérios de otimização.

- II) O método escolhido para o desenvolvimento do HMS se mostrou suficiente para atender todo o ciclo de vida do projeto (desde a parte de levantamentos de requisitos até a implementação e implantação). Pelo fato do método ASPECS ser genérico e não específico para a manufatura, foram identificados alguns pontos não contemplados em sua cobertura, em relação ao o que considerar para realizar a atividade. Um exemplo disto é que na IRI (atividade 3 da primeira fase) observou-se a necessidade de incluir os recursos físicos da fábrica (como por exemplo, máquinas, AGVs, rotas, entre outros) como sendo papéis para que fosse possível a criação da holarquia na HD (atividade 10 da segunda fase).
- III) Analisar a arquitetura holônica, simulando seu comportamento diante eventos não planejados e verificá-lo diante de um ambiente dinâmico. Foi observado que a arquitetura holônica se adaptou ao novo ambiente quando possível, atingindo assim os benefícios almejados para este trabalho, quando se optou pelos sistemas holônicos.

Como contribuições deste trabalho, pode-se listar:

- Detalhamento de uma arquitetura holônica para controlar o despacho de veículos;
- Uma estrutura que permite o ajuste dinâmico entre as variáveis de otimização selecionadas para orientar o despacho, de acordo com a saída do sistema de produção;
- Definição de regras da arquitetura para lidar com o ambiente dinâmico quando há uma falha na comunicação entre o *holon* que representa o AGV e o *holon* responsável pelo despacho; e outra relacionada à saturação do tempo de reposta de uma decisão (neste caso sobre decisões de despacho) aumentando assim a autonomia do *holon* para realizar a tarefa de acordo com suas competências (atendendo otimizações locais).

### 6.3 Trabalhos futuros

Durante o desenvolvimento da proposta foram identificados vários pontos a serem estendidos, acrescentados e ou melhorados e que são listados como possíveis trabalhos futuros:

- I) Adição de novos elementos de controle na arquitetura, como por exemplo: controle para o roteamento; controle para realizar a programação reativa. Todos estes controles podem ser adicionados à arquitetura proposta, seguindo o método de desenvolvimento citado, adicionando novos *holons* específicos para realizar cada um desses pontos levantados.
- II) Criação e adição de um módulo de despacho de veículos otimizado, aproveitando toda a arquitetura já proposta.
- III) Análise de impacto de outros critérios de otimização e uma regra de auto ajuste mais refinada para realizar o ajuste fino de acordo com as estratégias de negócio da fábrica. Este ajuste fino pode ser realizado por meio de um algoritmo genético para otimizar os pesos de entrada de cada uma das variáveis do sistema *fuzzy*; ou pode ser um sistema baseado em casos que realiza um estudo no histórico de produção e ajusta as entradas.
- IV) Adição de uma camada superior na holarquia que seja responsável por verificar um novo critério de otimização e que auxilie o despacho com uma nova abordagem, como por exemplo, um *holon* que verifique o nível de ocupação de máquinas, para realizar um balanceamento de cargas e com isso auxiliar na definição da tarefa a ser atendida.
- V) Adição de um *holon* responsável por verificar a carga de ocupação das máquinas e caso esteja com um nível baixo, o mesmo verifica se é melhor realizar um balanceamento das cargas ou realizar a reconfiguração. Esta análise pode ser por exemplo baseada no tempo em que as máquinas estão com um nível baixo de ocupação, caso não seja algo que perdure, realiza o balanceamento de cargas, caso contrário realiza a reconfiguração.
- VI) Adaptação do método ASPECS, para trabalhar especificamente com a manufatura, incluindo ou adaptando os termos usados para representar melhor o ambiente trabalhado. Adição de preocupações ou necessidades identificadas em cada passo, geradas pelo domínio da manufatura.

# REFERÊNCIAS

---

ABDI, M. R.; LABIB, A. W. Performance evaluation of reconfigurable manufacturing systems via holonic architecture and the analytic network process. **International Journal of Production Research**, v. 49, n. 5, p. 1319 – 1335, 2011.

ADAM, E.; BERGER, T.; SALLEZ, Y.; TRENTESAUX, D. Role-based manufacturing control in a holonic multi-agent system. **International Journal of Production Research**, v. 49, n. 5, p. 1455 – 1468, 2011.

ALVAREZ, R. D. R.; ANTUNES JR., J. A. V. *Takt-time*: conceitos e contextualização dentro do Sistema Toyota de Produção. **Gestão & Produção**, v. 8, p. 1-18, 2001.

BABICEANU, R. F.; CHEN, F. F. Development and applications of holonic manufacturing systems: a survey. **Journal of Intelligent Manufacturing**, v. 17, n. 1, p. 111 – 131, 2006.

BAL, M.; HASHEMIPOUR, M. Implementation of holonic scheduling and control in flow-line manufacturing systems: die-casting case study. **Production Planning & Control**, v. 22, n. 2, p. 108 – 123, 2011

BARBOSA, J.; LEITÃO, P.; TRENTESAUX, D.; ADAM, E. Enhancing ADACOR with biology insights towards reconfigurable manufacturing systems. In: IECON, 37<sup>th</sup> Annual Conference of the IEEE Industrial Electronics Society, 2011, Melbourne, Australia. **Proceedings** of Piscataway: IEEE Computer Society, 2011, p. 2746 – 2751.

BARBOSA, J.; LEITÃO, P.; ADAM, E.; TRENTESAUX, D. Self-organized Holonic Multi-agent Manufacturing System: the behavioral perspective. **Computers in Industry**, v. 66, p. 99 – 111, 2015.

BARTHOLDI III, J. J.; PLATZMAN, L. K., Decentralized control of automated guided vehicles on a simple loop. **IIE Transactions (Institute of Industrial Engineers)**, v. 21, n. 1, p. 76 – 81, 1989.

BENICASA, A. X.; MORANDIN JR, O.; KATO, E. R. R. Reactive *Fuzzy* dispatching rule for automated guided vehicles. In: IEEE International Conference on Systems Man and Cybernetics, 2003, Washington, USA. **Proceedings** of Piscataway: IEEE Inc., 2003, p. 4375-4380.

BERMAN, S.; EDAN, Y. Decentralized autonomous AGV system for material handling. **International Journal of Production Research**, v. 40, n. 15, p. 3995 – 4006, 2002.

BONGAERTS, L. Integration of Scheduling and Control in Holonic Manufacturing Systems. 1998. 223 f. Tese (Doutorado em Engenharia Mecânica) – Departamento de Engenharia Mecânica, Universidade Católica de Leuven, Leuven, Bélgica, 1998.

BONGAERTS, L.; VALCKENAERS, P.; VAN BRUSSEL, H.; WYNS, J. Schedule Execution for a Holonic Shop Floor Control System. In: Proceedings of the 1998 IEEE International Conference on Robotics and Automation (Part 1), 1998, Leuven, Belgium. **Proceedings of Piscataway: IEEE**, 1998, p. 2747 – 2752.

BORANGIU, T.; RAILEANU, S.; BERGER, T.; TRENTESAUX, D. Switching mode control strategy in manufacturing execution systems. **International Journal of Production Research**, v. 53, n. 7, 2015.

BOTTI, V.; GIRET A. **ANEMONA: A multi-agent methodology for Holonic Manufacturing Systems**. Springer, 2008, 230 p.

CARIDÁ, V. **Modelo Adaptativo Fuzzy Genético, Preditivo e Multiobjetivo para Despacho de AGVs**. 2011. 123f. Dissertação (Mestrado em computação) – Departamento de computação, Universidade Federal de São Carlos, São Carlos, 2011.

CHENG, F.; CHANG, C.; WU, S. Development of holonic manufacturing execution systems. **Journal of Intelligent Manufacturing**, n. 15, p. 253 – 267, 2004.

CHIBA, R.; ARAI, T., OTA, J. Integrated design for automated guided vehicle systems using cooperative co-evolution. **Advanced Robotics**, v. 24, n. 1-2, p. 25 – 45, 2010.

COSENTINO, M.; GAUD, N.; HILAIRE, V.; GALLAND, S.; KOUKAM, A. ASPECS: an agent-oriented software process for engineering complex systems. **Autonomous Agents and Multi-Agent Systems**. v. 20, n. 2, p. 260 – 304, 2010a.

COSENTINO, M.; GALLAND, S.; GAUD, N.; HILAIRE, V.; KOUKAM, A. A Glimpse of the ASPECS Process documented with the FIPA DPDF Template. In: Multi-Agent Logics, Languages, and Organisations Federated Workshops, 2010b, Lyon, França. Proceedings of CEUR-WS, 2010b.

DANTZING, G. B.; RAMSER, J. H. The truck dispatching problem. **Management**, v. 6, n. 1, p. 80 – 91, 1959.

DE WOLF, T.; SAMAIEY, G.; HOLVOET, T.; ROOSE, D. Decentralised Autonomic Computing: Analysing Self-Organising Emergent Behaviour using Advanced Numerical Methods. In: International Conference on Autonomic Computing, 2005, Seattle, EUA. **Proceedings of Piscataway: IEE Computer Society**, 2005, p. 52 – 63.

DEMANGE, J. **Un modèle d'environnement pour la simulation multiniveau – Application à la simulation de foules**. 2012. 248f. Tese (Doutorado em computação) – Universidade de Belfrot – Montbeliard, 2012, França.

EGBELU P. J. Pull versus push strategy for automated guided vehicle load movement in a batch manufacturing system. **Journal of Manufacturing Systems**, v. 6, n. 3, p. 209 – 221, 1987.

EGBELU, P. J.; TANCHOCO, J. M. A. A characterization of automatic guided vehicle dispatching rules. **International Journal of Production Research**, v. 22, n. 3, p. 359 – 374, 1984.

FAZLOLLAHTABAR, H.; REZAIE, B.; KALANTARI, H. Mathematical programming approach to optimize material flow in an AGV-based flexible jobshop manufacturing system with performance analysis. **International Journal of Advanced Manufacturing Technology**, v. 51, n. 9-12, p. 1149 – 1158, 2010.

FERBER, J. **Multi-agent systems. An introduction to distributed artificial intelligence**. Inglaterra: Addison Wesley, 1999, 509 p.

FISCHER, K. Agent-based design of holonic manufacturing systems. **Robotics and Autonomous Systems**, v. 27, n. 1, p. 3 – 13, 1999.

GALLAND, S.; GAUD, N.; DEMANGE, J.; KOUKAM, A. Environment Model for Multiagent-Based Simulation of 3D Urban Systems. **In: Proceedings of 7<sup>th</sup> European Workshop on Multiagent Systemas**, Ayia Napa, Chipre, paper 36, 2009.

GASSER, L. Boundaries, identity, and aggregation. Plurality issues in multi-agent systems. **In: Proceedings of the European Workshop on Modelling Autonomous Agents in a Multi-Agent World**, 1991, Kaiserslautern, Alemanha. **Proceedings** of Elsevier Science Publ BV, 2002, p. 199

GERBER, C., SIEKMANN, J. H., AND VIERKE, G. **Holonic multi-agent systems**. Technical Report DFKI-RR-99-03, Deutsches Forschungszentrum für Künstliche Intelligenz - GmbH, Postfach 20 80, 67608 Kaiserslautern, FRG, 1999.

GIRET, A.; BOTTI, V. *Holons* and agents. **Journal of Intelligent Manufacturing**, v. 15, n.5, p. 645 – 659, 2004

GIRET, A.; BOTTI, V. Analysis and design of holonic manufacturing systems. **In: International Conference on Production Research**, 2005, Palermo, Itália. **Proceedings** of 18th International Conference on Production Research, p. 1 – 10, 2005.



GIRET, A.; BOTTI, V. Engineering Holonic Manufacturing Systems. **Computers in Industry**, v. 60, n. 6, p. 428 – 440, 2009.

GIUNCHIGLIA, F.; MYLOPOULOS, J.; PERINI, A. The Tropos Software Development Methodology: Processes, Models and Diagrams. In: Agent-Oriented Software Engineering III. **Proceedings of Springer Berlin Heidelberg**, 2003. p. 162-173. 2002

GROOVER, M. P. **Automation production systems and computer-integrated manufacturing**. 3<sup>a</sup> ed. Englewood Cliffs: Prentice Hall, 2007, 856 p.

GUAN, X.; DAI, X. Deadlock-free multi-attribute dispatching method for AGV systems. **International Journal of Advanced Manufacturing Technology**, v. 45, n. 5-6, p. 603 – 615, 2009.

HAN, M. H.; MCGINNIS, L. F. Control material handling transporter in automated manufacturing. **IIE Transactions (Institute of Industrial Engineers)**, v.21, n. 2, p. 184 – 190, 1989.

HERRERO-PÉREZ, D.; MATÍNEZ-BARBERÁ, H. Decentralized coordination of autonomous AGVs in flexible manufacturing systems. In: International Conference on Intelligent Robots and Systems, 2008, Nice, França. **Proceedings of Piscataway: The IEEE Computer Society**, 2008. p. 3674 – 3679.

HIDEHIKO, Y. Addition and deletion of agent memories to realize autonomous decentralized manufacturing system. In: International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, 2008, Wroclaw, Poland. **Proceedings of Heidelberg: Springer Verlag, Heidelberg, Germany**, 2008, p. 738 – 748.

HO, Y.; LIAO, T. Zone design and control for vehicle collision prevention and load balancing in a zone control AGV system. **Computers and Industrial Engineering**, v. 56, n. 1, p. 417 – 432, 2009.

HO, Y.; LIU, H. The performance of load-selection rules and pickup-dispatching rules for multiple-load AGVs. **Journal of Manufacturing Systems**, v. 28, n. 1, p. 1 – 10, 2009.

HOSHINO, T.; TSUBOI, K.; YAMANAKA, K.; HAMAMATSU, Y. Operation of vehicles in AGV system and method for analyzing operation. **Electrical Engineering in Japan**, v. 178, n. 1, p. 56 – 68, 2012.

HSIEH, F.-S. Model and control holonic manufacturing systems based on fusion of contract nets and Petri nets. **Automatica**, v. 40, n. 1, p. 51 – 57, 2004.

- HSUEH, C. A simulation study of a bi-directional load-exchangeable automated guided vehicle system. **Computers & Industrial Engineering**, v. 58, n. 4, p. 594 – 601, 2010.
- IWAMURA, K.; MAYUMI, N.; TANIMIZU, Y.; SUGIMURA, N. A study on real-time scheduling for holonic manufacturing systems - Determination of utility values based on multi-agent reinforcement learning. In: 4th International Conference on Industrial Applications of Holonic and Multi-Agent Systems, 2009, Linz, Austria. **Proceedings of Heidelberg**: Springer Verlag, 2009, p. 135 – 144.
- JANA, T. K.; BIRAGI, B.; PAUL, S.; SARKAR, B., SAHA, J. Dynamic Scheduling execution in an agent based holonic manufacturing. **Journal of Manufacturing Systems**, v. 32, n. 4, 801-816, 2013.
- JOVANOVIC, M.; ZUPAN, S. STARBEK, M.; PREBIL, I. Virtual approach to holonic control of the tyre-manufacturing system. **Journal of Manufacturing Systems**, v. 33, n. 1, p. 116 – 128, 2014.
- KESEN, S. E.; BAYKOÇ, O. F. Simulation of automated guided vehicle (AGV) systems based on just-in-time (JIT) philosophy in a job-shop environment. **Simulation Modelling Practice and Theory**, v. 15, n. 3, p. 272 – 284, 2007.
- KIM, C. W.; TANCHOCO, J. M. A.; KOO, P. H. AGV dispatching based on workload balancing. **International Journal of Production Research**, v. 37, n. 17, p. 4053 – 4066, 1999.
- KIM, S. H.; HWANG, H. Adaptive dispatching algorithm for automated guided vehicles based on an evolutionary process. **International Journal of Production Economics**, v. 60, p. 465 – 472, 1999.
- KIZIL, M.; ÖZBAYRAK, M.; PAPADOPOULOU, T. C. Evaluation of dispatching rules for cellular manufacturing. **International Journal of Advanced Manufacturing Technology**, v. 28, n. 9, p. 985 – 992, 2006.
- KOESTLER, A. **O fantasma da máquina**. Rio de Janeiro: Zahar, 1969, 421 p.
- LAKATOS, E. M.; MARCONI, M. A. **Fundamentos de metodologia científica**. 5 ed. São Paulo: Atlas, 2003. 312 p.
- LE-ANH, T.; DE KOSTER, M. B. M. A review of design and control of automated guided vehicle systems. **European Journal of Operational Research**, v. 171, n.1 p. 1-23, 2006.
- LEITÃO, P. A holonic disturbance management architecture for flexible manufacturing systems. **International Journal of Production Research**, v. 49, n. 5, p. 1269 – 1284, 2011.

LEITÃO, P. Holonic Rationale and Self-organization on Design of Complex Evolvable Systems. In: Marik, V., *et al.* **Holonic and Multi-Agent Systems for Manufacturing**. Springer Berlin Heidelberg, 2009. p. 1- 12.

LEITÃO, P. RESTIVO, F. Implementation of a holonic control system in a flexible manufacturing system. **IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews**, v. 38, n. 5, p. 699 – 709, 2008.

LEITÃO, P. COLOMBO, A. W.; RESTIVO, F. Formal specification of ADACOR holonic control system: Coordination models. In: 44th IEEE Conference on Decision and Control, and the European Control Conference, 2005, Sevilha, Espanha. **Proceedings** of Piscataway: Inst. of Elec. and Elec. Eng. Computer Society, 2005, p. 2137 – 2142.

LEITÃO, P.; RESTIVO, F. ADACOR: A holonic architecture for agile and adaptive manufacturing control. **Computers in Industry**, v. 57, n. 2, p. 121 – 130, 2006.

LIN, L; SHIN, S. W.; GEN, M.; HWANG, H. Network model and effective evolutionary approach for AGV dispatching in manufacturing system. **Journal of Intelligent Manufacturing**, v. 17, n. 4, p. 465 – 477, 2006.

LIU, S.; GRUVER, W. A.; KOTAK, D.; BARDI, S. Holonic manufacturing system for distributed control of automated guided vehicles. In: IEEE International Conference on Systems, Man and Cybernetics, 2000, Nashville, EUA. **Proceedings** of Piscataway: IEEE, 2000, p. 1727 – 1732.

LIU, Q.; LIU D.; HUANG, D. Methods to support *holon* aggregation in holonic manufacturing system. In: International Symposium on Computational Intelligence an Design, 2008, Wuhan, China. **Proceedings** of Washington: IEEE Computer Society, 2008, p.215 – 219.

MARTINEZ-BARBERA, H.; HERRERO-PEREZ, D. Development of a flexible AGV for flexible manufacturing systems. **Industrial Robot**, v. 37, n. 5, p. 459 – 468, 2010.

MONTOYA-TORRES, J. R., BELLO, G. O. Integer linear programming formulation of the vehicle positioning problem in automated manufacturing systems. **Journal of Intelligent Manufacturing**, v. 22, n. 4, p. 545 – 552, 2011.

MORANDIN JR., O.; CASTRO, P. A. D.; KATO, E. R. R.; CAMARGO, H. A. A genetic *Fuzzy* system for defining a reactive dispatching rule for AGVs. In: IEEE International Conference on Systems, Man and Cybernetics, 2006, Taipei, Taiwan. **Proceedings** of New York: IEEE Inc., 2006, p. 56 – 61.

MORANDIN JR, O. ; SANCHES, D. S. ; DERIZ, A. C. ; KATO, E. R. R. ; TSUNAKI, R. H. . An Adaptive Genetic Algorithm Based Approach for Production Reactive Scheduling of

Manufacturing Systems. In: IECON 2008 – 34<sup>th</sup> Annual Conference of IEEE Industrial Electronics Society, 2008, Orlando, Florida. **Proceedings** Piscataway: IEEE Computer Society, 2008. p. 1461 – 1466.

MORANDIN, O.; CARIDA, V. F.; KATO, E. R. R.; TUMA, C. C. M. Adaptive genetic *Fuzzy*, predictive and multiobjective approach for AGVs dispatching. In: IECON, 37<sup>th</sup> Annual Conference on IEEE Industrial Electronics Society, 2011, Melbourne, Australia. **Proceedings** of Piscataway: IEEE Computer Society, 2011, p. 2317 – 2322.

NASO, D.; TURCHIANO, B. Multicriteria meta-heuristics for AGV dispatching control based on computational intelligence. **IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics**, v. 35, n. 2, p. 208-226, 2005.

PANETTO, H.; MOLINA, A. Enterprise integration and interoperability in manufacturing systems: Trends and issues. **Computers in Industry**, v. 59, n. 7, p. 641 – 646, 2008.

PHILIPS, J.; Saint Germain, B.; Belle, J. V.; Valckenaers P. Traffic Radar: A Holonic Traffic Coordination System Using PROSA++ and D-MAS. In: 6th International Conference on Industrial Applications of Holonic and Multi-Agent Systems, In: 2013, Praga, Republica Tcheca. **Proceedings** of Heidelberg: Springer Berlin, 2013, p. 163-174.

POSLAND, S. Specifying protocols for multi-agent system interaction. **ACM Trans. Autonom. Adapt. Syst.**, v. 2, n. 4, Article Number 15, 2007.

REY, G. Z.; PACH, C.; AISSANI, N.; BEKRAR, A.; BERGER, T.; TRENTSEAUX, D. The control of myopic behavior in semi-Heterarchical production systems: A holonic framework. **Engineering Applications of Artificial Intelligence**, v. 6, n. 2, 2013.

RODRIGUEZ, S. A. **From analysis to design of holonic multi-agent systems: A framework, methodological guidelines and applications**. 2005, 233 f. Tese (Doutorado em Informática) – Universidade Tecnológica Belfort-Montbéliard, Belfort, França, 2005

RUSSEL, S.; NORVIG, P. **Artificial Intelligence: A Modern Approach**. 3 ed. Prentice Hall, 2009, 1152 p.

SANCHES, D. S. ; MORANDIN JR, O. ; MUNIZ, B. D. ; KATO, E. R. R. . Modeling Strategy by Adaptive Genetic Algorithm for Production Reactive Scheduling with Simultaneous use of Machines and AGVs. In: SMC 2009 - IEEE International Conference on Systems, Man and Cybernetics, 2009, San Antonio. **Proceedings** of Piscataway: IEEE Computer Society, 2009. p. 249 – 254.

SILVA, E. D.; MENEZES, E. M. **Metodologia da Pesquisa e Elaboração de Dissertação**. 3<sup>a</sup> ed. Florianópolis: UFSC, 2005. 138 p.

SMOLICK-ROCAK, N.; BOGDAN, S.; KOVACIC, Z.; PETROVIC, T. Time windows based dynamic routing in multi-AGV system. **IEEE Transactions on Automation Science and Engineering**, v. 7, n. 1, p. 151 – 155, 2010.

SONG, S.; LI, A.; XU, L. AGV Dispatching Strategy Based on Theory of Constraints. In: Robotics, Automation and Mechatronics, 2008, Chengdu, China. **Proceedings of Piscataway: IEEE Computer Society**, 2008, p. 922 – 925.

TAGHABONI, F. **Scheduling and control of manufacturing systems with critical material handling**. 1989. 201 f. Tese (Doutorado em Engenharia Industrial) – Departamento de Engenharia Industrial, Purdue University, West Lafayette, USA, 1989.

TAN, K. K.; TANG, K. Z. Simulation of an evolutionary tuned *Fuzzy* dispatching system for automated guided vehicles. In: Winter Simulation Conference, 2000, Orlando, Florida, USA. **Proceedings of Piscataway: IEEE**, 2000, p. 1339-1343.

UMASHANKAR, N.; KARTHIK, V. N. Multi-criteria intelligent dispatching control of automated guided vehicles in FMS. In: IEEE Conference on Cybernetics and Intelligent Systems, 2006, Bangkok, Thailand. **Proceedings of Piscataway: IEEE Computer Society**, 2006. n. 4017851.

VAN BRUSSEL, H. Holonic Manufacturing Systems. In: LAPERRIÈRE, L.; REINHART, G. **CIRP Encyclopedia of Production Engineering**. New York: Springer, 2014. p. 654-659.

VAN BRUSSEL, H.; WYNS, J.; VALCKENAERS, P.; BONGAERTS, L.; PEETERS, P. Reference Architecture for Holonic Manufacturing Systems: PROSA. **Computers in Industry**, v. 37, n. 3, p. 255 – 274, 1998.

VAN BRUSSEL, H.; VALCKENAERS, P.; BONGAERTS, L.; WYNS, J. **Architectural and system design issues in holonic manufacturing systems**. In: 3rd IFAC Workshop on Intelligent Manufacturing Systems, 1995, Bucareste, Romênia.

VERSTEEGH, F.; SALIDO, M. A.; GIRET, A. A holonic architecture for the global road transportation system. **Journal of Intelligent Manufacturing**. v. 21, n. 1 p. 133 – 144, 2010.

VIS, I. F. A. Survey of research in the design and control of automated guided vehicle systems. **European Journal of Operational Research**, v. 170, n. 3, p. 677 – 709, 2006.

WANG, K.; CHOI, S. H. A holonic approach to flexible flow shop scheduling under stochastic processing times. **Computers and Operations Research**, v. 43, n. 1, p. 157 – 168, 2014.

WOOLDRIDGE, M. **An introduction to MultiAgent Systems**. Londres: John Wiley & Sons Ltd., 2002. 349 p.

WOOLDRIDGE, M. JENNINGS, N. Intelligent Agents: Theory and Practice. **Knowledge Engineering Review**. v. 10, p. 115 – 152, 1995.

WYNS, J. **Reference Architecture for Holonic Manufacturing**. 1999. 219 f. Tese (Doutorado em Engenharia Mecânica) – Departamento de Engenharia Mecânica, Universidade Católica de Leuven, Leuven, Bélgica, 1999.

ZHAO, F.; ZHANG, Q.; WANG, L. Task allocation and evaluation model for Holonic manufacturing system based on multi-agent system. In: 7th World Congress on Intelligent Control and Automation, 2008, Chongqing, China. **Proceedings** of Piscataway: Institute of Electrical and Electronics Engineers Inc., 2008. n. 4593720.

## Apêndice A\*

# MODELAGEM DA ARQUITETURA HOLÔNICA DE DESPACHO

---

---

Neste apêndice são abordados todos os modelos gerados durante este projeto para a composição da arquitetura holônica de despacho proposta. Serão apresentados seguindo as fases e atividades do ASPECS, método seguido para o desenvolvimento deste trabalho.

O apêndice será dividido em 2 seções, sendo elas referentes as 2 fases de modelagem do ASPECS: I) Requisitos do sistema e II) Sociedade de agentes. Em cada uma das seções, serão divididas em subseções de acordo com a quantidade de atividades presentes na fase.

### **A.1 Requisitos do sistema**

Nesta atividade foram modelados sob diferentes óticas, o domínio do problema, começando com os casos de uso, até se ter uma noção geral de papéis, organizações e capacidades.

#### **A.1.1 Descrição dos requisitos do domínio (DRD - *Domain Requirements***

##### ***Description)***

Nesta atividade foram gerados os casos de uso, assim como os seus diagramas. O documento de requisitos é apresentado a seguir.

##### **Requisitos Funcionais**

<<funcionalidades presentes no sistema – identificadas a partir da descrição do cenário do problema>>

- (RF01) Despachar produto;
- (RF02) Processar programação da produção;
- (RF03.1) Selecionar máquina;
- (RF03.2) Verificar disponibilidade de máquina;
- (RF03.3) Verificar roteiro alternativo de produção;
- (RF04.1) Selecionar AGV;
- (RF04.2) Gerar lista priorizada de AGVs;
- (RF04.3) Verificar AGVs disponíveis;
- (RF04.4) Calcular prioridade de AGV;
- (RF05) Definir Rotas;
- (RF06) Transportar produto;
- (RF07) Processar produto.
- (RF08) Atualizar produto

### **Requisitos não funcionais**

- Até o momento não foram identificados requisitos não funcionais

### **Descrição dos casos de uso**

- **Identificador:** RF01.
  - **Nome:** Despachar produto.
  - **Tipo:** Abstrato
  - **Descrição:** É selecionada a máquina de destino para processar o produto de acordo com a programação da produção (caso não seja possível é verificado um roteiro alternativo); Em seguida é selecionado um AGV para transportar este produto até a máquina definida, seguindo uma rota também definida neste momento.
  - **Atores:** A01, A02.
  - **Pré-condições:** Chegada de um produto inacabado no buffer de saída ou a existência de um pedido de produção de um produto.
  - **Pós-condições:** Finalizada uma etapa do processo de produção de um produto; Produto pronto para ser transportado.
  - **Fluxo normal:**
    - 1. Verificar produto a ser produzido e etapa de produção atual (RF02);
    - 2. Verificar a máquina planejada na programação da produção (RF03.1);
      - 2.a. Caso haja máquina, designá-la como destino;
      - 2.b. Caso contrário esperar até que haja máquina disponível.
    - 3. Selecionar AGV (RF04.1);
    - 4. Selecionar a rota utilizada (RF5);
    - 5. Transportar o produto (RF06).
  - **Variações/Fluxo alternativo:** Inexistente.
  - **Não funcional:** inexistente.
  - **Autor e Data:** Flavio Montoro – 12/12/2014.
- 
- **Identificador:** RF02.
  - **Nome:** Processar programação da produção.
  - **Tipo:** Abstrato
  - **Descrição:** Recebe a programação da produção e verifica o produto a ser produzido.



- **Atores:** inexistente.
  - **Pré-condições:** Ter recebido a programação da produção e existir uma requisição de despacho de produto.
  - **Pós-condições:** Ter informado as máquinas envolvidas na operação.
  - **Fluxo normal:**
    - 1. Receber pedido de verificação de produção de produto (RF01);
    - 2. Verificar máquinas envolvidas naquela etapa;
    - 3. Informar máquinas envolvidas (RF01).
  - **Variações/Fluxo alternativo:** inexistente.
  - **Não funcional:** inexistente.
  - **Autor e Datas:** Flavio Montoro – 12/12/2014.
- 
- **Identificador:** RF03.1.
  - **Nome:** Selecionar máquina.
  - **Tipo:** Abstrato.
  - **Descrição:** Seleciona a máquina de destino mais apropriada para processar o produto.
  - **Atores:** A03.
  - **Pré-condições:** Ter uma requisição de despacho de produto.
  - **Pós-condições:** Informar a máquina de destino; avisar indisponibilidade de máquina.
  - **Fluxo normal:**
    - 1. Receber pedido de contratação de máquina (RF01);
    - 2. Verificar se a máquina está disponível (RF04);
    - 2.a Caso esteja disponível, contratar máquina;
    - 3. Informar máquina contratada (RF01).
  - **Variações/Fluxo alternativo:**
    - 2.b.1. Caso contrário, verificar roteiro alternativo (RF05);
    - 2.b.2. Se tiver roteiro alternativo, Repetir passo 2 com a nova máquina;
    - 2.b.3. Caso contrário, enviar mensagem de indisponibilidade de máquina (RF01);
  - **Não funcional:** inexistente.
  - **Autor e Datas:** Flavio Montoro – 12/12/2014.
- 
- **Identificador:** RF03.2.
  - **Nome:** Verificar disponibilidade de máquina.
  - **Tipo:** Abstrato
  - **Descrição:** Responsável por verificar se a máquina desejada está disponível.
  - **Atores:** A03.
  - **Pré-condições:** Ter uma requisição de verificação de disponibilidade de máquina.
  - **Pós-condições:** Ter verificado a disponibilidade da máquina.
  - **Fluxo normal:**
    - 1. Receber pedido de verificação de máquina (RF03.1);
    - 2. Verificar se a máquina está disponível;
    - 3.a. Caso esteja, informar disponibilidade (RF03.1);
    - 3.b. Caso contrário, informar indisponibilidade (RF03.1).
  - **Variações/Fluxo alternativo:** inexistente.
  - **Não funcional:** inexistente.
  - **Autor e Datas:** Flavio Montoro – 12/12/2014.
- 
- **Identificador:** RF03.3.
  - **Nome:** Verificar roteiro alternativo de produção.
  - **Tipo:** Abstrato.
  - **Descrição:** Verificar roteiro alternativo para o produto a ser processado.
  - **Atores:** inexistente.
  - **Pré-condições:** Ter uma tarefa de produção; Roteiro principal impossibilitado.
  - **Pós-condições:** Selecionar um roteiro de produção alternativo; Informar inexistência de roteiro alternativo.
  - **Fluxo normal:**

- 1. Verificar pedido de roteiro alternativo (RF03.1);
    - 2. Verificar produto a ser processado;
    - 3. Consultar base de roteiros dos produtos;
    - 4.a. Se existir roteiro alternativo
      - 4.a.1. Informar roteiro alternativo (RF03.1).
    - 4.b. Senão
      - 4.b.1. Informar inexistência de roteiro alternativo (RF03.1).
  - **Variações/Fluxo alternativo:** inexistente.
  - **Não funcional:** inexistente.
  - **Autor e Datas:** Flavio Montoro – 12/12/2014.
- 
- **Identificador:** RF04.1.
  - **Nome:** Selecionar AGV.
  - **Tipo:** Abstrato.
  - **Descrição:** Responsável por selecionar o AGV mais indicado naquele momento para a tarefa em questão.
  - **Atores:** A04.
  - **Pré-condições:** Ter uma requisição de AGV ativa.
  - **Pós-condições:** Ter contratado um AGV para o processamento.
  - **Fluxo normal:**
    - 1. Verificar pedido de AGV (RF01);
    - 2. Requisitar lista priorizada de AGV (RF04.2);
    - 3. Tentar contratar AGV de maior prioridade;
    - 4.a. Caso AGV disponível, Contratar AGV;
    - 5. Enviar id do AGV contratado (RF01).
  - **Variações/Fluxo alternativo:**
    - 4.b.1. Caso contrário, Remover AGV da lista;
    - 4.b.2. Executar passo 3.
  - **Não funcional:** inexistente.
  - **Autor e Datas:** Flavio Montoro – 12/12/2014.
- 
- **Identificador:** RF04.2.
  - **Nome:** Gerar lista priorizada de AGVs.
  - **Tipo:** Abstrato.
  - **Descrição:** Responsável por gerar uma lista priorizada de AGVs a partir dos AGVs disponíveis para executar uma tarefa específica.
  - **Atores:** inexistente.
  - **Pré-condições:** Ter uma requisição de AGV ativa.
  - **Pós-condições:** Ter realizado a verificação e gerado a lista priorizada.
  - **Fluxo normal:**
    - 1. Verificar pedido de geração de lista priorizada de AGVs (RF04.1);
    - 2. Verificar AGVs disponíveis (RF04.3);
    - 3. Avaliar AGVs disponíveis (RF04.4);
    - 4. Gerar lista priorizada de AGVs;
    - 5. Encaminhar lista priorizada (RF04.1).
  - **Variações/Fluxo alternativo:** inexistente.
  - **Não funcional:** inexistente.
  - **Autor e Datas:** Flavio Montoro – 12/12/2014.
- 
- **Identificador:** RF04.3.
  - **Nome:** Verificar AGVs disponíveis.
  - **Tipo:** Abstrato.
  - **Descrição:** Responsável por verificar a disponibilidade de cada AGV existente na fábrica. A disponibilidade é definida pela inatividade do AGV ou se o AGV está executando uma tarefa próxima de ser finalizada (janela de tempo de 5 minutos).
  - **Atores:** A04.

- **Pré-condições:** Ter uma requisição de verificação de disponibilidade de AGVs.
  - **Pós-condições:** Ter listado todos os AGVs disponíveis.
  - **Fluxo normal:**
    - 1. Receber pedido de verificação de disponibilidade (RF04.2);
    - 2. Enviar *broadcast* para todos os AGVs cadastrados;
    - 3. Analisar os status dos AGVs;
    - 4. Gerar lista de AGVs disponíveis;
    - 5. Encaminhar lista de AGVs disponíveis (RF04.2).
  - **Variações/Fluxo alternativo:** inexistente.
  - **Não funcional:** inexistente.
  - **Autor e Datas:** Flavio Montoro – 12/12/2014.
- 
- **Identificador:** RF04.4.
  - **Nome:** Calcular prioridade de AGV.
  - **Tipo:** Abstrato.
  - **Descrição:** Responsável por calcular a prioridade do AGV. Esta prioridade se dá por uma nota atribuída pelo sistema *FUZZY*.
  - **Atores:** inexistente.
  - **Pré-condições:** Ter uma requisição de verificação de prioridade do AGV.
  - **Pós-condições:** Ter definido uma nota de prioridade para o AGV.
  - **Fluxo normal:**
    - 1. Receber pedido de avaliação de AGV (RF04.2);
    - 2. Calcular notas de prioridade;
    - 3. Encaminhar lista priorizada de AGVs (RF04.2).
  - **Variações/Fluxo alternativo:** inexistente.
  - **Não funcional:** inexistente.
  - **Autor e Datas:** Flavio Montoro – 12/12/2014.
- 
- **Identificador:** RF05.
  - **Nome:** Definir Rotas.
  - **Tipo:** Abstrato.
  - **Descrição:** Responsável por selecionar a rota mais indicada naquele momento para a tarefa em questão.
  - **Atores:** inexistente.
  - **Pré-condições:** Ter uma requisição de rota ativa.
  - **Pós-condições:** Ter realizado a verificação.
  - **Fluxo normal:**
    - 1. Verificar pedido de rota (RF01);
    - 2. Avaliar rotas;
    - 3. Selecionar melhor rota;
    - 4. Enviar id da rota selecionada (RF01).
  - **Variações/Fluxo alternativo:** inexistente.
  - **Não funcional:** inexistente.
  - **Autor e Datas:** Flavio Montoro – 12/12/2014.
- 
- **Identificador:** RF06.
  - **Nome:** Transportar produto.
  - **Tipo:** Abstrato.
  - **Descrição:** Responsável por transportar o produto para a máquina destino.
  - **Atores:** A05.
  - **Pré-condições:** Ter uma requisição de despacho.
  - **Pós-condições:** Deixar o produto no buffer de entrada da máquina destino.
  - **Fluxo normal:**
    - 1. Verificar pedido de transporte (RF01);
    - 2. Buscar produto no buffer de saída (ou buffer de matéria prima);

- 3. Transportar o produto até o buffer de entrada da máquina de destino.
  - **Variações/Fluxo alternativo:** inexistente.
  - **Não funcional:** inexistente.
  - **Autor e Datas:** Flavio Montoro – 12/12/2014.
- 
- **Identificador:** RF07.
  - **Nome:** Processar Produto.
  - **Tipo:** Abstrato.
  - **Descrição:** Responsável por processar o produto.
  - **Atores:** A01, A05.
  - **Pré-condições:** Ter um produto no buffer de entrada e estar disponível.
  - **Pós-condições:** enviar produto processado para o buffer de saída.
  - **Fluxo normal:**
    - 1. Chegada de produto no buffer de entrada;
    - 2. Caso a máquina esteja disponível;
      - 2.a Processar produto;
      - 2.a.1 Enviar produto pro buffer de saída;
    - 2.b Caso contrário;
      - 2.b.1 Esperar máquina terminar o processamento.
      - 2.b.2 Repetir passo 2.
    - 3. Atualizar status do produto (RF08).
  - **Variações/Fluxo alternativo:** inexistente
  - **Não funcional:** inexistente.
  - **Autor e Datas:** Flavio Montoro – 24/12/2014.
- 
- **Identificador:** RF08.
  - **Nome:** Atualizar produto.
  - **Tipo:** Abstrato
  - **Descrição:** Após o processamento do produto pela máquina, o status do produto é atualizado.
  - **Atores:** inexistente.
  - **Pré-condições:** A máquina ter finalizado o processamento.
  - **Pós-condições:** Status do produto atualizado.
  - **Fluxo normal:**
    - 1. Receber pedido de atualização do status do produto.
    - 2. Atualizar status do produto.
  - **Variações/Fluxo alternativo:** Inexistente.
  - **Não funcional:** inexistente.
  - **Autor e Datas:** Flavio Montoro – 24/12/2014.

### Descrição dos atores

- **Estereótipo:** Elemento físico;
- **Tipo:** Concreto. Primário;
- **Identificador:** A01;
- **Forma do Ator:** Iniciador;
- **Contato:** Buffer de saída;
- **Definição/Descrição:** Ator que representa cada buffer de saída existente em cada máquina;
- **Responsabilidades:** RF01, RF07;
- **Relações:** Cliente da aplicação;
- **Especialidade requerida:** não aplicável;
- **Autor e Datas:** Flavio Montoro – 12/12/2014.

- **Estereótipo:** Elemento físico;
- **Tipo:** Concreto. Primário;
- **Identificador:** A02;
- **Forma do Ator:** Iniciador;
- **Contato:** Buffer de matéria prima;
- **Definição/Descrição:** Ator que representa o buffer de matéria prima existente na fábrica;
- **Responsabilidades:** RF01;
- **Relações:** Cliente da aplicação;
- **Especialidade requerida:** não aplicável;
- **Autor e Datas:** Flavio Montoro – 27/06/2014.

- **Estereótipo:** Elemento físico;
- **Tipo:** Concreto. Secundário;
- **Identificador:** A03;
- **Forma do Ator:** Servidor;
- **Contato:** Máquina;
- **Definição/Descrição:** Ator que representa as máquinas existentes na fábrica;
- **Responsabilidades:** RF03.1, RF03.2;
- **Relações:** Servidor da aplicação;
- **Especialidade requerida:** não aplicável;
- **Autor e Datas:** Flavio Montoro – 12/12/2014.

- **Estereótipo:** Elemento físico;
- **Tipo:** Concreto. Secundário;
- **Identificador:** A04;
- **Forma do Ator:** Servidor;
- **Contato:** AGV;
- **Definição/Descrição:** Ator que representa os AGVs existentes na fábrica;
- **Responsabilidades:** RF04.1, RF04.3;
- **Relações:** Servidor da aplicação;
- **Especialidade requerida:** não aplicável;
- **Autor e Datas:** Flavio Montoro – 12/12/2014.

- **Estereótipo:** Elemento físico;
- **Tipo:** Concreto. Primário;
- **Identificador:** A05;
- **Forma do Ator:** Iniciador;
- **Contato:** Buffer de entrada;
- **Definição/Descrição:** Ator que representa cada buffer de saída existente em cada máquina;
- **Responsabilidades:** RF06, RF07;
- **Relações:** Cliente da aplicação;
- **Especialidade requerida:** não aplicável;
- **Autor e Datas:** Flavio Montoro – 12/12/2014.

Após os casos de uso e o levantamento de requisitos, foi gerado o diagrama de caso de uso, conforme apresentado na Figura. 1.

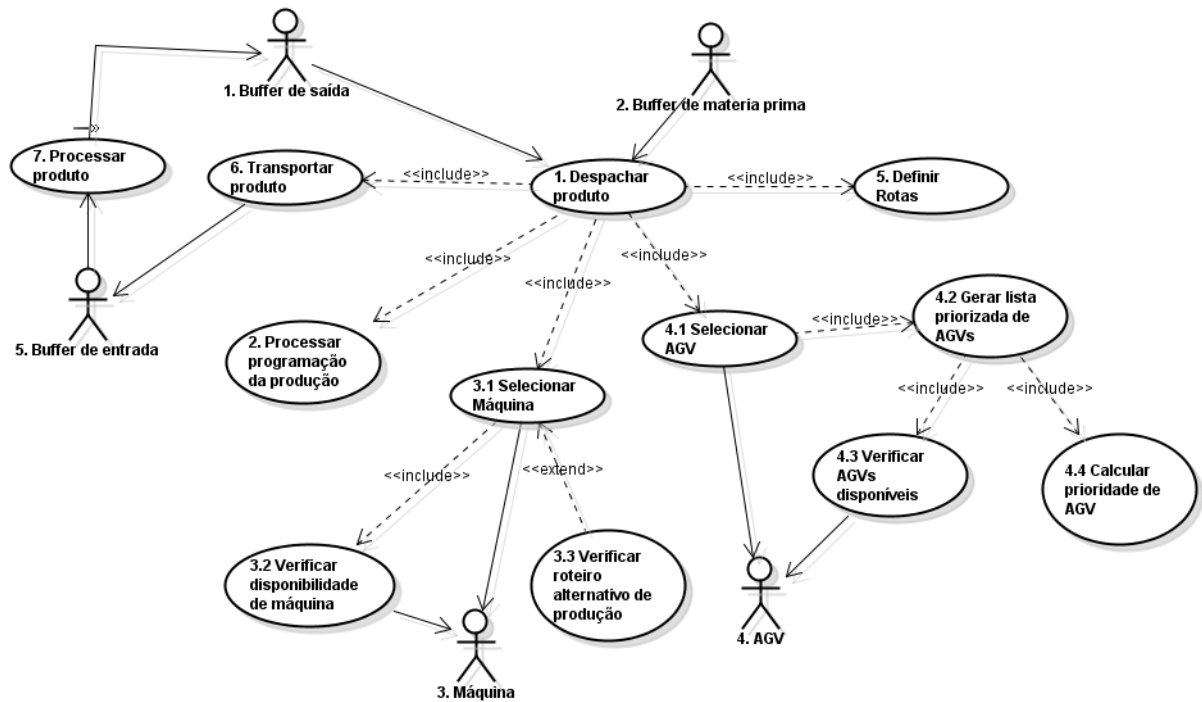


Figura. 1. Diagrama de caso de uso do domínio do problema

Neste diagrama apresentado na Figura. 1, são apresentadas as principais funções levantadas no documento de requisitos.

### A.1.2 Descrição da ontologia do problema (*POD - Problem Ontology Description*)

Após a identificação de todos os requisitos, foi gerado a ontologia do problema, apresentando uma outra visão de elementos que contem no domínio. A Figura. 2 apresenta a ontologia gerada para o domínio do problema.

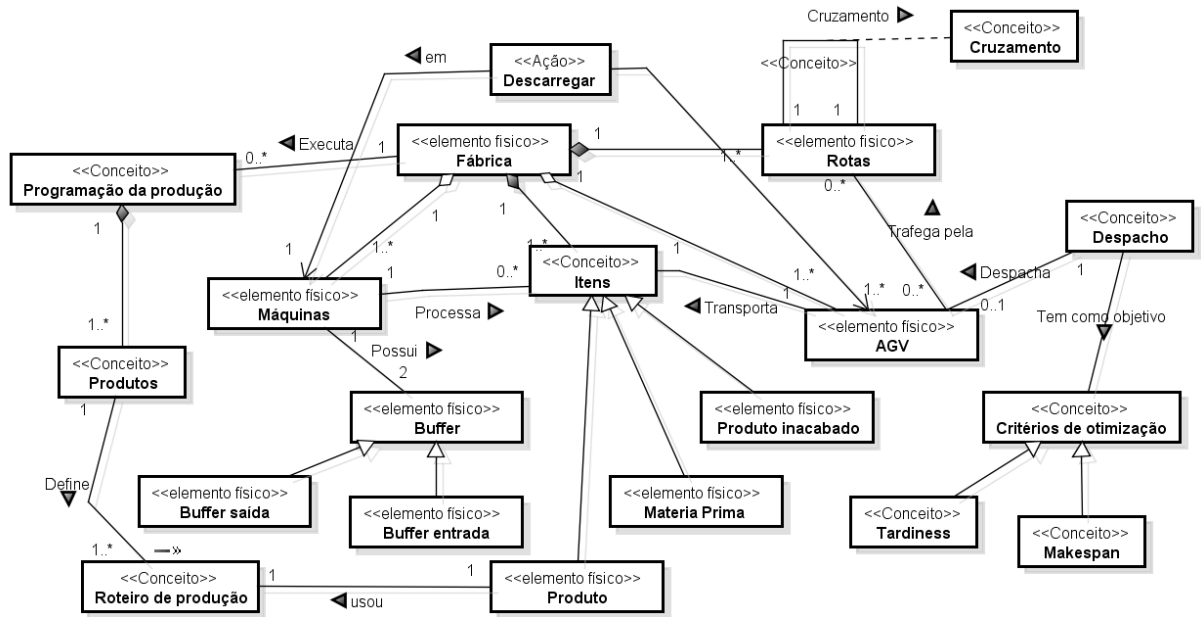


Figura. 2. Descrição da ontologia do problema

### A.1.3 Identificação das organizações (OID - Organisation Identification)

Após identificados os casos de uso e a ontologia, foi realizado o agrupamento das funcionalidades que tinham funções próximas e geradas as organizações, conforme apresentado na Figura 3.

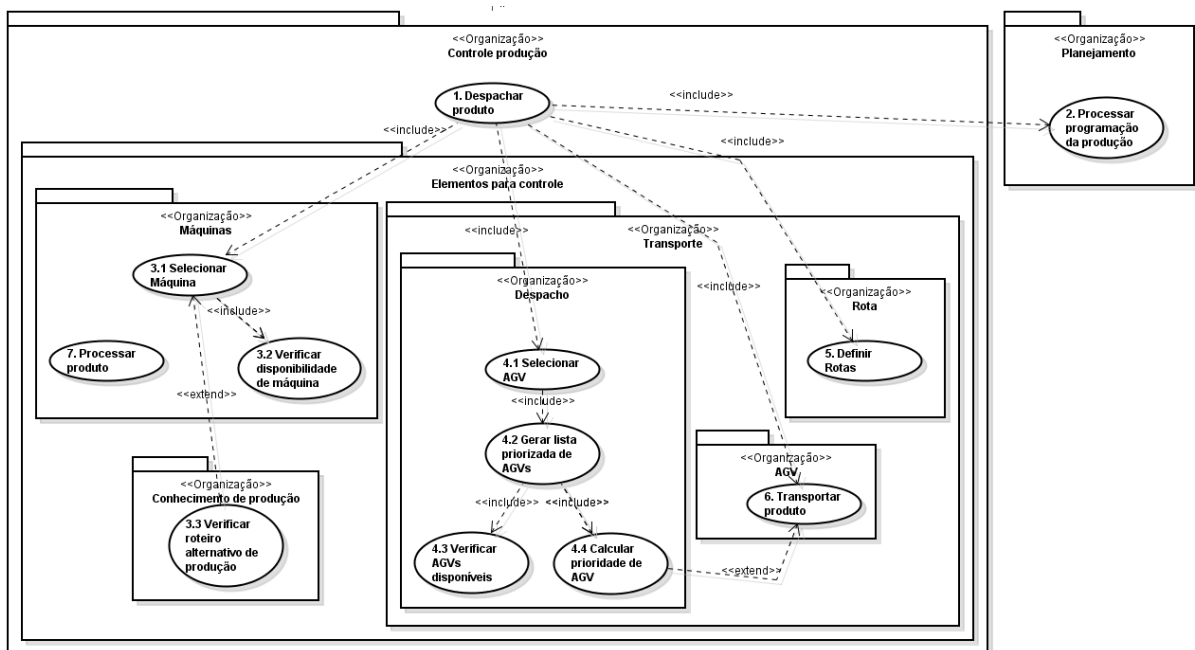


Figura. 3. Identificação das organizações.

### A.1.4 Identificação dos papéis e interações (IRI - Interactions and Role Identification)

Após identificadas as organizações presentes no sistema, foram identificados os papéis que representam o comportamento global da organização, conforme apresentado na Figura 4.

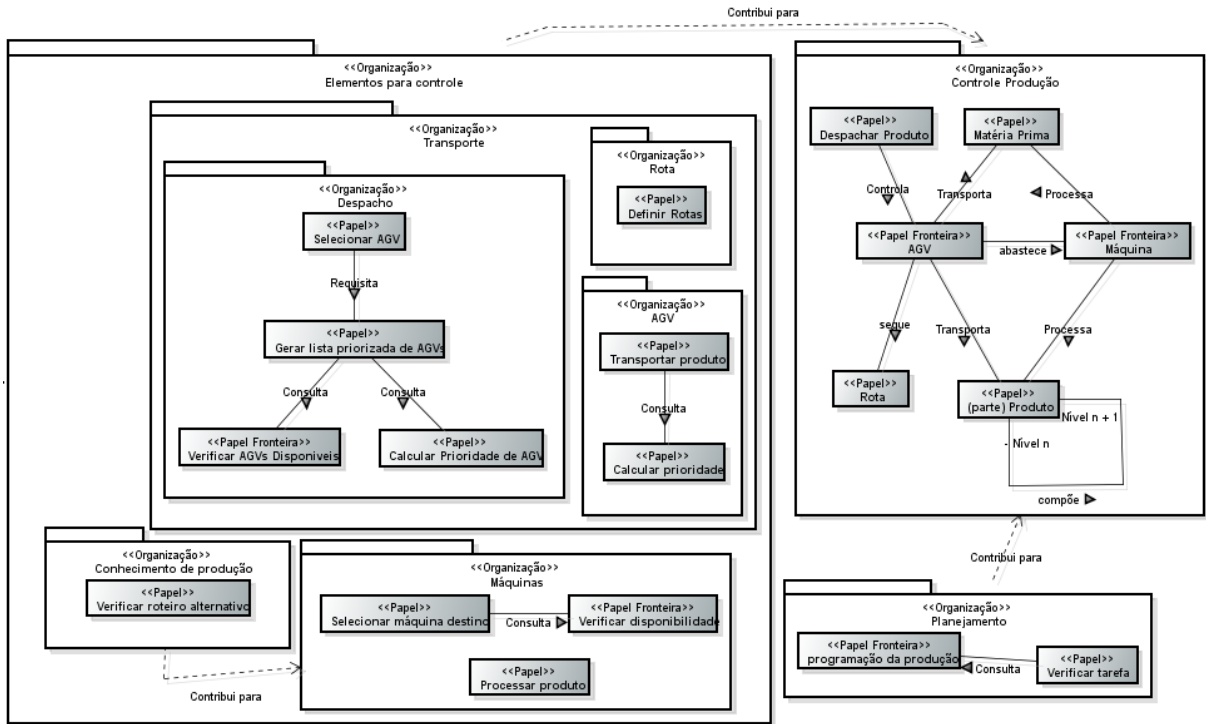


Figura 4. Identificação dos papéis e suas interações

### A.1.4 Descrição de cenários (SD - Scenario Description)

Com os papéis já identificados, assim como suas interações, foram gerados diagramas de sequência para modelar a sequência de interações dos papéis para cada uma das organizações, conforme apresentado nas Figuras 5, 6, 7, 8, 9, 10 e 11.



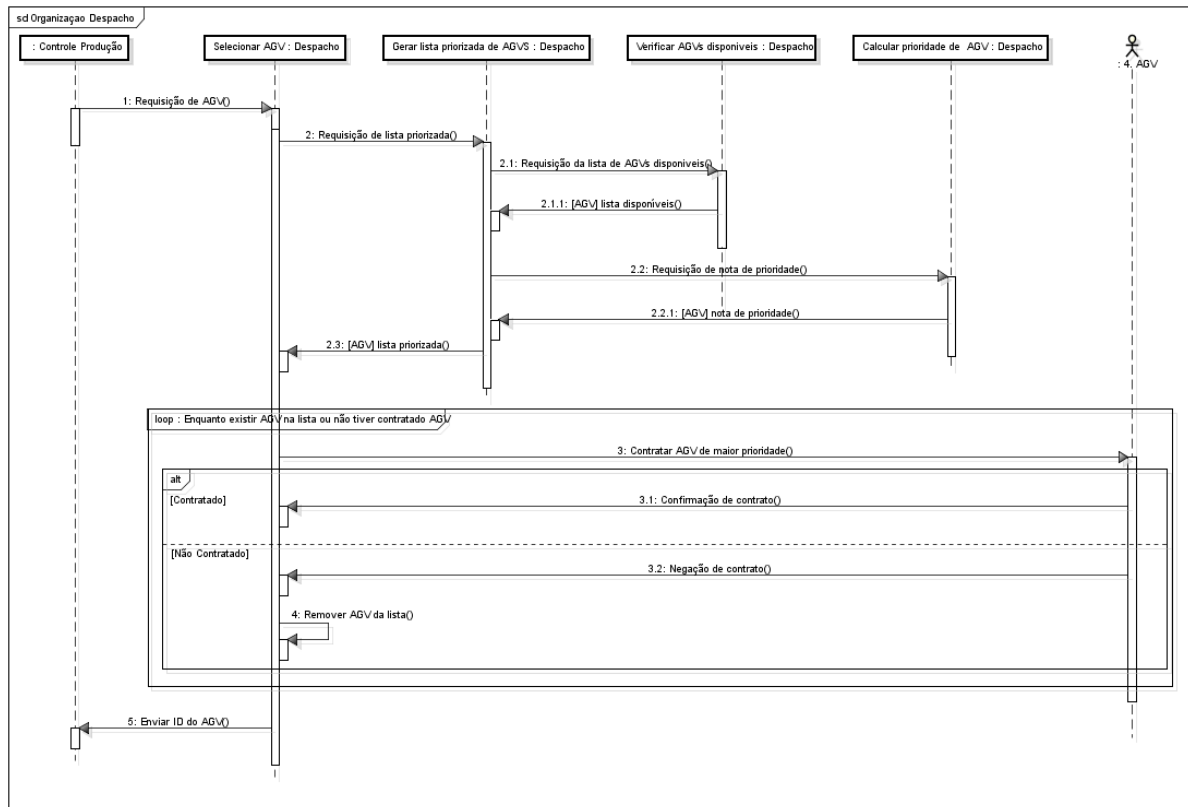


Figura. 5 Descrição do cenário da organização Despacho

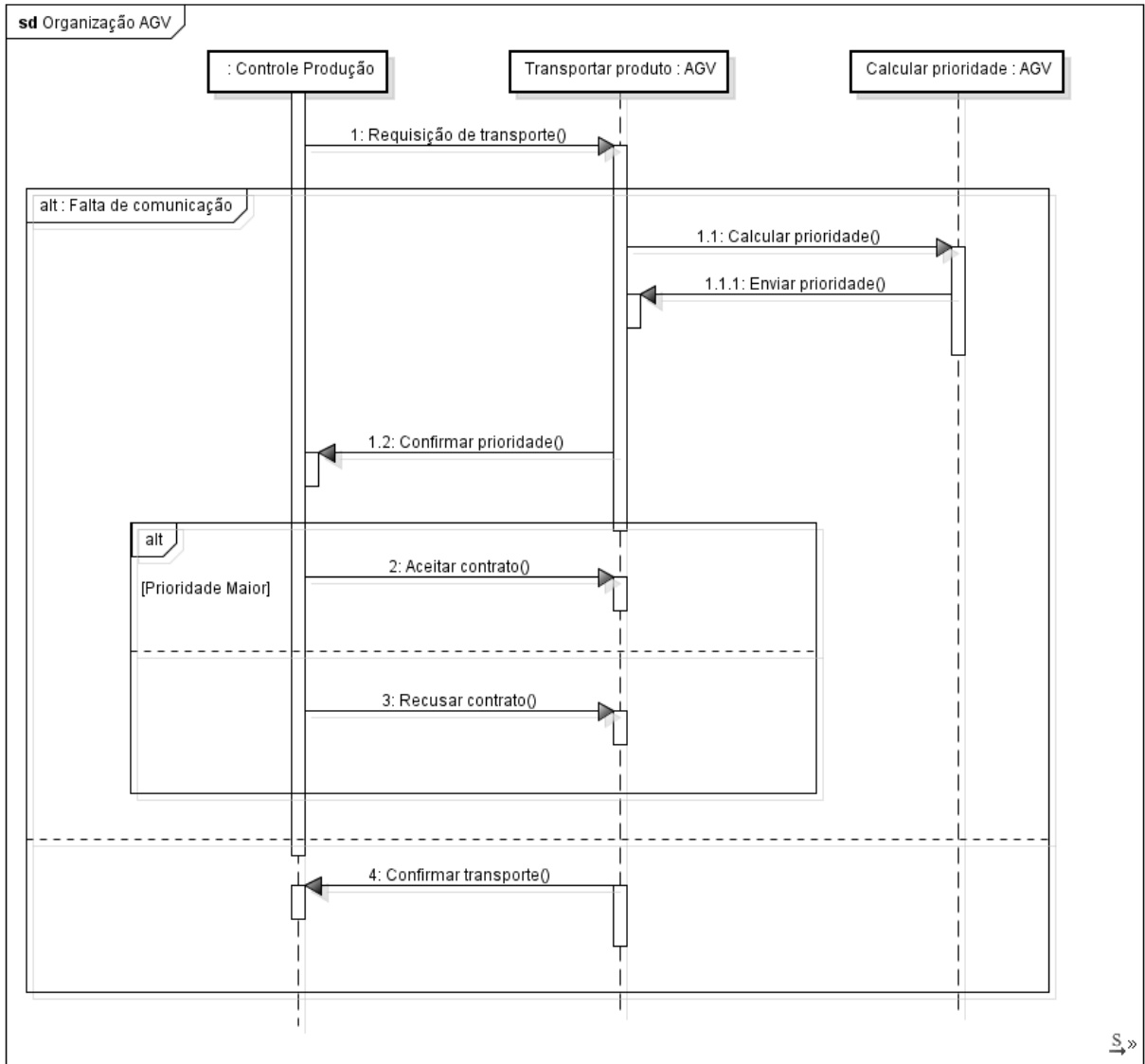


Figura. 6. Descrição do cenário da organização AGV.

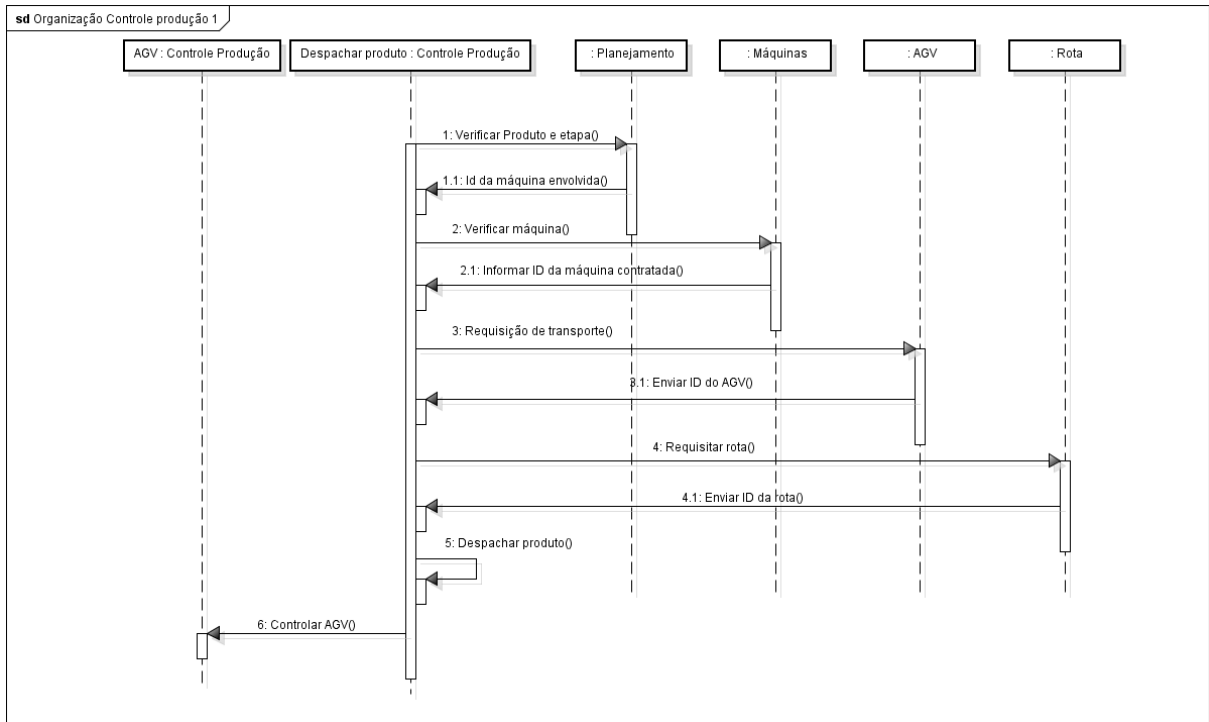


Figura. 7. Descrição do cenário da organização Controle produção (parte 1)

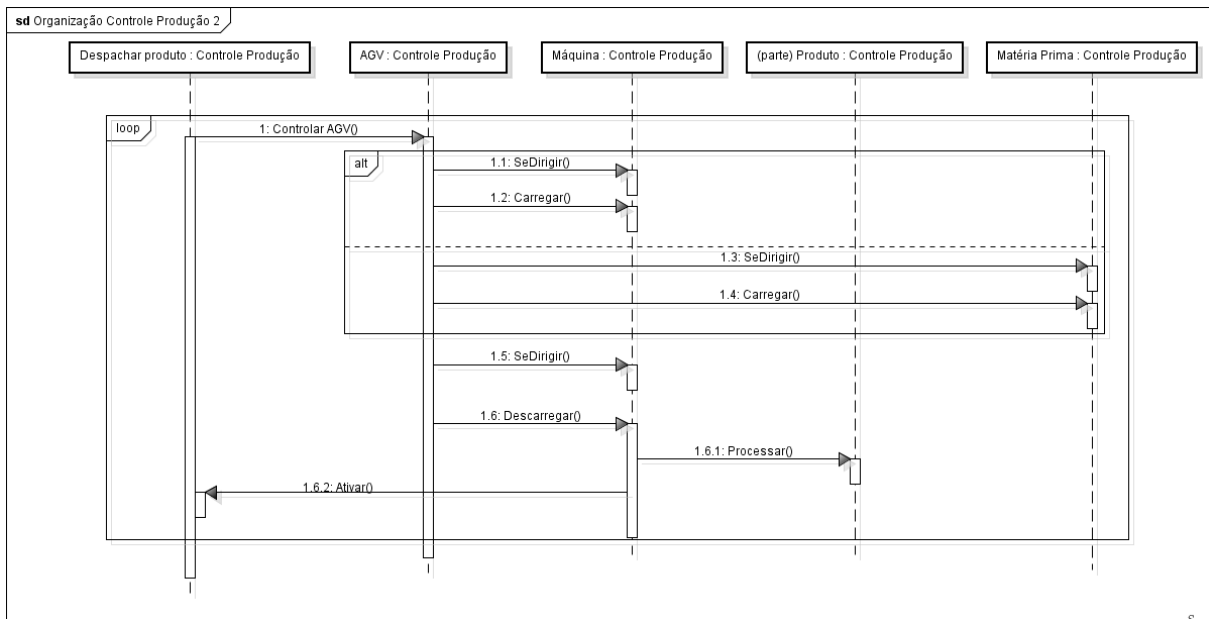


Figura. 8. Descrição do cenário da organização Controle produção (parte 2)

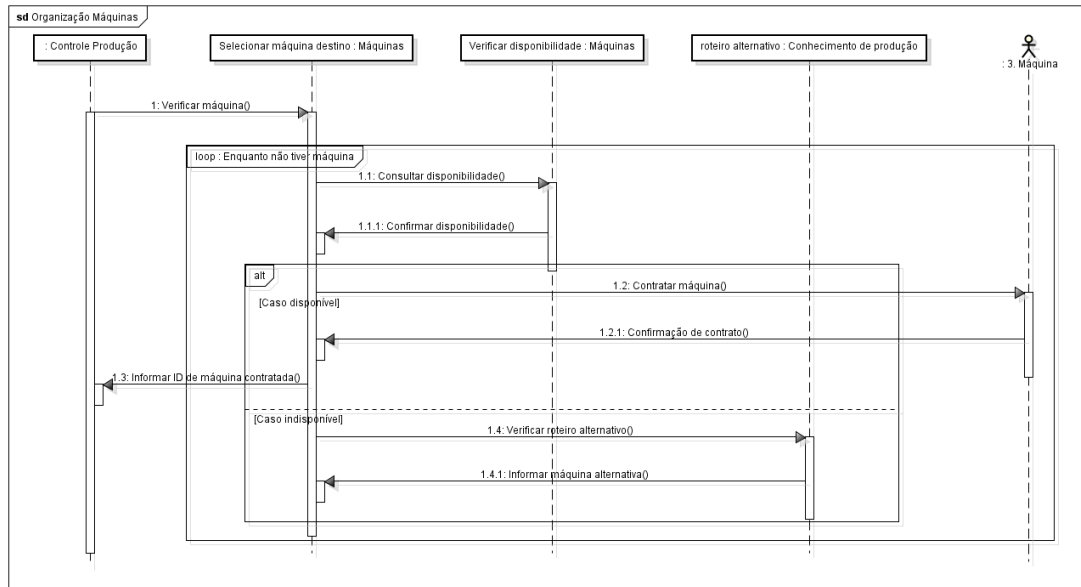


Figura. 9. Descrição dos cenários da organização máquinas.

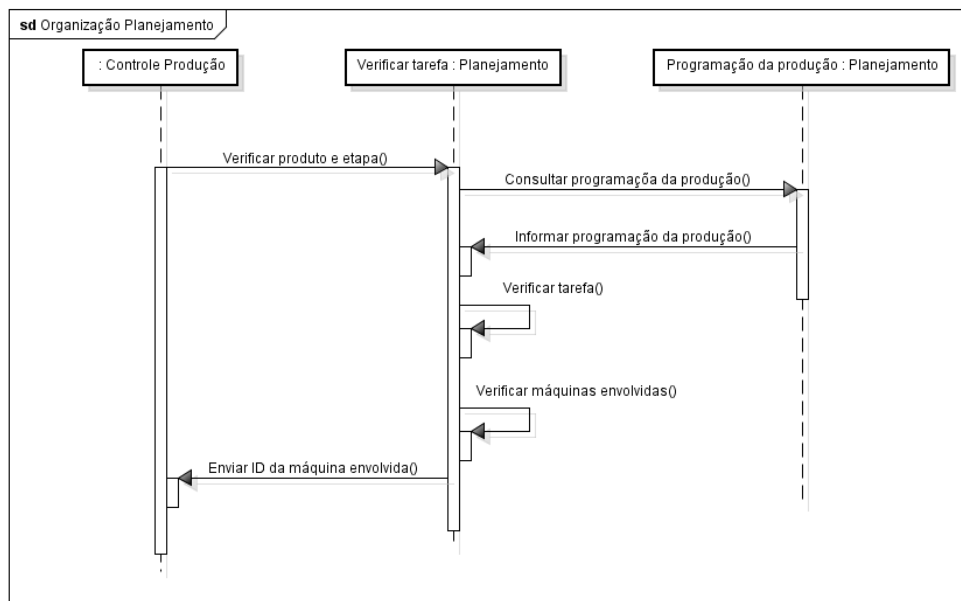


Figura. 10. Descrição do cenário da organização Planejamento

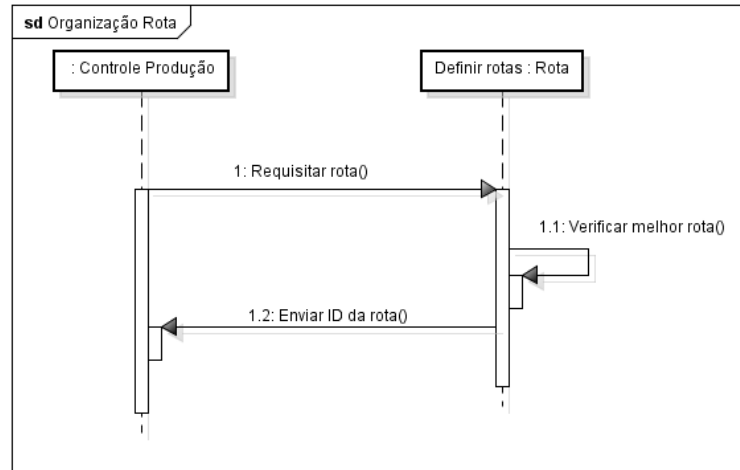


Figura. 11. Descrição do cenário da organização Rota

### A.1.6 Plano dos papéis (RP - Role Plan)

Com os cenários descritos, indicando a interação entre os papéis, nesta atividade foram gerados diagramas de atividade para representar as interações internas dentro de cada um dos papéis pertencentes às organizações, conforme apresentados nas Figuras 12, 13, 14, 15, 16, 17 e 18.

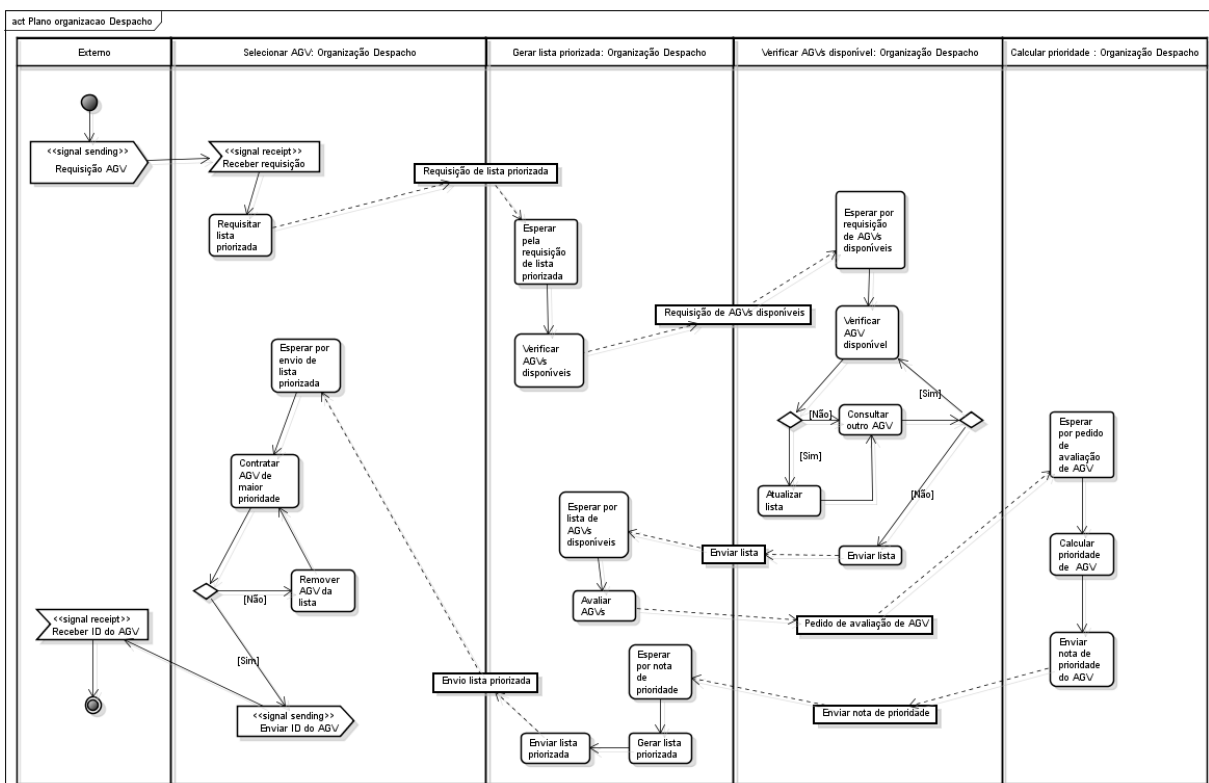


Figura. 12. Plano de papel da organização despacho.

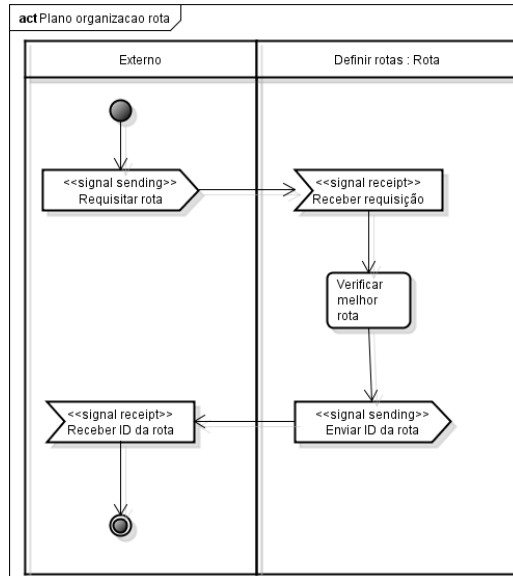


Figura. 13. Plano de papel da organização Rota

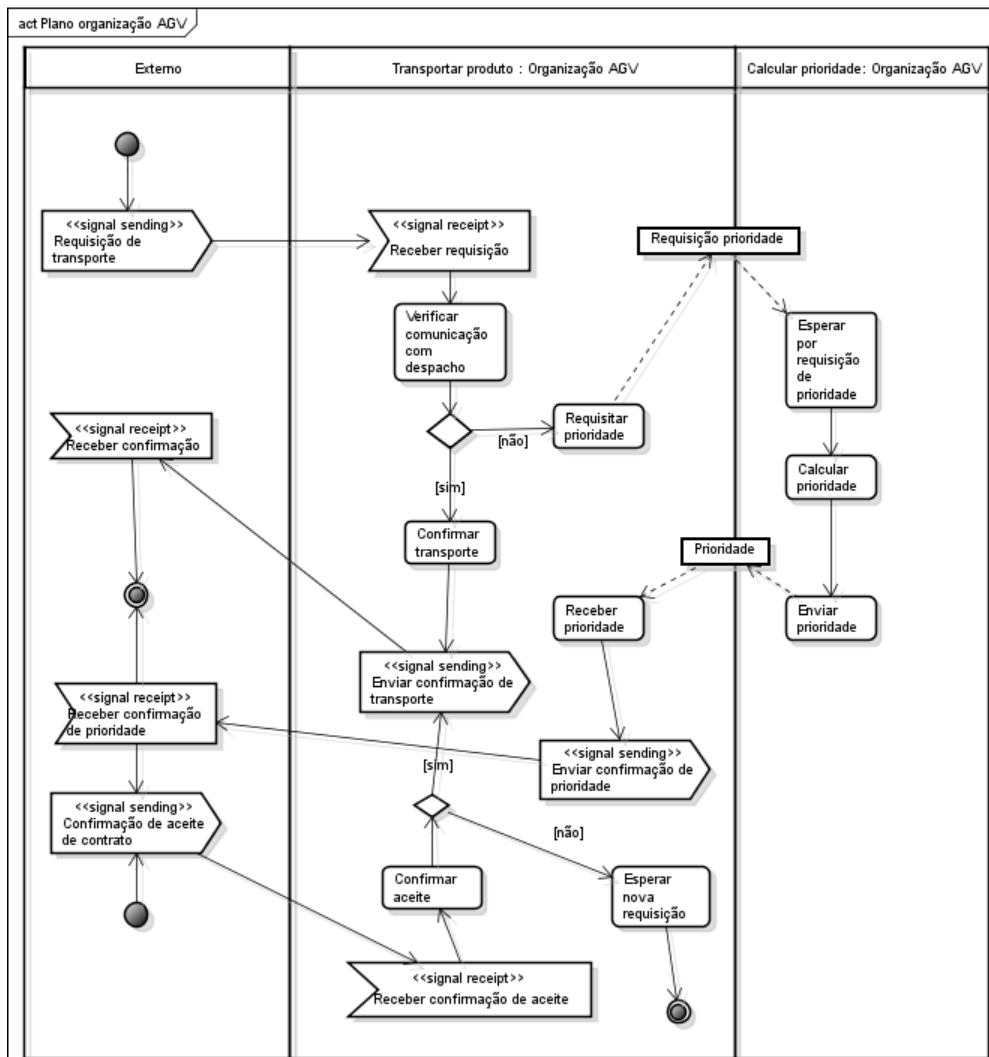


Figura. 14. Plano de papel da organização AGV.

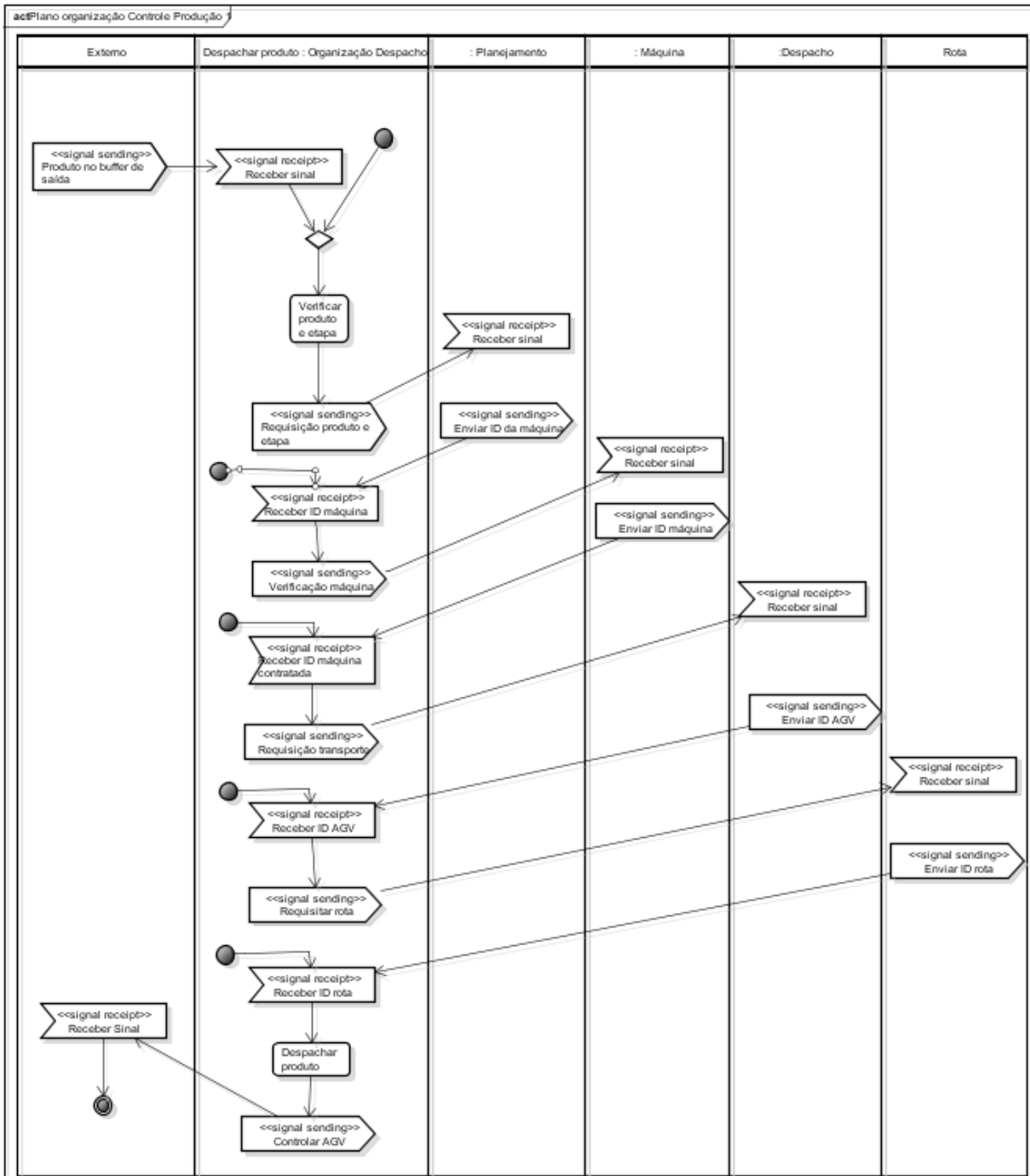


Figura. 15. Plano de papel da organização controle produção (parte 1)

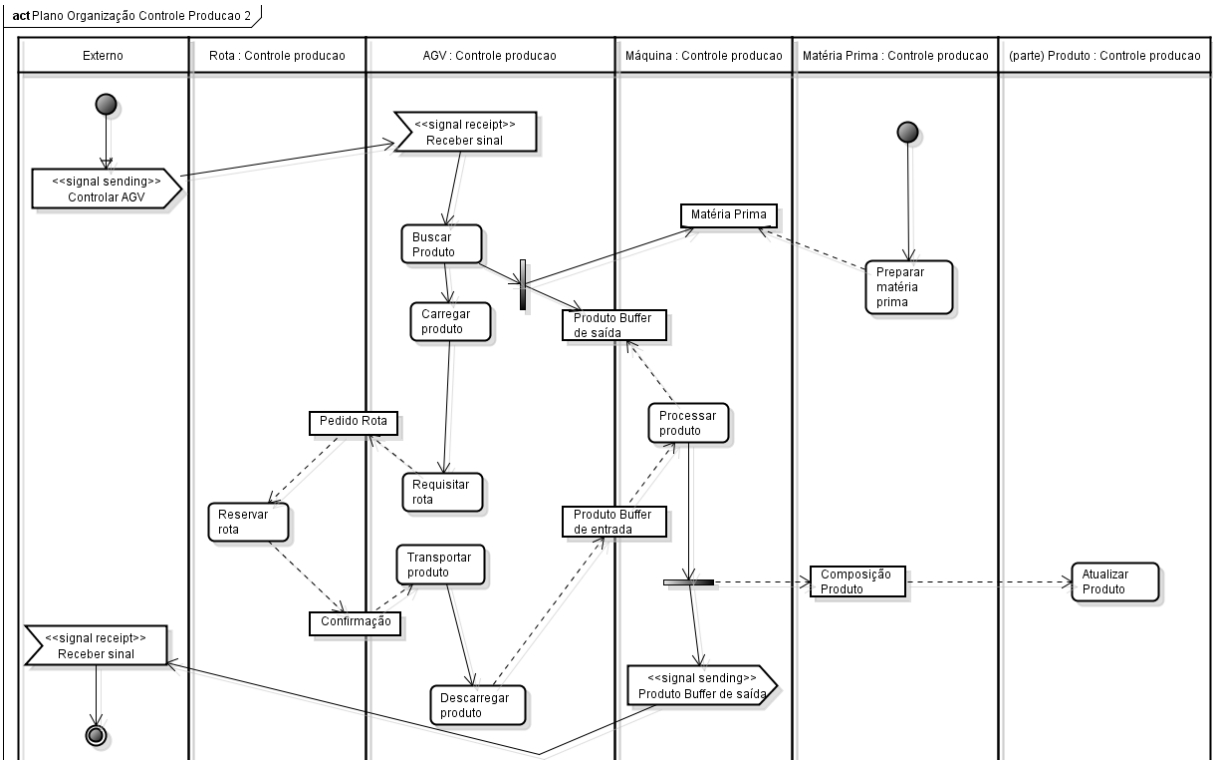


Figura. 16. Plano de papel da organização controle produção (parte 2)

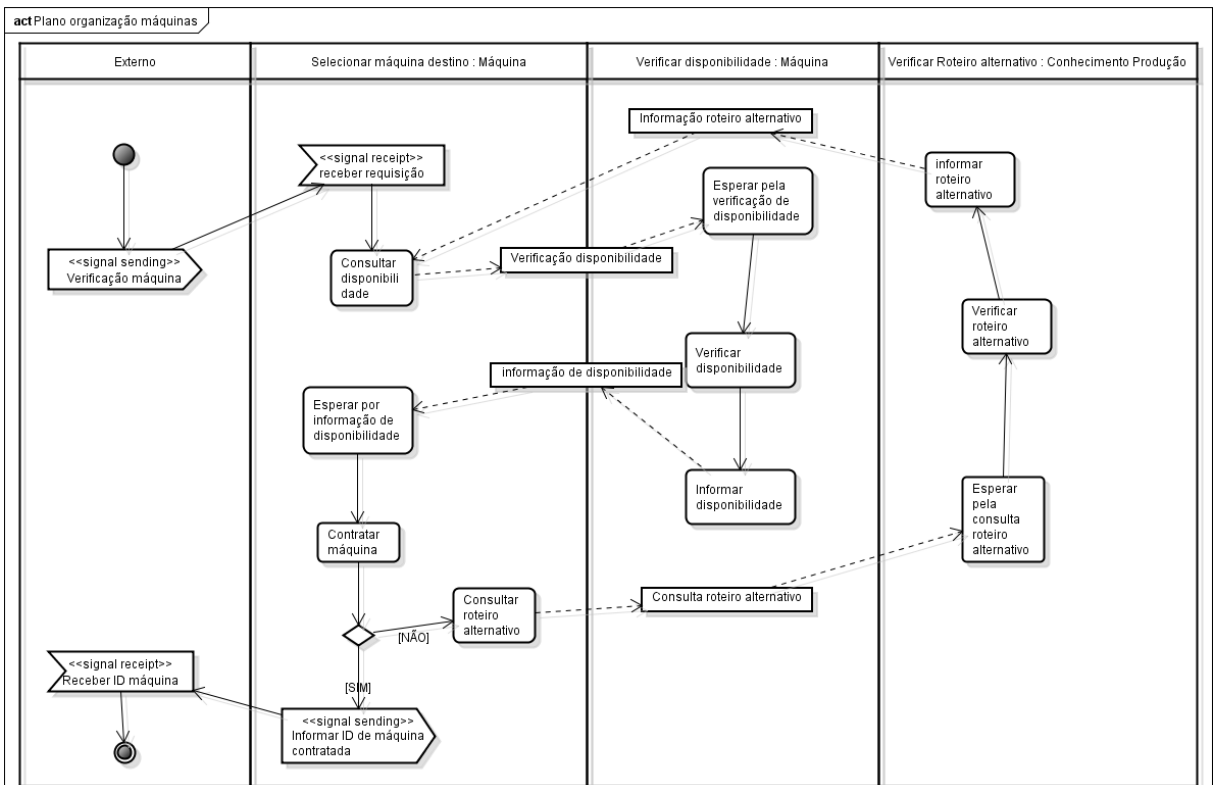


Figura. 17. Plano de papel da organização Máquinas.



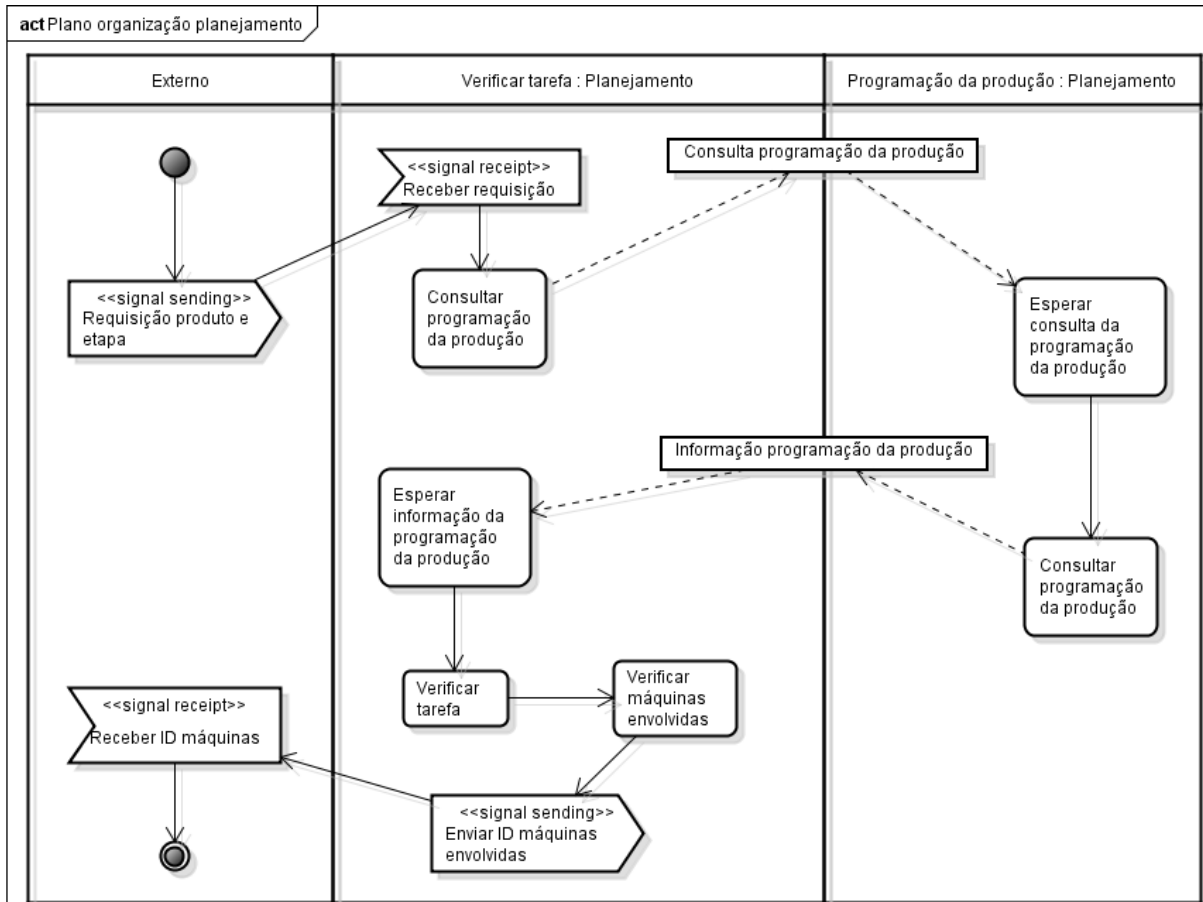


Figura. 18. Plano de papel da organização Planejamento.

### A.1.7 Identificação das capacidades (CP – Capacity Identification)

Nesta etapa, após todos os modelos demonstrando os papéis, organizações e as interações, foi definido os comportamentos genéricos dos papéis mais relevantes para o domínio do problema. Estes comportamentos genéricos foram representados por meios de capacidades, conforme apresentado na Figura 19.

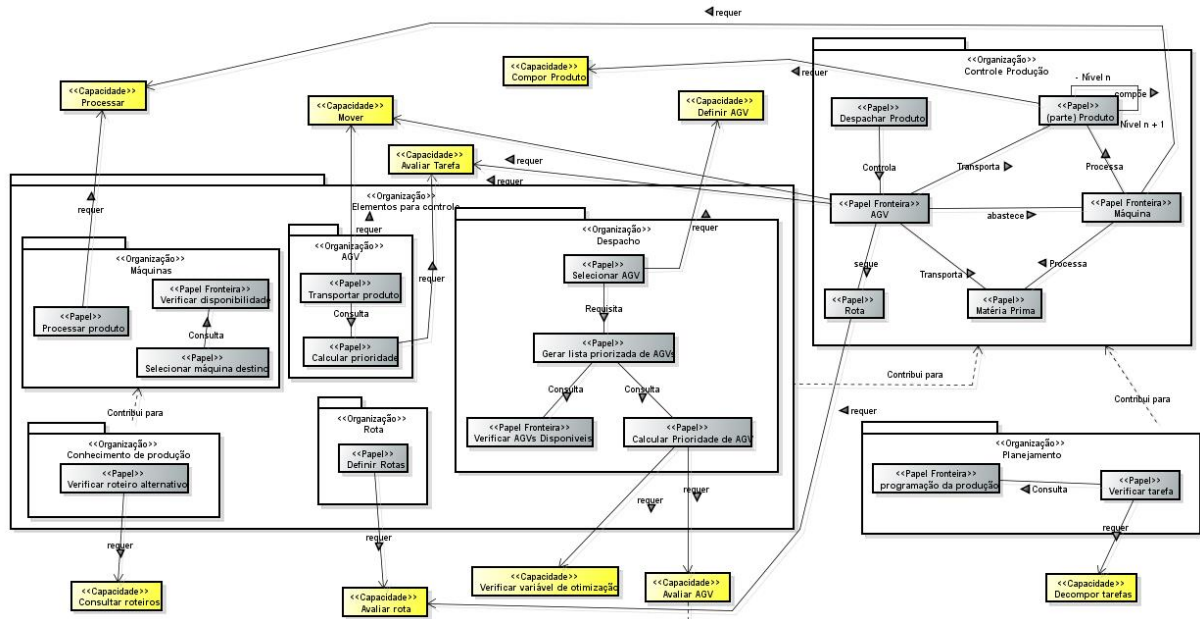


Figura. 19. Identificação das capacidades

## A.2. Sociedade de agentes

Nesta fase, foram refinados os diagramas gerados na fase anterior e adicionados alguns novos diagramas para modelar a solução do problema já orientado aos agentes.

### A.2.1. Descrição da ontologia de solução (SOD – *Solution Ontology Description*)

Nesta atividade, foi refinada a ontologia proposta na primeira fase, acrescentando elementos de solução do problema, conforme apresentado na Figura 20.

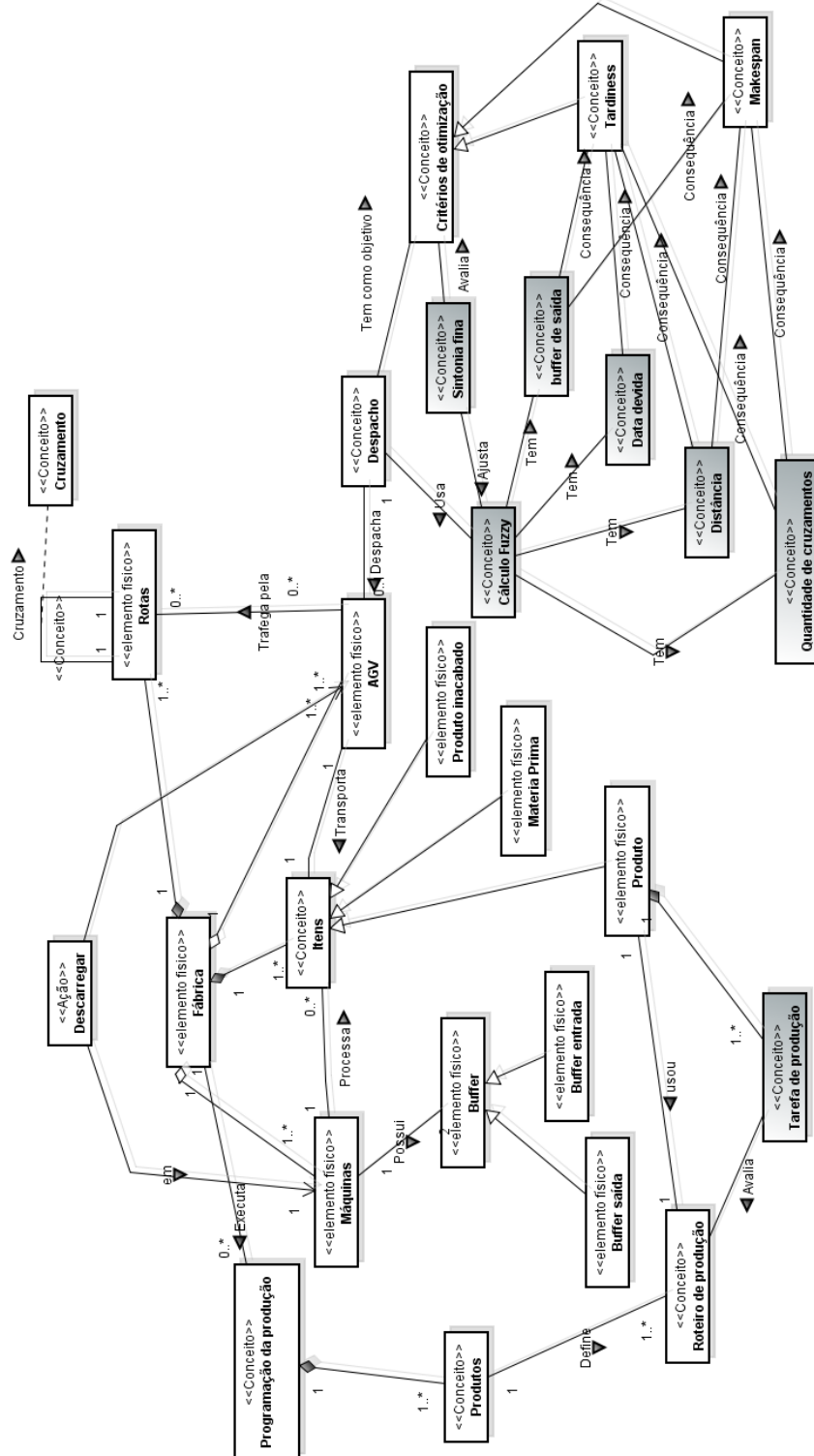


Figura. 20. Descrição da ontologia de solução.

### A.2.2 (Projeto de agentes) Identificação de agentes (*AI – Agent Identification*)

Nesta atividade, são identificados os agentes que irão compor a arquitetura holônica e que irão executar os papéis descritos para realizar o despacho de veículos, conforme apresentado na Figura 21.

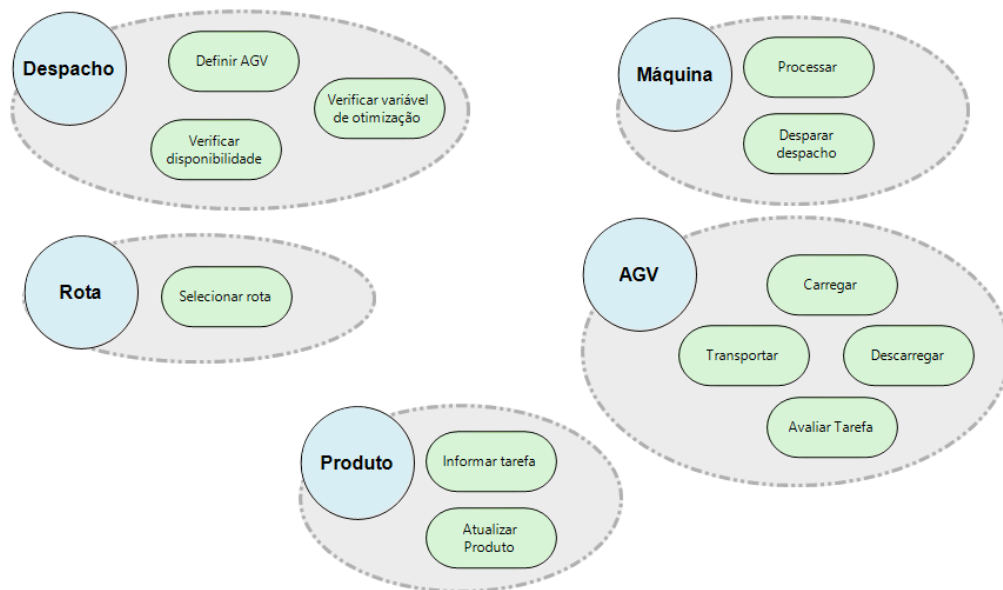


Figura. 21 Agentes do sistema

### A.2.3. Descrição da arquitetura de agentes (*AAD – Agent Architecture Description*)

Nesta atividade foram gerados diagramas de classe para representar cada um de seus agentes e suas capacidades de execução, conforme apresentado nas Figuras 22, 23, 24, 25 e 26.

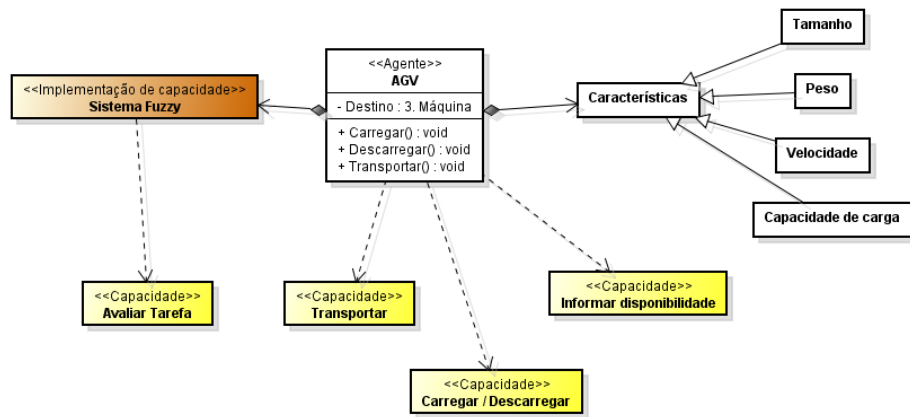


Figura. 22. Arquitetura do agente AGV

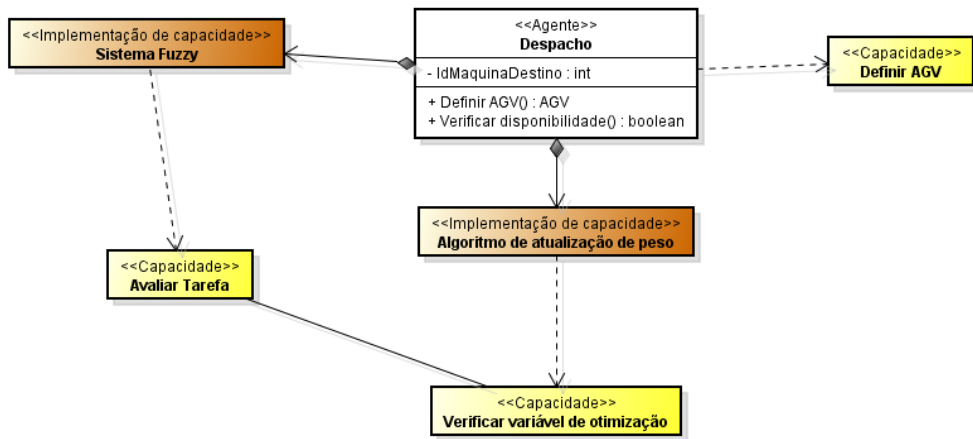


Figura. 23. Arquitetura do agente Despacho

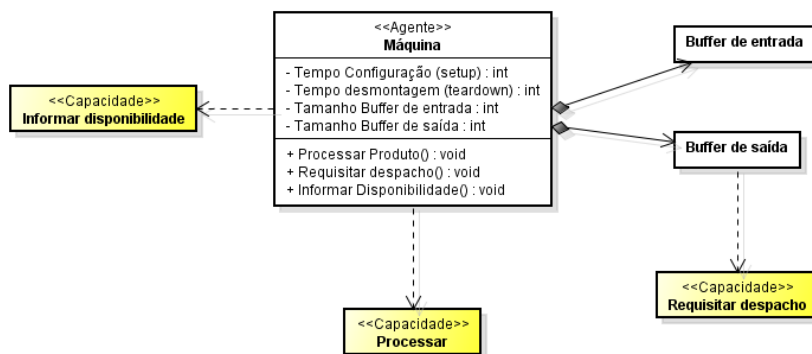


Figura. 24Arquitetura do agente Maquina

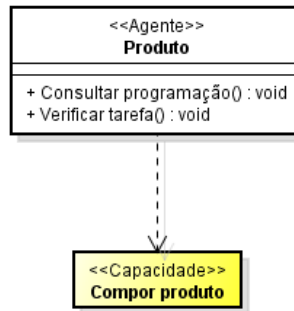


Figura. 25. Arquitetura do agente Produto.

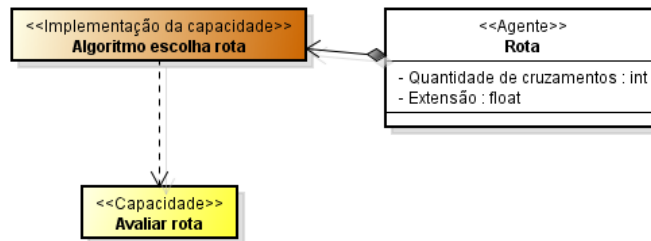


Figura. 26. Arquitetura do agente rota.

**A.2.4 (Projeto de organizações) Descrição da Ontologia de Comunicação (COD – *Communication Ontological Description*)**

Nesta atividade foram descritas todas as comunicações entre os papéis, definindo o conteúdo da comunicação por meio de uma linguagem, ontologia e uma codificação, conforme apresentado nas Figuras 27, 28, 29, 30, 31 e 32.

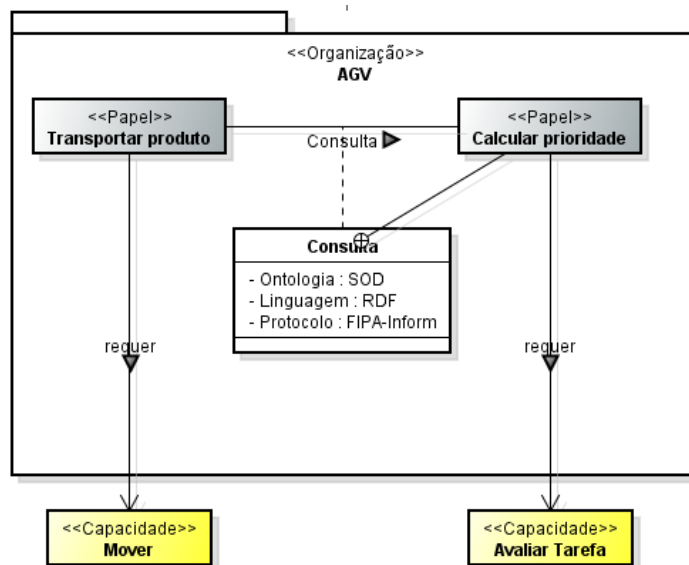


Figura. 27. COD organização AGV

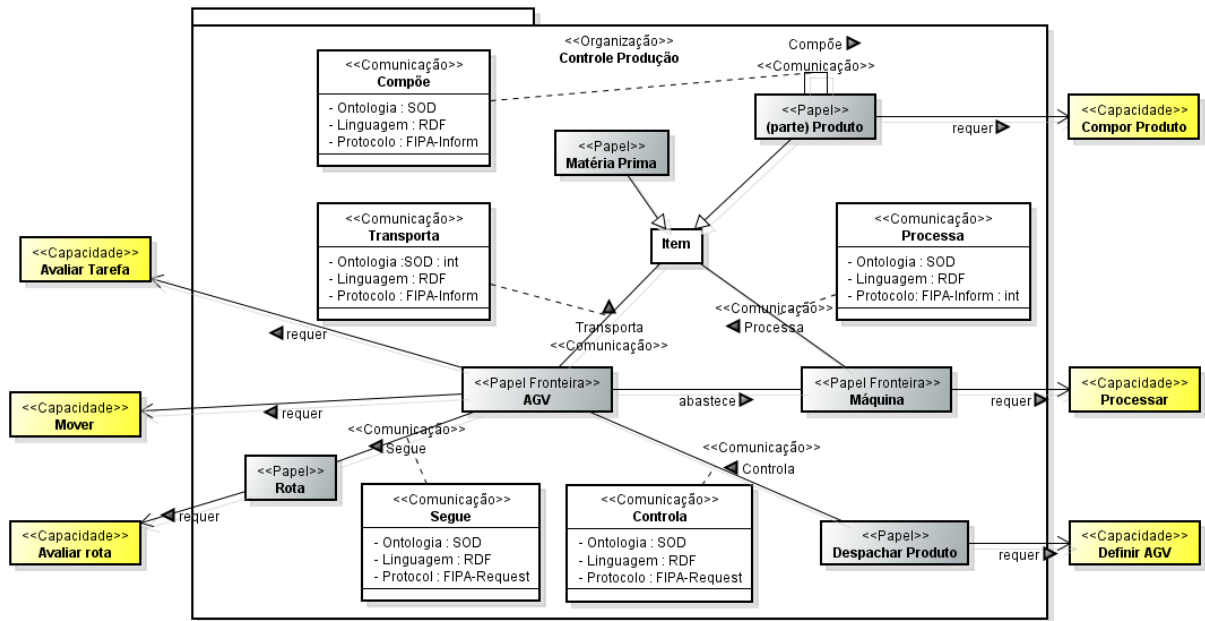


Figura. 28. COD da organização de controle da produção.

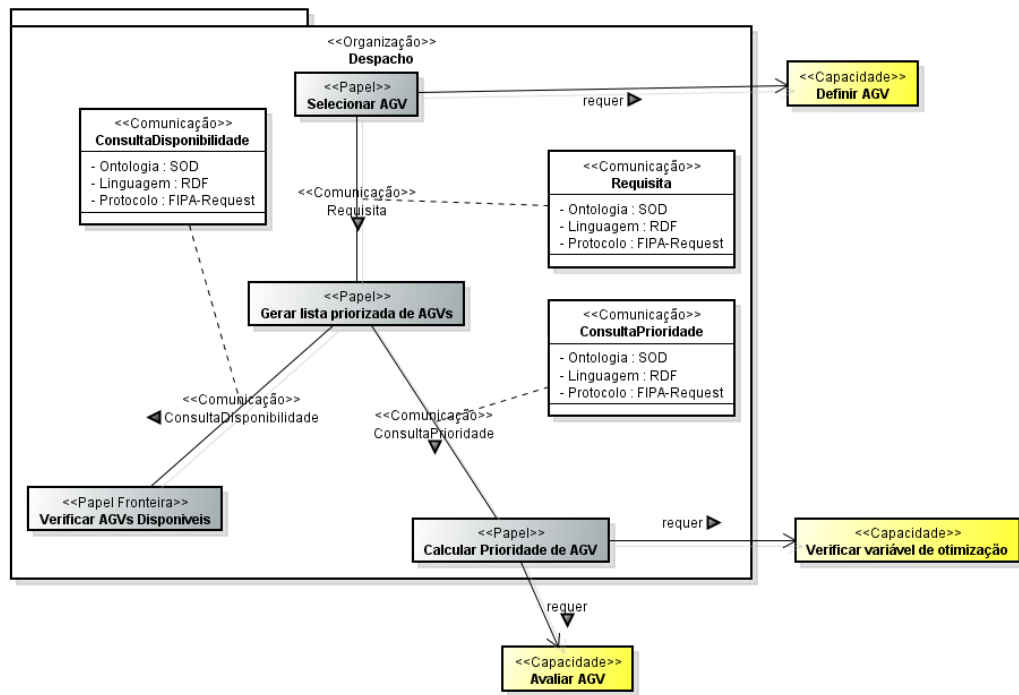


Figura. 29. COD organização despacho

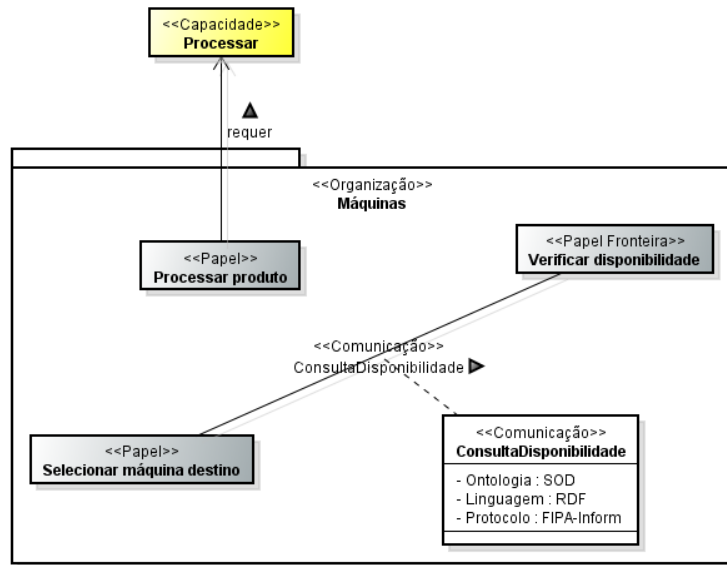


Figura. 30. COD organização máquinas.

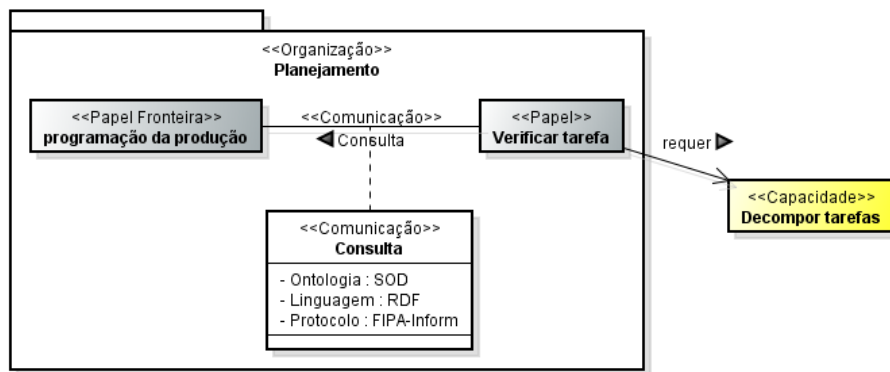


Figura. 31. COD organização Planejamento.

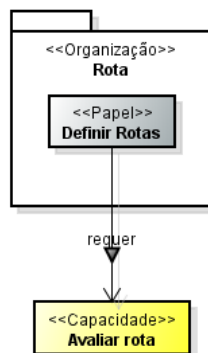


Figura. 32. COD organização Rota.

### A.2.5 (Projeto de organizações) Descrição do Comportamento do Papel (RBD – Role Behaviour Description)



Nesta atividade, foi descrito o ciclo de vida completo de um papel, usando o diagrama de atividades da UML, conforme apresentado nas Figuras 33, 34, 35, 36, 37, 38 e 39.

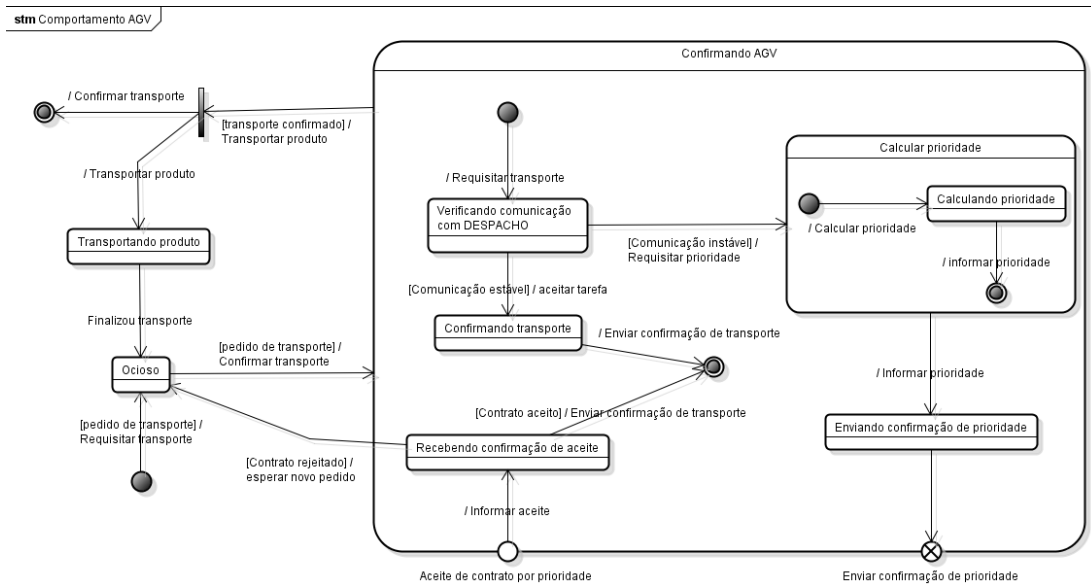


Figura. 33. Descrição do comportamento do AGV

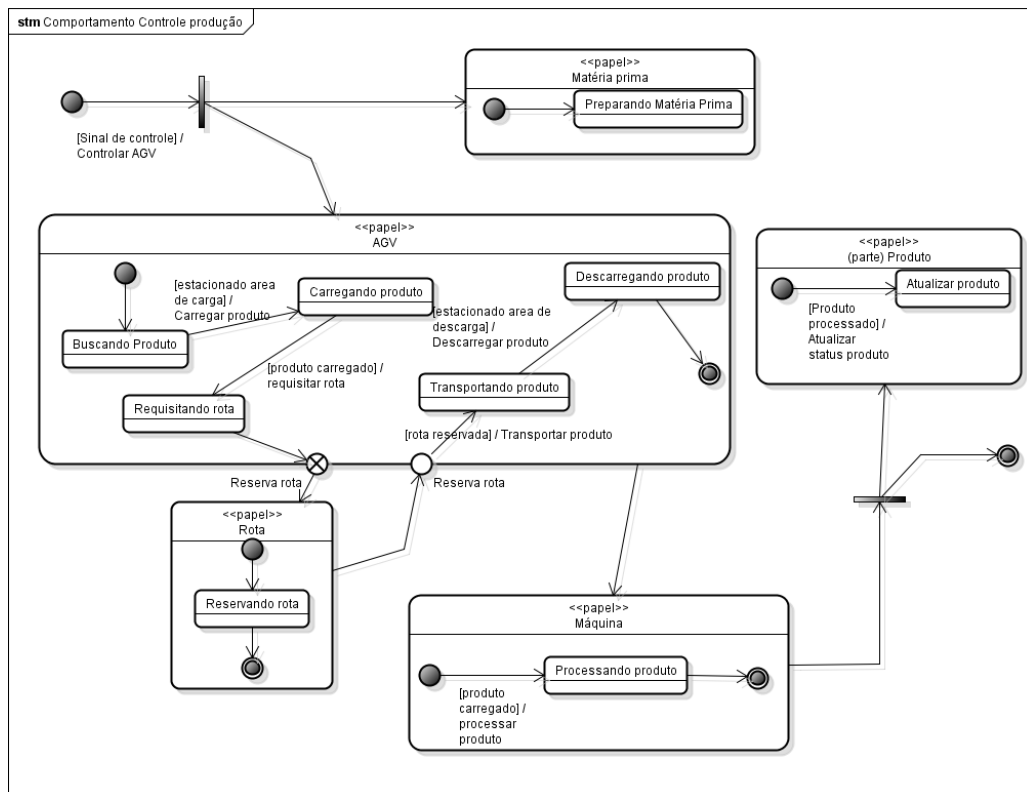


Figura. 34. Descrição do comportamento do controle de produção

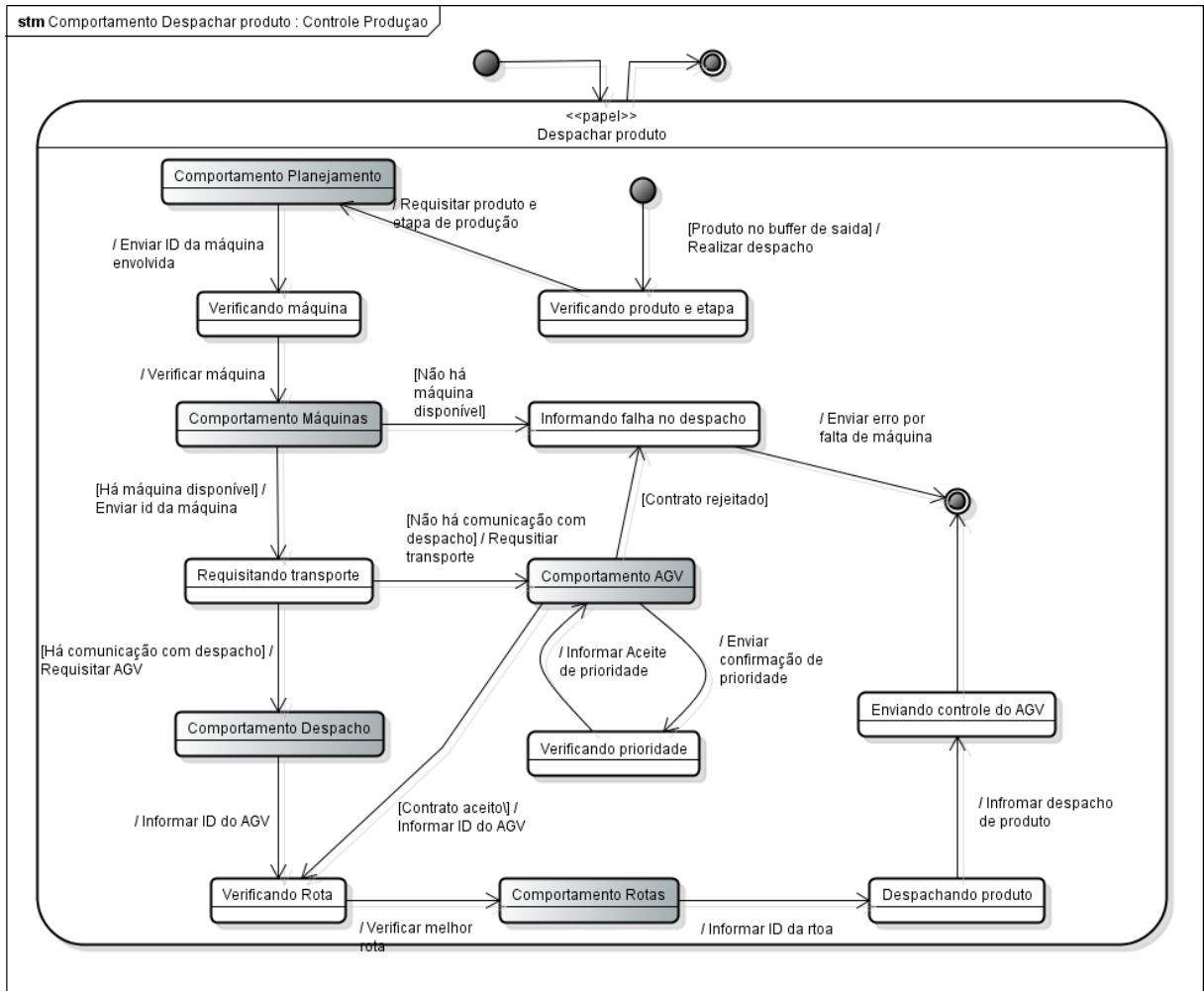


Figura. 35. Descrição do comportamento do Despatchar produto

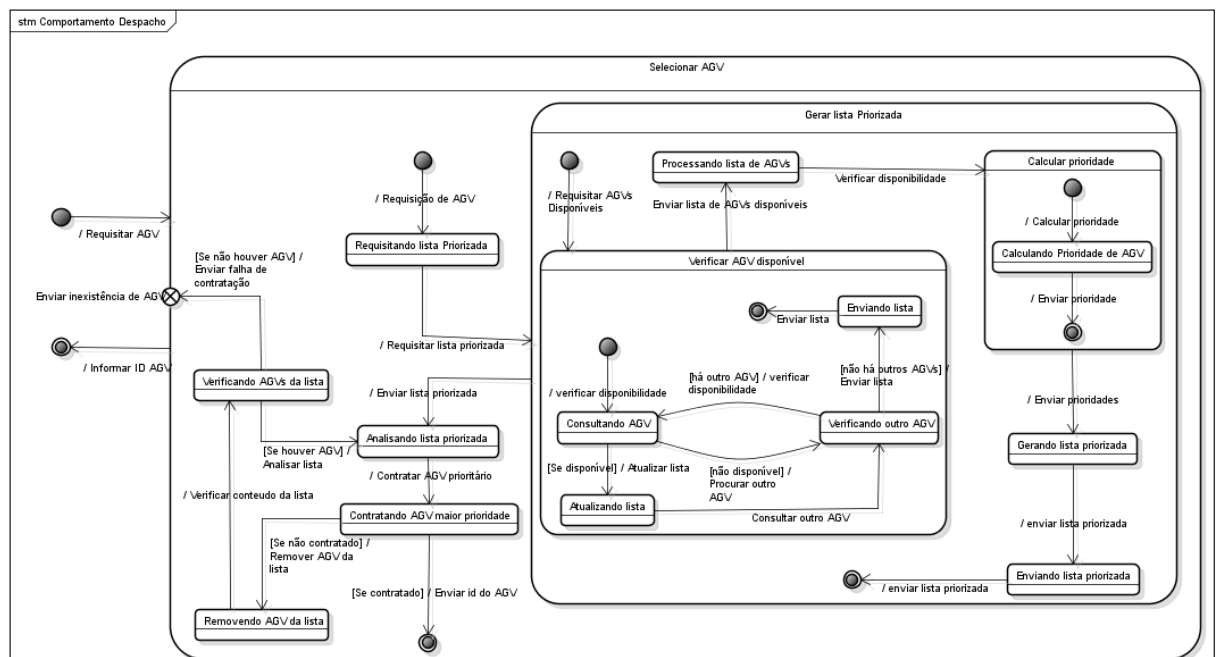


Figura. 36. Descrição do comportamento do despacho de veículos.

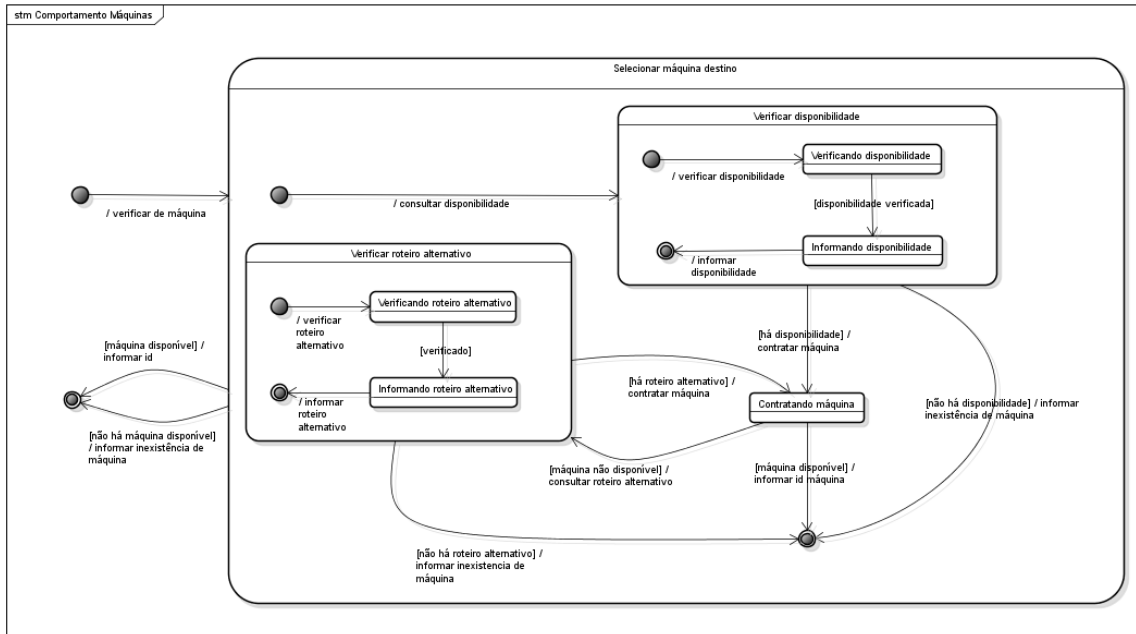


Figura. 37. Descrição do comportamento Máquinas.

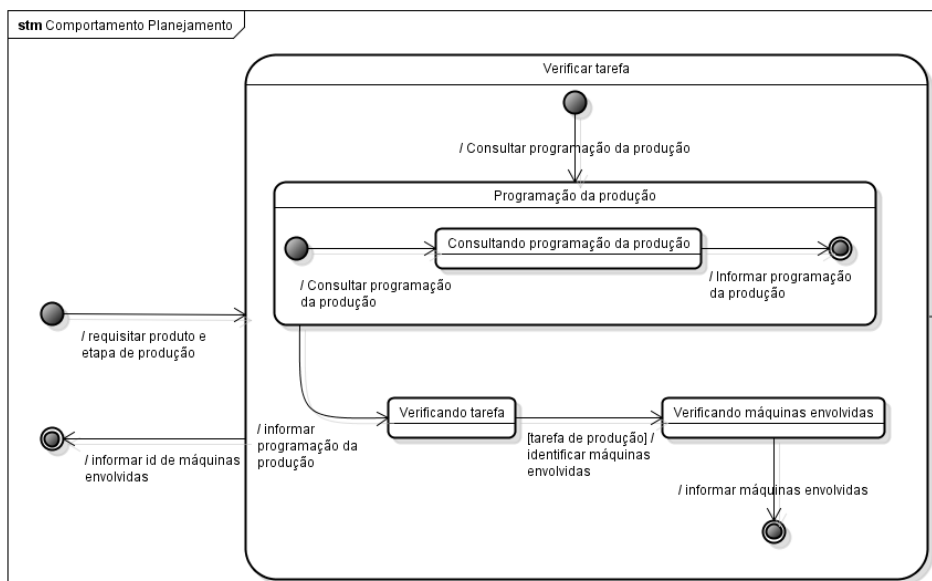


Figura. 38. Descrição do comportamento do Planejamento.

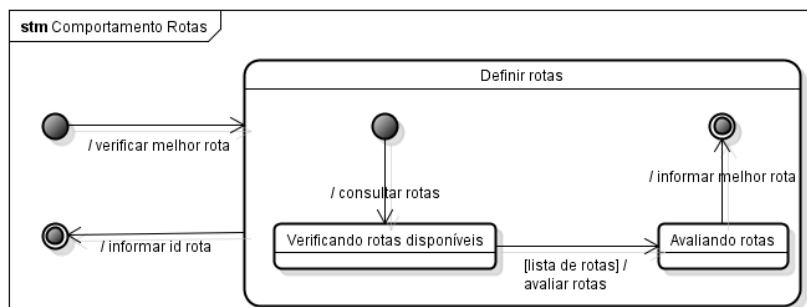


Figura. 39. Descrição do comportamento Rotas.

### A.2.6. (Projeto de organizações) Descrição do Protocolo (PD – *Protocol description*)

Nesta atividade não foi gerado nenhum diagrama, pois toda a comunicação realizada entre os agentes, seguiu o padrão de protocolos de comunicação FIPA.

### A.2.7. (Projeto de organizações) Descrição das Dependências da organização (ODD – *Organisation Dependencies Description*)

Nesta atividade foi definida para cada papel, as capacidades requisitadas por eles, assim como os serviços que realizam, conforme apresentado nas Figuras 40, 41, 42, 43, 44 e 45.

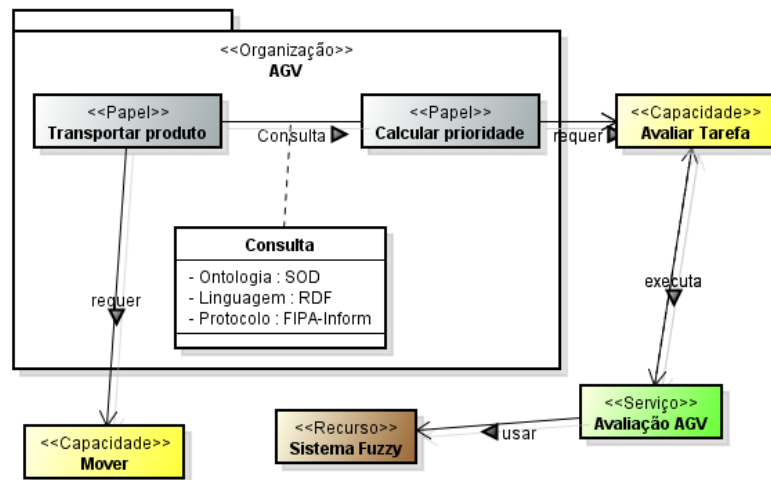


Figura. 40. Dependências dos AGVs

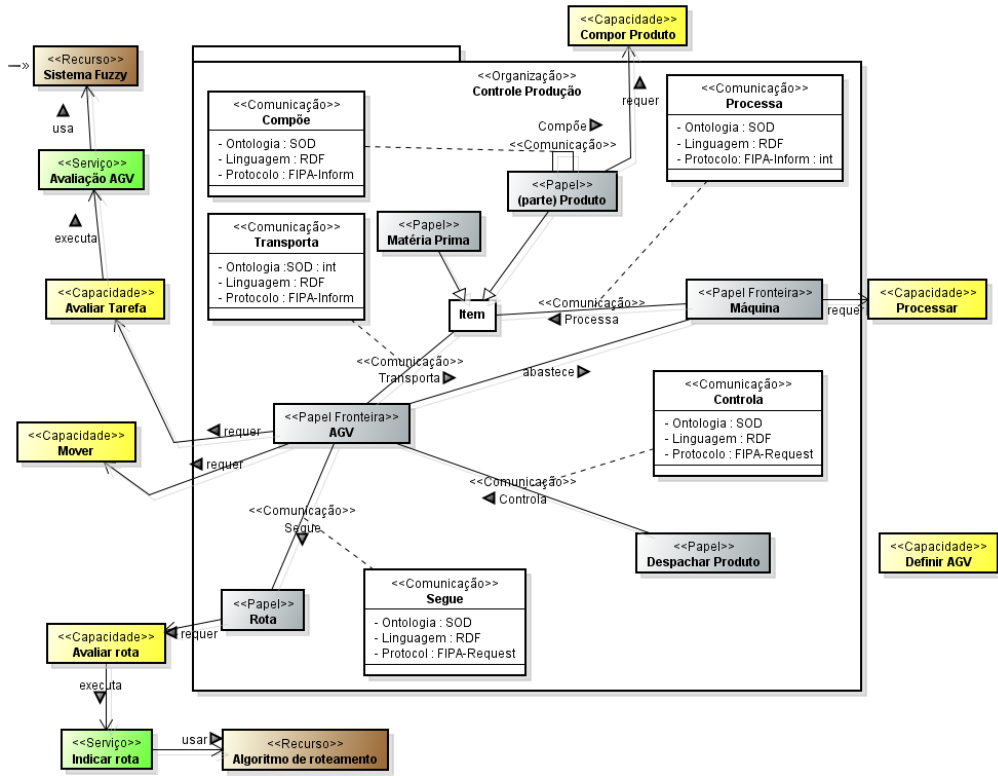


Figura. 41. Dependências do controle de produção.

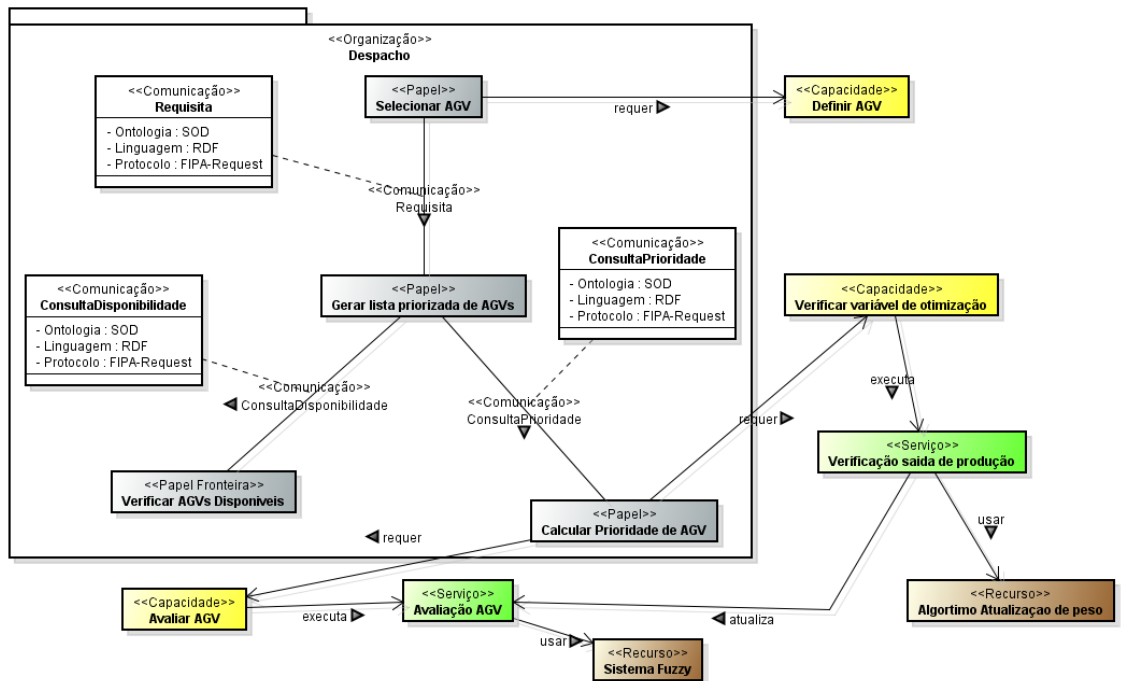


Figura. 42. Dependências do Despacho.

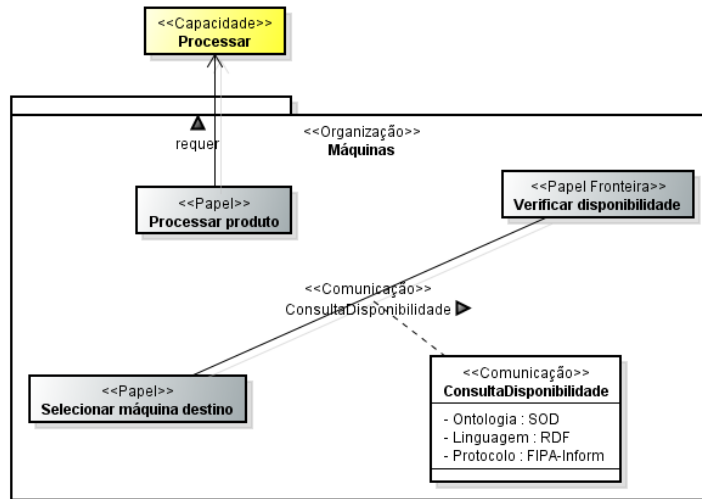


Figura. 43. Dependência das máquinas.

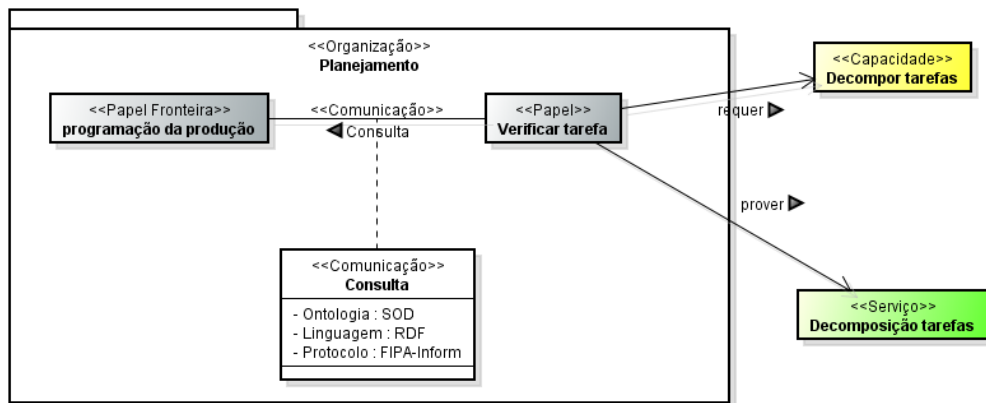


Figura. 44. Dependências do planejamento

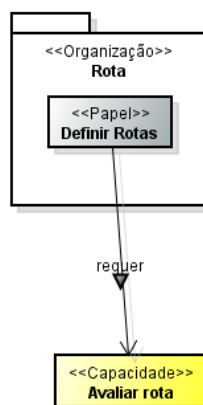


Figura. 45. Dependências da Rota.

### **A.2.8. Identificação das restrições dos papéis (RCI – Role Constraints Identification)**

Nesta atividade foram identificadas as restrições existentes entre os papéis, conforme apresentado na Figura 46.

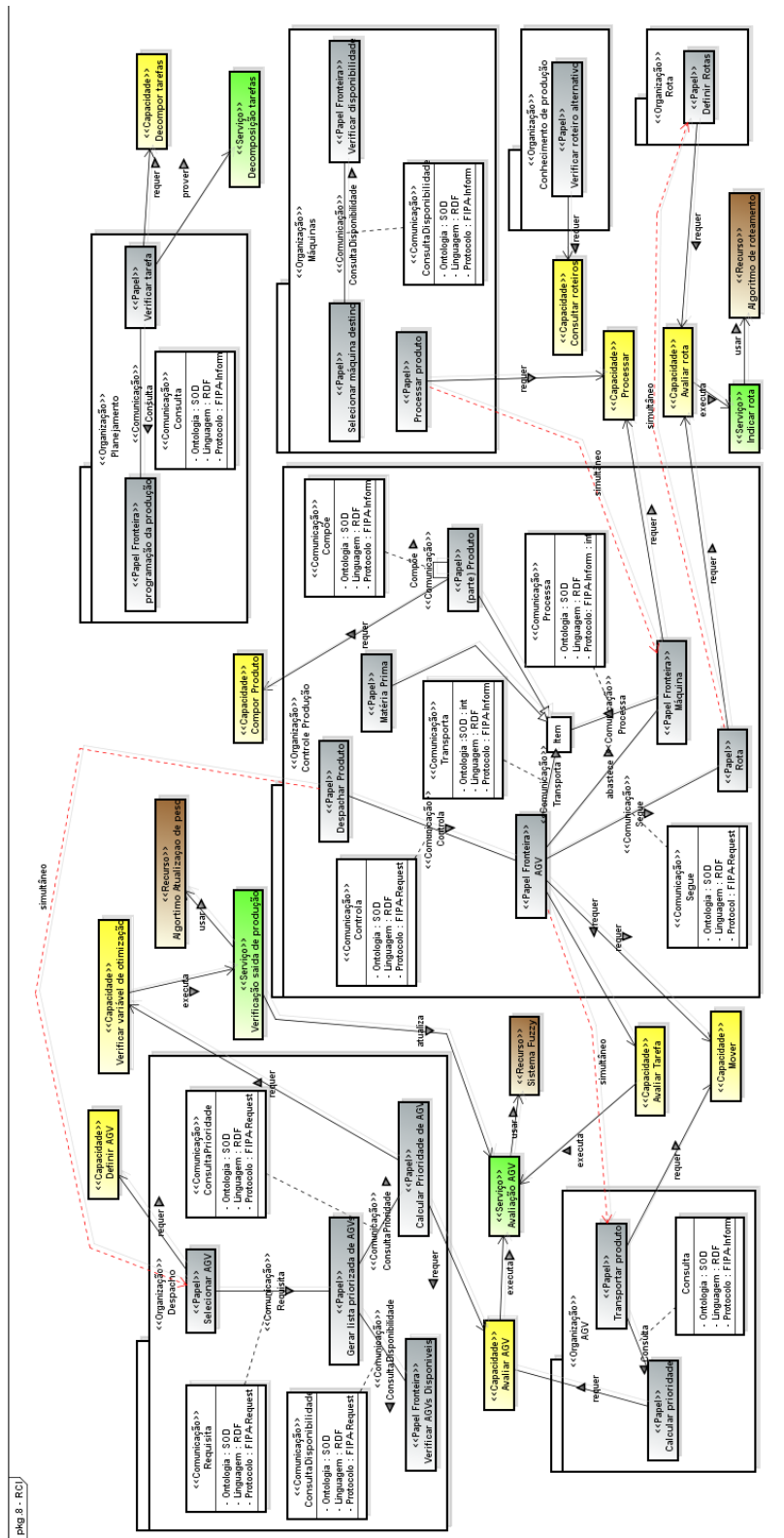


Fig. 8 - RCJ



Figura. 46. Restrições de execução dos papéis.

### A.2.9. Descrição do plano de agente (APD – *Agent Plan Description*)

Nesta atividade, foram associados os papéis aos agentes, para descrever as ações a serem tomadas por eles para serem executados de acordo com suas capacidades. Os diagramas são apresentados nas Figuras 47, 48, 49, 50 e 51.

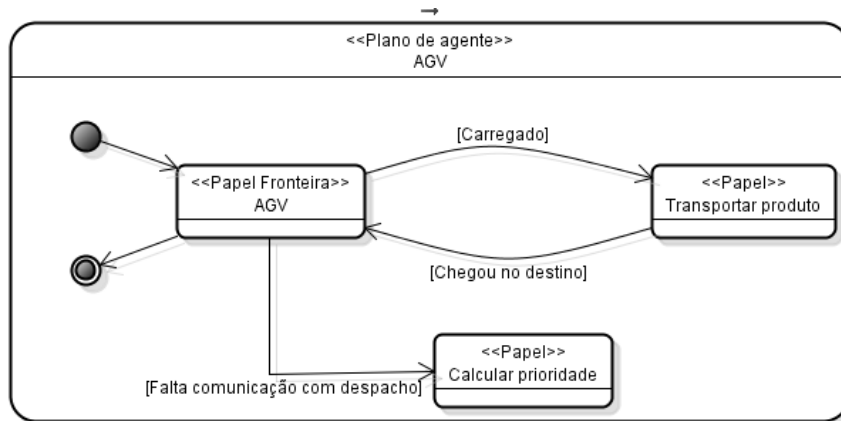


Figura. 47. Plano de agentes do AGV.

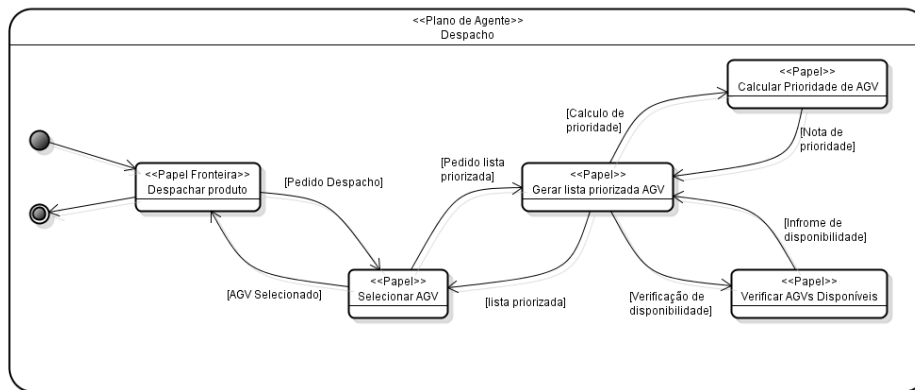


Figura. 48. Plano de agentes do despacho.

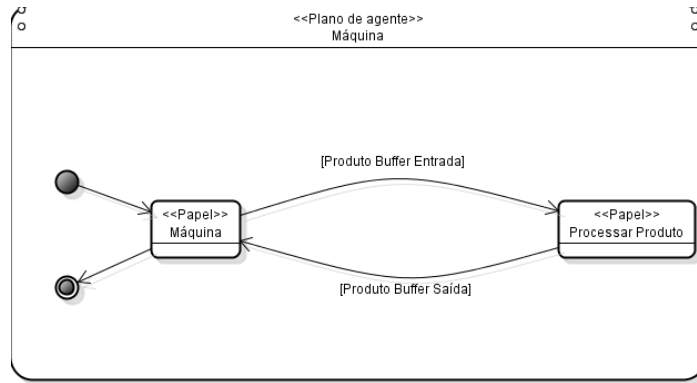


Figura. 49. Plano de agentes das Máquinas.

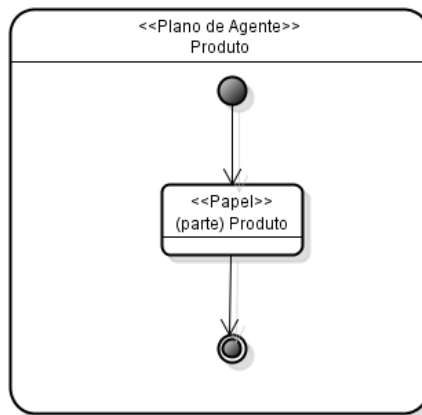


Figura. 50. Plano de agentes do produto.

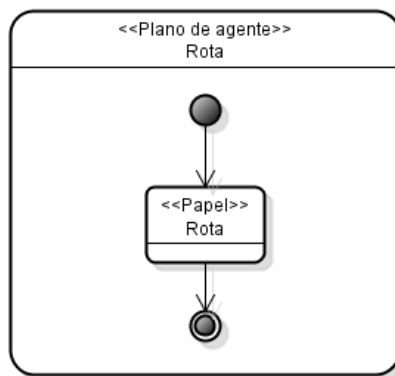


Figura. 51. Plano de agentes Rota.

### A.2.10. Projeto da holarquia (HD – Holarchy Design)

Por fim, a última atividade da fase 2 de modelagem do domínio de solução é a holarquia gerada com base nos diagramas produzidos anteriormente. A holarquia proposta pode ser vista na Figura 52.

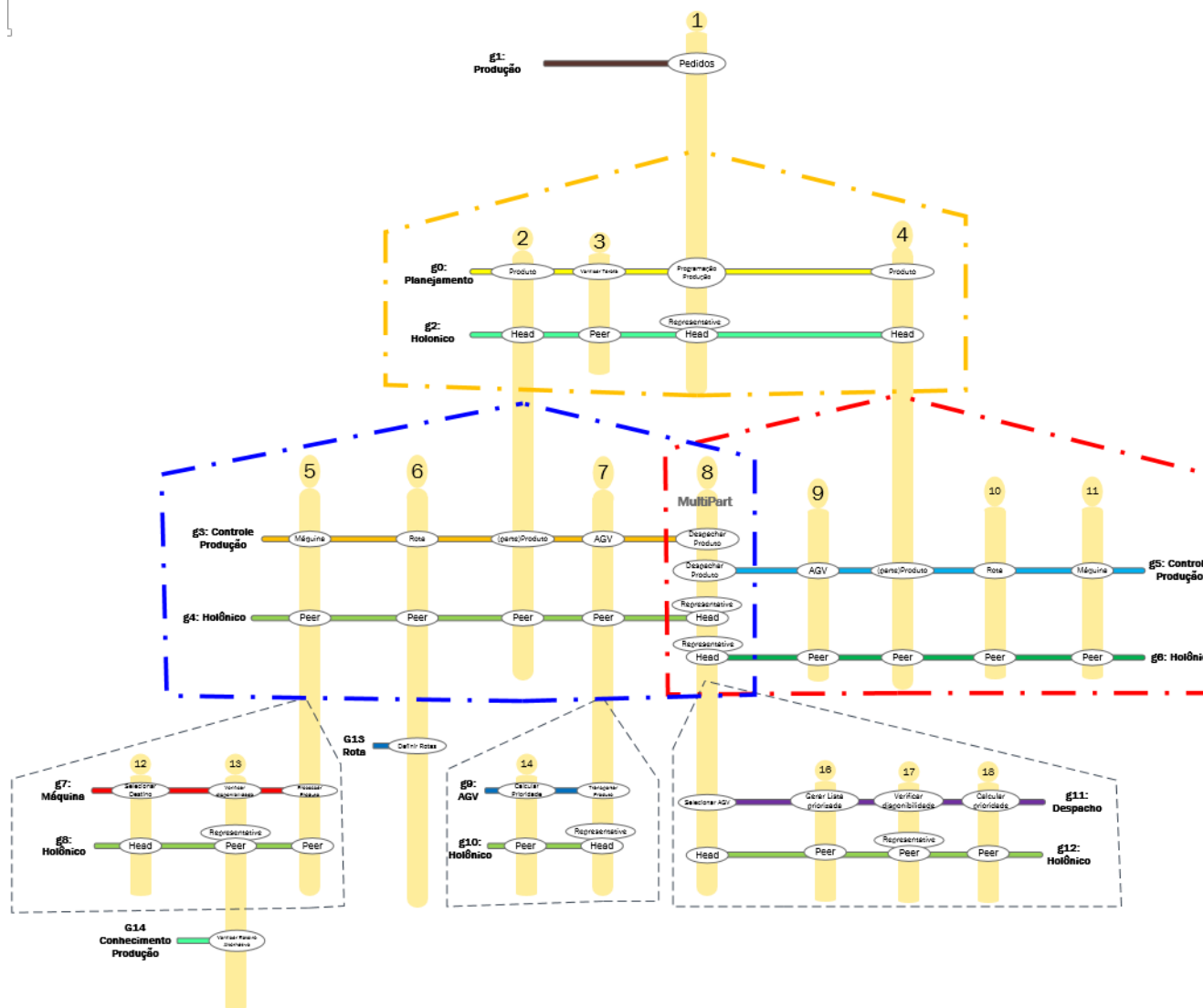


Figura. 52. Holarquia da arquitetura proposta