

- AGLETS Software Development Kit, versão 2, IBM Corporation. Disponível em: <<http://www.trl.ibm.com/aglets>>. Acessado em: 14 fevereiro 2004.
- BARBIER, F.; HENDERSON-SELLERS, B.; LE PARC, A.; BRUEL, J. Formalization of the Whole-Part Relationship in the Unified Modeling Language. Em: **IEEE Transaction on Software Engineering**, v.29, n.5, p.459-470. 2003.
- BAUER, B. UML Class Diagrams Revisited in the Context of Agent-Based Systems. Em: **Agent-Oriented Software Engineering**, WOOLDRIDGE, M.; WEISS, G.; CIANCARINI, P.(Eds). Second International Workshop, AOSE 2001, LNCS 2222 Springer, Canada, p. 101-118. 2002.
- BAUER, B.; MULLER, J.; ODELL, J. Agent UML: A Formalism for Specifying Multiagent Interaction. Em: **Agent-Oriented Software Engineering**, WOOLDRIDGE, M.; WEISS, G.; CIANCARINI, P.(Eds). Second International Workshop, AOSE 2001, LNCS 2222 Springer, Canada, p. 91-103. 2002.
- BOOCH, G. **Object-oriented Analysis and Design with Applications**. 2ed. USA: The Benjamin/Cummings Publishing Company, Inc., 1994.
- BOOCH, G.; RUMBAUGH, J.; JACOBSON I. The Unified Modeling Language User Guide. 1ed. Massachusetts: Addison-Wesley Longman Publishing Co, Inc., 1999.
- BRADSHAW, J. KAoS: An open agent architecture supporting reuse, interoperability, and extensibility. Em: **Tenth Knowledge Acquisition for Knowledge-Based Systems Workshop**, 1996.
- CAIRE, G; CHAINHO, F.; EVANS, R. Agent-oriented analysis using Message/UML. In: **Agent-Oriented Software Engineering**, WOOLDRIDGE, M.; WEISS, G.; CIANCARINI, P.(Eds). Second International Workshop, AOSE 2001, LNCS 2222 Stringer, Canada, p. 119-135. 2002.
- CASTRO, J.; KOLP, M.; MYLOPOULOS, J. Towards Requirements-Driven Information Systems Engineering: The Tropos Project. Em: **The 13th international conference on advanced information systems engineering**, Oxford: Elsevier Science Ltd, v.27, n.6. p. 365-389. 2002.
- CARLEY, K. Computational organizational theory. Em: **Multiagent Systems – A Modern Approach to Distributed Artificial Intelligence**. MIT Press. 1999.
- CHEN P. The Entity Relationship Model – Towards a Unified View of Data. **ACM Transactions on Database Systems**.v.1,n.1, p. 9-36. 1976.
- DARDENNE, A.; LAMSWEERDE, A.; FICKAS, S. Goal-directed Requirements Acquisition. **Science of Computer Programming**, v.20, p.3-50. 1993.
- DELOACH, S. A. Multiagent Systems Engineering: a Methodology and Language for Designing Agent Systems. In: WAGNER, G.; YU, E. (Eds), **Anais do Agent Oriented Information Systems**, Agent-oriented Information System (AOIS99), Washington. 1999.

- DELOACH, S. A. Analysis and Design of Multiagent Systems Using Hybrid Coordination Media. Em: **Anais do 6th World Multi-Conference on Systemic**, Cybernetics And Informatics (SCI 2002), Florida. p.14-18. 2002.
- DEPKE, R.; HECKEL, R.; HUSTER, J. M. Roles in Agent-Oriented Modeling. **International Journal of Software Engineering and Knowledge Engineering**, v.11, n.3, p. 281-302. 2001.
- DEPKE, R.; HECKEL, R.; HUSTER, J. M. Formal agent-oriented modeling with UML and graph transformation. Em: **Science of Computer Programming archive: Special issue on applications of graph transformations (GRATRA 2000)**, Netherlands: Elsevier, v.44, n.2, p. 229-252. 2002.
- D'INVERNO, M.; LUCK, M. **Understanding Agent Systems**. New York: Springer. 2001.
- ELAMMARI, M.; LALONDE, W. An Agent-Oriented Methodology: High-level and Intermediate Models. In: WAGNER, G.; YU, E. (Eds), **Anais do Agent Oriented Information Systems**, Agent-oriented Information System (AOIS99), Washington. 1999.
- EVANS, A. S.; FRANCE, R. B.; LANO, K. C.; RUMPE, B. Developing the UML as a formal modeling notation. In: MULER, P.; BERZIVIN, J. (Eds.), **UML'98 Beyond the notation - International Workshop**, France. 1998.
- EVANS, A. S.; FRANCE, R. B.; LANO, K. C.; RUMPE, B. Metamodeling semantics of UML. In: KILOV, H. (Ed.), **Behavioral Specifications for Businesses and Systems**. Kluwer, 1999.
- FERBER, J.; GUTKNECHT, O.; MICHAEL F. From Agents to Organizations: an Organizational View of Multi-Agent Systems. Em: **Anais do Fourth International Workshop on Agent-Oriented Software Engineering (AOSE)**, Australia, 2003.
- FIPA Abstract Architecture Specifications, version L-2002, FIPA. Disponível em: <<http://www.fipa.org/repository/architecturespecs.html>>. Acessado em: 14 de fevereiro 2004.
- FIPA Modeling: Agent Class Superstructure Metamodel, version 2003-02-10, FIPA. Disponível em: <<http://www.fipa.org/activities/modeling.html>>. Acessado em: 14 de fevereiro 2004.
- FOX, M. S.; BARBUCEANU, M.; TEIGEN, R. Agent-oriented Supply-chain Management. **The International Journal of Flexible Manufacturing**, v.12, p.165-188. 2000.
- FRANCE, R.; EVANS, A.; LANO, K.; RUMPE, B. The UML as a formal modeling notation. **Computer Standards & Interfaces**. v.19, p.325-334. 1998.
- GAMMA, E; HELM, R.; JOHNSON, R.; VLISSIDES, J. **Design Patterns**. 1ed. Addison-Wesley Pub Co. 1995.
- GARLAN, D.; PERRY, D. Introduction to the Special Issue on Software Architecture. Em: **IEEE Transactions on Software Engineering**. v.21, n.4. 1995.
- GLASER, N.; MORIGNOT, P. The Reorganization of Societies of Autonomous Agents Em: BOMAN, M.; VELDE, W. (Edts.) **Anais do 8th European Workshop on Modeling Autonomous Agents in a Multi-Agent World**, Sweden, p. 98- 111. 1997.
- GOMAA, A. **Designing Concurrent, Distributed, And Real-Time Applications with UML**. 1ed. Addison-Wesley Pub Co, 2000.
- HAREL, D. Statecharts: A visual formalism for complex systems. **Science of Computer Programming**. v.8, p.231-274. 1987.

- HE, M.; JENNINGS, N.; LEUNG, H. On agent-mediated electronic commerce. Em: **IEEE Transaction on Knowledge and Data Engineering**, v.15, n.4, p.985-1003. 2003.
- HOWDEN, N.; RÖNNQUIST, R.; HODGSON, A.; LUCAS, A. JACK Intelligent Agents – Summary of an Agent Infrastructure. Em: **Second International Workshop on Infrastructure for Agents, MAS, and scalable MAS**, 5th International Conference on Autonomous Agents, Canada, 2001.
- HUGET, M. Agent UML Class Diagrams Revisited. In: BAUER, B.; FISCHER, K.; MULLER, J.; RUMPE, B. (Eds.) **Anais do Agent Technology and Software Engineering (AgeS)**, Alemanha, 2002a.
- HUGET, M. An Application of Agent UML to Supply Chain Management Em: GIORGINI, P.; LESPERANCE, Y.; WAGNER, G.; YU, E. (Eds.) **Anais of Agent Oriented Information System (AOIS-02)**, Itália, 2002b.
- HUGET, M. Generating Code for Agent UML Sequence Diagrams. Em: BAUER, B.; FISCHER, K.; MULLER, J.; RUMPE, B. (Eds.) **Anais do Agent Technology and Software Engineering (AgeS)**, Alemanha, 2002c.
- HUHNS, M.; SINGH, M. Agents and Multi-agent Systems: Themes, Approaches and Challenges. Em: **Readings in Agents**, HUNS, M.; SINGH, M. (Eds.), Morgan Kaufmann, p.1-23, 1998.
- IBM Rational Rose XDE Developer, version 2003, IBM Corporation. Disponível em: <<http://www.ibm.com/rational>>. Acessado em: 14 de fevereiro 2004.
- IGLESIAS, C.; GARIJO, M.; GONZALES, J. A survey of agent-oriented methodologies. Em: MULLER, J.; SINGH, M.; RAO, A. (Eds.) **Anais of the Fifth International Workshop on Agent Theories, Architectures and Languages**, LNAI, Springer. 1999.
- ISHIDA, I.; GASSER, L.; YOKOO, M. Organization self design of production systems. Em: **IEEE Transaction on Knowledge and Data Engineering**, v.4, n.2, p.123-134. 1992.
- JADE: Java Agent Development Framework, version 3.1, Telecom Italian Lab. Disponível em: <<http://sharon.cselt.it/projects/jade>>. Acessado em: 14 de fevereiro 2004.
- JENNINGS, N. Agent-Oriented Software Engineering. Em: **Anais do Twelfth International Conference on Industrial and Engineering Applications of Artificial Intelligence**, Egito, p.4-10. 1999.
- JENNINGS, N. On Agent-based Software Engineering. Em: **Artificial Intelligence**, v.117, n.2, p. 277-296, 2000.
- JENNINGS, N.; WOOLDRIDGE, M. Applications of Intelligent Agents. Em: JENNINGS, J.; WOOLDRIDGE, M. (Eds.), **Agent Technology: Foundations, Applications, and Markets**, v.3, n.28. 1998.
- KINNY, D. The # Calculus: An Algebraic Agent Language. Em: **Intelligent Agents VIII**, Springer, v.2333, p.32-50. 2002.
- KRISTENSEN, B. Object-Oriented Modeling with Roles. Em: **Anais do 2nd International Conference on Object-Oriented Information Systems**, Irlanda, 1995.
- KRISTENSEN, B. Subject Composition by Roles. Em: **Anais Object-Oriented Information Systems**, Australia, 1997.
- KRISTENSEN, B.; OSTERBYE, K. Roles: Conceptual Abstraction Theory and Practical Language Issues. Em: **Theory and Practice of Object Systems**, v.2, n.3, p. 143-160. 1996.

- KUNCAK, V.; LAM, P.; RINARD, M. Role analysis. Em: **Anais 29th Annual ACM Symposium on the Principles of Programming Languages**, Estados Unidos, 2002.
- LETIER, E.; LAMSWEERDE, A. Agent-based Tactics for Goal-Oriented Requirements Elaboration. Em: **Anais of International Conference on Software Engineering (ICSE02)**, Florida, 2002.
- LIND, J. **MASSIVE: Software Engineering for Multi-agent Systems**. PhD Dissertation, Universität des Saarlandes, Saarbrücken, Alemanha, 2000.
- LIND, J. Issues in agent-oriented software engineering. Em: CIANCARINI, P.; WOOLDRIDGE, M. (Eds.) **Agent-Oriented Software Engineering**, LNCS 1957, Alemanha: Springer, p.45-58. 2001.
- LOMUSCIO, A. R.; WOOLDRIDGE, M.; JENNINGS, N. A classification scheme for negotiation in electronic commerce. Em: **International Journal of Group Decision and Negotiation**, v.12, n.1, p.31-56. 2003.
- MASIF: Mobile Agent Facility Specification, version 2000-01-02, OMG. Disponível em: <http://www.omg.org/technology/documents/formal/mobile_agent_facility.htm>. Acessado em: 14 de fevereiro 2004.
- MOF: Meta Object Facility Specification, version 1.4, OMG. Disponível em: <<http://www.omg.org/cwm>>. Acessado em: 14 de fevereiro 2004.
- MYLOPOULOS, J., KOLPS, M., CASTRO, J. UML for Agent-Oriented Software Development: the Tropos Proposal. Em: **Anais of the Fourth International Conference on the Unified Modeling Language**, Canada, 2001.
- ODELL, J.; PARUNAK, H.; BAUER, B. Extending UML for Agents. Em: WAGNER, G.; LESPERANCE, Y.; YU, E. (Eds.), **Anais of the Agent-Oriented Information Systems Workshop**, Austria, p. 3-17. 2000.
- ODELL, J.; PARUNAK, H.; BAUER, B. Representing agent interaction protocols in UML. Em: CIANCARINI, P.; WOOLDRIDGE, M. (Eds.) **Agent-Oriented Software Engineering**, LNCS 1957, Alemanha: Springer, p.121-140. 2001.
- ODELL, J.; PARUNAK, H.; FLEISHER, M. The Role of Roles in Designing Effective Agent Organizations. Em: GARCIA, A.; LUCENA, C.; ZAMBONELI, F.; OMICINI, A; CASTRO, J., (Eds.) **Software Engineering for Large-Scale Multi-Agent Systems**. LNCS 2603, Alemanha: Springer, 2003.
- OMG: Object Management Group. Disponível em: <<http://www.omg.org>>. Acessado em: 14 de fevereiro 2004.
- OMG. Agent Platform Special Interest Group: Agent Technology. Em: **Green Paper**. version 1.0, 2000.
- PADGHAM, L; WINIKOFF, M. Prometheus: A Methodology for Developing Intelligent Agents, Em: **Proceedings of the First International Joint Conference on Autonomous Agents and Multi-Agent Systems**, Itália. 2002.
- PARUNAK, H.; ODELL, J. Representing social structures in UML. Em: WOOLDRIDGE, M.; WEISS, G.; CIANCARINI, P. (Eds.) **Agent-Oriented Software Engineering II**, LNCS 2222, Alemanha: Springer, p.1-16. 2002.
- PETRIE, C. Agent-Based Software Engineering. Em CIANCARINI, P.; WOOLDRIDGE, M. (Eds.) **Agent-Oriented Software Engineering**, LNCS 1957, Alemanha: Springer, p.59-76. 2001.
- PRECISE UML Group, Disponível em: <<http://www.puml.org/>>, Acessado em: 14 de fevereiro 2004.

- SHOHAM, Y. Agent0: A Simple Agent Language and its Interpreter. Em: **Proceedings of the Ninth National Conference on Artificial Intelligence**, p.704–709, 1991.
- SHOHAM, Y. Agent-Oriented Programming. **Artificial Intelligence**, v.60, 1993.
- SILVA, V.; CHOREN, R.; LUCENA, C. Using the MAS-ML to Model a Multi-Agent System. Em: LUCENA, C.; GARCIA, A.; ROMANOVSKY, A.; CASTRO J.; ALENCAR, P. (Eds.) **Software Engineering for Large-Scale Multi-Agent Systems II**, LNCS 2940, Springer, 2004a.
- SILVA, V.; LUCENA, C. Extending the UML Sequence Diagram to Model the Dynamic Aspects of Multi-Agent Systems, Em: **Software and System Modeling**, 2004b. (em revisão).
- SILVA, V.; LUCENA, C. From a Conceptual Framework for Agents and Objects to a Multi-Agent System Modeling Language, Em: SYCARA, K.; WOOLDRIDGE, M. (Edts.), **Journal of Autonomous Agents and Multi-Agent Systems**, Kluwer Academic Publishers, 2004c.
- SILVA, V.; LUCENA, C.; ALENCAR, P.; COWAN, C. **A Model-Based Transformational Approach for Implementing Multi-Agent Systems**, Technical Report CS2004-12, School of Computer Science, University of Waterloo, Canada, 2004d.
- SILVA, V.; GARCIA, A.; BRANDAO, A.; CHAVEZ, C.; LUCENA, C.; ALENCAR, P. Taming Agents and Objects in Software Engineering. Em: GARCIA, A.; LUCENA, C.; ZAMBONELI, F.; OMICINI, A.; CASTRO, J., (Eds.) **Software Engineering for Large-Scale Multi-Agent Systems**. LNCS 2603, Berlin: Springer, 2003.
- TOGETHER Tool, version 6.0, Borland Software Corporation. Disponível em: <<http://www.borland.com/together>>. Acessado em: 14 de fevereiro 2004.
- TURSKI, W; MAIBAUM, T. **Specification of computer programs**, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, 1987.
- TXL: The TXL programming language, version 10.3, Software Technology Laboratory. Disponível em: <<http://www.txl.ca>>. Acessado em: 14 de fevereiro 2004.
- UML: Unified Modeling Language Specification, versão 2.0, OMG, Disponível em: <<http://www.omg.org/uml/>>. Acessado em: 14 de fevereiro 2004.
- XMI: XML Metada Interchange, versão 2.0, OMG. Disponível em: <<http://www.omg.org/technology/documents/formal/xmi.htm>>. Acessado em: 14 de fevereiro 2004.
- WAGNER, G. The Agent-Object-Relationship Metamodel. Em: **Proceedings of the 2nd International Symposium: From Agent Theory to Agent Implementation**, Austrália, 2000.
- WAGNER, G. A UML Profile for External AOR Models. Em: **Proceedings of 3rd International Workshop on Agent-Oriented Software Engineering (AOSE-2002)**, Itália, 2002.
- WAGNER, G. The Agent-Object-Relationship Metamodel: Towards a Unified View of State and Behavior. In: **Information Systems**, v.28, n.5, 2003.
- WEERASOORIYA, D.; RAO, A; RAMMAMOHANARAO, K. Design of a concurrent agent-oriented language. Em: WOOLDRIDGE, M., JENNINGS, N. (Eds) **Intelligent Agents**, Alemanha: Springer, p. 386-402, 1995.
- WEISS, G. **Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence**. MIT Press, Massachusetts, 1999.

- WOOD, M.; DELOACH, S. An overview of the multi-agent systems engineering methodology. Em: CIANCARINI, P.; WOOLDRIDGE, M., (Eds.) **Agent-Oriented Software Engineering**, Springer, 2001.
- WOOLDRIDGE, M.; JENNINGS, N.; KINNY, D. The Gaia methodology for agent-oriented analysis and design. Em: **Journal of Autonomous Agents and Multi-Agent Systems**, v.3, p.285–312, 2000.
- WOOLDRIDGE, M.; CIANCARINI, P. Agent-Oriented Software Engineering: the State of the Art. Em: CIANCARINI, P.; WOOLDRIDGE, M. (Eds.) **Agent-Oriented Software Engineering**, LNCS 1957, Berlin: Springer, p. 1-28. 2001.
- YU, L.; SCHMID, B. A Conceptual Framework for Agent-Oriented and Role-Based Work on Modeling. Em: WAGNER, G.; YU, E. (Eds.). **Anais do 1st International Workshop on Agent-Oriented Information Systems**, 1999.
- ZAMBONELLI, F.; JENNINGS, N.; WOOLDRIDGE, M. Organizational abstractions for the analysis and design of multi-agent systems. Em: CIANCARINI, P.; WOOLDRIDGE, M. (Eds.) **Agent-Oriented Software Engineering**, LNCS 1957, Berlin: Springer, p. 127-141. 2001.
- ZAMBONELLI, F.; PARUNAK, H. From design to intention: signs of a revolution. Em: **Anais of the First International Conference on Autonomous Agents and Multi-Agent Systems**, p. 455-456. 2002.

Apêndice I

A Gramática MAS-ML

Este apêndice apresenta duas gramáticas parcialmente descritas no Capítulo 5.3. A gramática MAS-ML ilustrada na Figura 119 define a estrutura das entidades, propriedades e relacionamentos descritos na entrada da fase de transformação das entidades e a estrutura dos relacionamentos e classes gerados na saída. A gramática descrita na Figura 120, que é um sub-conjunto da gramática MAS-ML, define a estrutura dos relacionamentos e classes descritos na entrada da fase de transformação dos relacionamentos e a estrutura do conjunto de classes de saída.

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Grammar
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

compounds
    != // /* */ /** <{ }>
        @associates @link @label @clientRole @clientCardinality
    @supplierRole @supplierCardinality
end compounds

tokens
    multi  "\\d+\\.\\d+" | "\\d+\\.\\*" | "\\d+"
end tokens

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% PROGRAM
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

define program
    [relationshipDescription+] [basicEntities] [otherEntities]
end define

define basicEntities
    [environment] [NL] [NL]
    [organization] [NL] [NL]
    [agentRole] [NL] [NL]
    [agent] [NL] [NL]
end define

define otherEntities
    [entity*]
end define

define entity
    [environment] [NL] [NL]
    | [organization] [NL] [NL]

```

```

        | [agentRole] [NL][NL]
        | [agent] [NL][NL]
        | [objectRole] [NL][NL]
        | [entityClass] [NL][NL]
    end define

    define relationshipDescription
        'RELATIONSHIP [relationship][NL] [NL]
    end define

    define relationship
        [ownership]
        | [inhabit]
        | [play]
        | [control]
        | [dependency]
        | [association]
        | [aggregation]
        | [specialization]
    end define

    %%%%%%%%%%%%%%%%%%%%%%%%%%
    % RELATIONSHIPS
    %%%%%%%%%%%%%%%%%%%%%%%%%%

    %%%%%%%%%%%%%%%%%%%%%%%%%%
    % ownership
    define ownership
        [organizationMainID] '1 'OWNS [multiplicity][agentRoleID]
        | [organizationID] '1 'OWNS [multiplicity][objectRoleID]
    end define

    define organizationMainID
        [organizationID]
    end define

    %%%%%%%%%%%%%%%%%%%%%%%%%%
    % inhabit
    define inhabit
        [agentID][multiplicity] 'INHABITS '1 [environmentID]
        | [organizationMainID][multiplicity] 'INHABITS '1
        [environmentID]
        | [entityClassID][multiplicity] 'INHABITS '1 [environmentID]
    end define

    %%%%%%%%%%%%%%%%%%%%%%%%%%
    % play
    define play
        [agentID] '1 'PLAYS [multiplicity][agentRoleID] 'IN
        [organizationID]
        | [organizationID] '1 'PLAYS [multiplicity][agentRoleID] 'IN
        [organizationID]
        | [entityClassID] '1 'PLAYS [multiplicity][objectRoleID] 'IN
        [organizationID]
    end define

    %%%%%%%%%%%%%%%%%%%%%%%%%%
    % specialization
    define specialization
        [agentID] 'SPECIALIZES [agentID]
        | [organizationID] 'SPECIALIZES [organizationID]

```

```

| [environmentID] 'SPECIALIZES [environmentID]
| [agentRoleID] 'SPECIALIZES [agentRoleID]
| [objectRoleID] 'SPECIALIZES [objectRoleID]
| [entityClassID] 'SPECIALIZES [entityClassID]
end define

%%%%%%%%%%%%%%
% control
define control
| [agentRoleID] [multiplicity] 'CONTROLS
[multiplicity] [agentRoleID] [labelRel]
end define

%%%%%%%%%%%%%%
% dependency
define dependency
| [agentRoleID] [multiplicity] [role] 'DEPENDS 'ON
[role] [multiplicity] [agentRoleID] [labelRel]
| [agentRoleID] [multiplicity] [role] 'DEPENDS 'ON
[role] [multiplicity] [objectRoleID] [labelRel]
| [objectRoleID] [multiplicity] [role] 'DEPENDS 'ON
[role] [multiplicity] [objectRoleID] [labelRel]
| [entityClassID] [multiplicity] [role] 'DEPENDS 'ON
[role] [multiplicity] [entityClassID] [labelRel]
end define

%%%%%%%%%%%%%%
% association
define association
| [objectRoleID] [multiplicity] [role] 'ASSOCIATED 'WITH
[role] [multiplicity] [objectRoleID] [labelRel]
| [agentRoleID] [multiplicity] [role] 'ASSOCIATED 'WITH
[role] [multiplicity] [agentRoleID] [labelRel]
| [agentRoleID] [multiplicity] [role] 'ASSOCIATED 'WITH
[role] [multiplicity] [objectRoleID] [labelRel]
| [environmentID] [multiplicity] [role] 'ASSOCIATED 'WITH
[role] [multiplicity] [environmentID] [labelRel]
| [agentID] [multiplicity] [role] 'ASSOCIATED 'WITH
[role] [multiplicity] [entityClassID] [labelRel]
| [organizationID] [multiplicity] [role] 'ASSOCIATED 'WITH
[role] [multiplicity] [entityClassID] [labelRel]
| [entityClassID] [multiplicity] [role] 'ASSOCIATED 'WITH
[role] [multiplicity] [entityClassID] [labelRel]
end define

%%%%%%%%%%%%%%
% aggregation
define aggregation
| [objectRoleID] [multiplicity] [role] 'AGGREGATES
[role] [multiplicity] [objectRoleID] [labelRel]
| [agentRoleID] [multiplicity] [role] 'AGGREGATES
[role] [multiplicity] [objectRoleID] [labelRel]
| [entityClassID] [multiplicity] [role] 'AGGREGATES
[role] [multiplicity] [entityClassID] [labelRel]
end define

define role
| [id]
| [empty]
end define

```

```

define multiplicity
    [multi]
    | [empty]
end define

define labelRel
    [id]
    | [empty]
end define

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% ENTITIES
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% AGENT
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

define agent
    'AGENT          [NL]          '(          [agentID]
[IN] [NL] [goal*] [NL] [belief*] [NL] [plan*] [NL] [actionPrePostCondition
*] [NL] [EX] ')
    | [entityClass*] %agent se transforma em class
end define

define agentID
    [id]
end define

define goal
    [leafGoal]
    | [compositeGoal]
end define

define leafGoal
    'GOAL '( [attrBody] [goalPlans+] ')'
    | 'objectGoal '= 'new 'LeafGoal '( [stringlit] ', [stringlit]
', [stringlit] ');[NL]
    'this.goals.add(objectGoal);[NL]
    [goalPlans+]
    | 'objectSubGoal '= 'new 'LeafGoal '( [stringlit] ',
[stringlit] ', [stringlit] ');[NL]
    'objectCompositeGoal.setSubGoal(objectSubGoal);[NL]
    [goalPlans+]
end define

define compositeGoal
    'GOAL '( [attrBody] [goalPlans*] 'SUBGOAL '( [goal+] ')')
    | 'objectCompositeGoal '= 'new 'CompositeGoal '( [stringlit]
', [stringlit] ', [stringlit] ');[NL]
    'this.goals.add(objectCompositeGoal);[NL]
    [goalPlans+]
    [goal*]
    | 'objectSubCompositeGoal '= 'new 'CompositeGoal '(
[stringlit] ', [stringlit] ', [stringlit] ');[NL]
    'objectCompositeGoal.setSubGoal(objectSubCompositeGoal);[NL]
    [goalPlans+]
    [goal*]
end define

define goalPlans

```

```

        'RELATED 'TO 'PLAN [stringlit]
        |'objectGoal.setPlan( [stringlit] ');[NL]
        |'objectCompositeGoal.setPlan( [stringlit] ');[NL]
        |'objectSubGoal.setPlan( [stringlit] ');[NL]
        |'objectSubCompositeGoal.setPlan( [stringlit] ');[NL]
    end define

    define attrBody
        [varType] ': [varName] '= [varValue]
    end define

    define varType
        [stringlit]
    end define

    define varName
        [stringlit]
    end define

    define varValue
        [stringlit]
        | 'null
    end define

    define belief
        'BELIEF '( [attrBody] '
        |'this.beliefs.add(new 'Belief '( [stringlit]',
        [stringlit]', [stringlit]'));[NL]
        |'this.beliefs.add(new 'Belief '( [stringlit]',
        [stringlit]', 'null '));[NL]
    end define

    define plan
        'PLAN '( [planName] [planActions*] [planGoal]')
        |[entityClass*]
        |'objectPlan '= 'new [planName] '();[NL]
        |'objectPlan.setAgent(this);[NL]
        |'this.plans.add(objectPlan);[NL]
        [planActions*]
        [planGoal]
        |'objectPlan '= 'new [planName] '();[NL]
        |'objectPlan.setOrganization(this);[NL]
        |'this.plans.add(objectPlan);[NL]
        [planActions*]
        [planGoal]
    end define

    define planActions
        'COMPOSED 'OF 'ACTION [stringlit]
        |'actionAux '= 'null;[NL]
        |'enumActions '= 'this.actions.elements();[NL]
        |'while(enumActions.hasMoreElements()) [NL]
        '{ [NL] [IN]
            |'actionAux '= '(Action)enumActions.nextElement();[NL]
            |'if
            [stringlit] ')) [NL]
            '{ [NL] [IN]
                |'objectPlan.setAction(actionAux);[NL] [EX]
            '} [NL] [EX]
        '} [NL]
    end define

```

```

define planGoal
    'RELATED 'TO 'GOAL [varName]
    |
    'goalAux '=' 'null;[NL]
    'enumGoals '=' 'this.goals.elements();[NL]
    'while(enumGoals.hasMoreElements()) [NL]
    '{ [NL] [IN]
        'goalAux '=' '(Goal)enumGoals.nextElement();[NL]
        'if      '(goalAux.getName().equals(      [stringlit]
    ')) [NL]
        '{ [NL] [IN]
            'objectPlan.setGoal(goalAux);[NL] [EX]
        '} [NL]
        'else[NL]
        '{ [NL] [IN]
            'if
[stringlit] ')) [NL]
                '{ [NL] [IN]
                    'Vector
                    'vGoals
                    '='
'goalAux.getSubGoals();[NL]
                    'Enumeration
                    'enumvGoals
                    '='
'vGoals.elements ();[NL]
                    'while  '(enumvGoals.hasMoreElements
'()) [NL]
                        '{ [NL] [IN]
                            'Goal  'theGoal  '='  '(Goal)
'enumvGoals.nextElement();[NL]
                            'if(theGoal.getName().equals(
[stringlit] ')) [NL] [IN]
                                'objectPlan.setGoal(theGoal);[NL] [EX] [EX]
                                '} [NL] [EX]
                            '} [NL] [EX]
                        '} [NL] [EX]
                    '} [NL]
end define

define goalName
    [stringlit]
end define

define planName
    [id]
end define

define actionPrePostCondition
    [action] [NL] [preCondition*] [postCondition*]
    |[entityClass*]
end define

define action
    'ACTION [actionAssignment]
    |'objectAction '=' 'new [actionAssignment] '(); [NL]
    'this.actions.add(objectAction);
end define

define actionAssignment
    [id]
end define

define preCondition

```

```

        'PRECONDITION '( [attrBody] '
        |'objectCondition '= 'new 'Condition '( [stringlit] ',
[stringlit] ', [stringlit]');[NL]
        'objectAction.setPreCondition '(objectCondition);[NL]
end define

define postCondition
    'POSTCONDITION '( [attrBody] '
    |'objectCondition '= 'new 'Condition '( [stringlit] ',
[stringlit] ', [stringlit]');[NL]
    'objectAction.setPostCondition '(objectCondition);[NL]
end define

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% ORGANIZATION

define organizationMain
    [organization]
end define

define organization
    'ORGANIZATION                                [NL]                                '(
[organizationID] [IN] [NL] [goal*] [NL] [belief*] [NL] [plan*] [NL] [action
PrePostCondition*] [NL] [axiom*] [NL] [EX] ' )
    | [entityClass*]
end define

define organizationID
    [id]
end define

define axiom
    'AXIOM '( [attrBody] '
    |'this.axioms.add(new 'Axiom '( [stringlit]', [stringlit]',
[stringlit]'));[NL]
    |'this.axioms.add(new 'Axiom '( [stringlit]', [stringlit]',
'null '));[NL]
end define

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% AGENT ROLE

define agentRole
    'AGENTROLE                                [NL]
'([agentRoleID] [IN] [NL] [roleGoal+] [NL] [belief*] [NL] [duty*] [NL] [rig
ht*] [NL] [protocolMessage*] [NL] [EX] ' )
    | [entityClass*] %agentRole se transforma em class
end define

define agentRoleID
    [id]
end define

define roleGoal
    [roleLeafGoal]
    | [roleCompositeGoal]
end define

define roleLeafGoal
    'ROLEGOAL '( [attrBody] '

```

```

        |'objectGoal '=' 'new 'LeafGoal '( [stringlit] ', [stringlit]
        ', [stringlit] ');[NL]
        'this.goals.add(objectGoal);[NL]
        |'objectSubGoal '=' 'new 'LeafGoal '( [stringlit] ',
        [stringlit] ', [stringlit] ');[NL]
        'objectCompositeGoal.setSubGoal(objectSubGoal);[NL]
    end define

    define roleCompositeGoal
        'ROLEGOAL '( [attrBody] 'SUBGOAL '( [roleGoal+] ')')
        |'objectCompositeGoal '=' 'new 'CompositeGoal '( [stringlit]
        ', [stringlit] ', [stringlit] ');[NL]
        'this.goals.add(objectCompositeGoal);[NL]
        [roleGoal*]
        |'objectSubCompositeGoal '=' 'new 'CompositeGoal '(
        [stringlit] ', [stringlit] ', [stringlit] ');[NL]
        'objectCompositeGoal.setSubGoal(objectSubCompositeGoal);[NL]
        [roleGoal*]
    end define

    define duty
        'DUTY [stringlit]
        |'this.duties.add(new 'Duty '( [stringlit] ')); [NL]
    end define

    define right
        'RIGHT [stringlit]
        |'this.rights.add(new 'Right '( [stringlit] ')); [NL]
    end define

    define protocolMessage
        [protocol][message+]
        |[entityClass*]
    end define

    define protocol
        'PROTOCOL [id]
        |'objectProtocol '=' 'new [id] '(); [NL]
        'this.protocols.add(objectProtocol); [NL]
    end define

    define message
        'MESSAGE [IN][NL]'( [labelMessage][NL][IN][opt
        content][NL][sender][NL][receiver][NL][EX] ') [NL][EX]
        |'objectMessage '=' 'new 'Message '( [stringlit] ',
        [stringlit] ', [stringlit] ', [stringlit] '); [NL]
        'objectProtocol.setMessage(objectMessage); [NL]
    end define

    define labelMessage
        'LABEL [stringlit]
    end define

    define content
        'CONTENT [stringlit]
    end define

    define sender
        'SENDER [stringlit]
    end define

```

```

define receiver
    'RECEIVER [stringlit]
end define

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% OBJECT ROLE

define objectRole
    'OBJECTROLE [NL] '('
    [objectRoleID] '(' [NL] [IN] [attributeDescription*] [NL] [methodDescription*] [NL] [EX] ')' ')'
end define

define objectRoleID
    [id]
end define

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% CLASS

define entityClass
    [opt importDef] [static] [visibility] [concret] 'class
    [entityClassID] [inheritance*] [NL] '{ [NL] [IN] [attributeDescription*]
    [methodDescription*] [EX] '}' [NL] [NL]
    [entityClass*] % uma classe pode virar mais de uma classe
    [actionPrePostCondition*] % action se transforma em class
    [plan*] % plano se transforma em class
    [protocolMessage*] % protocolo se transforma em class
end define

define importDef
    'import 'java.util.*; [NL]
end define

define inheritance
    'extends [entityClassID]
end define

define entityClassID
    [id]
end define

define static
    'static
    | [empty]
end define

define concret
    'abstract
    | [empty]
end define

define visibility
    'public
    | 'private
    | 'protected
end define

define attributeDescription

```

```

[opt
[static][visibility][concret][attribType][attribName]
[attribValue]';[NL]
end define

define coments
  '/* [NL]
  '* '@associates <{[id]}>[NL]
  '* '@link [id][NL]
  [opt comentsLabel]
  [opt comentsClientRole]
  [opt comentsClientCardinality]
  [opt comentsSupplierRole]
  [opt comentsSupplierCardinality]
  '*/[NL]
end define

define comentsLabel
  '* '@label [id][NL]
end define

define comentsClientRole
  '* '@clientRole [id][NL]
end define

define comentsClientCardinality
  '* '@clientCardinality [multi][NL]
end define

define comentsSupplierRole
  '* '@supplierRole [id][NL]
end define

define comentsSupplierCardinality
  '* '@supplierCardinality [multi][NL]
end define

define methodDescription
  [static][visibility][concret][outParameter][assignment]'(
[list inParameter] ') [method*][NL]
end define

define method
  [NL]'{ [NL][IN][methodBody*][EX][NL] ' }
  | ';'
end define

define methodBody
  [not 'import][not visibility][not 'ENVIRONMENT][not
'ORGANIZATION][not 'AGENTROLE][not 'AGENT]
  [not 'OBJECTROLE][token]
  |
  [goal+][belief*][actionPrePostCondition+][plan+][axiom*]
  | [goal+][belief*][actionPrePostCondition+][plan+]
  | [roleGoal+][belief*][duty*][right*][protocolMessage*]
end define

define assignment
  [id]
end define

```

```

define outParameter
    [id]
    | 'void
    | [empty]
end define

define inParameter
    [attribType][attribName]
    | [empty]
end define

define attribType
    [id]
end define

define attribName
    [id]
end define

define attribValue
    [id]
    | [stringlit]
    | 'new 'Vector()
end define

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% ENVIRONMENT

define environment
    [proactiveEnvironment]
    | [reactiveEnvironment]
end define

define environmentID
    [id]
end define

define proactiveEnvironment
    'ENVIRONMENT [NL]'( [NL][IN][agent] [NL][EX]')
    | [agent*] %environment se transforma em agent para depois
se transformar em classe
end define

define reactiveEnvironment
    'ENVIRONMENT [NL]'( [NL][IN][entityClass] [NL][EX]')
    | [entityClass*] %environment se transforma em classe
end define

```

Figura 119 – A gramática MAS-ML usada na fase de transformação das entidades.

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Grammar
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

compounds
    != // /* */ /** <{ }> /*#
    @associates @link @label @clientRole @clientCardinality
    @supplierRole @supplierCardinality
end compounds

tokens
    multi  "\\d+\\.\\d+" | "\\d+\\.\\*" | "\\d+"

```

```

end tokens

%%%%%%%%%%%%%%
% PROGRAM
%%%%%%%%%%%%%%

define program
    [UMLprogram*]
end define

define UMLprogram
    [relationshipDescription*][entityClass*]
end define

define relationshipDescription
    'RELATIONSHIP [relationship] [NL] [NL]
end define

define relationship
    [ownership]
    | [inhabit]
    | [play]
    | [control]
    | [dependency]
    | [association]
    | [aggregation]
    | [specialization]
end define

%%%%%%%%%%%%%%
% RELATIONSHIPS
%%%%%%%%%%%%%%

%%%%%%%%%%%%%%
% ownership
define ownership
    [id] [multiplicity] 'OWNS [multiplicity] [id]
end define

%%%%%%%%%%%%%%
% inhabit
define inhabit
    [id] [multiplicity] 'INHABITS '1 [id]
end define

%%%%%%%%%%%%%%
% play
define play
    [id] [multiplicity] 'PLAYS [multiplicity] [id] 'IN [id]
end define

%%%%%%%%%%%%%%
% specialization
define specialization
    [id] 'SPECIALIZES [id]
end define

%%%%%%%%%%%%%%
% control
define control
    [id] [multiplicity] 'CONTROLS [multiplicity][id][labelRel]

```

```

end define

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% dependency
define dependency
    [id][multiplicity][role]                'DEPENDS                'ON
    [role][multiplicity][id][labelRel]
end define

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% association
define association
    [id][multiplicity][role]                'ASSOCIATED                'WITH
    [role][multiplicity][id][labelRel]
end define

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% aggregation
define aggregation
    [id][multiplicity][role]                'AGGREGATES
    [role][multiplicity][id][labelRel]
end define

define role
    [id]
    | [empty]
end define

define multiplicity
    [multi]
    | [empty]
end define

define labelRel
    [id]
    | [empty]
end define

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% ENTITIES
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% CLASS

define entityClass
    [opt      importDef][static][visibility][concret]    'class
    [entityClassID][inheritance*][NL]'{[NL][IN]
    [comentsAttrib*][methodDescription*][EX]'}[NL][NL]
end define

define comentsAttrib
    [generalComents*][attributeDescription*]
end define

define importDef
    'import java.util.*;[NL]
end define

define static
    'static

```

```

    | [empty]
end define

define visibility
    'public
    | 'private
    | 'protected
end define

define concret
    'abstract
    | [empty]
end define

define entityClassID
    [id]
end define

define inheritance
    'extends [entityClassID]
end define

define attributeDescription
    [static][visibility][concret][attribType][attribName]      '='
    [attribValue]';[NL]
end define

define generalComents
    '/*

    [NL]
    [coments*]
    '* '@link [id][NL]
    [opt comentsLabel]
    [opt comentsClientRole]
    [opt comentsClientCardinality]
    [opt comentsSupplierRole]
    [opt comentsSupplierCardinality]
    '*/[NL]
    [comentsDepAttrib*]
end define

define coments
    '* '@associates <{[id]}>[NL]
    | '* '@associates [token][NL]
end define

define comentsLabel
    '* '@label [labelRel][NL]
end define

define comentsClientRole
    '* '@clientRole [role][NL]
end define

define comentsClientCardinality
    '* '@clientCardinality [multiplicity][NL]
end define

define comentsSupplierRole
    '* '@supplierRole [role][NL]

```

```

end define

define comentsSupplierCardinality
    '* '@supplierCardinality [multiplicity] [NL]
end define

define comentsDepAttrib
    '/*# [id] [id] '; '*/[NL]
    |[attributeDescription]
end define

define varType
    [stringlit]
    |[id]
end define

define varName
    [stringlit]
    |[id]
end define

define varValue
    [stringlit]
    |[id]
    | 'null
end define

define methodDescription
    [static] [visibility] [concret] [outParameter] [assignment] ' (
    [list inParameter] ') [method*] [NL]
end define

define outParameter
    [id]
    | 'void
    | [empty]
end define

define assignment
    [id]
end define

define inParameter
    [attribType] [attribName]
    | [empty]
end define

define attribType
    [id]
end define

define attribName
    [id]
end define

define attribValue
    [id]
    |[stringlit]
    | 'new 'Vector()
end define

```

```

define method
    [NL] '{
[NL] [IN] [partialAttribDescription*] [methodBody*] [EX] [NL] ' }
    | ';
end define

define partialAttribDescription
    [attribType] [attribName] '= [attribValue]'; [NL]
end define

define methodBody
    [token]'; [NL]
    | [NL] '{ [NL] [IN] [token]
    | [NL] [EX] ' } [NL]
    | [not 'import] [not visibility] [token]
end define

```

Figura 120 – O sub-conjunto da gramática MAS-ML usada na fase de transformação dos relacionamentos.

Apêndice II

A Descrição Textual do Exemplo do Virtual Marketplace

Os diagramas estruturais de MAS-ML que modelam o exemplo *Virtual Marketplace* foram descritos usando a gramática MAS-ML (Apêndice I). A descrição textual do exemplo está ilustrada na Figura 121.

```

RELATIONSHIP General_Store 1..* INHABITS 1 Virtual_Marketplace
RELATIONSHIP User_Agent 0..* INHABITS 1 Virtual_Marketplace
RELATIONSHIP Store_Agent 0..* INHABITS 1 Virtual_Marketplace
RELATIONSHIP Second_hand_Bookstore 0..* INHABITS 1
Virtual_Marketplace
RELATIONSHIP Imported_Bookstore 0..* INHABITS 1
Virtual_Marketplace

RELATIONSHIP General_Store 1 OWNS 0..* Buyer
RELATIONSHIP General_Store 1 OWNS 0..* Seller
RELATIONSHIP General_Store 1 OWNS 0..* Market_of_Special_Goods
RELATIONSHIP General_Store 1 OWNS 0..* Market_of_Used_Goods
RELATIONSHIP General_Store 1 OWNS 0..* Desire
RELATIONSHIP General_Store 1 OWNS 0..* Offer

RELATIONSHIP User_Agent 1 PLAYS 0..* Buyer IN General_Store
RELATIONSHIP Store_Agent 1 PLAYS 0..* Seller IN General_Store
RELATIONSHIP Book 1 PLAYS 0..* Desire IN General_Store
RELATIONSHIP Book 1 PLAYS 0..* Offer IN General_Store

RELATIONSHIP Second_hand_Bookstore 1 PLAYS 0..*
Market_of_Used_Goods IN General_Store
RELATIONSHIP Imported_Bookstore 1 PLAYS 0..*
Market_of_Special_Goods IN General_Store

RELATIONSHIP Second_hand_Bookstore 1 OWNS 0..*
Buyer_of_Second_hand_Books
RELATIONSHIP Second_hand_Bookstore 1 OWNS 0..*
Seller_of_Second_hand_Books
RELATIONSHIP Second_hand_Bookstore 1 OWNS 0..*
Desire_of_Second_hand_Books
RELATIONSHIP Second_hand_Bookstore 1 OWNS 0..*
Offer_of_Second_hand_Books

RELATIONSHIP User_Agent 1 PLAYS 0..* Buyer_of_Second_hand_Books
IN Second_hand_Bookstore
RELATIONSHIP User_Agent 1 PLAYS 0..* Seller_of_Second_hand_Books
IN Second_hand_Bookstore
RELATIONSHIP Second_hand_Book 1 PLAYS 0..*
Desire_of_Second_hand_Books IN Second_hand_Bookstore
RELATIONSHIP Second_hand_Book 1 PLAYS 0..*
Offer_of_Second_hand_Books IN Second_hand_Booksstore

RELATIONSHIP Imported_Bookstore 1 OWNS 0..*
Buyer_of_Imported_Books

```

```

RELATIONSHIP Imported_Bookstore 1 OWNS 0..*
    Seller_of_Imported_Books
RELATIONSHIP Imported_Bookstore 1 OWNS 0..*
    Desire_of_Imported_Books
RELATIONSHIP Imported_Bookstore 1 OWNS 0..*
    Offer_of_Imported_Books

RELATIONSHIP User_Agent 1 PLAYS 0..* Buyer_of_Imported_Books IN
    Imported_Bookstore
RELATIONSHIP Store_Agent 1 PLAYS 0..* Seller_of_Imported_Books IN
    Imported_Bookstore
RELATIONSHIP Imported_Book 1 PLAYS 0..* Desire_of_Imported_Books
    IN Imported_Bookstore
RELATIONSHIP Imported_Book 1 PLAYS 0..* Offer_of_Imported_Books
    IN Imported_Booksstore

RELATIONSHIP Desire_of_Second_hand_Books SPECIALIZES Desire
RELATIONSHIP Desire_of_Imported_Books SPECIALIZES Desire
RELATIONSHIP Offer_of_Second_hand_Books SPECIALIZES Offer
RELATIONSHIP Offer_of_Imported_Books SPECIALIZES Offer

RELATIONSHIP Seller_of_Imported_Books SPECIALIZES Seller
RELATIONSHIP Seller_of_Second_hand_Books SPECIALIZES
    Seller_of_Imported_Books

RELATIONSHIP Buyer_of_Second_hand_Books 0..* ASSOCIATED WITH 0..*
    Desire_of_Second_hand_Books
RELATIONSHIP Seller_of_Second_hand_Books 0..* ASSOCIATED WITH 0..*
    Offer_of_Second_hand_Books
RELATIONSHIP Buyer 1 ASSOCIATED WITH 0..* Desire
RELATIONSHIP Seller 1 ASSOCIATED WITH 0..* Offer

RELATIONSHIP Seller 0..* ASSOCIATED WITH 0..* Buyer

RELATIONSHIP Buyer 0..* ASSOCIATED WITH 0..*
    Market_of_Special_Goods
RELATIONSHIP Buyer 0..* ASSOCIATED WITH 0..* Market_of_Used_Goods
RELATIONSHIP Seller_of_Second_hand_Books 0..* ASSOCIATED WITH 1..*
    Market_of_Used_Goods
RELATIONSHIP Seller_of_Second_hand_Books 0..* ASSOCIATED WITH 0..*
    Buyer_of_Second_hand_Books

RELATIONSHIP Buyer 1 AGGREGATES 0..* Buyer_of_Second_hand_Books
RELATIONSHIP Buyer 1 AGGREGATES 0..* Buyer_of_Imported_Books

RELATIONSHIP Buyer_of_Imported_Books 1 ASSOCIATED WITH 0..*
    Desire_of_Imported_Books
RELATIONSHIP Market_of_Special_Goods 1 ASSOCIATED WITH 0..*
    Buyer_of_Imported_Books
RELATIONSHIP Market_of_Special_Goods 1 ASSOCIATED WITH 0..*
    Seller_of_Imported_Books
RELATIONSHIP Buyer_of_Imported_Books 1 ASSOCIATED WITH 0..*
    Seller_of_Imported_Books
RELATIONSHIP Seller_of_Imported_Books 1 ASSOCIATED WITH 0..*
    Offer_of_Imported_Books

RELATIONSHIP Book SPECIALIZES Item
RELATIONSHIP Imported_Book SPECIALIZES Book
RELATIONSHIP Second_hand_Book SPECIALIZES Book

```

```

RELATIONSHIP Virtual_Marketplace 0..* ASSOCIATED WITH 0..*
                                Virtual_Marketplace
RELATIONSHIP Virtual_Marketplace 1 habitat ASSOCIATED WITH member
0..* Item

ENVIRONMENT
(
import java.util.*;
public class Virtual_Marketplace
{
    public Vector getItems ()
    {
        return this.theItem;
    }
    public void setItems (Item newItem)
    {
        this.theItem.add (newItem);
    }
    public void sellItem (Item newItem)
    {
        newItem.setSold();
    }
    public Vector getItems ()
    {
        return this.theItem;
    }
    public Vector getOtherEnvironments ()
    {
        return this.theVirtual_Marketplace;
    }
    public void setOtherEnvironments (Environment newEnvironment)
    {
        this.theVirtual_Marketplace.add (newEnvironment);
    }
    public boolean checkAgentCanLeave (Agent newAgent)
    {
    }
}
)

ORGANIZATION
( General_Store
    GOAL ( "boolean" : "management_of_sellers" = "true"
          RELATED TO PLAN "Creating_sellers"
        )
    GOAL ( "boolean" : "management_of_orders" = "true"
          RELATED TO PLAN "Update_environment_plan"
        )
    GOAL ( "boolean" : "management_of_profit" = "true"
          RELATED TO PLAN "Evaluating_profit"
        )
    BELIEF ("boolean" : "management" = "false")
    BELIEF ("boolean" : "management_of_sellers" = "false")
    BELIEF ("boolean" : "management_of_orders" = "false")
    BELIEF ("Vector" : "sales" = null)
    BELIEF ("Vector" : "sellers" = null)
    BELIEF ("Vector" : "buyers" = null)
    PLAN ( Creating_sellers
          COMPOSED OF ACTION "Create_seller"
          COMPOSED OF ACTION "Inform_buyer_of_seller"
        )
)

```

```

        RELATED TO GOAL "management_of_sellers"
    )

    PLAN ( Update_environment_plan
        COMPOSED OF ACTION "Update_environment_action"
        RELATED TO GOAL "management_of_orders"
    )

    PLAN ( Evaluating_profit
        COMPOSED OF ACTION "Evaluate_profit"
        COMPOSED OF ACTION "Register_sale"
        RELATED TO GOAL "management_of_profit"
    )

    ACTION Create_seller
    ACTION Inform_buyer_of_seller
    ACTION Update_environment_action
    ACTION Evaluate_profit
    ACTION Register_order
    AXIOM ("boolean" : "Send_information_about_sale" = "true")
)

AGENTROLE
( Buyer
    ROLEGOAL ("boolean" : "itemBought" = "true"
    )
    DUTY "Search_for_seller"
    RIGHT "Send_answer_proposal_accept"
    RIGHT "Send_answer_proposal_reject"
    PROTOCOL SimpleNegotiation
    MESSAGE
        (LABEL "Request"
            CONTENT "ItemDescription"
            SENDER "Buyer"
            RECEIVER "Seller"
        )
    MESSAGE
        (LABEL "Proposal"
            CONTENT "ItemDescription"
            SENDER "Seller"
            RECEIVER "Buyer"
        )
    MESSAGE
        (LABEL "AnswerProposal"
            CONTENT "RejectItemDescription"
            SENDER "Buyer"
            RECEIVER "Seller"
        )
    MESSAGE
        (LABEL "AnswerProposal"
            CONTENT "AcceptItemDescription"
            SENDER "Buyer"
            RECEIVER "Seller"
        )
    MESSAGE
        (LABEL "Inform"
            CONTENT "Bill"
            SENDER "Seller"
            RECEIVER "Buyer"
        )
    MESSAGE
        (LABEL "Inform"
            CONTENT "Payment"

```

```

        SENDER "Buyer"
        RECEIVER "Seller"
    )

    PROTOCOL To_request_to_enter_in_market_of_Special_Goods
    MESSAGE
    (LABEL "Request"
        CONTENT "OrgGoals"
        SENDER "Buyer"
        RECEIVER "Market_of_Special_Goods"
    )
    MESSAGE
    (LABEL "Inform"
        CONTENT "OrgGoals"
        SENDER "Market_of_Special_Goods"
        RECEIVER "Buyer"
    )
    MESSAGE
    (LABEL "Request"
        CONTENT "Roles"
        SENDER "Buyer"
        RECEIVER "Market_of_Special_Goods"
    )
    MESSAGE
    (LABEL "Inform"
        CONTENT "Roles"
        SENDER "Market_of_Special_Goods"
        RECEIVER "Buyer"
    )
    MESSAGE
    (LABEL "Request"
        CONTENT "RoleToPlay"
        SENDER "Buyer"
        RECEIVER "Market_of_Special_Goods"
    )
    MESSAGE
    (LABEL "Inform"
        CONTENT "PermissionAccept"
        SENDER "Market_of_Special_Goods"
        RECEIVER "Buyer"
    )
    MESSAGE
    (LABEL "Inform"
        CONTENT "PermissionReject"
        SENDER "Market_of_Special_Goods"
        RECEIVER "Buyer"
    )

    PROTOCOL To_enter_in_market_of_Used_Goods
    MESSAGE
    (LABEL "Resquest"
        CONTENT "OrgGoals"
        SENDER "Buyer"
        RECEIVER "Market_of_Used_Goods"
    )
    MESSAGE
    (LABEL "Inform"
        CONTENT "OrgGoals"
        SENDER "Market_of_Used_Goods"
        RECEIVER "Buyer"
    )

```

```

MESSAGE
  (LABEL "Request"
    CONTENT "Roles"
    SENDER "Buyer"
    RECEIVER "Market_of_Used_Goods"
  )
MESSAGE
  (LABEL "Inform"
    CONTENT "Roles"
    SENDER "Market_of_Used_Goods"
    RECEIVER "Buyer"
  )
MESSAGE
  (LABEL "Inform"
    CONTENT "RoleToPlay"
    SENDER "Buyer"
    RECEIVER "Market_of_Used_Goods"
  )
)

AGENT
( User_Agent
  GOAL ( "boolean" : "itemBought" = "true"
    SUBGOAL (
      GOAL ( "boolean" : "searchSeller" = "true"
        RELATED TO PLAN "Searching_seller"
      )
      GOAL ( "boolean" : "negotiate" = "true"
        RELATED TO PLAN "Buying_item"
        RELATED TO PLAN "Complex_Buying_item"
      )
      GOAL ( "boolean" : "enterOrganization" = "true"
        RELATED TO PLAN "Entering_organization"
        RELATED TO PLAN "Leaving_environment"
      )
    )
  )
  GOAL ( "boolean" : "itemSold" = "true"
    SUBGOAL (
      GOAL ( "boolean" : "announce" = "false"
        RELATED TO PLAN "Announcing_item"
      )
      GOAL ( "boolean" : "negotiate" = "true"
        RELATED TO PLAN "Selling_item"
      )
    )
  )
  BELIEF ("Item" : "itemBought" = "false")
  BELIEF ("Item" : "itemSold" = "false")
  BELIEF ("Item" : "item" = "null")
  PLAN ( Searching_seller
    COMPOSED OF ACTION "Search_for_seller"
    RELATED TO GOAL "searchSeller"
  )
  PLAN ( Buying_item
    COMPOSED OF ACTION "Create_item"
    COMPOSED OF ACTION "Send_request_item"
    COMPOSED OF ACTION "Eval_proposal"
    COMPOSED OF ACTION "Send_answer_proposal_accept"
    COMPOSED OF ACTION "Send_answer_proposal_reject"
  )
)

```

```

        COMPOSED OF ACTION "Check_bill"
        COMPOSED OF ACTION "Send_payment"
        RELATED TO GOAL "negotiate"
    )
    PLAN ( Complex_Buying_item
        COMPOSED OF ACTION "Create_item"
        COMPOSED OF ACTION "Send_request_item"
        COMPOSED OF ACTION "Eval_proposal"
        COMPOSED OF ACTION "Create_counter_proposal"
        COMPOSED OF ACTION "Check_bill"
        COMPOSED OF ACTION "Send_payment"
        RELATED TO GOAL "negotiate"
    )
    PLAN ( Entering_organization
        COMPOSED OF ACTION "Search_for_organization"
        COMPOSED OF ACTION "Check_organization_goals"
        COMPOSED OF ACTION "Check_organization_roles"
        COMPOSED OF ACTION "Select_role_to_play"
        COMPOSED OF ACTION
    "Register_itself_in_organization"
        COMPOSED OF ACTION "Search_for_announcement"
        RELATED TO GOAL "enterOrganization"
    )
    PLAN ( Leaving_environment
        COMPOSED OF ACTION "Search_for_environment"
        COMPOSED OF ACTION "Search_for_organization"
        COMPOSED OF ACTION "Check_organization_goals"
        COMPOSED OF ACTION "Check_organization_roles"
        COMPOSED OF ACTION "Select_role_to_play"
        COMPOSED OF ACTION
    "Register_itself_in_organization"
        RELATED TO GOAL "enterOrganization"
    )
    PLAN ( Announcing_item
        COMPOSED OF ACTION "Announce"
        RELATED TO GOAL "announce"
    )
    PLAN ( Selling_item
        COMPOSED OF ACTION "Create_proposal"
        COMPOSED OF ACTION "Eval_counter_proposal"
        COMPOSED OF ACTION
    "Send_answer_counter_proposal_accept"
        COMPOSED OF ACTION
    "Send_answer_counter_proposal_reject"
        COMPOSED OF ACTION "Calculate_bill"
        COMPOSED OF ACTION "Inform_organization"
        RELATED TO GOAL "negotiate"
    )
    ACTION Search_for_seller POSTCONDITION ( "boolean" :
    "seller" = "true")
    ACTION Create_item
    ACTION Send_request_item
    ACTION Eval_proposal PRECONDITION ( "boolean" : "proposal" =
    "true")
    ACTION Send_answer_proposal
    ACTION Send_answer_proposal_accept
    ACTION Send_answer_proposal_reject
    ACTION Check_bill
    ACTION Create_item
    ACTION Eval_proposal
    ACTION Create_counter_proposal

```

```

ACTION Send_payment
ACTION Search_for_organization
ACTION Check_organization_goals
ACTION Check_organization_roles
ACTION Select_role_to_play
ACTION Register_itself_in_organization
ACTION Search_for_announcement
ACTION Search_for_environment
ACTION Announce
ACTION Create_proposal
ACTION Eval_counter_proposal
ACTION Send_answer_counter_proposal_accept
ACTION Send_answer_counter_proposal_reject
ACTION Calculate_bill
ACTION Inform_organization
)

AGENT
( Store_Agent
  GOAL ( "boolean" : "itemSold" = "true"
    RELATED TO PLAN "Selling_item"
    RELATED TO PLAN "Informing_organization_about_sale"
  )
  BELIEF ("Item" : "itemSold" = "false")

  PLAN ( Selling_item
    COMPOSED OF ACTION "Create_proposal"
    COMPOSED OF ACTION "Calculate_bill"
    COMPOSED OF ACTION "Inform_organization"
    RELATED TO GOAL "itemSold"
  )
  PLAN ( Informing_organization_about_sale
    COMPOSED OF ACTION "Inform_organization"
    RELATED TO GOAL "itemSold"
  )
  ACTION Create_proposal
  ACTION Calculate_bill
  ACTION Inform_organization
)

AGENTROLE
( Seller
  ROLEGOAL ("boolean" : "itemSold" = "true"
  )
  DUTY "Inform_organization"
  PROTOCOL SimpleNegotiation
  MESSAGE
    (LABEL "Request"
      CONTENT "ItemDescription"
      SENDER "Buyer"
      RECEIVER "Seller"
    )
  MESSAGE
    (LABEL "Proposal"
      CONTENT "ItemDescription"
      SENDER "Seller"
      RECEIVER "Buyer"
    )
  MESSAGE
    (LABEL "AnswerProposal"
      CONTENT "RejectItemDescription"

```

```

        SENDER "Buyer"
        RECEIVER "Seller"
    )
    MESSAGE
    (LABEL "AnswerProposal"
        CONTENT "AcceptItemDescription"
        SENDER "Buyer"
        RECEIVER "Seller"
    )
    MESSAGE
    (LABEL "Inform"
        CONTENT "Bill"
        SENDER "Seller"
        RECEIVER "Buyer"
    )
    MESSAGE
    (LABEL "Inform"
        CONTENT "Payment"
        SENDER "Buyer"
        RECEIVER "Seller"
    )
)

ORGANIZATION
( Second_hand_Bookstore
    GOAL ( "boolean" : "management_of_entrance" = "true"
        RELATED TO PLAN "Managing_entrance"
    )
    GOAL ( "boolean" : "management_of_announcement" = "true"
        RELATED TO PLAN "Managing_announcement"
    )
    GOAL ( "boolean" : "management_of_order" = "true"
        RELATED TO PLAN "Registering_sale"
    )
    BELIEF ( "boolean" : "management_of_entrance" = "false")
    BELIEF ( "boolean" : "management_of_announcement" = "false")
    BELIEF ( "boolean" : "management_of_order" = "false")
    BELIEF ( "Vector" : "sales" = null)
    BELIEF ( "Vector" : "announcements" = null)

    PLAN ( Managing_entrance
        COMPOSED OF ACTION "Get_Organization_Goals"
        COMPOSED OF ACTION "Get_Organization_Roles"
        COMPOSED OF ACTION "Register_new_buyer"
        RELATED TO GOAL "management_of_entrance"
    )
    PLAN ( Managing_announcement
        COMPOSED OF ACTION "Register_announcement"
        COMPOSED OF ACTION "Get_Announcement"
        COMPOSED OF ACTION "Inform_Announcement_and_Seller"
        RELATED TO GOAL "management_of_announcement"
    )
    PLAN ( Registering_sale
        COMPOSED OF ACTION "Register_sale"
        COMPOSED OF ACTION "Inform_supplier_organization"
        RELATED TO GOAL "management_of_order"
    )
    ACTION Get_Organization_Goals
    ACTION Get_Organization_Roles
    ACTION Register_new_buyer

```

```

ACTION Register_announcement
ACTION Get_Announcement
ACTION Inform_Announcement_and_Seller
ACTION Register_sale
ACTION Inform_supplier_organization
AXIOM ("boolean" : "Send_information_about_sale" = "true")
)

ORGANIZATION
( Imported_Bookstore
  GOAL ( "boolean" : "management_of_buyers" = "true"
    RELATED TO PLAN "Analyzing_entrance"
  )
  GOAL ( "boolean" : "management_of_sellers" = "true"
    RELATED TO PLAN "Creating_sellers"
  )
  GOAL ( "boolean" : "management_of_orders" = "true"
    RELATED TO PLAN "Registering_sale"
  )
  BELIEF ( "boolean" : "management_of_buyers" = "false")
  BELIEF ( "boolean" : "management_of_sellers" = "false")
  BELIEF ( "boolean" : "management_of_orders" = "false")
  BELIEF ( "Vector" : "sellers" = null)
  BELIEF ( "Vector" : "buyers" = null)
  BELIEF ( "Vector" : "sales" = null)
  BELIEF ( "Organization" : "main_organization" = null)
  PLAN ( Analyzing_entrance
    COMPOSED OF ACTION "Get_Organization_Goals"
    COMPOSED OF ACTION "Get_Organization_Roles"
    COMPOSED OF ACTION "Analyze_entrance"
    COMPOSED OF ACTION "Register_new_buyer"
    RELATED TO GOAL "management_of_buyers"
  )
  PLAN ( Creating_sellers
    COMPOSED OF ACTION "Create_seller"
    COMPOSED OF ACTION "Inform_buyer_of_seller"
    RELATED TO GOAL "management_of_sellers"
  )
  PLAN ( Registering_sale
    COMPOSED OF ACTION "Register_sale"
    COMPOSED OF ACTION "Inform_supplier_organization"
    COMPOSED OF ACTION "Update_environment_action"
    RELATED TO GOAL "management_of_order"
  )

  ACTION Get_Organization_Goals
  ACTION Get_Organization_Roles
  ACTION Analyze_entrance
  ACTION Register_new_buyer
  ACTION Create_seller
  ACTION Inform_buyer_of_seller
  ACTION Register_sale
  ACTION Inform_supplier_organization
  ACTION Update_environment_action
  AXIOM ("boolean" : "Send_information_about_sale" = "true")
)

AGENTROLE
( Market_of_Used_Goods
  ROLEGOAL ("boolean" : "management_of_entrance" = "true"
  )
)

```

```

ROLEGOAL ("boolean" : "management_of_announcement" = "true"
)
ROLEGOAL ( "boolean" : "management_of_orders" = "true"
)
PROTOCOL To_enter_in_market_of_Used_Goods
  MESSAGE
    (LABEL "Request"
      CONTENT "OrgGoals"
      SENDER "Buyer"
      RECEIVER "Market_of_Used_Goods"
    )
  MESSAGE
    (LABEL "Inform"
      CONTENT "OrgGoals"
      SENDER "Market_of_Used_Goods"
      RECEIVER "Buyer"
    )
  MESSAGE
    (LABEL "Request"
      CONTENT "Roles"
      SENDER "Buyer"
      RECEIVER "Market_of_Used_Goods"
    )
  MESSAGE
    (LABEL "Inform"
      CONTENT "Roles"
      SENDER "Market_of_Used_Goods"
      RECEIVER "Buyer"
    )
  MESSAGE
    (LABEL "Inform"
      CONTENT "RoleToPlay"
      SENDER "Buyer"
      RECEIVER "Market_of_Used_Goods"
    )
)

PROTOCOL Buyer_registration
  MESSAGE
    (LABEL "Inform"
      CONTENT "Buyer_of_Second_hand_Books"
      SENDER "Buyer_of_Second_hand_Books"
      RECEIVER "Market_of_Used_Goods"
    )
)

PROTOCOL To_announce
  MESSAGE
    (LABEL "Inform"
      CONTENT "Offer_Seller"
      SENDER "Seller_of_Second_hand_Books"
      RECEIVER "Market_of_Used_Goods"
    )
)

PROTOCOL To_search_announcement
  MESSAGE
    (LABEL "Request"
      CONTENT "Announcement"
      SENDER "Buyer_of_Second_hand_Books"
      RECEIVER "Market_of_Used_Goods"
    )
  MESSAGE
    (LABEL "Inform"
      CONTENT "Announcement"
      SENDER "Market_of_Used_Goods"
    )
)

```

```

        RECEIVER "Buyer_of_Second_hand_Books"
    )
    MESSAGE
    (LABEL "Inform"
        CONTENT "Seller"
        SENDER "Market_of_Used_Goods"
        RECEIVER "Buyer_of_Second_hand_Books"
    )
    PROTOCOL To_register_sale
    MESSAGE
    (LABEL "Inform"
        CONTENT "ItemSold_Buyer_Payment"
        SENDER "Seller_of_Second_hand_Books"
        RECEIVER "Market_of_Used_Goods"
    )
)

AGENTROLE
( Market_of_Special_Goods
    ROLEGOAL ("boolean" : "management_of_buyers" = "true"
    )
    ROLEGOAL ( "boolean" : "management_of_sellers" = "true"
    )
    ROLEGOAL ( "boolean" : "management_of_orders" = "true"
    )
    DUTY "Analyze_entrance"

    PROTOCOL To_request_to_enter_in_market_of_Special_Goods
    MESSAGE
    (LABEL "Request"
        CONTENT "OrgGoals"
        SENDER "Buyer"
        RECEIVER "Market_of_Special_Goods"
    )
    MESSAGE
    (LABEL "Inform"
        CONTENT "OrgGoals"
        SENDER "Market_of_Special_Goods"
        RECEIVER "Buyer"
    )
    MESSAGE
    (LABEL "Request"
        CONTENT "Roles"
        SENDER "Buyer"
        RECEIVER "Market_of_Special_Goods"
    )
    MESSAGE
    (LABEL "Inform"
        CONTENT "Roles"
        SENDER "Market_of_Special_Goods"
        RECEIVER "Buyer"
    )
    MESSAGE
    (LABEL "Request"
        CONTENT "RoleToPlay"
        SENDER "Buyer"
        RECEIVER "Market_of_Special_Goods"
    )
    MESSAGE
    (LABEL "Inform"
        CONTENT "PermissionAccept"

```

```

        SENDER "Market_of_Special_Goods"
        RECEIVER "Buyer"
    )
    MESSAGE
    (LABEL "Inform"
        CONTENT "PermissionReject"
        SENDER "Market_of_Special_Goods"
        RECEIVER "Buyer"
    )
    PROTOCOL Buyer_registration
    MESSAGE
    (LABEL "Inform"
        CONTENT "Buyer_of_Imported_Books"
        SENDER "Buyer_of_Imported_Books"
        RECEIVER "Market_of_Special_Goods"
    )
    PROTOCOL To_search_for_seller
    MESSAGE
    (LABEL "Request"
        CONTENT "Seller?"
        SENDER "Buyer_of_Imported_Books"
        RECEIVER "Market_of_Special_Goods"
    )
    MESSAGE
    (LABEL "Inform"
        CONTENT "Seller"
        SENDER "Market_of_Special_Goods"
        RECEIVER "Buyer_of_Imported_Books"
    )
    PROTOCOL To_register_sale
    MESSAGE
    (LABEL "Inform"
        CONTENT "ItemSold_Buyer_Payment"
        SENDER "Seller_of_Imported_Books"
        RECEIVER "Market_of_Special_Goods"
    )
)

AGENTROLE
( Buyer_of_Imported_Books
    ROLEGOAL ("boolean" : "itemBought" = "true"
    )
    DUTY "Register_itself_in_organization"
    PROTOCOL Buyer_registration
    MESSAGE
    (LABEL "Inform"
        CONTENT "Buyer_of_Imported_Books"
        SENDER "Buyer_of_Imported_Books"
        RECEIVER "Market_of_Special_Goods"
    )
    PROTOCOL To_search_for_seller
    MESSAGE
    (LABEL "Request"
        CONTENT "Seller?"
        SENDER "Buyer_of_Imported_Books"
        RECEIVER "Market_of_Special_Goods"
    )
    MESSAGE
    (LABEL "Inform"
        CONTENT "Seller"
        SENDER "Market_of_Special_Goods"

```

```

        RECEIVER "Buyer_of_Imported_Books"
    )
    PROTOCOL SimpleNegotiation
    MESSAGE
    (LABEL "Request"
        CONTENT "ItemDescription"
        SENDER "Buyer_of_Imported_Books"
        RECEIVER "Seller_of_Imported_Books"
    )
    MESSAGE
    (LABEL "Proposal"
        CONTENT "ItemDescription"
        SENDER "Seller_of_Imported_Books"
        RECEIVER "Buyer_of_Imported_Books"
    )
    MESSAGE
    (LABEL "AnswerProposal"
        CONTENT "RejectItemDescription"
        SENDER "Buyer_of_Imported_Books"
        RECEIVER "Seller_of_Imported_Books"
    )
    MESSAGE
    (LABEL "AnswerProposal"
        CONTENT "AcceptItemDescription"
        SENDER "Buyer_of_Imported_Books"
        RECEIVER "Seller_of_Imported_Books"
    )
    MESSAGE
    (LABEL "Inform"
        CONTENT "Bill"
        SENDER "Seller_of_Imported_Books"
        RECEIVER "Buyer_of_Imported_Books"
    )
    MESSAGE
    (LABEL "Inform"
        CONTENT "Payment"
        SENDER "Buyer_of_Imported_Books"
        RECEIVER "Seller_of_Imported_Books"
    )
)

AGENTROLE
( Buyer_of_Second_hand_Books
    ROLEGOAL ("boolean" : "itemBought" = "true"
    )
    DUTY "Register_itself_in_organization"
    PROTOCOL Buyer_registration
    MESSAGE
    (LABEL "Inform"
        CONTENT "Buyer_of_Second_hand_Books"
        SENDER "Buyer_of_Second_hand_Books"
        RECEIVER "Market_of_Used_Goods"
    )
    PROTOCOL SimpleNegotiation
    MESSAGE
    (LABEL "Request"
        CONTENT "ItemDescription"
        SENDER "Buyer_of_Second_hand_Books"
        RECEIVER "Seller_of_Second_hand_Books"
    )
    MESSAGE

```

```

        (LABEL "Proposal"
          CONTENT "ItemDescription"
          SENDER "Seller_of_Second_hand_Books"
          RECEIVER "Buyer_of_Second_hand_Books"
        )
      MESSAGE
      (LABEL "AnswerProposal"
        CONTENT "RejectItemDescription"
        SENDER "Buyer_of_Second_hand_Books"
        RECEIVER "Seller_of_Second_hand_Books"
      )
      MESSAGE
      (LABEL "AnswerProposal"
        CONTENT "AcceptItemDescription"
        SENDER "Buyer_of_Second_hand_Books"
        RECEIVER "Seller_of_Second_hand_Books"
      )
      MESSAGE
      (LABEL "Inform"
        CONTENT "Bill"
        SENDER "Seller_of_Second_hand_Books"
        RECEIVER "Buyer_of_Second_hand_Books"
      )
      MESSAGE
      (LABEL "Inform"
        CONTENT "Payment"
        SENDER "Buyer_of_Second_hand_Books"
        RECEIVER "Seller_of_Second_hand_Books"
      )
    )
  PROTOCOL ComplexNegotiation
  MESSAGE
  (LABEL "Call_for_proposal"
    CONTENT "Item"
    SENDER "Buyer_of_Second_hand_Books"
    RECEIVER "Seller_of_Second_hand_Books"
  )
  MESSAGE
  (LABEL "Proposal"
    CONTENT "Item"
    SENDER "Seller_of_Second_hand_Books"
    RECEIVER "Buyer_of_Second_hand_Books"
  )
  MESSAGE
  (LABEL "Counter_proposal"
    CONTENT "Item"
    SENDER "Buyer_of_Second_hand_Books"
    RECEIVER "Seller_of_Second_hand_Books"
  )
  MESSAGE
  (LABEL "Reject_counter_proposal"
    CONTENT ""
    SENDER "Seller_of_Second_hand_Books"
    RECEIVER "Buyer_of_Second_hand_Books"
  )
  MESSAGE
  (LABEL "Accept_counter_proposal"
    CONTENT ""
    SENDER "Seller_of_Second_hand_Books"
    RECEIVER "Buyer_of_Second_hand_Books"
  )
)

```

```

MESSAGE
  (LABEL "Inform"
    CONTENT "Bill"
    SENDER "Seller_of_Second_hand_Books"
    RECEIVER "Buyer_of_Second_hand_Books"
  )
MESSAGE
  (LABEL "Inform"
    CONTENT "Payment"
    SENDER "Buyer_of_Second_hand_Books"
    RECEIVER "Seller_of_Second_hand_Books"
  )
MESSAGE
  (LABEL "Accept_propose"
    CONTENT "Item"
    SENDER "Buyer_of_Second_hand_Books"
    RECEIVER "Seller_of_Second_hand_Books"
  )
)
PROTOCOL To_search_announcement
MESSAGE
  (LABEL "Request"
    CONTENT "Announcement"
    SENDER "Buyer_of_Second_hand_Books"
    RECEIVER "Market_of_Used_Goods"
  )
MESSAGE
  (LABEL "Inform"
    CONTENT "Announcement"
    SENDER "Market_of_Used_Goods"
    RECEIVER "Buyer_of_Second_hand_Books"
  )
MESSAGE
  (LABEL "Inform"
    CONTENT "Seller"
    SENDER "Market_of_Used_Goods"
    RECEIVER "Buyer_of_Second_hand_Books"
  )
)

```

AGENTROLE

```

( Seller_of_Imported_Books
  ROLEGOAL ("boolean" : "itemSold" = "true"
  )
  DUTY "Inform_organization"
  PROTOCOL SimpleNegotiation
  MESSAGE
    (LABEL "Request"
      CONTENT "ItemDescription"
      SENDER "Buyer_of_Imported_Books"
      RECEIVER "Seller_of_Imported_Books"
    )
  MESSAGE
    (LABEL "Proposal"
      CONTENT "ItemDescription"
      SENDER "Seller_of_Imported_Books"
      RECEIVER "Buyer_of_Imported_Books"
    )
  MESSAGE
    (LABEL "AnswerProposal"
      CONTENT "RejectItemDescription"
      SENDER "Buyer_of_Imported_Books"
    )
  )

```

```

        RECEIVER "Seller_of_Imported_Books"
    )
    MESSAGE
    (LABEL "AnswerProposal"
        CONTENT "AcceptItemDescription"
        SENDER "Buyer_of_Imported_Books"
        RECEIVER "Seller_of_Imported_Books"
    )
    MESSAGE
    (LABEL "Inform"
        CONTENT "Bill"
        SENDER "Seller_of_Imported_Books"
        RECEIVER "Buyer_of_Imported_Books"
    )
    MESSAGE
    (LABEL "Inform"
        CONTENT "Payment"
        SENDER "Buyer_of_Imported_Books"
        RECEIVER "Seller_of_Imported_Books"
    )
    PROTOCOL To_register_sale
    MESSAGE
    (LABEL "Inform"
        CONTENT "ItemSold_Buyer_Payment"
        SENDER "Seller_of_Imported_Books"
        RECEIVER "Market_of_Special_Goods"
    )
)

AGENTROLE
( Seller_of_Second_hand_Books
    ROLEGOAL ("boolean" : "itemSold" = "true"
    )
    DUTY "Inform_organization"
    PROTOCOL SimpleNegotiation
    MESSAGE
    (LABEL "Request"
        CONTENT "ItemDescription"
        SENDER "Buyer_of_Second_hand_Books"
        RECEIVER "Seller_of_Second_hand_Books"
    )
    MESSAGE
    (LABEL "Proposal"
        CONTENT "ItemDescription"
        SENDER "Seller_of_Second_hand_Books"
        RECEIVER "Buyer_of_Second_hand_Books"
    )
    MESSAGE
    (LABEL "AnswerProposal"
        CONTENT "RejectItemDescription"
        SENDER "Buyer_of_Second_hand_Books"
        RECEIVER "Seller_of_Second_hand_Books"
    )
    MESSAGE
    (LABEL "AnswerProposal"
        CONTENT "AcceptItemDescription"
        SENDER "Buyer_of_Second_hand_Books"
        RECEIVER "Seller_of_Second_hand_Books"
    )
    MESSAGE
    (LABEL "Inform"

```

```

        CONTENT "Bill"
        SENDER "Seller_of_Second_hand_Books"
        RECEIVER "Buyer_of_Second_hand_Books"
    )
    MESSAGE
    (LABEL "Inform"
        CONTENT "Payment"
        SENDER "Buyer_of_Second_hand_Books"
        RECEIVER "Seller_of_Second_hand_Books"
    )
    PROTOCOL To_register_sale
    MESSAGE
    (LABEL "Inform"
        CONTENT "ItemSold_Buyer_Payment"
        SENDER "Seller_of_Imported_Books"
        RECEIVER "Market_of_Special_Goods"
    )
    PROTOCOL ComplexNegotiation
    MESSAGE
    (LABEL "Call_for_proposal"
        CONTENT "Item"
        SENDER "Buyer_of_Second_hand_Books"
        RECEIVER "Seller_of_Second_hand_Books"
    )
    MESSAGE
    (LABEL "Proposal"
        CONTENT "Item"
        SENDER "Seller_of_Second_hand_Books"
        RECEIVER "Buyer_of_Second_hand_Books"
    )
    MESSAGE
    (LABEL "Counter_proposal"
        CONTENT "Item"
        SENDER "Buyer_of_Second_hand_Books"
        RECEIVER "Seller_of_Second_hand_Books"
    )
    MESSAGE
    (LABEL "Reject_counter_proposal"
        CONTENT ""
        SENDER "Seller_of_Second_hand_Books"
        RECEIVER "Buyer_of_Second_hand_Books"
    )
    MESSAGE
    (LABEL "Accept_counter_proposal"
        CONTENT ""
        SENDER "Seller_of_Second_hand_Books"
        RECEIVER "Buyer_of_Second_hand_Books"
    )
    MESSAGE
    (LABEL "Inform"
        CONTENT "Bill"
        SENDER "Seller_of_Second_hand_Books"
        RECEIVER "Buyer_of_Second_hand_Books"
    )
    MESSAGE
    (LABEL "Inform"
        CONTENT "Payment"
        SENDER "Buyer_of_Second_hand_Books"
        RECEIVER "Seller_of_Second_hand_Books"
    )
    MESSAGE

```

```

        (LABEL "Accept_propose"
          CONTENT "Item"
          SENDER "Buyer_of_Second_hand_Books"
          RECEIVER "Seller_of_Second_hand_Books"
        )
      PROTOCOL To_announce
      MESSAGE
      (LABEL "Offer_Seller"
        CONTENT "Offer_Seller"
        SENDER "Seller_of_Second_hand_Books"
        RECEIVER "Market_of_Used_Goods"
      )
    )

import java.util.*;
public class Item
{
    protected boolean sold = false;
    public boolean getSold ()
    {
        return this.sold;
    }
    public void setSold ()
    {
        this.sold = true;
    }
}

import java.util.*;
public class Book extends Item
{
    protected String title = null;
    protected String author = null;
    protected String ISBN = null;
    protected int price = null;
    public String getTitle ()
    {
        return this.title;
    }
    public String getAuthor ()
    {
        return this.author;
    }
    public String getISBN ()
    {
        return this.ISBN;
    }
    public int getPrice ()
    {
        return this.price;
    }
    public void setTitle (String title)
    {
        this.title = title;
    }
    public void setAuthor (String author)
    {
        this.author = author;
    }
    public void setISBN (String ISBN)
    {

```

```

        this.ISBN = ISBN;
    }
    public void setPrice (int price)
    {
        this.price = price;
    }
}

import java.util.*;
public class Imported_Book extends Book
{
    protected String origin = null;
    public String getOrigin ()
    {
        return this.origin;
    }
    public void setOrigin (String origin)
    {
        this.origin = origin;
    }
}

import java.util.*;
public class Second_hand_Book extends Book
{
    protected String appearance = null;
    public String getAppearance ()
    {
        return this.appearance;
    }
    public void setAppearance (String appearance)
    {
        this.appearance = appearance;
    }
}

OBJECTROLE
( Desire (
    public int getPrice ()
    {
        return object.getPrice();
    }
    public void setTitle (String title)
    {
        object.setTitle(title);
    }
    public void setAuthor (String author)
    {
        object.setAuthor(author);
    }
    public void setISBN (String ISBN)
    {
        object.setISBN(ISBN);
    }
})

OBJECTROLE
( Offer (
    public void setPrice (int price)
    {
        object.setPrice(price);
    }

```

```

    }
    public String getTitle ()
    {
        return this.title;
    }
    public String getAuthor ()
    {
        return this.author;
    }
    public String getISBN ()
    {
        return this.ISBN;
    }
}
))

OBJECTROLE
( Desire_of_Imported_Books (
    public void setOrigin (String origin)
    {
        object.setOrigin(origin);
    }
)
))

OBJECTROLE
( Offer_of_Imported_Books (
    public String getOrigin ()
    {
        return object.getOrigin();
    }
)
))

OBJECTROLE
( Desire_of_Second_hand_Books (
    public void setAppearance (String appearance)
    {
        object.setAppearance(appearance);
    }
)
))

OBJECTROLE
( Offer_of_Second_hand_Books (
    public String getAppearance ()
    {
        return object.getAppearance();
    }
)
))

```

Figura 121 – Descrição textual do exemplo do *Virtual Marketplace*.