9
Referências

BAADER, F.; CALVANESE, D.; MCGUINNESS, D.L.; NARDI, D.; PATEL-SCHNEIDER, P.F. Eds. **The Description Logic Handbook: Theory, Implementation and Applications**: Cambridge University Press, 2003.

BASILI, V., SELBY, R e HUTCHENS, D. Experimentation in software engineering **IEEE Transactions on Software Engineering**, v.12, n.7, p.733-743. 1986.

BAUER, B. Class diagrams revisited in the context of agent-based systems. In: M. Wooldridge, G. Weiss, et al (Ed.). **Agent-Oriented Software Engineering**. Berlin: Springer-Verlag, v.2222, p.101-118. (LNCS), 2002.

BAUER, B., MULLER, J.; ODELL, J.. Agent UML: A formalism for specifying multiagent interaction. In: M. Wooldridge, G. Weiss, et al (Ed.). **Agent-oriented Software Engineering**. Berlin: Springer-Verlag, v.2222, p.91-103. (LNCS), 2002.

BAUER, B., MÜLLER, J; ODELL, J. Agent UML: A Formalism for Specifying Multiagent Software Systems. **Lecture Notes in Computer Science**, v.1957, p.91. 2001.

BELLIFEMINE, F., G. CAIRE, POGGI, A. e RIMASSA, G. **JADE - A White Paper**. TILAB (Telecom Italia Lab). Turin, Italy, p.141. 2003. (3)

BELLIFEMINE, F., A. POGGI, RIMASSA, G. JADE - FIPA Compliant Agent Framework. **Proceedings of the PAAM**, 97-108 p, 1999.

BERARDI, D. Using DLs to reason on UML Class Diagrams. ADL 2002 - 2002 **Workshop on Applications of Description Logics**. Toulouse, France 2002.

BERARDI, D., CALVANESE, D. e DE GIACOMO, G. Reasoning on UML Class Diagrams Using Description Logic Based Systems. **KI-2001 Workshop on Applications of Description Logics**. Vienna, Austria: Ceur-ws.org 2001.

BERNARAS, A., LARESGOITI, I e CORERA, J. Building and reusing ontologies for electrical network applications. **European Conference on Artificial Intelligence (ECAI'96)**. Budapest, Hungary 1996.

BOOCH, G., RUMBAUGH, J. e JACOBSON, R. **The Unified Modeling Language User Guide**: Addison-Wesley. 1999. 482 p.

BRANDÃO, A. A. F., ALENCAR, P. e LUCENA, C. Extending Object-Z for Multi-Agent Systems Specification. **Workshop on Agent-Oriented Information Systems at CAiSE 2004**. Riga, Latvia, 280-294 p., 2004a.

______. AgentZ: Extending Object-Z for Multi-agent Systems Specification. In: P. Bresciani, P. Giorgini, et al (Ed.). **AOIS 2004**: Springer, LNCS, v.3508, p.125-139, 2005c.

BRANDÃO, A. A. F., SILVA, V. e LUCENA, C. A Knowledge-based Approach to the Verification of Multi-Agent Systems Design. **The 17th International Conference on Software Engineering and Knowledge Engineering**. Taipei, Taiwan, Republic of China: Knowledge Systems Institute Graduate School, 602-607 p, 2005a.

______. **An Ontology-based Method for Structuring Multi-Agent Systems Formal Specifications**. Pontifícia Universidade Católica do Rio de Janeiro. Rio de Janeiro, RJ, Brasil, p.12. (MCC 25/05), 2005b.

______.Ontologies as Specifications for the Verification of Multi-Agent Systems Design. **Workshop on Ontologies as Software Engineering Artifacts, at OOPSLA 2004**. Vancouver, Canada 2004. disponível em http://swt-www.informatik.uni-hamburg.de/conferences/oopsla%202004%20-%20accepted/BrandaoSilvaLucena.pdf , último acesso em 18/11/2005.

______. A Knowledge-based Approach to the Specification and Verification of MAS Design. **4th International Joint Conference on Autonomous Agents and Multi-Agent Systems - AAMAS 2005**. Utrecht, The Neetherlands: ACM-Press, 1373 p, 2005.

BUNGE, M. A. **Treatise on Basic Philosophy: Ontology I: The Furniture of the World**. Dordrecht, Holland: D. Reidel Publishing Company. 1977

______. **Treatise on Basic Philosophy: Ontology II: A World of Systems**. Dordrecht, Holland: D. Reidel Publishing Company. 1979

CHAVEZ, C.V.F.C **Uma abordagem orientada a modelos para design orientado a aspectos**, Tese de Doutorado. Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, RJ, Brasil, 2004

CHOREN, R. e LUCENA, C. Modeling Multi-Agent Systems with Anote. **Journal of Software and Systems Modeling**, v.4, n.2, p.199-208. 2004.

COLLIS, J., NDUMU, D.; NWANA, H. e LEE, L. The ZEUS Agent Building Tool-kit. **BT Technology Journal** v.16, n.3, p.60-68. 1998.

DAML+OIL. The DARPA Agent Markup Language (DAML) Homepage. http://www.daml.org, último acesso em 08/11/2005.

DELOACH, S. A. Analysis and Design of Multiagent Systems Using Hybrid Coordination Media **Software Engineering in Multiagent Systems (SEMAS 2002)**. Orlando, Florida, USA 2002.

DONG, J., LEE, C.; LI, Y-F e WANG, H. Verifying DAML+OIL and Beyond in Z/EVES. **26th International Conference on Software Engineering (ICSE 2004)**. Edinburgh, United Kingdom: IEEE Computer Society 2004.

EKENBERG, L. e JOHANNESSON, P. A framework for determining design correctness. **Knowledge-Based Systems**, v.17, n.7-8, p.249-262. 2004.

EVERMANN, J. e WAND, Y. Toward Formalizing Domain Modeling Semantics in Language Syntax. **IEEE Transactions on Software Enginering**, v.31, n.1, p.21-37. 2005.

FaCT. The FaCT – Fast Classification of Terminologies Homepage, 2003 http://www.cs.man.ac.uk/~horrocks/FaCT , último acesso em 08/11/2005.

FIPA. The FIPA : Foundation For Intelligent Physical Agents Homepage 2004 http://www.fipa.org, último acesso em 08/11/2005.

GARCIA, A. **De objetos a agentes: uma abordagem orientada a aspectos**, Tese de Doutorado. Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, RJ, Brasil, 2004.

GENESERETH, M. R. Knowledge Interchange Format: draft proposed **American National Standard** (dpANS) 2004.

GOMÉZ-PÉREZ, A. Towards a framework to verify knowledge sharing technology. **Expert Systems with Applications**, v.11, n.4, p.519-529. 1996.

GÓMEZ-PÉREZ, A. Ontological Engineering: A State of the Art. **Expert Update**, v.2, n.3, p.33-43. 1999.

GRUBER, T. R. A translation approach to portable ontology specifications. **Knowledge Acquisition**, v.5, n.2, p.199-220. 1993.

HAARSLEV, V., R. MÖLLER, STRAETEN, R. e WESSEL, M. Extended Query Facilities for Racer and an Application to Software-Engineering Problems **2004 International Workshop on Description Logics (DL2004).** Whistler, British Columbia, Canada: CEUR-WS.ORG 2004.

HENDERSON-SELLERS, B. Advanced OO modelling: metamodels and notations for the new millennium. **Proceedings of the 21st International Conference on Conceptual Modeling - ER 2002**: Springer-Verlag, 2002. 11 p.

HUGET, M. Agent UML Class diagrams revisited. **Agent Technology and Software Engineering. Germany**, 2002a. p.

______. Generating Code from Agent UML Sequence Diagrams **Agent Technology and Software Engineering**. Germany, 2002b. p.

Icse. International Conference on Software Engineering.

d'INVERNO, M. e LUCK, M. **Understanding Agent Systems**, Springer Verlag, 2001.

ISHIDA, I., GASSER, L. e YOKOO, M. Organization self design of production systems. **IEEE Transactions on Knowledge and Data Engeneering**, v.4, n.2, p.123-134. 1992.

KALFOGLOU, Y. e D. ROBERTSON. A case study in applying Ontologies to augment and reason about the correctness of Specifications. **11th International Conference on Software Engineering and Knowledge Engineering (SEKE99).** Kaiserslautern, Germany 1999.

KIM, S.-K. e CARRINGTON, D. A Formal Mapping between UML Models and Object-Z Specifications. **First International Conference of B and Z Users on Formal Specification and Development in Z and B:** Springer-Verlag 2000.

LUCK, M., MCBURNEY, P, SHEHORY, O. e WILLMOTT, S. **A Roadmap for Agent Based Computing** AgentLink 2005.

MARIA, B. D., SILVA, V. e LUCENA, C.. VisualAgent: A Software Development Environment for Multi-Agent Systems. **Simpósio Brasileiro de Engenharia de Software (SBES 2005)**. Uberlândia - Minas Gerais 2005.

MILLS, K. L. e GOMAA, H. Knowledge-based automation of a design method for concurrent systems. **IEEE Transactions on Software Enginering**, v.28, n.3, p.228-255. 2002.

NECHES, R., FIKES, R., FININ, T., GRUBER, T., PATIL, R.; SENATOR, T.; e SWARTOUT, W. Enabling technology for knowledge sharing. **AI Magazine,** v.12, n.3, p.36-56. 1991.

ODELL, J., PARUNAK, H. e BAUER, B. Extending UML for Agents. **Agent-oriented Information Systems Workshop**. Austin - Texas, 2000. 3-17 p.

______. The role of roles in designing effective agent organizations. Berlin: Springer-Verlag, v.2603. 2003 (LNCS).

OILED OilEd Ontology Editor 2002, http://oiled.man.ac.uk/ último acesso em 18/11/2005.

OMG. Meta Object Facility (MOF) 2.0 Core Specification: Object Management Group 2003, ultimo acesso em 18/11/2005 e disponível em http://www.omg.org/technology/documents/modeling_spec_catalog.htm#MOF

______. Unified Modeling Language (UML): Superstructure version 2.0 Object Management Group 2004. disponível em http://www.uml.org/#UML2.0 , ultimo acesso em 18/11/2005

PARUNAK, H. e ODELL, J. Representing social structures in UML. In: M. Wooldridge, G. Weiss, et al (Ed.). **Agent-oriented Software Engineering**. Berlin: Springer-Verlag, (LNCS), v.2222, p.1-16, 2002.

POWERLOOM. PowerLoom Knowledge Representation System. http://www.isi.edu/isd/LOOM/PowerLoom/index.html último acesso em 16/11/2005.

PROTEGÉ. The Protegé 2000 Editor home page, 2004 http://protege.stanford.edu/ último acesso em 18/11/2005.

PUML. Precise UML http://www.puml.org/ último acesso em 08/11/2005.

RACER. The RACER - Renamed Abox and Concept Expression Reasoner homepage 2004. http://www.racer-systems.com/ último acesso e, 18/11/2005.

RICE. The RICE - RACER Interactive Client Environment homepage, 2004 http://www.cs.concordia.ca/~haarslev/racer/rice.jar, último acesso em 08/11/2005.

SHOHAM, Y. Agent-Oriented Programming. **Artificial Intelligence**, v.60. 1993.

SILVA, V. **Uma Linguagem de Modelagem para Sistemas Multiagentes Baseada em um Framework Conceitual para Agentes e Objetos**. Tese de Doutorado. Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, RJ, Brasil, 2004. 252 p.

SILVA, V., GARCIA, A.; BRANDÃO, A.; CHAVEZ, C. LUCENA, C. e ALENCAR, P. Taming Agents and Objects in Software Engineering. In: A. Garcia, C. Lucena, et al (Ed.). **Software Engineering for Large-Scale Multi-Agent Systems: Research Issues and Practical Applications**: Springer-Verlag GmbH, LNCS, v. 2603, p.1-26, 2003.

SILVA, V. e LUCENA, C. From a Conceptual Framework for Agents and Objects to a Multi-Agent System Modeling Language. **Journal of Autonomous Agents and Multi-Agent Systems,** v.9, n.1-2, p.145-189. 2004.

SILVA, V., CHOREN, R. e LUCENA, C. Using the UML 2.0 activity diagram to model agent plans and actions **International Joint Conference on Autonomous Agents and Multiagent Systems**. Utrecht - The Netherlands ACM Press 2005.

SMITH, G. **The Object-Z Specification Language**: Springer, v.1. 1999. 160 p. (Advances in Formal Methods)

STRAETEN, R., SIMMONDS, J. e MENS, T. Detecting Inconsistencies between UML Models Using Description Logic. **International Workshop on Description Logics**. Rome, Italy: http://ceur-ws.org/ 2003. último acesso em 08/11/2005.

SYCARA, K., PAOLUCCI, M.; VELSEN, M. e GIAMPAPA, J. The RETSINA MAS Infrastructure. **Journal of Autonomous Agents and Multi-Agent Systems**, v.7, n.1-2, p.29-48. 2003.

USCHOLD, M. e KING, M. Towards a Methodology for Building Ontologies. **IJCAI95's Workshop on Basic Ontological Issues in Knowledge Sharing.** Montreal - Canada 1995.

W3C. OWL Web Ontology Language Guide - W3C Recommendation 10 February 2004: World Wide Web Consortium 2004a.

______. OWL Web Ontology Language Overview: W3C Recommendation 10 February 2004 World Wide Web Consortium 2004b.

______. OWL Web Ontology Language Semantics and Abstract Syntax: W3C Recommendation 10 February 2004 World Wide Web Consortium 2004c.

______. RDF (Resource Description Framework) Primer: W3C Recommendation 10 February 2004. B. Mcbride: World Wide Web Consortium 2004d.

WAGNER, G. A UML Profile for External AOR Models. **Third Workshop on Agent-Oriented Software Engineering**. Italy, 2002. p.

______. The agent-object-relationship metamodel: towards a unified view of state and behavior. **Information Systems**, v.28, n.5, p.475-504. 2003.

WEISS, G. **Multiagent systems: A Modern Approach to Distributed Artificial Intelligence**. Boston: MIT Press. 1999

WOOD, M. e DELOACH, S. An overview of the multiagent systems engineering methodology **First International workshop on Agent-oriented software engineering** , AOSE 2000 Limerick, Ireland: Springer-Verlag 2001.

WOOLDRIDGE, M. e CIANCARINI, P. Agent-Oriented Software Engineering: The State of the Art. In: P. Ciancarini e M. Wooldridge (Ed.). **Agent-Oriented Software Engineering**. Berlin: Springer Verlag, (LNCS), v.1957, p.1-28., 2001.

WOOLDRIDGE, M., JENNINGS, N. e KINNY, D. The Gaia Methodology for Agent-Oriented Analysis and Design. **Autonomous Agents and Multi-Agent Systems**, v.3, n.3, p.285-312. 2000.

YU, L. e SCHMID, B. A conceptual framework for agent-oriented and role-based work on modeling. **1st International Workshop on Agent-Oriented Information Systems** 1999.

ZAMBONELLI, F., JENNINGS, N. e WOOLDRIDGE, M. Organizational Abstractions for the Analysis and Design of Multi-agent Systems. In: P. Ciancarini e M. Wooldridge (Ed.). **Agent-oriented Software Engineering** Berlin: Springer, LNCS, v.1957, p.127-141. 2001.

Apêndice A

A Ontologia Ont1 para MAS-ML

Neste apêndice encontra-se o código da Ontologia Ont1, construída durante a aplicação do Observed-MAS para a linguagem MAS-ML.

```
(in-tbox MASML-TBox)
(signature
         :atomic-concepts (;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                            ;;;;;;;  TAO concepts
                            ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                           entity
                            class
                             citizen-class
                                 agent-class
                                 organization-class
                                   main-organization-class
                                    sub-organization-class
                                 object-class
                               environment-class
                                 active-environment-class
                                 passive-environment-class
                              role-class
                                 agent-role-class
                                 object-role-class
                            class-instance
                             agent
                               a-environment
                               p-environment
                               organization
                                main-organization
                                sub-organization
                               object
                               agent-role
                               object-role
                           features
                             goal
                               belief
                               action
                               plan
                               duty
                               right
                               protocol
                               axiom
                               attribute
                               method
                               condition
                            relationship
                              inhabit
                                play
                                ownership
                                specialization
                                association
                              aggregation
                                control
                                dependency
                            msg
                              agent-msg
                              object-msg
                         ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                          ;;;;;;  MAS-ML concepts
                          ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                           model
                              static-model
                             class-model
                               organization-model
                               role-model
                            sequence-model
                           stereotyped-msg
                              create-msg
                                destroy-msg
                                role-commit-msg
                              role-cancel-msg
                                role-activate-msg
                                role-deactivate-msg
                              role-change-msg
                            path
                           class-path
                                agent-class-path
                                organization-class-path
                                  main-organization-class-path
                                  sub-organization-class-path
                                object-class-path
                                active-environment-class-path
                              passive-environment-class-path
                                agent-role-class-path
                                object-role-class-path
                             instance-path
                               agent-instance-path
                               organization-instance-path
                                 main-organization-instance-path
                                 sub-organization-instance-path
                               object-instance-path
                               a-environment-instance-path
                             p-environment-instance-path
                               agent-role-instance-path
                               object-role-instance-path
                           sequence
                           seq-element
                          )
          :roles ( ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                   ;;;;;;;;;; TAO properties
                   ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                   (has-goal :domain (or agent-class organization-class agent-role-class
                                        active-environment-class)
                             :range goal)
                    (has-belief :domain (or agent-class organization-class
                                        agent-role-class active-environment-class)
                               :range belief)
                    (has-axiom :domain organization-class
                              :range axiom)
                    (has-duty :domain agent-role-class
                             :range duty)
                    (has-right :domain agent-role-class
                              :range right)
                  (has-action :range action
                              :parent has-element)
                   (has-precond :domain action
                                 :range condition)
                    (has-poscond :domain action
                                :range condition)
                    (has-plan :range plan)
                  (has-protocol :range protocol)
                    (has-end1 :domain relationship
                            :range class
                              :inverse is-end1)
                  (has-end2 :domain relationship
                            :range class
                              :inverse is-end2)
                  (is-end1 :domain class
                            :range relationship
                             :inverse has-end1)
                    (is-end2 :domain class
                            :range relationship
                             :inverse has-end2)
                  (has-msg :range msg
                           :parent has-element)
                   (has-msg-end :domain msg
                               :inverse is-msg-end)
                  (is-msg-end :inverse has-msg-end)
                  (has-msg-sender :parent has-msg-end)
                  (has-msg-receiver :parent has-msg-end)
                  (is-msg-sender :inverse has-msg-sender)
                  (is-msg-receiver :inverse has-msg-receiver)
                  (is-instanceOf :domain class-instance
                                  :range class)
                    (has-attribute   :domain   (or   object-class   object-role-class   passive-
environment-class)
                                 :range attribute)
                    (has-method :domain (or object-class object-role-class passive-environment-
class)
                                 :range method)
                  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                   ;;;;;;;;;; MAS-ML properties
                   ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                   (has-class :domain static-model
                             :range class
                             :inverse is-in-static-model)
                  (is-in-static-model :inverse has-class)
                  (has-relationship :domain static-model
                                    :range relationship
                                      :inverse is-relationship-of)
                   (is-relationship-of :inverse has-relationship)
                   (has-path ;:domain sequence-model
                            :range path)
                  (has-head :domain path
                             :range (or class-instance class))
                   (has-tail :domain path
                             :range path)
                  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                   ;;;;;;;;;; ordering properties
                   ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                   (has-structure :range sequence)
                   (has-element :range seq-element)
                   (has-first :domain sequence
                              :range seq-element
                              :parent has-element
                              :inverse is-first)
                   (is-first :inverse has-first)
                   (has-last :domain sequence
                             :range seq-element
                             :parent has-element
                             :inverse is-last)
                   (is-last :inverse has-last)
                   (is-before :domain seq-element
                              :range seq-element
                              :transitive t
                              :inverse is-after)
                   (is-after  :domain seq-element
                              :range seq-element
                              :transitive t
                              :inverse is-before)
                  )
 )
 ;; concepts restrictions
 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
 ;;;;;;;;; TAO concepts taxonomy
 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
 (implies class entity)
 (implies class-instance entity)

 (implies environment-class class)
 (implies role-class class)
 (implies citizen-class class)

 (implies active-environment-class environment-class)
 (implies passive-environment-class environment-class)

 (implies agent-role-class role-class)
 (implies object-role-class role-class)

 (implies agent-class citizen-class)
 (implies organization-class citizen-class)
 (implies object-class citizen-class)

 (implies main-organization-class organization-class)
 (implies sub-organization-class organization-class)

 (implies inhabit relationship)
 (implies play relationship)
 (implies ownership relationship)
 (implies specialization relationship)
 (implies association relationship)
 (implies aggregation relationship)
 (implies control relationship)
 (implies dependency relationship)

 (implies goal features)
 (implies belief features)
 (implies action features)
 ;(implies duty features)
 ;(implies right features)
   (implies duty action)
   (implies right action)
 (implies plan features)

 (implies protocol features)
 (implies axiom features)
 (implies attribute features)
 (implies method features)
 (implies condition features)

 (implies agent class-instance)
 (implies a-environment class-instance)
 (implies p-environment class-instance)
 (implies organization class-instance)
 (implies main-organization organization)
 (implies sub-organization organization)
```

```
(implies object class-instance)
(implies agent-role class-instance)
(implies object-role class-instance)

(implies agent-msg msg)
(implies object-msg msg)
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;; MAS-ML concepts taxonomy (it reuses some TAO concepts)
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(implies stereotyped-msg msg)

(implies static-model model)
(implies sequence-model model)

(implies class-model static-model)
(implies organization-model static-model)
(implies role-model static-model)

(implies create-msg stereotyped-msg)
(implies destroy-msg stereotyped-msg)
(implies role-commit-msg stereotyped-msg)
(implies role-cancel-msg stereotyped-msg)
(implies role-activate-msg stereotyped-msg)
(implies role-deactivate-msg stereotyped-msg)
(implies role-change-msg stereotyped-msg)

(equivalent path (or class-path instance-path))

(implies citizen-class-path class-path)
(implies role-class-path class-path)
(implies environment-class-path class-path)
(implies agent-class-path citizen-class-path)

(implies organization-class-path citizen-class-path)
(implies object-class-path citizen-class-path)

(implies agent-role-class-path role-class-path)
(implies object-role-class-path role-class-path)

(implies main-organization-class-path organization-class-path)
(implies sub-organization-class-path organization-class-path)

(implies active-environment-class-path environment-class-path)
(implies passive-environment-class-path environment-class-path)

(implies citizen-instance-path instance-path)
(implies role-instance-path instance-path)
(implies environment-instance-path instance-path)

(implies agent-instance-path citizen-instance-path)
(implies organization-instance-path citizen-instance-path)
(implies object-instance-path citizen-instance-path)

(implies main-organization-instance-path organization-instance-path)
(implies sub-organization-instance-path organization-instance-path)

(implies agent-role-instance-path role-instance-path)
(implies object-role-instance-path role-instance-path)

(implies a-environment-instance-path environment-instance-path)
(implies p-environment-instance-path environment-instance-path)

;;; axiom that guarantee the possibility of the dynamic structure definition
(implies msg seq-element)
(implies action seq-element)
;;;;;;;;;;;;;;;;;;;;;;;;;;
;; ontology axioms
;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;               TAO disjoint restrictions
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(disjoint class class-instance features msg relationship)
(disjoint citizen-class environment-class role-class)
(disjoint agent-class organization-class object-class)
(disjoint main-organization-class sub-organization-class)
(disjoint active-environment-class passive-environment-class)
(disjoint agent-role-class object-role-class)
(disjoint agent a-environment p-environment main-organization
          sub-organization object agent-role object-role)
(disjoint agent-msg object-msg)

(disjoint goal belief protocol attribute)
(disjoint inhabit play ownership specialization association aggregation
          control dependency)
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;             MAS-ML disjoint restrictions
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(disjoint static-model sequence-model)
```

```
(disjoint class-model organization-model role-model)
(disjoint create-msg destroy-msg role-commit-msg
          role-cancel-msg role-activate-msg role-deactivate-msg
          role-change-msg)
(disjoint class-path instance-path)
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;                                              ;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;      Instances-related axioms                ;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;                                              ;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(implies agent (all is-instanceOf agent-class))
(implies organization (all is-instanceOf organization-class))
(implies main-organization (all is-instanceOf main-organization-class))
(implies sub-organization (all is-instanceOf sub-organization-class))
(implies object (all is-instanceOf object-class))
(implies a-environment (all is-instanceOf active-environment-class))
(implies p-environment (all is-instanceOf passive-environment-class))
(implies agent-role (all is-instanceOf agent-role-class))
(implies object-role (all is-instanceOf object-role-class))
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; path axioms
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; pathnames are defined recursively
(implies agent-class-path (and (all has-head agent-class)
                               (some has-tail agent-role-class-path)))
(implies object-class-path (and (all has-head object-class)
                                (some has-tail object-role-class-path)))
(implies main-organization-class-path (and (all has-head main-organization-class)
                                           (some has-tail environment-class-path)))
(implies sub-organization-class-path
             (and (all has-head sub-organization-class)
                  (some has-tail (or environment-class-path organization-class-path))))
(implies agent-role-class-path (and (all has-head agent-role-class)
                                    (some has-tail organization-class-path)))
(implies object-role-class-path (and (all has-head object-role-class)
                                     (some has-tail organization-class-path)))
(implies active-environment-class-path (and (all has-head active-environment-class)
                                            (all has-tail *bottom*)))
(implies passive-environment-class-path (and (all has-head passive-environment-class)
                                             (all has-tail *bottom*)))
(implies agent-instance-path (and (all has-head agent)
                                  (some has-tail agent-role-instance-path)))
(implies object-instance-path (and (all has-head object)
                                   (some has-tail object-role-instance-path)))
(implies main-organization-instance-path (and (all has-head main-organization)
                                              (some has-tail environment-instance-path)))
(implies sub-organization-instance-path
        (and (all has-head sub-organization)
             (some has-tail (or organization-instance-path environment-instance-path))))
(implies agent-role-instance-path (and (all has-head agent-role)
                                       (some has-tail organization-instance-path)))
(implies object-role-instance-path (and (all has-head object-role)
                                        (some has-tail organization-instance-path)))
(implies a-environment-instance-path (and (all has-head a-environment)
                                          (all has-tail *bottom*)))
(implies p-environment-instance-path (and (all has-head p-environment)
                                          (all has-tail *bottom*)))
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;                                order axioms                                ;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; semantics of the first element of a sequence
(implies (and (some is-before *top*) (some is-after *bottom*)) (some is-first *top*))

;; semantics of the last element of a sequence
(implies (and (some is-after *top*) (some is-before *bottom*)) (some is-last *top*))

(implies (some has-first msg) (some has-msg msg))
(implies (some has-last msg) (some has-msg msg))

(implies (some has-first action) (some has-action action))
(implies (some has-last action) (some has-action action))
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;; end of tbox ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(classify-tbox)
(tbox-classified-p)
(tbox-coherent-p)
(tbox-cyclic-p)
```

Apêndice B

QV1 para o método aplicado a MAS-ML

Neste apêndice encontram-se as consultas referentes à fase F1 do método Observed-MAS aplicado a MAS-ML.

Consultas referentes ao domínio de SMAs

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;; This file contains general queries that check the existence of ;;;;;;;;;;
;;;;;; inconsistencies that are model independent. Therefore, these  ;;;;;;;;;;
;;;;;; queries analyze general aspects of a MAS design.             ;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;; Q1 checks relationships that are not in any static-diagram
;; PS: the has-relationship has static-diagram as its domain
(retrieve (?relation)
          (and (?relation relationship)
               (nil ?relation has-relationship)))

;;Q2 checks bad-defined inhabit relationships
(retrieve (?end1 ?end2 ?inh)
        (and (?inh inhabit)
              (?end1 ?inh is-end1)
              (?end2 ?inh is-end2)
              (or (?end1 (or environment-class role-class))
                  (?end2 (or citizen-class role-class)))))

;;Q3 checks  bad-defined play relationships
(retrieve ($?end1 $?end2 ?play)
        (and (?play play)
              ($?end1 ?play is-end1)
              ($?end2 ?play is-end2)
             (not (and ($?end1 (or agent-class sub-organization-class))
                       ($?end2 agent-role-class)))
             (not (and ($?end1 object-class) ($?end2 object-role-class)))))

; the following query has the same meaning of the above one.
;(retrieve ($?end1 $?end2 ?play)
;        (and (?play play)
;             ($?end1 ?play is-end1)
;             ($?end2 ?play is-end2)
;             (or ($?end1 (or role-class environment-class main-organization-class))
;                 ($?end2 (or citizen-class environment-class))
;                 (and ($?end1 (or agent-class sub-organization-class))
;                      (not ($?end2 agent-role-class)))
;                 (and ($?end1 object-class) (not ($?end2 object-role-class))))))

;;Q4 checks  bad-defined ownership relationships
(retrieve ($?end1 $?end2 ?own)
      (and (?own ownership)
            ($?end1 ?own is-end1)
            ($?end2 ?own is-end2)
            (or ($?end1 (or role-class environment-class agent-class object-class))
                  ($?end2 (or citizen-class environment-class)))))

;;Q5 checks  bad-defined specialization relationships
(retrieve (?end1 ?end2 ?spec)
        (and (?spec specialization)
              (?end1 ?spec is-end1)
              (?end2 ?spec is-end2)
              (or (and (?end1 object-class)
                       (?end2 (or organization-class agent-class
                                  environment-class role-class)))
                  (and (?end1 agent-class)
                        (?end2 (or organization-class object-class
                                   environment-class role-class)))
                  (and (?end1 main-organization-class)
                       (?end2 (or agent-class object-class sub-organization-class
                                  environment-class role-class)))
                  (and (?end1 sub-organization-class)
                       (?end2 (or agent-class object-class main-organization-class
                                  environment-class role-class)))
                  (and (?end1 passive-environment-class)
                       (?end2 (or agent-class organization-class object-class
                                  active-environment-class role-class)))
                  (and (?end1 agent-role-class)
                       (?end2 (or citizen-class environment-class object-role-class)))
                  (and (?end1 object-role-class)
                       (?end2 (or citizen-class environment-class agent-role-class))))))

;;Q6 checks  bad-defined aggregation relationships
(retrieve (?end1 ?end2 ?agg)
        (and (?agg aggregation)
              (?end1 ?agg is-end1)
              (?end2 ?agg is-end2)
              (or (and (?end1 object-class)
                       (?end2 (or organization-class agent-class environment-class
                                  role-class)))
                  (and (?end1 agent-role-class)
                       (?end2 (or citizen-class environment-class object-role-class)))
                  (and (?end1 object-role-class)
                       (?end2 (or citizen-class environment-class agent-role-class)))
                 (?end1 (or agent-class organization-class environment-class)))))

;;Q7 checks  bad-defined control relationships
(retrieve (?end1 ?end2 ?ctrl)
        (and (?ctrl control)
              (?end1 ?ctrl is-end1)
              (?end2 ?ctrl is-end2)
             (or (?end1 (or citizen-class environment-class object-role-class))
                  (and (?end1 agent-role-class)
                       (?end2 (or citizen-class environment-class object-role-class))))))

;;Q8 checks  bad-defined dependency relationships
(retrieve (?end1 ?end2 ?dep)
        (and (?dep dependency)
              (?end1 ?dep is-end1)
              (?end2 ?dep is-end2)
              (or (?end1 (or agent-class organization-class environment-class))
                 (and (?end1 object-class)
                       (?end2 (or organization-class agent-class environment-class
                                  role-class)))
                  (and (?end1 agent-role-class)
                       (?end2 (or citizen-class environment-class)))
                  (and (?end1 object-role-class)
                       (?end2 (or citizen-class environment-class agent-role-class))))))


;; Q10 checks  classes that are not in any static-diagram
(retrieve (?class-alone)
            (and (?class-alone class)
                  (NIL ?class-alone has-class)))

;; Q11-A checks agent-classes that don't have assigned goals
(retrieve (?agentwithoutgoal)
            (and (?agentwithoutgoal agent-class)
                  (?agentwithoutgoal NIL has-goal)))

;; Q11-B checks organization-classes that don't have assigned goals
(retrieve (?orgwithoutgoal)
            (and (?orgwithoutgoal organization-class)
                  (?orgwithoutgoal NIL has-goal)))

;; Q11-C checks agent-role-classes that don't have assigned goals
(retrieve (?rolewithoutgoal)
            (and (?rolewithoutgoal agent-role-class)
                  (?rolewithoutgoal NIL has-goal)))

;; Q12 checks citizens that don't inhabit any environemnt
(retrieve (?no-citizen)
          (and (?no-citizen citizen-class)
                (not (?no-citizen (some is-end1
                                 (and inhabit (some has-end2 environment-class)))))))

;; Q12A checks environments that aren't the habitat of any citizen
(retrieve (?no-habitat)
          (and (?no-habitat environment-class)
              (not (?no-habitat (some is-end2 (and inhabit (some has-end1 citizen-class)))))))

;; Q13 checks agents that don't have assigned roles
(retrieve (?agentwithoutrole)
           (and  (?agentwithoutrole agent-class)
                 (not (?agentwithoutrole (some is-end1
                         (and play (some has-end2 agent-role-class)))))))
```

```
;; Q14 checks organizations that don't own roles
(retrieve (?org-no-owner)
            (and  (?org-no-owner organization-class)
                  (not (?org-no-owner (some is-end1 (and ownership (some
                                       has-end2 agent-role-class)))))))

;; Q15 checks main-organizations that play roles
(retrieve (?org ?play)
            (and (?org main-organization-class)
                  (?play (and play (some has-end2 agent-role-class)))
                  (?org ?play is-end1)))

;; Q16 checks sub-organizations that don't have assigned roles
(retrieve (?org-no-play)
            (and (?org-no-play sub-organization-class)
                 (not (?org-no-play (some is-end1 play)))))

;; Q17 checks organization-classes that don't have assigned axioms
(retrieve (?org)
            (and (?org organization-class)
                  (?org NIL has-axiom)))

;; Q18 checks role-classes that aren't owned by any organization
(retrieve (?role-no-owned)
            (and (?role-no-owned role-class)
                 (not (?role-no-owned (some is-end2 ownership)))))

;; Q19 checks paths that aren't part of any sequence-model
(retrieve (?path-without-model)
     (and (?path-without-model path)
          (nil ?path-without-model has-path)))

;; Q20 checks for bad-defined object-msgs
(retrieve (?bad-obj-msg)
     (and (?bad-obj-msg object-msg)
          ($?sender ?bad-obj-msg is-msg-sender)
           ($?receiver ?bad-obj-msg is-msg-receiver)
          (not (and  ($?sender  (or citizen-class-path citizen-instance-path
                              environment-class-path environment-instance-path))
                    ($?receiver (or object-class-path object-instance-path
                              passive-environment-class-path p-environment-instance-path))))))

;; Q20A same object as sender and receiver
(retrieve (?bad-obj-msg)
     (and (?bad-obj-msg object-msg)
          (?sender-receiver ?bad-obj-msg is-msg-sender)
           (?sender-receiver ?bad-obj-msg is-msg-receiver)
          (?sender-receiver (or object-class-path object-instance-path
                               passive-environment-class-path p-environment-instance-path))))

;; Q21 checks for bad-defined agent-msgs
(retrieve ($?sender $?receiver ?ag-msg)
        (and (?ag-msg agent-msg)
              ($?sender ?ag-msg is-msg-sender)
              ($?receiver ?ag-msg is-msg-receiver)
              (or (and ($?sender (or agent-role-class-path agent-role-instance-path))
                     ($?receiver (or agent-class-path agent-instance-path
                                  object-class-path object-instance-path
                                  sub-organization-class-path sub-organization-instance-path
                                  passive-environment-class-path p-environment-instance-path
                                  object-role-class-path object-role-instance-path)))
                  (and ($?sender (or active-environment-class-path a-environment-instance-path
                                main-organization-class-path main-organization-instance-path))
                       ($?receiver (or agent-class-path agent-instance-path
                                  sub-organization-class-path sub-organization-instance-path
                                  object-class-path object-instance-path
                                  passive-environment-class-path p-environment-instance-path
                                  object-role-class-path object-role-instance-path)))
                  (not ($?sender (or agent-role-class-path agent-role-instance-path
                               active-environment-class-path a-environment-instance-path
                               main-organization-class-path main-organization-instance-path)))
                 (not ($?receiver (or agent-role-class-path agent-role-instance-path
                               active-environment-class-path a-environment-instance-path
                            main-organization-class-path main-organization-instance-path))))))

;; Q22 checks for bad-defined create-msgs
(retrieve (?sender ?receiver ?create)
      (and (?create create-msg)
           (?sender ?create is-msg-sender)
           (?receiver ?create is-msg-receiver)
           (or (and (?sender (or main-organization-class-path main-organization-instance-path
                                   agent-class-path agent-instance-path))
                   (?receiver (or role-class-path role-instance-path)))
              (and (?sender (or sub-organization-class-path sub-organization-instance-path))
             (?receiver (or main-organization-class-path main-organization-instance-path
                                  role-class-path role-instance-path)))
          (and (?sender (or environment-class-path environment-instance-path))
               (?receiver (or role-class-path role-instance-path)))
          (and (?sender (or object-class-path object-instance-path))
               (?receiver (or agent-class-path agent-instance-path
                                organization-class-path organization-instance-path
                                environment-class-path environment-instance-path
                                role-class-path role-instance-path)))
                 (?sender (or role-class-path role-instance-path)))))

;; Q22-A checks for bad-defined create-msgs (auto-creation)
(retrieve (?sender ?create)
        (and (?create create-msg)
              (?sender ?create is-msg-sender)
              (?sender ?create is-msg-receiver)))

;; Q23 checks for bad-defined destroy-msgs
(retrieve (?sender ?receiver ?destroy)
        (and (?destroy destroy-msg)
              (?sender ?destroy is-msg-sender)
              (?receiver ?destroy is-msg-receiver)
           (or (and (?sender (or main-organization-class-path main-organization-instance-path
                                   agent-class-path agent-instance-path))
                       (?receiver (or role-class-path role-instance-path)))
               (and (?sender (or sub-organization-class-path sub-organization-instance-path))
              (?receiver (or main-organization-class-path main-organization-instance-path
                                     role-class-path role-instance-path)))
                  (and (?sender (or environment-class-path environment-instance-path))
                       (?receiver (or role-class-path role-instance-path)))
                  (and (?sender (or object-class-path object-instance-path))
                       (?receiver (or agent-class-path agent-instance-path
                                     organization-class-path organization-instance-path
                                     environment-class-path environment-instance-path
                                     role-class-path role-instance-path)))
                 (?sender (or role-class-path role-instance-path)))))

;; Q24 checks for bad-defined role-commit-msgs
;; 24A wrong sender
(retrieve (?bad-commit-ag-role ?wrong-sender)
 (and (?bad-commit-ag-role role-commit-msg)
      (?bad-commit-ag-role ?wrong-sender has-msg-sender)
      (not (?wrong-sender (or agent-class-path agent-instance-path)))))

;; 24B sender ok, but wrong receivers (agents)
(retrieve (?bad-commit-ag-role ?sender ?wrong-receiver)
 (and (?bad-commit-ag-role role-commit-msg)
      (?bad-commit-ag-role ?sender has-msg-sender)
      (?bad-commit-ag-role ?wrong-receiver has-msg-receiver)
      (?sender (or agent-class-path agent-instance-path))
      (not (?wrong-receiver (or agent-class-path agent-instance-path)))))

;; 24B sender ok, but wrong receivers (sub-orgs)
(retrieve (?bad-commit-ag-role ?sender ?wrong-receiver)
 (and (?bad-commit-ag-role role-commit-msg)
      (?bad-commit-ag-role ?sender has-msg-sender)
      (?bad-commit-ag-role ?wrong-receiver has-msg-receiver)
      (?sender (or sub-organization-class-path sub-organization-instance-path))
     (not (?wrong-receiver (or sub-organization-class-path sub-organization-instance-path)))))

;; 24C sender and receiver ok but representing different agents
(retrieve (?bad-commit-ag-role ?sender ?receiver)
 (and (?bad-commit-ag-role role-commit-msg)
      (?bad-commit-ag-role ?sender has-msg-sender)
      (?bad-commit-ag-role ?receiver has-msg-receiver)
      (?sender (or agent-class-path agent-instance-path))
      (?receiver (or agent-class-path agent-instance-path))))

;; 24C sender and receiver ok but representing different sub-orgs
(retrieve (?bad-commit-ag-role ?sender ?receiver)
 (and (?bad-commit-ag-role role-commit-msg)
      (?bad-commit-ag-role ?sender has-msg-sender)
      (?bad-commit-ag-role ?receiver has-msg-receiver)
      (?sender (or sub-organization-class-path sub-organization-instance-path))
      (?receiver (or sub-organization-class-path sub-organization-instance-path))))

;; 24D sender and receiver are the same agent playing the role they're trying to commit
(retrieve (?bad-commit-ag-role $?sender $?receiver)
 (and (?bad-commit-ag-role role-commit-msg)
      (?bad-commit-ag-role $?sender has-msg-sender)
      (?bad-commit-ag-role $?receiver has-msg-receiver)
      ($?sender agent-instance-path)
      ($?receiver agent-instance-path)
      ($?sender ?head has-head) ($?receiver ?head has-head)
      (?role agent-role-instance-path)
      ($?sender ?role has-tail) ($?receiver ?role has-tail)))


;; 24D sender and receiver are the same agent playing the role they're trying to commit
(retrieve (?bad-commit-ag-role $?sender $?receiver)
 (and (?bad-commit-ag-role role-commit-msg)
      (?bad-commit-ag-role $?sender has-msg-sender)
      (?bad-commit-ag-role $?receiver has-msg-receiver)
      ($?sender sub-organization-instance-path)
      ($?receiver sub-organization-instance-path)
      ($?sender ?head has-head) ($?receiver ?head has-head)
      (?role agent-role-instance-path)
      ($?sender ?role has-tail) ($?receiver ?role has-tail)))

;; 24E COMMIT-TO-OBJECT-ROLE WRONG SENDER
(retrieve (?bad-commit-obj-role ?sender ?receiver)
 (and (?bad-commit-obj-role role-commit-msg)
      (?bad-commit-obj-role ?sender has-msg-sender)
      (?bad-commit-obj-role ?receiver has-msg-receiver)
      (?sender (or object-class-path object-instance-path
                   environment-class-path environment-instance-path
                   role-class-path role-instance-path))))

;; 24F COMMIT-TO-OBJECT-ROLE SENDER OK AND WRONG RECEIVER
(retrieve (?bad-commit-obj-role ?sender ?receiver)
 (and (?bad-commit-obj-role role-commit-msg)
      (?bad-commit-obj-role ?sender has-msg-sender)
      (?bad-commit-obj-role ?receiver has-msg-receiver)
      (?sender (or agent-class-path agent-instance-path
                   organization-class-path organization-instance-path))
      (not (?receiver (or object-class-path object-instance-path)))))

;; Q25 checks for bad-defined role-cancel-msgs
(retrieve (?bad-cancel-ag-role ?sender ?receiver)
 (and (?bad-cancel-ag-role role-cancel-msg)
      (?bad-cancel-ag-role ?sender has-msg-sender)
      (?bad-cancel-ag-role ?receiver has-msg-receiver)
      (or (?sender (or object-class-path object-instance-path
                       environment-class-path environment-instance-path
                       role-class-path role-instance-path
                       main-organization-class-path main-organization-instance-path))
          (and (?sender (or agent-class-path agent-instance-path))
               (?receiver (or agent-class-path agent-instance-path))
               (?sender ?head-s has-head) (?receiver ?head-r has-head))
          (and (?sender (or sub-organization-class-path sub-organization-instance-path))
               (?receiver (or sub-organization-class-path sub-organization-instance-path))
               (?sender ?head-s has-head) (?receiver ?head-r has-head)))))

(retrieve (?bad-cancel-obj-role ?sender ?receiver)
 (and (?bad-cancel-obj-role role-cancel-msg)
      (?bad-cancel-obj-role ?sender has-msg-sender)
      (?bad-cancel-obj-role ?receiver has-msg-receiver)
      (or (?sender (or object-class-path object-instance-path
                       environment-class-path environment-instance-path
                       role-class-path role-instance-path
                       main-organization-class-path main-organization-instance-path))
          (and (?sender (or agent-class-path agent-instance-path
                            organization-class-path organization-instance-path))
               (?receiver (not (or object-class-path object-instance-path)))))))

;; Q26 checks for bad-defined role-activate-msgs
(retrieve (?sender ?receiver ?activate)
        (and (?activate role-activate-msg)
              (?sender ?activate is-msg-sender)
              (?receiver ?activate is-msg-receiver)
              (or (?sender (or main-organization-class-path main-organization-instance-path
                                role-class-path role-instance-path
                                environment-class-path environment-instance-path
                                object-class-path object-instance-path))
                  (and (?sender (or agent-class-path agent-instance-path
                                    sub-organization-class-path sub-organization-instance-path))
                       (?receiver (or citizen-class-path citizen-instance-path
                                      environment-class-path environment-instance-path
                                      object-role-class-path object-role-instance-path))))))

;; Q27 checks for bad-defined role-deactivate-msgs
(retrieve (?sender ?receiver ?deactivate)
        (and (?deactivate role-deactivate-msg)
              (?sender ?deactivate is-msg-sender)
              (?receiver ?deactivate is-msg-receiver)
              (or (?sender (or main-organization-class-path main-organization-instance-path
                                role-class-path role-instance-path
                                environment-class-path environment-instance-path
                                object-class-path object-instance-path))
                  (and (?sender (or agent-class-path agent-instance-path
                                    sub-organization-class-path sub-organization-instance-path))
                       (?receiver (or citizen-class-path citizen-instance-path
                                      environment-class-path environment-instance-path
                                      object-role-class-path object-role-instance-path))))))

;; Q28 checks for bad-defined role-change-msgs
(retrieve (?sender ?receiver ?change)
        (and (?change role-change-msg)
              (?sender ?change is-msg-sender)
              (?receiver ?change is-msg-receiver)
              (or (?sender (or main-organization-class-path main-organization-instance-path
                               role-class-path role-instance-path
                               environment-class-path environment-instance-path
                               object-class-path object-instance-path))
                   (and (?sender (or agent-class-path agent-instance-path
                                   sub-organization-class-path sub-organization-instance-path))
                        (?receiver (or citizen-class-path citizen-instance-path
                                      environment-class-path environment-instance-path
                                      object-role-class-path object-role-instance-path))))))

;; Paths identify instances of MAS-ML entities, therefore there is the need of
;; check if all instances have an associated id.
;; Q29 checks for class-instances without paths
(retrieve (?path-without-source)
          (and (?path-without-source instance-path)
               (?path-without-source ?head has-head)
               (?head nil is-instanceOf)))

;; Messages must be sent and received by paths
;; Q30 checks for bad-defined msgs
(retrieve (?msg-bad-def ?sender ?receiver)
  (and (?msg-bad-def msg)
          (?msg-bad-def ?sender has-msg-sender) (?msg-bad-def ?receiver has-msg-receiver)
          (or (not (?sender path)) (not (?receiver path)))))

;; Q31 checks agent-msgs that are in an agent-role-class but aren't in any protocol
(retrieve (?ag-msg ?ag-role-class)
          (and (?ag-msg agent-msg)
               (?ag-role-class agent-role-class)
               (?ag-role-class ?ag-msg has-msg)
               (not (?ag-msg (some (inv has-msg) protocol)))))

;; Q32 checks protocols that aren't in any agent-role-class
(retrieve (?prtcl-alone)
          (and (?prtcl-alone protocol)
               (not (?prtcl-alone (some (inv has-protocol) agent-role-class)))))

;; Q33 checks plans that aren't in any agent-class, organization-class or
;; active-environment-class
(retrieve (?plan-alone)
          (and (?plan-alone plan)
               (not (?plan-alone (some (inv has-plan)
                                 (or agent-class organization-class
                                     active-environment-class))))))

;; Q34 checks actions that aren't in any agent-class, organization-class or
;; active-environment-class or plan.
(retrieve (?action-alone)
(and (?action-alone action)
               (not (?action-alone (some (inv has-action)
                                 (or agent-class organization-class
                                     active-environment-class))))))

;; Q35 checks agent-msgs that aren't in any agent-role-class,
;; main-organization-class or active-environment-class
(retrieve (?ag-msg-alone)
          (and (?ag-msg-alone agent-msg)
               (not (?ag-msg-alone (some (inv has-msg)
                                         (or agent-role-class
                                             main-organization-class
                                             active-environment-class))))))

;; Q36 checks paths without heads (their id)
(retrieve (?path-without-head)
   (and (?path-without-head path)
        (?path-without-head nil has-head)))

;; Q37 checks protocols without structure
(retrieve (?prtcl-without-struct)
   (and (?prtcl-without-struct protocol)
        (?prtcl-without-struct nil has-structure)))

;; Q38 checks plans without structure
(retrieve (?plan-without-struct)
   (and (?plan-without-struct plan)
        (?plan-without-struct nil has-structure)))

;; Q39 checks bad-defined protocol structures
;; it shows a triple containing the protocol, its structure and the elements which
;; must have be in both and are not.
(retrieve (?bad-struct ?prtcl ?element)
   (and (?bad-struct sequence)
        (?prtcl protocol)
        (?prtcl ?bad-struct has-structure)
        (or (and (?bad-struct ?element has-element)
                 (not (?prtcl ?element has-msg)))
            (and (?prtcl ?element has-msg)
                 (not (?bad-struct ?element has-element))))))

;; Q40 checks bad-defined structures
(retrieve (?bad-struct)
   (and (?bad-struct sequence)
        (?bad-struct ?element1 has-first)
        (?bad-struct ?element2 has-last)
        (?element2 ?element1 is-before)))

;; Q41 checks agent-role-classes without assigned protocols
(retrieve (?ag-role-no-prtcl)
   (and (?ag-role-no-prtcl agent-role-class)
        (?ag-role-no-prtcl nil has-protocol)))

;; Q42 agents must play a role in an organization and this must be
;; modeled. This query checks if it isn't modeled properly and returns
;; the duple agent, role which role doesn't belong to any organization
(retrieve (?agent ?role)
           (and  (?agent agent-class)
                 (?pl play)
                 (?role agent-role-class)
                 (?agent ?pl is-end1)
                 (?role ?pl is-end2)
                 (not (?role (some is-end2
                                (and ownership (some has-end1 organization-class)))))))

;; Q42A sub-orgs must play a role in an organization and this must be
;; modeled. This query checks if it isn't modeled properly and returns
;; the duple sub-org, role which role doesn't belong to any organization
(retrieve (?sub-org ?role)
           (and  (?sub-org sub-organization-class)
                 (?pl play)
                 (?role agent-role-class)
                 (?sub-org ?pl is-end1)
                 (?role ?pl is-end2)
                 (not (?role (some is-end2
                                   (and ownership (some has-end1 organization-class)))))))

;; Q43 protocols don't have first or last msgs, but their structure!
(retrieve (?bad-prtcl ?first-or-last)
   (and (?bad-prtcl protocol)
        (?first-or-last msg)
        (or (?bad-prtcl ?first-or-last has-first)
            (?bad-prtcl ?first-or-last has-last))))
```

Consultas referentes as propriedades intra-diagramas MAS-ML

```
;; These queries analyze the class diagrams structures and their
;; results are the inconsistencies that were found in the diagram

;; Q1 checks for role-classes definition in a class-diagram
(retrieve (?rolecl ?clmd)
        (and (?rolecl role-class)
              (?clmd class-model)
              (?clmd ?rolecl has-class)
        )
)

;; Q2 checks for play relationships definition in a class-diagram
(retrieve (?pl ?clmd)
        (and (?pl play)
              (?clmd class-model)
              (?clmd ?pl has-relationship)
        )
)

;; Q3 checks for ownership relationships definition in a class-diagram
(retrieve (?own ?clmd)
        (and (?own ownership)
              (?clmd class-model)
              (?clmd ?own has-relationship)
        )
)

;; Q4 checks for control relationships definition in a class-diagram
(retrieve (?ctrl ?clmd)
        (and (?ctrl control)
              (?clmd class-model)
              (?clmd ?ctrl has-relationship)
        )
)

;;Q5 checks for bad-defined inhabit relationships in a class-diagram
```

```
(retrieve (?inh ?clmd)
          (and (?inh inhabit)
                (?clmd class-model)
                (?clmd ?inh has-relationship)
               (?end1 ?inh is-end1)
                (?end2 ?inh is-end2)
               (or (?end1 (or agent-class organization-class role-class environment-class))
                   (?end2 (or citizen-class role-class)))
          )
)

;;Q6 checks for bad-defined association relationships in a class-diagram
(retrieve (?ass ?clmd)
          (and (?ass association)
                (?clmd class-model)
                (?clmd ?ass has-relationship)
               (?end1 ?ass is-end1)
                (?end2 ?ass is-end2)
               (or (and (?end1 (or agent-class organization-class))
                        (?end2 (or agent-class organization-class environment-class)))
                   (and (?end1 environment-class)
                        (?end2 citizen-class))
                   (?end1 role-class)
                   (?end2 role-class)
               )
          )
)

;;Q7 checks for bad-defined specialization relationships in a class-diagram
(retrieve (?spec ?clmd)
          (and (?spec specialization)
                (?clmd class-model)
                (?clmd ?spec has-relationship)
               (?end1 ?spec is-end1)
                (?end2 ?spec is-end2)
               (or (?end1 role-class) (?end2 role-class)
                 (and (?end1 agent-class)
                      (?end2 (or object-class organization-class environment-class)))
                 (and (?end1 main-organization-class)
                      (?end2 (or object-class sub-organization-class
                                 agent-class environment-class)))
                 (and (?end1 sub-organization-class)
                      (?end2 (or object-class main-organization-class
                                 agent-class environment-class)))
                 (and (?end1 active-environment-class)
                      (?end2 (or citizen-class passive-environment-class)))
                 (and (?end1 passive-environment-class)
                      (?end2 (or citizen-class active-environment-class)))
               )
          )
)

;;Q8 checks for bad-defined aggregation relationships in a class-diagram
(retrieve (?agg ?clmd)
          (and (?agg aggregation)
                (?clmd class-model)
                (?clmd ?agg has-relationship)
               (?end1 ?agg is-end1)
                (?end2 ?agg is-end2)
               (or (?end1 (or role-class agent-class organization-class environment-class))
                   (?end2 (or role-class agent-class organization-class environment-class))
               )
          )
)

;;Q9 checks for bad-defined dependency relationships in a class-diagram
(retrieve (?dep ?clmd)
          (and (?dep dependency)
                (?clmd class-model)
                (?clmd ?dep has-relationship)
               (?end1 ?dep is-end1)
                (?end2 ?dep is-end2)
               (or (?end1 (or role-class agent-class organization-class environment-class))
                   (?end2 (or role-class agent-class organization-class environment-class))
               )
          )
)

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; These queries analyze the organization diagrams structures and their
;; results are the inconsistencies that were found in the diagram
;; Organization diagrams can have all class types defined in the MAS-ML metamodel
;; and only the relationships inhabit, play and ownership

;; Q1 checks for specialization relationships definition in an organization-diagram
(retrieve (?spec ?orgmd)
        (and (?spec specialization)
              (?orgmd organization-model)
              (?orgmd ?spec has-relationship)
        )
)

;; Q2 checks for association relationships definition in an organization-diagram
(retrieve (?assoc ?orgmd)
        (and (?assoc association)
              (?orgmd organization-model)
              (?orgmd ?assoc has-relationship)
        )
)

;; Q3 checks for aggregation relationships definition in an organization-diagram
(retrieve (?agg ?orgmd)
        (and (?agg aggregation)
              (?orgmd organization-model)
              (?orgmd ?agg has-relationship)
        )
)

;; Q4 checks for control relationships definition in an organization-diagram
(retrieve (?ctrl ?orgmd)
        (and (?ctrl control)
              (?orgmd organization-model)
              (?orgmd ?ctrl has-relationship)
        )
)

;; Q5 checks for dependency relationships definition in an organization-diagram
(retrieve (?dep ?orgmd)
        (and (?dep dependency)
              (?orgmd organization-model)
              (?orgmd ?dep has-relationship)
        )
)


;; Q6 checks for relationships defined between roles
(retrieve (?rel ?orgmd)
        (and (?rel relationship)
              (?orgmd organization-model)
              (?orgmd ?rel has-relationship)
              (?end1 ?rel is-end1)
             (?end2 ?rel is-end2)
             (?end1 role-class) (?end2 role-class)
        )
)

;; Q7 checks for relationships defined between agents
(retrieve (?rel ?orgmd)
        (and (?rel relationship)
              (?orgmd organization-model)
              (?orgmd ?rel has-relationship)
              (?end1 ?rel is-end1)
             (?end2 ?rel is-end2)
             (?end1 agent-class) (?end2 agent-class)
        )
)

;; Q8 checks for relationships defined between environment
(retrieve (?rel ?orgmd)
        (and (?rel relationship)
              (?orgmd organization-model)
              (?orgmd ?rel has-relationship)
              (?end1 ?rel is-end1)
             (?end2 ?rel is-end2)
             (?end1 environment-class) (?end2 environment-class)
        )
)

;;Q9 checks for bad-formed organization diagrams
(retrieve (?rel ?orgmd-bad)
        (and (?rel relationship)
              (?orgmd-bad organization-model)
              (?orgmd-bad ?rel has-relationship)
             (?rel (or association aggregation specialization control dependency))
        )
)

;;Q10 checks organization-models which model main-organizations and
;;    didn't have inhabit relationship
(retrieve (?orgmd-without-env ?main-org)
  (and (?orgmd-without-env organization-model)
       (?main-org main-organization-class)
       (?orgmd-without-env ?main-org has-class)
       (or (not (?main-org (some is-end1 inhabit)))
           (not (?orgmd-without-env (some has-relationship inhabit)))
       )
  )
)
;; In an organization diagram the internal properties of the organization
;; which is being modeled as well as the internal properities of the roles
;; it defines are considered in isolation and, therefore, are in the
;; GeneralQueriesFinal.racer


;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; These queries analyze the role diagrams structures and their
;; results are the inconsistencies that were found in the diagram
;; Role diagrams can have role-class and object-class
;; and the relationships control, dependency, association, aggregation
;; and specialization

;; Q1 checks for agent-classes definition in a role-diagram
(retrieve (?agcl ?rlmd)
        (and (?agcl agent-class)
              (?rlmd role-model)
              (?rlmd ?agcl has-class)
        )
)

;; Q2 checks for organization-classes definition in a role-diagram
(retrieve (?orgcl ?rlmd)
        (and (?orgcl organization-class)
              (?rlmd role-model)
              (?rlmd ?orgcl has-class)
        )
)

;; Q3 checks for environment-classes definition in a role-diagram
(retrieve (?envcl ?rlmd)
        (and (?envcl environment-class)
              (?rlmd role-model)
              (?rlmd ?envcl has-class)
        )
)

;; Q4 checks for play relationships definition in a role-diagram
(retrieve (?pl ?rlmd)
        (and (?pl play)
              (?rlmd role-model)
              (?rlmd ?pl has-relationship)
        )
)

;; Q5 checks for ownership relationships definition in a role-diagram
(retrieve (?own ?rlmd)
        (and (?own ownership)
              (?rlmd role-model)
              (?rlmd ?own has-relationship)
        )
)

;; Q6 checks for inhabit relationships definition in a role-diagram
(retrieve (?inh ?rlmd)
        (and (?inh inhabit)
              (?rlmd role-model)
              (?rlmd ?inh has-relationship)
        )
)

;; the control relationship was previously analyzed by the general queries

;;Q7 checks for bad-defined dependency relationships in role diagrams
(retrieve (?dep ?end1 ?end2 ?rlmd)
        (and (?dep dependency)
              (?rlmd role-model)
              (?rlmd ?dep has-relationship)
             (?end1 ?dep is-end1)
              (?end2 ?dep is-end2)
             (or (?end1 (or citizen-class environment-class))
                 (?end2 (or citizen-class environment-class))
                 (and (?end1 agent-role-class) (?end2 (or citizen-class environment-class)))
                 (and (?end1 object-role-class)
                      (?end2 (or agent-role-class citizen-class environment-class)))
             )
        )
)

;;Q8 checks for bad-defined aggregation relationships
(retrieve (?agg ?end1 ?end2 ?rlmd)
        (and (?agg aggregation)
              (?rlmd role-model)
              (?rlmd ?agg has-relationship)
             (?end1 ?agg is-end1)
              (?end2 ?agg is-end2)
             (or (?end1 (or citizen-class environment-class))
                 (?end2 (or citizen-class environment-class))
                 (and (?end1 agent-role-class) (not (?end2 agent-role-class)))
                 (and (?end1 object-role-class) (not (?end2 object-role-class)))
             )
        )
)

;;Q9 checks for bad-defined specialization relationships
(retrieve (?spec ?end1 ?end2 ?rlmd)
        (and (?spec specialization)
              (?rlmd role-model)
              (?rlmd ?spec has-relationship)
             (?end1 ?spec is-end1)
              (?end2 ?spec is-end2)
             (or (?end1 (or citizen-class environment-class))
                 (?end2 (or citizen-class environment-class))
                 (and (?end1 agent-role-class) (not (?end2 agent-role-class)))
                 (and (?end1 object-role-class) (not (?end2 object-role-class)))
             )
        )
)

;;Q10 checks for bad-defined association relationships
(retrieve (?assoc ?end1 ?end2 ?rlmd)
        (and (?assoc association)
              (?rlmd role-model)
              (?rlmd ?assoc has-relationship)
             (?end1 ?assoc is-end1)
              (?end2 ?assoc is-end2)
             (or (?end1 (or agent-class organization-class environment-class))
                 (?end2 (or agent-class organization-class environment-class))
                 (and (?end1 object-class) (not (?end2 object-class)))
             )
        )
)

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; This file contains queries related to the msgs from MAS-ML, which are described
;; in sequence diagrams.
;; Sequence diagrams also describe the execution of plans and protocols. In this
;; sense, a sequence diagram is composed of class-paths (of agents, organizations
;; and active-environments) when we are modeling plans, and agent-role-class-paths
;; when we are modeling protocols.

;; Q1 check stereotyped and object-msgs that aren't part of any sequence-model
(retrieve (?msg-without-model)
     (and (?msg-without-model (or stereotyped-msg object-msg))
           (not (?msg-without-model (some (inv has-msg) sequence-model)))
     )
)

;; Q2 check agent-msgs that are in a sequence-model and are not in an
;; agent-role-class or main-organization-class or active-environment-class
;; ps: the inclusion of agent-msgs in main-org and act-env are new
(retrieve (?msg-without-owner ?seq-md ?sender)
     (and (?msg-without-owner agent-msg)
          (?seq-md sequence-model)
          (?seq-md ?msg-without-owner has-msg)
          (?msg-without-owner ?sender has-msg-sender)
           (not (?msg-without-owner
               (some (inv has-msg) (or agent-role-class main-organization-class
                                       active-environment-class))))
     )
)

;; Q3 check agent-msgs that are in a sequence-model, which sender is an agent-role
;; and which didn't belong to ANY protocol
(retrieve (?msg-without-prtcl ?ag-role-sender ?seq-md)
     (and (?msg-without-prtcl agent-msg)
          (?ag-role-sender agent-role-class-path)
          (?seq-md sequence-model)
          (?seq-md ?msg-without-prtcl has-msg)
          (?ag-role-sender ?msg-without-prtcl is-msg-sender)
           (not (?msg-without-prtcl (some (inv has-msg) protocol)))
     )
)

;; Q3A check agent-msgs that are in a sequence-model, which sender is an agent-role
;; but the msg didn't belong to the sender protocols
;; This query also return the Q3 queries results!
(retrieve (?msg-without-prtcl ?ag-role-sender ?seq-md)
     (and (?msg-without-prtcl agent-msg)
          (?ag-role-sender agent-role-class-path)
          (?seq-md sequence-model)
          (?prtcl protocol)
          (?seq-md ?msg-without-prtcl has-msg)
          (?seq-md ?ag-role-sender has-path)
          (?ag-role-sender ?msg-without-prtcl is-msg-sender)
          (?ag-role-sender ?ag-role has-head)
          (?ag-role ?prtcl has-protocol)
           (not (?prtcl ?msg-without-prtcl has-msg))
     )
)

;; Q4 check seq-models which have plans and don't have any agent/org or
;; act-env-class-paths
(retrieve (?seq-md ?plan-without-owner)
     (and (?seq-md sequence-model)
          (?plan-without-owner plan)
          (?seq-md ?plan-without-owner has-plan)
          (not (?seq-model (some (inv has-path) (or agent-class-path
                       organization-class-path active-environment-class-path))))
     )
)

;; Q5 check paths that aren't part of any sequence-model
(retrieve (?path-without-model)
     (and (?path-without-model path)
           (not (?path-without-model (some (inv has-path) sequence-model)))
     )
)

;; Q6 check sequence-models without structure
(retrieve (?seq-without-struct)
     (and (?seq-without-struct sequence-model)
           (?seq-without-struct nil has-structure)
     )
)

;; Q7 check bad-structured sequences in sequence-models
(retrieve (?seq-md-loop ?element1 ?element2)
   (and (?seq-md sequence-model)
        (?seq sequence)
        (?seq-md-loop ?seq has-structure)
        (?seq ?element1 has-first)
        (?seq ?element2 has-last)
        (?element2 ?element1 is-before)
   )
)

;; Q8 check bad-defined sequence-model which models a protocol
(retrieve (?seqmd ?role-without-prtcl ?prtcl)
 (and (?seqmd sequence-model)
      (?role-without-prtcl agent-role-class)
      (?prtcl protocol)
      (?role-path agent-role-class-path)
      (?seqmd ?role-path has-path)
      (?role-path ?role-without-prtcl has-head)
      (?struct sequence)
      (?seqmd ?struct has-structure)
      (?prtcl ?struct has-structure)
      (not (?role-without-prtcl ?prtcl has-protocol))
 )
)
```

Apêndice C
A ontologia Ont2 para MAS-ML

Neste apêndice encontra-se o código da ontologia Ont2 construída durante a aplicação do Observed-MAS para MAS-ML.

```
(in-tbox MASML-2-TBox)

(signature
        :atomic-concepts (;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                            ;;;;;;;  TAO concepts
                            ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                           class
                            citizen-class
                                agent-class
                                organization-class
                                  main-organization-class
                                   sub-organization-class
                                object-class
                              environment-class
                                active-environment-class
                                passive-environment-class
                             role-class
                                agent-role-class
                                object-role-class
                            class-instance
                            agent
                              a-environment
                              p-environment
                              organization
                               main-organization
                               sub-organization
                              object
                              agent-role
                              object-role
                          features
                            goal
                              belief
                              action
                              plan
                              duty
                              right
                              protocol
                              axiom
                              attribute
                              method
                            relationship
                             inhabit
                               play
                               ownership
                               specialization
                               association
                             aggregation
                               control
                               dependency
                             play-in
                            msg
                             agent-msg
                             object-msg
                        ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                          ;;;;;;  MAS-ML concepts
                          ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                          model
                             static-model
                            class-model
                              organization-model
                              role-model
                           sequence-model
                          stereotyped-msg
                             create-msg
                               destroy-msg
                               role-commit-msg
```

```
                    role-cancel-msg
                      role-activate-msg
                      role-deactivate-msg
                    role-change-msg
                  path
                 class-path
                     citizen-class-path
                      agent-class-path
                      organization-class-path
                        main-organization-class-path
                        sub-organization-class-path
                      object-class-path
                     environment-class-path
                       active-environment-class-path
                     passive-environment-class-path
                     role-class-path
                       agent-role-class-path
                       object-role-class-path
                   instance-path
                    citizen-instance-path
                     agent-instance-path
                     organization-instance-path
                       main-organization-instance-path
                       sub-organization-instance-path
                     object-instance-path
                    environment-instance-path
                     a-environment-instance-path
                   p-environment-instance-path
                    role-instance-path
                     agent-role-instance-path
                     object-role-instance-path
                 sequence
                 seq-element
                )
:roles ( ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
         ;;;;;;;;;; TAO properties
         ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
         (has-goal :range goal)
          (has-belief :range belief)
          (has-axiom  :range axiom)
          (has-duty :domain agent-role-class
                   :range duty)
          (has-right :domain agent-role-class
                    :range right)
        (has-action :range action
                    :parent has-element)
          (has-plan :range plan)
        (has-protocol :range protocol)
          (has-end :domain relationship
                  :range class)
          (has-end1 :parent has-end
                    :inverse is-end1)
        (has-end2 :parent has-end
                    :inverse is-end2)
        (is-end :domain class
                 :range relationship
                  :inverse has-end)
          (is-end1 :domain class
                  :range relationship
                   :inverse has-end1)
          (is-end2 :domain class
                  :range relationship
                   :inverse has-end2)
        (has-msg :range msg
                 :parent has-element)
         (has-msg-end :domain msg
                     :inverse is-msg-end)
        (is-msg-end :inverse has-msg-end)
        (has-msg-sender :parent has-msg-end)
        (has-msg-receiver :parent has-msg-end)
        (is-msg-sender :inverse has-msg-sender)
        (is-msg-receiver :inverse has-msg-receiver)
        (is-instanceOf :domain class-instance
                        :range class)
          (has-attribute :range attribute)
          (has-method :range method)
        ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
         ;;;;;;;;; MAS-ML properties
         ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
         (has-class :domain static-model
                   :range class
                   :inverse is-in-static-model)
        (is-in-static-model :inverse has-class)
        (has-relationship :domain static-model
                          :range relationship
                            :inverse is-relationship-of)
          (is-relationship-of :inverse has-relationship)
          (has-path :range path)
```

```
                 (has-head :domain path
                           :range (or class-instance class))
                  (has-tail :domain path
                           :range path)
                 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                  ;;;;;;;;;; ordering properties
                  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                  (has-structure :range sequence)
                  (has-element :range seq-element)
                  (has-first :domain sequence
                             :range seq-element
                             :parent has-element
                             :inverse is-first)
                  (is-first :inverse has-first)
                  (has-last :domain sequence
                            :range seq-element
                            :parent has-element
                            :inverse is-last)
                  (is-last :inverse has-last)
                  (is-before :domain seq-element
                             :range seq-element
                             :transitive t
                             :inverse is-after)
                  (is-after  :domain seq-element
                             :range seq-element
                             :transitive t
                             :inverse is-before)
                 )
)

;; concepts restrictions

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;; TAO concepts taxonomy
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(implies environment-class class)
(implies role-class class)
(implies citizen-class class)

(implies active-environment-class environment-class)
(implies passive-environment-class environment-class)

(implies agent-role-class role-class)
(implies object-role-class role-class)

(implies agent-class citizen-class)
(implies organization-class citizen-class)
(implies object-class citizen-class)

(implies main-organization-class organization-class)
(implies sub-organization-class organization-class)

(implies inhabit relationship)
(implies play relationship)
(implies ownership relationship)
(implies specialization relationship)
(implies association relationship)
(implies aggregation relationship)
(implies control relationship)
(implies dependency relationship)
(implies play-in relationship)

(implies goal features)
(implies belief features)
(implies action features)
(implies plan features)
(implies duty features)
(implies right features)
(implies protocol features)
(implies axiom features)
(implies attribute features)
(implies method features)

(implies agent class-instance)
(implies a-environment class-instance)
(implies p-environment class-instance)
(implies organization class-instance)
(implies main-organization organization)
(implies sub-organization organization)
(implies object class-instance)
(implies agent-role class-instance)
(implies object-role class-instance)

(implies agent-msg msg)
(implies object-msg msg)

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;; MAS-ML concepts taxonomy (it reuses some TAO concepts)
```

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(implies stereotyped-msg msg)

(implies static-model model)
(implies sequence-model model)

(implies class-model static-model)
(implies organization-model static-model)
(implies role-model static-model)

(implies create-msg stereotyped-msg)
(implies destroy-msg stereotyped-msg)
(implies role-commit-msg stereotyped-msg)
(implies role-cancel-msg stereotyped-msg)
(implies role-activate-msg stereotyped-msg)
(implies role-deactivate-msg stereotyped-msg)
(implies role-change-msg stereotyped-msg)

(equivalent path (or class-path instance-path))

(implies citizen-class-path class-path)
(implies role-class-path class-path)
(implies environment-class-path class-path)

(implies agent-class-path citizen-class-path)
(implies organization-class-path citizen-class-path)
(implies object-class-path citizen-class-path)
(implies agent-role-class-path role-class-path)
(implies object-role-class-path role-class-path)

(implies main-organization-class-path organization-class-path)
(implies sub-organization-class-path organization-class-path)

(implies active-environment-class-path environment-class-path)
(implies passive-environment-class-path environment-class-path)

(implies citizen-instance-path instance-path)
(implies role-instance-path instance-path)
(implies environment-instance-path instance-path)

(implies agent-instance-path citizen-instance-path)
(implies organization-instance-path citizen-instance-path)
(implies object-instance-path citizen-instance-path)

(implies main-organization-instance-path organization-instance-path)
(implies sub-organization-instance-path organization-instance-path)

(implies agent-role-instance-path role-instance-path)
(implies object-role-instance-path role-instance-path)

(implies a-environment-instance-path environment-instance-path)
(implies p-environment-instance-path environment-instance-path)

;;; axiom that guarantee the possibility of the dynamic structure definition
(implies msg seq-element)
(implies action seq-element)

;;;;;;;;;;;;;;;;;;;;;;;;;;
;; ontology axioms
;;;;;;;;;;;;;;;;;;;;;;;;;;;

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;               TAO disjoint restrictions
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(disjoint class class-instance features msg relationship)

(disjoint citizen-class environment-class role-class)

(disjoint agent-class organization-class object-class)

(disjoint main-organization-class sub-organization-class)

(disjoint active-environment-class passive-environment-class)

(disjoint agent-role-class object-role-class)

(disjoint agent a-environment p-environment main-organization
          sub-organization object agent-role object-role)

(disjoint agent-msg object-msg)

(disjoint goal belief protocol attribute)

(disjoint inhabit play ownership specialization association aggregation
          control dependency)

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;             MAS-ML disjoint restrictions
```

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(disjoint static-model sequence-model)

(disjoint class-model organization-model role-model)

(disjoint create-msg destroy-msg role-commit-msg
          role-cancel-msg role-activate-msg role-deactivate-msg
          role-change-msg)

(disjoint class-path instance-path)

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;                                               ;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;      Instances-related axioms                 ;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;                                               ;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(implies agent (all is-instanceOf agent-class))
(implies organization (all is-instanceOf organization-class))
(implies main-organization (all is-instanceOf main-organization-class))
(implies sub-organization (all is-instanceOf sub-organization-class))
(implies object (all is-instanceOf object-class))
(implies a-environment (all is-instanceOf active-environment-class))
(implies p-environment (all is-instanceOf passive-environment-class))
(implies agent-role (all is-instanceOf agent-role-class))
(implies object-role (all is-instanceOf object-role-class))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; path axioms
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

; pathnames are defined recursively

(implies agent-class-path (and (all has-head agent-class)
                               (some has-tail agent-role-class-path)))
(implies object-class-path (and (all has-head object-class)
                                (some has-tail object-role-class-path)))
(implies main-organization-class-path (and (all has-head main-organization-class)
                                           (some has-tail environment-class-path)))
(implies sub-organization-class-path
             (and (all has-head sub-organization-class)
                  (some has-tail (or environment-class-path organization-class-path))))

(implies agent-role-class-path (and (all has-head agent-role-class)
                                    (some has-tail organization-class-path)))
(implies object-role-class-path (and (all has-head object-role-class)
                                     (some has-tail organization-class-path)))

(implies active-environment-class-path (and (all has-head active-environment-class)
                                            (all has-tail *bottom*)))
(implies passive-environment-class-path (and (all has-head passive-environment-class)
                                             (all has-tail *bottom*)))

(implies agent-instance-path (and (all has-head agent)
                                  (some has-tail agent-role-instance-path)))
(implies object-instance-path (and (all has-head object)
                                   (some has-tail object-role-instance-path)))
(implies main-organization-instance-path (and (all has-head main-organization)
                                              (some has-tail environment-instance-path)))
(implies sub-organization-instance-path
        (and (all has-head sub-organization)
             (some has-tail (or organization-instance-path environment-instance-path))))

(implies agent-role-instance-path (and (all has-head agent-role)
                                       (some has-tail organization-instance-path)))
(implies object-role-instance-path (and (all has-head object-role)
                                        (some has-tail organization-instance-path)))
(implies a-environment-instance-path (and (all has-head a-environment)
                                          (all has-tail *bottom*)))
(implies p-environment-instance-path (and (all has-head p-environment)
                                          (all has-tail *bottom*)))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;                                order axioms                                ;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;; semantics of the first element of a sequence
(implies (and (some is-before *top*) (some is-after *bottom*)) (some is-first *top*))

;; semantics of the last element of a sequence
(implies (and (some is-after *top*) (some is-before *bottom*)) (some is-last *top*))

;; as has-first and has-msg are children of has-element, this relation must be declared
;; a first-msg and a last-masg are msgs
(implies (some has-first msg) (some has-msg msg))
(implies (some has-last msg) (some has-msg msg))
```

```
;; the first element is always before the last element
(implies (some is-first *top*) (some is-before (some is-last *top*)))
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;                                          ;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;       Relationships semantics            ;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;                                          ;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;; inhabit ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(implies inhabit (and
                    (some has-end1 citizen-class)
                    (some has-end2 environment-class)
                  )
)
;; end inhabit;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;; play ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;(implies play
;  (or (and (some has-end1 agent-class) (some has-end2 agent-role-class))
;      (and (some has-end1 sub-organization-class) (some has-end2 agent-role-class))
;      (and (some has-end1 object-class) (some has-end2 object-role-class))
;  )
;)
; the semantics was hiden because it is given by the internal state of agents
; sub-organizations and object-roles (see it in 2ndOntology_3.racer)
;; end play ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;; ownership ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;(implies ownership (and
;                      (some has-end1 organization-class)
;                      (some has-end2 role-class)
;                  )
;)
; the semantics was hiden because it is given by the internal state of
; organizations and object-roles (see it in 2ndOntology_3.racer)
;; end ownership ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;; specialization ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(implies specialization
  (or (and (some has-end1 organization-class) (all has-end2 organization-class))
      (and (some has-end1 agent-class) (all has-end2 agent-class))
      (and (some has-end1 object-class) (all has-end2 object-class))
      (and (some has-end1 environment-class) (all has-end2 environment-class))
      (and (some has-end1 agent-role-class) (all has-end2 agent-role-class))
      (and (some has-end1 object-role-class) (all has-end2 object-role-class))
  )
)
;; end specialization ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;; association is commutative ;;;;;;;;;;;;;;;;;;;;;
(implies association
         (or
           (and (some has-end1 object-class) (some has-end2 citizen-class))
            (and (some has-end1 citizen-class) (some has-end1 object-class))
            (and (some has-end1 object-class) (some has-end2 role-class))
           (and (some has-end1 role-class) (some has-end1 object-class))
           (and (some has-end1 role-class) (some has-end2 role-class))
          )
)
;; end association ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;; aggregation ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(implies aggregation
            (or
              (and (some has-end1 object-class) (all has-end2 object-class))
              (and (some has-end1 agent-role-class) (all has-end2 agent-role-class))
              (and (some has-end1 object-role-class) (all has-end2 object-role-class))
            )
)
;; end aggregation ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;; control
(implies control (and
                   (some has-end1 agent-role-class)
                   (all has-end2 agent-role-class)
                 )
)
;; end control ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;; dependency ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(implies dependency
      (or (and (some has-end1 object-class) (all has-end2 object-class))
          (and (some has-end1 agent-role-class) (all has-end2 role-class))
          (and (some has-end1 object-role-class) (all has-end2 object-role-class))
      )
)
;; end dependency ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; play-in is an implicit relationship between agent or sub-organization-classes
;; and organization-classes
;(implies play-in
;      (or (and (some has-end1 agent-class) (all has-end2 organization-class))
;          (and (some has-end1 sub-organization-class) (all has-end2 organization-class))
;      )
;)


;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;                                          ;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;       Static diagrams semantics          ;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;                                          ;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; class-model axioms
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;; a class-model is a diagram composed of citizen classes and environment classes

(implies class-model (all has-class (or citizen-class environment-class)))

;; the allowed relationships are inhabit, association and specialization
;; inhabit is used in class-models just for objects
(implies class-model (some has-relationship (and inhabit (all has-end1 object-class))))

;; association is used in class-models between object classes and citizen classes

(implies class-model (some has-relationship
  (and association
     (or (and (some has-end1(or citizen-class environment-class))
              (some has-end2 object-class))
         (and (some has-end1 object-class)
              (some has-end2 (or citizen-class environment-class)))
      )
  )
))

;; specialization is used between classes of the same type that take part in the diagram.
(implies class-model
         (some has-relationship
           (or  (and specialization (some has-end1 organization-class)
                     (all has-end2 organization-class))
                (and specialization (some has-end1 agent-class) (all has-end2 agent-class))
                (and specialization (some has-end1 object-class) (all has-end2 object-class))
                (and specialization (some has-end1 environment-class)
                     (all has-end2 environment-class))
           )
          )
)

;; aggregation is used between object-classes
(implies class-model
    (some has-relationship (and aggregation (some has-end1 object-class)
                                            (all has-end2 object-class))
         )
)

;; dependency is used between object-classes
(implies class-model
   (some has-relationship
      (and dependency (some has-end1 object-class) (all has-end2 object-class)))
)


;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; end of class-model axioms
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; organization-model axioms
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;; an organization model is a diagram composed of all types of classes defined in the
;; MASML metamodel
(implies organization-model (all has-class class))

;; the allowed relationships in this diagram are ownership, play and inhabit and these
;; relationships haven't any additional constraint to the relationships definition ones

(implies organization-model (all has-relationship (or ownership play inhabit)))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; end of organization-model axioms
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
```

```
;; role-model axioms
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;; a role model is a diagram composed of role classes and object classes
(implies role-model (all has-class (or object-class role-class)))

;; the allowed relationships in this diagram are control, dependency, aggregation,
;; association and specialization

(implies role-model (some has-relationship control))

(implies role-model (some has-relationship
   (or (and dependency (some has-end1 agent-role-class) (all has-end2 role-class))
       (and dependency (some has-end1 object-role-class) (all has-end2 object-role-class)))))


(implies role-model (some has-relationship
   (or (and aggregation (some has-end1 agent-role-class) (all has-end2 agent-role-class))
       (and aggregation (some has-end1 object-role-class) (all has-end2 object-role-class)))))

;; association is used between agent-role classes and role classes,
;; between object-role classes, between object classes and role classes
(implies role-model (some has-relationship
  (and association
     (or (and (some has-end1 role-class)
              (some has-end2 object-class))
         (and (some has-end1 object-class)
              (some has-end2 role-class))
         (and (some has-end1 role-class)
              (some has-end2 role-class))
      )
   )
))

;; specialization is used between agent-role classes and between object-role classes
(implies role-model
    (some has-relationship
      (or (and specialization (some has-end1 agent-role-class) (all has-end2 agent-role-
class))
          (and specialization (some has-end1 object-role-class) (all has-end2 object-role-
class))
      )
    )
)
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; end of role-model axioms
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;
;; Sequence-model structure semantics
;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(implies sequence-model (some has-path path))
(implies sequence-model (some has-msg msg))
(implies sequence-model (some has-structure sequence))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;
;;              semantics of message definition
;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;; agent-msgs must be sent by agents-roles (through protocols), main-organizations or
;; active environments and they cannot be received by objects and object-roles

(implies agent-msg
  (or (and (some has-msg-sender (or agent-role-class-path agent-role-instance-path
                                     main-organization-class-path main-organization-instance-
path
                                     active-environment-class-path a-environment-instance-
path))
            (all has-msg-receiver (or agent-role-class-path agent-role-instance-path
                                      main-organization-class-path main-organization-instance-
path
                                      environment-class-path environment-instance-path)))
   )
)

;; agents, organizations and sub-organizations can create/destroy agents, objects, sub-
organizations
;; and environments that were different from the one they live in.
;; environments can create/destroy citizens and environments
;; objects can create/destroy only objects

(implies create-msg
  (or (and (some has-msg-sender (or main-organization-class-path main-organization-instance-
path
                                    agent-class-path agent-instance-path))
            (all has-msg-receiver (or agent-class-path agent-instance-path
                                     object-class-path object-instance-path
                                     sub-organization-class-path sub-organization-instance-
path
                                     environment-class-path environment-instance-path)))
      (and (some has-msg-sender (or environment-class-path environment-instance-path))
           (all has-msg-receiver (or citizen-class-path citizen-instance-path
                                     environment-class-path environment-instance-path)))
      (and (some has-msg-sender (or object-class-path object-instance-path))
           (all has-msg-receiver (or object-class-path object-instance-path)))
  )
)

(implies destroy-msg
  (or (and (some has-msg-sender (or main-organization-class-path main-organization-instance-
path
                                    agent-class-path agent-instance-path))
           (all has-msg-receiver (or agent-class-path agent-instance-path
                                     object-class-path object-instance-path
                                     organization-class-path organization-instance-path
                                     environment-class-path environment-instance-path)))
      (and (some has-msg-sender (or environment-class-path environment-instance-path))
           (all has-msg-receiver (or citizen-class-path citizen-instance-path
                                     environment-class-path environment-instance-path)))
      (and (some has-msg-sender (or object-class-path object-instance-path))
           (all has-msg-receiver (or object-class-path object-instance-path)))
  )
)

;; agents and sub-organizations can send commit-msg to other agents or sub-organizations
;; in order to play a role, or to objects in order to commit them to an object-role.
;; organizations can send the commit-msg to objects to associate them with object-roles.

(implies role-commit-msg
   (or (and (some has-msg-sender (or agent-class-path agent-instance-path))
            (all has-msg-receiver (or agent-class-path agent-instance-path
                                      object-class-path object-instance-path)))
       (and (some has-msg-sender (or sub-organization-class-path sub-organization-instance-
path))
            (all has-msg-receiver (or sub-organization-class-path sub-organization-instance-
path
                                      object-class-path object-instance-path)))
       (and (some has-msg-sender (or main-organization-class-path main-organization-instance-
path))
            (all has-msg-receiver (or object-class-path object-instance-path)))
   )
)

;; if the sender and the receiver of a role-commit-msg is an agent-path then their heads must
be
;; the same. This means that the same agent instance will play different agent-roles.
;; If the sender is an agent-path and the receiver is an object-path, there isn't constraints.

(implies (and (some has-head agent) (some is-msg-receiver role-commit-msg))
         (and agent-instance-path (some is-msg-sender role-commit-msg)))

(implies (and (some has-head agent-class) (some is-msg-receiver role-commit-msg))
         (and agent-class-path (some is-msg-sender role-commit-msg)))

;; if the sender and the receiver of a role-commit-msg is a sub-organization-path then their
heads must be
;; the same. This means that the same sub-organization instance will play different agent-
roles.
;; If the sender is a sub-organization-path and the receiver is an object-path, there isn't
constraints.

(implies (and (some has-head sub-organization) (some is-msg-receiver role-commit-msg))
         (and sub-organization-instance-path (some is-msg-sender role-commit-msg)))

(implies (and (some has-head sub-organization-class) (some is-msg-receiver role-commit-msg))
         (and sub-organization-class-path (some is-msg-sender role-commit-msg)))


;; the role-cancel msg has the same ends of the role-commit-msg and the same constrains

(implies role-cancel-msg
   (or (and (some has-msg-sender (or agent-class-path agent-instance-path))
            (all has-msg-receiver (or agent-class-path agent-instance-path
                                      object-class-path object-instance-path)))
       (and (some has-msg-sender (or sub-organization-class-path sub-organization-instance-
path))
            (all has-msg-receiver (or sub-organization-class-path sub-organization-instance-
path
                                      object-class-path object-instance-path)))
       (and (some has-msg-sender (or main-organization-class-path main-organization-instance-
path))
            (all has-msg-receiver (or object-class-path object-instance-path)))
   )
)

(implies (and (some has-head agent) (some is-msg-receiver role-cancel-msg))
         (and agent-instance-path (some is-msg-sender role-cancel-msg)))

(implies (and (some has-head agent-class) (some is-msg-receiver role-cancel-msg))
         (and agent-class-path (some is-msg-sender role-cancel-msg)))

(implies (and (some has-head sub-organization) (some is-msg-receiver role-cancel-msg))
         (and sub-organization-instance-path (some is-msg-sender role-cancel-msg)))

(implies (and (some has-head sub-organization-class) (some is-msg-receiver role-cancel-msg))
         (and sub-organization-class-path (some is-msg-sender role-cancel-msg)))

;; agents and sub-organizations can send role-activate-msgs to other agents or sub-
organizations
;; in order to activate an inactive role.

(implies role-activate-msg
   (or (and (some has-msg-sender (or agent-class-path agent-instance-path))
            (all has-msg-receiver (or agent-class-path agent-instance-path
                                      object-class-path object-instance-path)))
       (and (some has-msg-sender (or sub-organization-class-path sub-organization-instance-
path))
            (all has-msg-receiver (or sub-organization-class-path organization-instance-
path)))
   )
)

;; if the sender and the receiver of a role-activate-msg is an agent-path then their heads
must be
;; the same. This means that the same agent instance will activate an agent-role that was
inactive.

(implies (and (some has-head agent) (some is-msg-receiver role-activate-msg))
         (and agent-instance-path (some is-msg-sender role-activate-msg)))

(implies (and (some has-head agent-class) (some is-msg-receiver role-activate-msg))
         (and agent-class-path (some is-msg-sender role-activate-msg)))

(implies (and (some has-head sub-organization) (some is-msg-receiver role-activate-msg))
         (and sub-organization-instance-path (some is-msg-sender role-activate-msg)))

(implies (and (some has-head sub-organization-class) (some is-msg-receiver role-activate-msg))
         (and sub-organization-class-path (some is-msg-sender role-activate-msg)))

;; agents and sub-organizations can send role-deactivate-msgs to other agents or sub-
organizations
;; in order to deactivate an active role.

(implies role-deactivate-msg
   (or (and (some has-msg-sender (or agent-class-path agent-instance-path))
            (all has-msg-receiver (or agent-class-path agent-instance-path
                                      object-class-path object-instance-path)))
       (and (some has-msg-sender (or sub-organization-class-path sub-organization-instance-
path))
            (all has-msg-receiver (or sub-organization-class-path sub-organization-instance-
path)))
   )
)

;; if the sender and the receiver of a role-deactivate-msg is an agent-path then their heads
must be
;; the same. This means that the same agent instance will deactivate an agent-role that was
active.

(implies (and (some has-head agent) (some is-msg-receiver role-deactivate-msg))
         (and agent-instance-path (some is-msg-sender role-deactivate-msg)))

(implies (and (some has-head agent-class) (some is-msg-receiver role-deactivate-msg))
         (and agent-class-path (some is-msg-sender role-deactivate-msg)))

(implies (and (some has-head sub-organization) (some is-msg-receiver role-deactivate-msg))
         (and sub-organization-instance-path (some is-msg-sender role-deactivate-msg)))

(implies (and (some has-head sub-organization-class) (some is-msg-receiver role-deactivate-
msg))
         (and sub-organization-class-path (some is-msg-sender role-deactivate-msg)))

;; agents and sub-organizations can send role-change-msgs to other agents or sub-organizations
;; in order to:
;; cancel an agent-role and commit to another agent-role,
;; cancel an agent-role and activate a deactive role,
;; deactivate an agent-role and activate a deactive role,
;; deactivate an agent-role and commit to a new agent-role.

(implies role-change-msg
   (or (and (some has-msg-sender (or agent-class-path agent-instance-path))
            (all has-msg-receiver (or agent-class-path agent-instance-path
                                      object-class-path object-instance-path)))
       (and (some has-msg-sender (or sub-organization-class-path sub-organization-instance-
path))
            (all has-msg-receiver (or sub-organization-class-path sub-organization-instance-
path)))
   )
)

;; if the sender and the receiver of a role-change-msg is an agent-path then their heads must
be
;; the same. This means that the same agent instance will activate an agent-role that was
inactive.

(implies (and (some has-head agent) (some is-msg-receiver role-change-msg))
         (and agent-instance-path (some is-msg-sender role-change-msg)))

(implies (and (some has-head agent-class) (some is-msg-receiver role-change-msg))
         (and agent-class-path (some is-msg-sender role-change-msg)))

(implies (and (some has-head sub-organization) (some is-msg-receiver role-change-msg))
         (and sub-organization-instance-path (some is-msg-sender role-change-msg)))

(implies (and (some has-head sub-organization-class) (some is-msg-receiver role-change-msg))
         (and sub-organization-class-path (some is-msg-sender role-change-msg)))


;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;
;;                               internal properties
;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;;;;;;;;;;;;;
;;;;;;;;;;;;; agent state' semantics
;;;;;;;;;;;;;

;; an agent must have at least one goal
(implies agent-class (some has-goal goal))

;; an agent has at least one plan
(implies agent-class (some has-plan plan))

;; an agent has at least one action
(implies agent-class (some has-action action))

;; an agent has beliefs
(implies agent-class (some has-belief belief))

;; an agent play at least one agent-role in an organization
(implies agent-class
  (some is-end1 (and play (some has-end2 (and agent-role-class (some is-end2 ownership)))))
)

;; an agent inhabits an environment
(implies agent-class (some is-end1 (and inhabit (all has-end2 environment-class))))

;;;;;;;;;;;;;;
;;;;;;;;;;;;;; organization state' semantics
;;;;;;;;;;;;;;

;; an organization has at least one goal
(implies organization-class (some has-goal goal))

;; an organization has at least one plan
(implies organization-class (some has-plan plan))

;; an organization has at least one action
(implies organization-class (some has-action action))

;; an organization has axioms
(implies organization-class (some has-axiom axiom))

;; an organization has beliefs
(implies organization-class (some has-belief belief))

;;; an organization inhabits an environment
(implies organization-class (some is-end1 (and inhabit (all has-end2 environment-class))))

;;; an organization owns at least one role
(implies  organization-class (some is-end1 (and ownership (some has-end2 role-class))))

;; a sub-organization play at least one agent-role in an organization
(implies sub-organization-class
  (some is-end1 (and play (all has-end2 (and agent-role-class (some is-end2 ownership)))))
)

;; main-organizations have msgs
(implies main-organization-class (some has-msg msg))

;;;;;;;;;;;;;
;;;;;;;;;;;;; agent-role state' semantics
;;;;;;;;;;;;;

;; an agent-role must have at least one goal
(implies agent-role-class (some has-goal goal))

;; an agent-role has beliefs
(implies agent-role-class (some has-belief belief))

;; an agent-role has protocols
(implies agent-role-class (some has-protocol protocol))

;; an agent-role is owned by an organization
;; we don't say which are the ownership end-1 since it was done during the organization-class
;; semantics definition
(implies agent-role-class (some is-end2 ownership))

;; an agent-role is played by someone
(implies agent-role-class (some is-end2 play))

;;;;;;;;;;;;;
;;;;;;;;;;;;; object state' semantics
;;;;;;;;;;;;;

;; an object has attributes
(implies object-class (some has-attribute attribute))

;; an object has methods
(implies object-class (some has-method method))

;; an object plays object-roles
(implies object-class (some is-end1 (and play (all has-end2 object-role-class))))

;;;;;;;;;;;;
;;;;;;;;;;;; object-role state' semantics
;;;;;;;;;;;;

;; an object-role has attributes
(implies object-role-class (some has-attribute attribute))

;; an object-role has methods
(implies object-role-class (some has-method method))

;; an object-role is played by an object
(implies object-role-class (some is-end2 play))

;; an object-role is owned by an organization
(implies object-role-class (some is-end2 (and ownership (all has-end1 organization-class))))

;;;;;;;;;;;;
;;;;;;;;;;;; environment state' semantics
;;;;;;;;;;;;

;; a passive-environment has attributes
(implies passive-environment-class (some has-attribute attribute))

;; a passive-environment has methods
(implies passive-environment-class (some has-method method))

;; an active-environment has goals
(implies active-environment-class (some has-goal goal))

;; an active-environment has plans
(implies active-environment-class (some has-plan plan))

;; an active-environmentagent has actions
(implies active-environment-class (some has-action action))

;; an active-environmentagent has beliefs
(implies active-environment-class (some has-belief belief))

;; an active-environmentagent has msgs
;; this was introduced, since it isn't mentioned in the MAS-ML metamodel
(implies active-environment-class (some has-msg msg))
```

Apêndice D

QV2 para o método aplicado a MAS-ML

Neste apêndice encontram-se as consultas QV2, definidas durante a aplicação do método para a linguagem MAS-ML.

```
;;;;;;;;
;;; interdependências entre diagramas de classe e diagramas
;;; de organizações e de papéis
;;;;;;;;
(RETRIEVE (?CLMD) (?CLMD CLASS-MODEL))

(RETRIEVE (?OBJ-CLASS ?ORG-DIAGRAM ?CLMD)
  (AND (?OBJ-CLASS OBJECT-CLASS) (?ORG-DIAGRAM ORGANIZATION-MODEL)
       (?ORG-DIAGRAM ?OBJ-CLASS HAS-CLASS) (?CLMD CLASS-MODEL)
       (NOT (?CLMD ?OBJ-CLASS HAS-CLASS))))

(RETRIEVE (?OBJ-CLASS ?ROLE-DIAGRAM ?CLMD)
  (AND (?OBJ-CLASS OBJECT-CLASS) (?ROLE-DIAGRAM ROLE-MODEL)
       (?ROLE-DIAGRAM ?OBJ-CLASS HAS-CLASS) (?CLMD CLASS-MODEL)
       (NOT (?CLMD ?OBJ-CLASS HAS-CLASS))))

;;;;;;;;
;;; interdependências entre diagramas de organizações e diagramas
;;; de classes e de papéis
;;;;;;;;
(RETRIEVE (?ORGMD) (?ORGMD ORGANIZATION-MODEL))

(RETRIEVE (?AG-CLASS ?CLASS-DIAGRAM ?ORGMD)
   (AND (?AG-CLASS AGENT-CLASS) (?CLASS-DIAGRAM CLASS-MODEL)
        (?ORGMD ORGANIZATION-MODEL)(?CLASS-DIAGRAM ?AG-CLASS HAS-CLASS)
        (NOT (?ORG-MD ?AG-CLASS HAS-CLASS))))

(RETRIEVE (?ORG-CLASS ?CLASS-DIAGRAM ?ORGMD)
  (AND (?ORG-CLASS ORGANIZATION-CLASS) (?CLASS-DIAGRAM CLASS-MODEL)
       (?ORGMD ORGANIZATION-MODEL)(?CLASS-DIAGRAM ?ORG-CLASS HAS-CLASS)
       (NOT (?ORGMD ?ORG-CLASS HAS-CLASS))))

(RETRIEVE (?RCLASS ?ROLE-DIAGRAM ?ORGMD)
  (AND (?RCLASS ROLE-CLASS) (?ROLE-DIAGRAM ROLE-MODEL)
       (?ROLE-DIAGRAM ?RCLASS HAS-CLASS) (?ORGMD ORGANIZATION-MODEL)
       (NOT (?ORGMD ?RCLASS HAS-CLASS))))

;;;;;;;;
;;; interdependências entre diagramas estáticos e diagramas de
;;; sequências
;;;;;;;;

;;;; classless
(RETRIEVE (?IPATH ?SEQMD ?HEAD)
  (AND (?IPATH INSTANCE-PATH) (?SEQMD SEQUENCE-MODEL) (?SEQMD ?IPATH HAS-PATH)
       (?IPATH ?HEAD HAS-HEAD)(?HEAD NIL IS-INSTANCEOF)))

;;;; caso especial de classless onde quem recebe a msg de criação
;;;; de agente está associado a um papel de agente que é instância
;;;; de uma classe que não está modelada em diagramas estáticos.
(RETRIEVE (?SEQMD ?AGENT-IPATH ?CREATE ?ROLE)
  (AND (?SEQMD SEQUENCE-MODEL)
       (?AGENT-IPATH AGENT-INSTANCE-PATH)
       (?SEQMD ?AGENT-IPATH HAS-PATH)
       (?CREATE CREATE-MSG)
       (?SEQMD ?CREATE HAS-MSG)
       (?AGENT-IPATH ?CREATE IS-MSG-RECEIVER)
       (?AGENT-IPATH ?ROLE-IPATH HAS-TAIL)
       (?ROLE AGENT-ROLE)
       (?ROLE-IPATH ?ROLE HAS-HEAD)
       (?ROLE NIL IS-INSTANCEOF)))
```

Apêndice E
QD1 e QD2 para o método aplicado a MAS-ML

Neste apêndice encontram-se as consultas QD1 e QD2, sobre sugestões de boas práticas de modelagem, definidas durante a aplicação do método para a linguagem MAS-ML.

```
;;;;;;;;;;;;;;;;;;;;;;;;;; QD1
;; Guideline rules

;; Q1 helping the designer to build plans which have the set of
;; allowed actions already assigned

;; return a plan and the candidates to be its 1st actions
(retrieve (?plan ?1st-act)
  (and (?plan plan)
       (?1st-act action)
       (?plan ?1st-act has-action)
       (?1st-act NIL has-precond)
  )
)

;; return a plan and the candidates to be its 1last actions
(retrieve (?plan ?1last-act)
  (and (?plan plan)
       (?1last-act action)
       (?plan ?1last-act has-action)
       (?1last-act ?cond has-poscond)
       (?plan ?cond has-goal)
  )
)

;; return a plan with 3 or more actions and the possible ordering
;; between them. it shows to designer possible structures to the plan
(retrieve (?plan ?act1 ?act2 ?act3)
  (and (?plan plan)
       (?act1 action) (?act2 action) (?act3 action)
       (?plan ?act1 has-action)
       (?plan ?act2 has-action)
       (?plan ?act3 has-action)
       (?act1 ?cond1 has-poscond)
       (?act12 ?cond1 has-precond)
       (?act2 ?cond2 has-poscond)
       (?act3 ?cond2 has-precond)
  )
)

;; Q2 helping the designer to assign suited roles to the agent-classes
;; return a duple (agent, agent-role) which have at least a common goal
(retrieve (?agent ?role)
  (and (?agent agent-class)
       (?role agent-role-class)
       (?agent ?goal has-goal)
       (?role ?goal has-goal)
  )
)

;; Q3 helping the designer to relate agent-role-classes according to
;; the protocols they contain. This query can also be posted in the
;; execution time, considering given agent-role-class and protocol
;; in order to find agent-role classes to relate.
(retrieve (?role-1 ?role-2 ?prtcl)
  (and (?role-1 agent-role-class)
       (?role-2 agent-role-class)
       (?prtcl protocol)
       (?role-1 ?prtcl has-protocol)
       (?role-2 ?prtcl has-protocol)
   )
)
;;;;;;;;;;;;;;;;;;;;;;;;;; QD2
;;;;;;;;
```

```
;;;; consultas referentes à sugestões de boas práticas de
;;;; modelagem usando MAS-ML e considerando diagramas de
;;;; organização e papel
;;;;;;;;
(retrieve (?agrl ?orgmd)
    (and (?agrl agent-role-class)
         (?rlmd role-model)
         (?orgmd organization-model)
         (?orgmd ?agrl has-class)
         (not (?rolemd ?agrl has-class))))

(retrieve (?rolemd) (?rolemd role-model))

;;;;;;;;
;;;; consulta referente a boas práticas de modelagem, envolvendo
;;;; diagrama de papéis e de sequência.

(retrieve (?prtcl-not-modeled)
 (and (?rel (or association dependency aggregation control))
      (?rlmd role-model)
      (?rlmd ?rel has-relationship)
      (?role-class1 agent-role-class)
      (?role-class2 agent-role-class)
      (?rel ?role-class1 has-end1)
      (?rel ?role-class2 has-end2)
      (?prtcl-not-modeled protocol)
      (?role-class1 ?prtcl-not-modeled has-protocol)
      (?role-class2 ?prtcl-not-modeled has-protocol)
      (?struct sequence)
      (?prtcl-not-modeled ?struct has-structure)
      (?seqmd sequence-model)
      (not (?seqmd ?struct has-structure))
 )
)
```

Apêndice F
Resultados do estudo de caso sobre mercados virtuais usando o método aplicado a MAS-ML

Código da base de conhecimento usada na fase F1

Neste apêndice encontra-se o código da base de conhecimento usada no estudo de caso sobre mercados virtuais, usando a aplicação do método para a linguagem MAS-ML. Esta base refere-se à execução da fase F1 do Observed-MAS e possui erros de modelagem, cuja introdução foi deliberada.

```
(in-abox phase1-ABox MASML-TBox)
;; class-model
(instance class-diagram class-model)
;; organization-model
(instance org1-diagram organization-model)
;; role-model
(instance role-diagram role-model)
;; sequence-model
(instance seq-model sequence-model)
(instance seq-model-struct sequence)
(instance seq-model-struct-2 sequence)
;; virtual-market is an environment-class
(instance virtual-market passive-environment-class)
(instance supermarket passive-environment-class)
;; general-store, imported-bookstore, 2nd-hand-bookstore are organization-classes
(instance general-store main-organization-class)
(instance imported-bookstore sub-organization-class)
(instance 2nd-hand-bookstore sub-organization-class)
;; user-agent and store-agent are agent-classes
(instance user-agent agent-class)
(instance store-agent agent-class)
(instance super-user agent-class)
;; item and book are object-classes
(instance item object-class)
(instance book object-class)
(instance magazine object-class)
;; has-plan (GeneralQueries-Q33-OK)
(instance plan-1 plan)
(related user-agent plan-1 has-plan)
;; has-structure (GeneralQueries-Q38-OK)
(related plan-1 seq-model-struct-2 has-structure)
;; has-action (GeneralQueries-Q34-OK)
(instance action-1 action)
(related user-agent action-1 has-action)
;; ERROR has-plan (GeneralQueries-Q33-OK)
;; ERROR has-structure (GeneralQueries-Q38-OK)
(instance plan-2 plan)
;; ERROR has-plan (GeneralQueries-Q33-OK)
;; ERROR has-structure (GeneralQueries-Q38-OK)
(instance plan-3 plan)
(related book plan-1 has-plan)
;; ERROR has-action (GeneralQueries-Q34-OK)
(instance action-2 action)
;; ERROR has-action (GeneralQueries-Q34-OK)
(instance action-3 action)
(related book action-1 has-action)
;; roles
(instance seller agent-role-class)
(instance buyer agent-role-class)
(instance offer object-role-class)
(instance desire object-role-class)
(instance mediator agent-role-class)
;; agents
```

```
(instance mary agent)
(related mary user-agent is-instanceOf)
(instance mary-path agent-instance-path)
(related mary-path mary has-head)
(instance user-agent-path agent-class-path)
(related user-agent-path  user-agent has-head)
(instance john agent)
(related john user-agent is-instanceOf)
(instance john-path agent-instance-path)
(related john john-path has-path)
(instance vendor-americanas agent)
(related vendor-americanas store-agent is-instanceOf)
(instance vendor-americanas-path agent-instance-path)
(related vendor-americanas vendor-americanas-path has-path)
(related store-agent store-agent-path has-path)
;; defining a messages for an agent-role and inserting it into a protocol (GeneralQueries-Q31-
OK)
(instance agent-msg-1 agent-msg)
(related buyer agent-msg-1 has-msg)
(instance agent-msg-2 agent-msg)
(related buyer agent-msg-2 has-msg)
(instance agent-msg-3 agent-msg)
(related buyer agent-msg-3 has-msg)
(instance protocol-1 protocol)
(related buy protocol-1 has-protocol)
(related protocol-1 agent-msg-1 has-msg)
(related protocol-1 agent-msg-2 has-msg)
(related protocol-1 agent-msg-3 has-msg)

;; protocol has-structure (GeneralQueries-Q37-OK)
(related protocol-1 seq-model-struct has-structure)
;; sequence has-first (GeneralQueries-Q39-OK)
(related seq-model-struct agent-msg-3 has-first)
;; sequence has-last (GeneralQueries-Q39-OK)
(related seq-model-struct agent-msg-1 has-last)
;; sequence has-element
(related seq-model-struct agent-msg-2 has-element)
;; sequence is-before (GeneralQueries-Q39-OK)
(related agent-msg-1 agent-msg-2 is-before)
(related agent-msg-2 agent-msg-3 is-before)

;; ERROR sequence has-first (GeneralQueries-Q39)
;; ERROR sequence has-last (GeneralQueries-Q39)
 (instance protocol-2 protocol)
(related buy protocol-2 has-protocol)
(related protocol-2 seq-model-struct-2 has-structure)
;(related protocol-2 agent-msg-1 has-first)
;(related protocol-2 agent-msg-2 has-last)
;; ERROR message is not in a protocol (GeneralQueries-Q31-OK)
(instance agent-msg-4 agent-msg)
(related buyer agent-msg-4 has-msg)
;; ERROR protocol has-structure (GeneralQueries-Q37-OK)
(instance protocol-3 protocol)
(related seller protocol-3 has-protocol)

;; organization
(instance americanas-dotcom main-organization)
(instance americanas-dotcom-path main-organization-instance-path)
(related americanas-dotcom general-store is-instanceOf)
(related americanas-dotcom americanas-dotcom-path has-path)
(related general-store general-store-path has-path)
(instance intl-bookstore-dotcom sub-organization)
(instance intl-bookstore-dotcom-path sub-organization-instance-path)
(instance imported-bookstore-path sub-organization-class-path)
(related intl-bookstore-dotcom imported-bookstore is-instanceOf)
(related intl-bookstore-dotcom intl-bookstore-dotcom-path has-path)
(related imported-bookstore imported-bookstore-path has-path)
;; environment
(instance webshopping p-environment)
(instance webshopping-path p-environment-instance-path)
(instance virtual-market-path passive-environment-class-path)
(related webshopping virtual-market is-instanceOf)
(related webshopping webshopping-path has-path)
(related virtual-market virtual-market-path has-path)
;; objects
(instance sommerville2001 object)
(instance sommerville2001-path object-instance-path)
(instance book-path object-class-path)
(related sommerville2001 book is-instanceOf)
(related sommerville2001 sommerville2001-path has-path)
(related book book-path has-path)
;(instance comm-acm object)
(related comm-acm magazine is-instanceOf)
(instance comm-acm-path object-instance-path)
(instance magazine-path object-class-path)
(related comm-acm comm-acm-path has-path)
(related magazine magazine-path has-path)
;; roles
```

```
(instance americanas-seller agent-role)
(related americanas-seller seller is-instanceOf)
(instance americanas-seller-path agent-role-instance-path)
(instance seller-path agent-role-class-path)
(related americanas-seller americanas-seller-path has-path)
(related seller seller-path has-path)
(instance americanas-buyer agent-role)
(related americanas-buyer buyer is-instanceOf)
(instance americanas-buyer-path agent-role-instance-path)

;; >>>ERROR (SequenceModelQueriesFinal.racer Q3
;; >>> including the head of buyer-path
;; putting agent-msg-6 in a buyer protocol and using a msg from the seller protocol
;; as a sending msg from buyer, alternatively
(instance buyer-path agent-role-class-path)
(related buyer-path buyer has-head)
(instance prtcl-0 protocol)
(related buyer prtcl-0 has-protocol)
;(related prtcl-0 agent-msg-6 has-msg)
(instance ag-msg-7 agent-msg)
(related buyer-path ag-msg-7 is-msg-sender)
(related protocol-3 ag-msg-7 has-msg)
(related seq-model ag-msg-7 has-msg)

(related americanas-buyer americanas-buyer-path has-path)
(related buyer buyer-path has-path)
(instance intl-book-seller agent-role)
(instance intl-book-seller-path agent-role-instance-path)
(related  intl-book-seller seller is-instanceOf)
(instance intl-book-buyer agent-role)
(related intl-book-buyer buyer is-instanceOf)
(instance intl-book-buyer-path agent-role-instance-path)
(instance special-offer object-role)
(related  special-offer offer is-instanceOf)
(instance special-offer-path object-role-instance-path)
(instance offer-path object-role-class-path)
(related special-offer special-offer-path has-path)
(related offer offer-path has-path)
;; messages
(instance create-env-1 create-msg)
(related create-env-1 webshopping-path has-msg-sender)
(related create-env-1 webshopping-path has-msg-receiver)
(instance create-agent-1 create-msg)
(related create-agent-1 webshopping-path has-msg-sender)
(related create-agent-1 mary-path has-msg-receiver)
(instance create-agent-2 create-msg)
(related create-agent-2 americanas-buyer-path has-msg-sender)
(related create-agent-2 john-path has-msg-receiver)
(instance create-org-1 create-msg)
(related create-org-1 mary-path has-msg-sender)
(related create-org-1 americanas-dotcom-path has-msg-receiver)
(instance destroy-org-1 destroy-msg)
(related destroy-org-1 john-path has-msg-sender)
(related destroy-org-1 americanas-buyer-path has-msg-receiver)
(instance commit-role-1 role-commit-msg)
(related commit-role-1 john-path has-msg-sender)
(related commit-role-1  offer-path has-msg-receiver)
(instance commit-role-2 role-commit-msg)
(related commit-role-2 user-agent-path has-msg-sender)
(related commit-role-2 seller-path has-msg-receiver)
;; ERROR role-commit-msg (GeneralQueries-Q24)
 (instance commit-role-3 role-commit-msg)
(related commit-role-3 book-path has-msg-sender)
(related commit-role-3 user-agent-path has-msg-receiver)
(instance commit-ob-role-1 role-commit-msg)
(related commit-ob-role-1 intl-book-seller has-msg-sender)
(related commit-ob-role-1 special-offer-path has-msg-receiver)
(instance cancel-role-1 role-cancel-msg)
(related cancel-role-1 intl-book-seller has-msg-sender)
(related cancel-role-1 special-offer-path has-msg-receiver)
;; (GeneralQueries-Q25-OK)
(instance cancel-ob-role-2 role-cancel-msg)
(related cancel-ob-role-2 john-path has-msg-sender)
(related cancel-ob-role-2 special-offer-path has-msg-receiver)
;; ERROR cancel-ob-role (GeneralQueries-Q25-OK)
(instance cancel-ob-role-3 role-cancel-msg)
(related cancel-ob-role-3 special-offer-path has-msg-sender)
(related cancel-ob-role-3 john-path has-msg-receiver)
;; role-activate-msg (GeneralQueries-Q26-OK)
(instance activate-role-1 role-activate-msg)
(related activate-role-1 john-path has-msg-sender)
(related activate-role-1 john-path has-msg-receiver)
;; ERROR role-activate-msg (GeneralQueries-Q26-OK)
(instance activate-role-2 role-activate-msg)
(related activate-role-2 john-path has-msg-sender)
(related activate-role-2 special-offer-path has-msg-receiver)
;; ERROR role-activate-msg (GeneralQueries-Q26-OK)
(instance activate-role-3 role-activate-msg)
```

```
(related activate-role-3 john-path has-msg-sender)
(related activate-role-3 mary-path has-msg-receiver)
;; role-deactivate-msg (GeneralQueries-Q27-OK)
(instance deactivate-role-1 role-deactivate-msg)
(related deactivate-role-1 john-path has-msg-sender)
(related deactivate-role-1 john-path has-msg-receiver)
;; ERROR role-deactivate-msg (GeneralQueries-Q27-OK)
(instance deactivate-role-2 role-deactivate-msg)
(related deactivate-role-2 john-path has-msg-sender)
(related deactivate-role-2 special-offer-path has-msg-receiver)
(instance change-role-1 role-change-msg)
(related change-role-1 john-path has-msg-sender)
(related change-role-1 special-offer-path has-msg-receiver)
;; object-msg (GeneralQueries-Q20)
 (instance getTitle object-msg)
(related getTitle user-agent-path has-msg-sender)
(related getTitle book-path has-msg-receiver)
;; ERROR object-msg (GeneralQueries-Q20)
;; Não está dando erro em getAuthor mas está errado
;; Agent não pode ser object-msg-receiver
(instance getAuthor object-msg)
(related getAuthor user-agent-path has-msg-sender)
(related getAuthor user-agent-path has-msg-receiver)
;; ERROR object-msg (GeneralQueries-Q20)
;; Não está dando erro mas está errado
(instance getAbstract object-msg)
(related getAbstract book-path has-msg-sender)
(related getAbstract book-path has-msg-receiver)
;; agent-msg (GeneralQueries-Q21)
 (instance inform agent-msg)
(related inform user-agent-path has-msg-sender)
(related inform store-agent-path has-msg-receiver)
;; ERROR agent-msg (GeneralQueries-Q21-OK)
(instance proposal agent-msg)
(related proposal user-agent-path has-msg-sender)
(related proposal book-path has-msg-receiver)
;; ERROR agent-msg (GeneralQueries-Q21-OK)
(instance answerProposal agent-msg)
(related answerProposal book-path has-msg-sender)
(related answerProposal user-agent-path has-msg-receiver)
;; inhabit
(instance inhabit-1 inhabit)
(related inhabit-1 user-agent has-end1)
(related inhabit-1 virtual-market has-end2)
(instance inhabit-2 inhabit)
(related inhabit-2 book has-end1)
(related inhabit-2 item has-end2)
(instance inhabit-3 inhabit)
(related inhabit-3 seller has-end1)
(related inhabit-3 virtual-market has-end2)
;; specialization
(instance spec-1 specialization)
(related spec-1 book has-end1)
(related spec-1 item has-end2)
(instance spec-2 specialization)
(related spec-2 user-agent has-end1)
(related spec-2 super-user has-end2)
;; (ClassModelQuerires-Q7-OK)
(instance spec-3 specialization)
(related spec-3 buyer has-end1)
(related spec-3 seller has-end2)
;; ERROR specialization (GeneralModel-Q5-OK)
(instance spec-3 specialization)
(related spec-3 user-agent has-end1)
(related spec-3 imported-bookstore has-end2)
;; aggregation
(instance aggr-1 aggregation)
(related aggr-1 imported-bookstore has-end1)
(related aggr-1 2nd-hand-bookstore has-end2)
;; erro na (ClassModelQueries-Q8-OK)
(instance aggr-2 aggregation)
(related aggr-2 imported-bookstore has-end1)
(related aggr-2 2nd-hand-bookstore has-end2)
;; ownership
(instance own-1 ownership)
(related own-1 general-store has-end1)
(related own-1 buyer has-end2)
(instance own-2 ownership)
(related own-2 imported-bookstore has-end1)
(related own-2 desire has-end2)
;; ERROR (GeneralQueries-Q42-OK)
;; O papel seller está sendo desempenhado pelo agente store-agent mas não está definido em
nenhuma organização
;; (instance own-1 ownership)
;; (related own-1 general-store has-end1)
;; (related own-1 seller has-end2)
;; (GeneralQueries-Q42-OK)
(instance play-5 play)
```

```
(related play-5 store-agent has-end1)
(related play-5 seller has-end2)
;; ERROR ownership (GeneralQueries-Q4)
 (instance own-3 ownership)
(related own-3 imported-bookstore has-end1)
(related own-3 imported-bookstore has-end2)
;; play
(instance play-1 play)
(related play-1 user-agent has-end1)
(related play-1 buyer has-end2)
(instance play-2 play)
(related play-2 imported-bookstore has-end1)
(related play-2 seller has-end2)
;; ERROR play (GeneralQueries-Q3)
 (instance play-3 play)
(related play-3 user-agent has-end1)
(related play-3 user-agent has-end2)
;; ERROR (GeneralQueries-Q15-OK)
;; ERROR (GeneralQueries-Q3-OK)
(instance play-4 play)
(related play-4 general-store has-end1)
(related play-4 buyer has-end2)
;; control
(instance ctrl-1 control)
(related ctrl-1 buyer has-end1)
(related ctrl-1 buyer has-end2)
(instance ctrl-2 control)
(related ctrl-2 buyer has-end1)
(related ctrl-2 desire has-end2)
(instance ctrl-3 control)
(related ctrl-3 mediator has-end1)
(related ctrl-3 buyer has-end2)
;; dependency
(instance dep-1 dependency)
(related dep-1 book has-end1)
(related dep-1 item has-end2)
(instance dep-2 dependency)
(related dep-2 offer has-end1)
(related dep-2 desire has-end2)
;; ERROR dependency (GeneralQueries-Q8-OK)
(instance dep-3 dependency)
(related dep-3 offer has-end1)
(related dep-3 book has-end2)
;; objetivo é introduzir erro na (ClassModellQueries-Q9-OK)
(instance dep-4 dependency)
(related dep-4 offer has-end1)
(related dep-4 book has-end2)
;; association
(instance assoc-1 association)
(related assoc-1 user-agent has-end1)
(related assoc-1 general-store has-end2)
(instance assoc-2 association)
(related assoc-2 book has-end1)
(related assoc-2 magazine has-end2)
(instance assoc-3 association)
(related assoc-3 book has-end1)
(related assoc-3 desire has-end2)
(instance assoc-4 association)
(related assoc-4 supermarket has-end1)
(related assoc-4 virtual-market has-end2)
;; CLASS-MODEL-RELATIONSHIPS
(related class-diagram item has-class)
(related class-diagram book has-class)
(related class-diagram user-agent has-class)
(related class-diagram store-agent has-class)
(related class-diagram general-store has-class)
(related class-diagram imported-bookstore has-class)
(related class-diagram 2nd-hand-bookstore has-class)
(related class-diagram virtual-market has-class)
(related class-diagram seller has-class)
(related class-diagram mediator has-class)
(related class-diagram buyer has-class)
(related class-diagram super-user has-class)
(related class-diagram inhabit-1 has-relationship)
(related class-diagram inhabit-2 has-relationship)
(related class-diagram inhabit-3 has-relationship)
(related class-diagram spec-1 has-relationship)
(related class-diagram play-1 has-relationship)
(related class-diagram assoc-1 has-relationship)
(related class-diagram ctrl-3 has-relationship)
(related class-diagram spec-2 has-relationship)
;; ERROR (ClassModelQueries-Q7-OK)
(related class-diagram spec-3 has-relationship)
;; ERROR (ClassModelQueries-Q8)
(related class-diagram aggr-2 has-relationship)
;; ERROR (ClassModelQueries-Q8)
(related class-diagram dep-4 has-relationship)
;; ORGANIZATION-MODEL-RELATIONSHIPS
```

```
(related org1-diagram inhabit-1 has-relationship)
(related org1-diagram play-1 has-relationship)
(related org1-diagram play-2 has-relationship)
(related org1-diagram own-2 has-relationship)
;; ERROR (OrgModelQueries-Q1-OK)
;; ERROR (OrgModelQueries-Q7-OK)
(related org1-diagram spec-2 has-relationship)
;; ERROR (OrgModelQueries-Q2-OK)
(related org1-diagram assoc-2 has-relationship)
;; ERROR (OrgModelQueries-Q8-OK)
(related org1-diagram assoc-4 has-relationship)
;; ERROR (OrgModelQueries-Q3-OK)
(related org1-diagram aggr-1 has-relationship)
;; ERROR (OrgModelQueries-Q4-OK)
;; ERROR (OrgModelQueries-Q6-OK)
;; ERROR (OrgModelQueries-Q9-OK)
(related org1-diagram ctrl-3 has-relationship)
;; ERROR (OrgModelQueries-Q5-OK)
;; ERROR (OrgModelQueries-Q9-OK)
(related org1-diagram dep-1 has-relationship)
;; ORGANIZATION-MODEL-CLASS
(related org1-diagram item has-class)
(related org1-diagram book has-class)
(related org1-diagram super-user has-class)
(related org1-diagram seller has-class)
(related org1-diagram mediator has-class)
(related org1-diagram buyer has-class)
(related org1-diagram general-store has-class)
(related org1-diagram user-agent has-class)
(related org1-diagram offer has-class)
(related org1-diagram imported-bookstore has-class)
(related org1-diagram 2nd-hand-bookstore has-class)
(related org1-diagram virtual-market has-class)
(related org1-diagram supermarket has-class)
;; ROLE-MODEL-RELATIONSHIPS
(related role-diagram play-1 has-relationship)
(related role-diagram assoc-1 has-relationship)
(related role-diagram ctrl-1 has-relationship)
(related role-diagram aggr-1 has-relationship)
(related role-diagram inhabit-2 has-relationship)
(related role-diagram spec-1 has-relationship)
;; ERROR (RoleModelQueries-Q7-OK)
(related role-diagram dep-1 has-relationship)
;; ROLE-MODEL-CLASS
(related role-diagram book has-class)
(related role-diagram item has-class)
(related role-diagram offer has-class)
(related role-diagram desire has-class)
(related role-diagram seller has-class)
(related role-diagram user-agent has-class)
(related role-diagram general-store has-class)
(related role-diagram imported-bookstore has-class)
(related role-diagram 2nd-hand-bookstore has-class)
;; ERROR (RoleModelQueries-Q3-OK)
(related role-diagram virtual-market has-class)
;; SEQUENCE-MODEL-PATH
(related seq-model offer-path has-path)
(related seq-model buyer-path has-path)
(related seq-model americanas-buyer-path has-path)
(related seq-model book-path has-path)
(related seq-model mary-path has-path)
(related seq-model john-path has-path)
(related seq-model webshopping-path has-path)
(related seq-model general-store-path has-path)
(related seq-model user-agent-path has-path)
;; SEQUENCE-MODEL-MESSAGE
(related seq-model create-env-1 has-msg)
(related seq-model create-agent-1 has-msg)
(related seq-model create-agent-2 has-msg)
(related seq-model create-org-1 has-msg)
(related seq-model destroy-org-1 has-msg)
(related seq-model commit-role-1 has-msg)
(related seq-model commit-role-2 has-msg)
(related seq-model commit-ob-role-1 has-msg)
(related seq-model cancel-ob-role-1 has-msg)

;; ERROR (SequenceModelQueries-Q2 e Q3-OK)
(instance agent-msg-6 agent-msg)
(related seq-model agent-msg-6 has-msg)
(related buyer-path agent-msg-6 is-msg-sender)
(related seq-model buyer-path has-path)
;(related general-store agent-msg-6 has-msg)

(related protocol-1 seq-model-struct has-structure)
;; ERROR (SequenceModelQueries-Q7-OK)
(related seq-model-struct agent-msg-1 has-first)
(related seq-model-struct-2 agent-msg-2 has-last)
```

```
(related agent-msg-2 agent-msg-1 is-before)
```

Respostas às consultas QV1

Respostas obtidas a partir das consultas QV1 aplicadas à base de conhecimento descrita na seção anterior.

```
(IN-KNOWLEDGE-BASE MASML-TBOX PHASE1-ABOX) --> (MASML-TBOX PHASE1-ABOX)
(IN-TBOX MASML-TBOX) --> MASML-TBOX
(TBOX-CLASSIFIED-P) --> T
(TBOX-COHERENT-P) --> T
(TBOX-CYCLIC-P) --> NIL
(IN-ABOX PHASE1-ABOX MASML-TBOX) --> PHASE1-ABOX
(CHECK-ABOX-COHERENCE) --> T
(RETRIEVE (?RELATION) (AND (?RELATION RELATIONSHIP) (NIL ?RELATION HAS-RELATIONSHIP))) -->
(((?RELATION ASSOC-4)) ((?RELATION ASSOC-3)) ((?RELATION DEP-3)) ((?RELATION DEP-2))
((?RELATION DEP-1)) ((?RELATION CTRL-2)) ((?RELATION PLAY-4)) ((?RELATION PLAY-3)) ((?RELATION
OWN-3)) ((?RELATION PLAY-5)) ((?RELATION OWN-1)) ((?RELATION SPEC-3)))
(RETRIEVE (?END1 ?END2 ?INH) (AND (?INH INHABIT) (?END1 ?INH IS-END1) (?END2 ?INH IS-END2) (OR
(?END1 (OR ENVIRONMENT-CLASS ROLE-CLASS)) (?END2 (OR CITIZEN-CLASS ROLE-CLASS))))) -->
(((?END1 BOOK) (?END2 ITEM) (?INH INHABIT-2)) ((?END1 SELLER) (?END2 VIRTUAL-MARKET) (?INH
INHABIT-3)))
(RETRIEVE ($?END1 $?END2 ?PLAY) (AND (?PLAY PLAY) ($?END1 ?PLAY IS-END1) ($?END2 ?PLAY IS-
END2) (NOT (AND ($?END1 (OR AGENT-CLASS SUB-ORGANIZATION-CLASS)) ($?END2 AGENT-ROLE-CLASS)))
(NOT (AND ($?END1 OBJECT-CLASS) ($?END2 OBJECT-ROLE-CLASS))))) --> ((($?END1 USER-AGENT)
($?END2 USER-AGENT) (?PLAY PLAY-3)) (($?END1 GENERAL-STORE) ($?END2 BUYER) (?PLAY PLAY-4)))
(RETRIEVE ($?END1 $?END2 ?OWN) (AND (?OWN OWNERSHIP) ($?END1 ?OWN IS-END1) ($?END2 ?OWN IS-
END2) (OR ($?END1 (OR ROLE-CLASS ENVIRONMENT-CLASS AGENT-CLASS OBJECT-CLASS)) ($?END2 (OR
CITIZEN-CLASS ENVIRONMENT-CLASS))))) --> ((($?END1 IMPORTED-BOOKSTORE) ($?END2 IMPORTED-
BOOKSTORE) (?OWN OWN-3)))
(RETRIEVE (?END1 ?END2 ?SPEC) (AND (?SPEC SPECIALIZATION) (?END1 ?SPEC IS-END1) (?END2 ?SPEC
IS-END2) (OR (AND (?END1 OBJECT-CLASS) (?END2 (OR ORGANIZATION-CLASS AGENT-CLASS ENVIRONMENT-
CLASS ROLE-CLASS))) (AND (?END1 AGENT-CLASS) (?END2 (OR ORGANIZATION-CLASS OBJECT-CLASS
ENVIRONMENT-CLASS ROLE-CLASS))) (AND (?END1 MAIN-ORGANIZATION-CLASS) (?END2 (OR AGENT-CLASS
OBJECT-CLASS SUB-ORGANIZATION-CLASS ENVIRONMENT-CLASS ROLE-CLASS))) (AND (?END1 SUB-
ORGANIZATION-CLASS) (?END2 (OR AGENT-CLASS OBJECT-CLASS MAIN-ORGANIZATION-CLASS ENVIRONMENT-
CLASS ROLE-CLASS))) (AND (?END1 PASSIVE-ENVIRONMENT-CLASS) (?END2 (OR AGENT-CLASS
ORGANIZATION-CLASS OBJECT-CLASS ACTIVE-ENVIRONMENT-CLASS ROLE-CLASS))) (AND (?END1 AGENT-ROLE-
CLASS) (?END2 (OR CITIZEN-CLASS ENVIRONMENT-CLASS OBJECT-ROLE-CLASS))) (AND (?END1 OBJECT-
ROLE-CLASS) (?END2 (OR CITIZEN-CLASS ENVIRONMENT-CLASS AGENT-ROLE-CLASS)))))) --> (((?END1
USER-AGENT) (?END2 IMPORTED-BOOKSTORE) (?SPEC SPEC-3)))
(RETRIEVE (?END1 ?END2 ?AGG) (AND (?AGG AGGREGATION) (?END1 ?AGG IS-END1) (?END2 ?AGG IS-END2)
(OR (AND (?END1 OBJECT-CLASS) (?END2 (OR ORGANIZATION-CLASS AGENT-CLASS ENVIRONMENT-CLASS
ROLE-CLASS))) (AND (?END1 AGENT-ROLE-CLASS) (?END2 (OR CITIZEN-CLASS ENVIRONMENT-CLASS OBJECT-
ROLE-CLASS))) (AND (?END1 OBJECT-ROLE-CLASS) (?END2 (OR CITIZEN-CLASS ENVIRONMENT-CLASS AGENT-
ROLE-CLASS))) (?END1 (OR AGENT-CLASS ORGANIZATION-CLASS ENVIRONMENT-CLASS))))) --> (((?END1
IMPORTED-BOOKSTORE) (?END2 2ND-HAND-BOOKSTORE) (?AGG AGGR-1)))
(RETRIEVE (?END1 ?END2 ?CTRL) (AND (?CTRL CONTROL) (?END1 ?CTRL IS-END1) (?END2 ?CTRL IS-END2)
(OR (?END1 (OR CITIZEN-CLASS ENVIRONMENT-CLASS OBJECT-ROLE-CLASS)) (AND (?END1 AGENT-ROLE-
CLASS) (?END2 (OR CITIZEN-CLASS ENVIRONMENT-CLASS OBJECT-ROLE-CLASS)))))) --> (((?END1 BUYER)
(?END2 DESIRE) (?CTRL CTRL-2)))
(RETRIEVE (?END1 ?END2 ?DEP) (AND (?DEP DEPENDENCY) (?END1 ?DEP IS-END1) (?END2 ?DEP IS-END2)
(OR (?END1 (OR AGENT-CLASS ORGANIZATION-CLASS ENVIRONMENT-CLASS)) (AND (?END1 OBJECT-CLASS)
(?END2 (OR ORGANIZATION-CLASS AGENT-CLASS ENVIRONMENT-CLASS ROLE-CLASS))) (AND (?END1 AGENT-
ROLE-CLASS) (?END2 (OR CITIZEN-CLASS ENVIRONMENT-CLASS))) (AND (?END1 OBJECT-ROLE-CLASS)
(?END2 (OR CITIZEN-CLASS ENVIRONMENT-CLASS AGENT-ROLE-CLASS)))))) --> (((?END1 OFFER) (?END2
BOOK) (?DEP DEP-3)))
(RETRIEVE (?CLASS-ALONE) (AND (?CLASS-ALONE CLASS) (NIL ?CLASS-ALONE HAS-CLASS))) -->
(((?CLASS-ALONE MAGAZINE)))
(RETRIEVE (?AGENTWITHOUTGOAL) (AND (?AGENTWITHOUTGOAL AGENT-CLASS) (?AGENTWITHOUTGOAL NIL HAS-
GOAL))) --> (((?AGENTWITHOUTGOAL SUPER-USER)) ((?AGENTWITHOUTGOAL STORE-AGENT))
((?AGENTWITHOUTGOAL USER-AGENT)))
(RETRIEVE (?ORGWITHOUTGOAL) (AND (?ORGWITHOUTGOAL ORGANIZATION-CLASS) (?ORGWITHOUTGOAL NIL
HAS-GOAL))) --> (((?ORGWITHOUTGOAL 2ND-HAND-BOOKSTORE)) ((?ORGWITHOUTGOAL IMPORTED-BOOKSTORE))
((?ORGWITHOUTGOAL GENERAL-STORE)))
(RETRIEVE (?ROLEWITHOUTGOAL) (AND (?ROLEWITHOUTGOAL AGENT-ROLE-CLASS) (?ROLEWITHOUTGOAL NIL
HAS-GOAL))) --> (((?ROLEWITHOUTGOAL MEDIATOR)) ((?ROLEWITHOUTGOAL BUYER)) ((?ROLEWITHOUTGOAL
SELLER)))
(RETRIEVE (?NO-CITIZEN) (AND (?NO-CITIZEN CITIZEN-CLASS) (NOT (?NO-CITIZEN (SOME IS-END1 (AND
INHABIT (SOME HAS-END2 ENVIRONMENT-CLASS))))))) --> (((?NO-CITIZEN BOOK)) ((?NO-CITIZEN
MAGAZINE)) ((?NO-CITIZEN ITEM)) ((?NO-CITIZEN STORE-AGENT)) ((?NO-CITIZEN SUPER-USER)) ((?NO-
CITIZEN IMPORTED-BOOKSTORE)) ((?NO-CITIZEN 2ND-HAND-BOOKSTORE)) ((?NO-CITIZEN GENERAL-STORE)))
(RETRIEVE (?NO-HABITAT) (AND (?NO-HABITAT ENVIRONMENT-CLASS) (NOT (?NO-HABITAT (SOME IS-END2
(AND INHABIT (SOME HAS-END1 CITIZEN-CLASS))))))) --> (((?NO-HABITAT SUPERMARKET)))
(RETRIEVE (?AGENTWITHOUTROLE) (AND (?AGENTWITHOUTROLE AGENT-CLASS) (NOT (?AGENTWITHOUTROLE
(SOME IS-END1 (AND PLAY (SOME HAS-END2 AGENT-ROLE-CLASS))))))) --> (((?AGENTWITHOUTROLE SUPER-
USER)))
```

```
(RETRIEVE (?ORG-NO-OWNER) (AND (?ORG-NO-OWNER ORGANIZATION-CLASS) (NOT (?ORG-NO-OWNER (SOME
IS-END1 (AND OWNERSHIP (SOME HAS-END2 AGENT-ROLE-CLASS))))))) --> (((?ORG-NO-OWNER IMPORTED-
BOOKSTORE)) ((?ORG-NO-OWNER 2ND-HAND-BOOKSTORE)))
(RETRIEVE (?ORG ?PLAY) (AND (?ORG MAIN-ORGANIZATION-CLASS) (?PLAY (AND PLAY (SOME HAS-END2
AGENT-ROLE-CLASS))) (?ORG ?PLAY IS-END1))) --> (((?ORG GENERAL-STORE) (?PLAY PLAY-4)))
(RETRIEVE (?ORG-NO-PLAY) (AND (?ORG-NO-PLAY SUB-ORGANIZATION-CLASS) (NOT (?ORG-NO-PLAY (SOME
IS-END1 PLAY))))) --> (((?ORG-NO-PLAY 2ND-HAND-BOOKSTORE)))
(RETRIEVE (?ORG) (AND (?ORG ORGANIZATION-CLASS) (?ORG NIL HAS-AXIOM))) --> (((?ORG 2ND-HAND-
BOOKSTORE)) ((?ORG IMPORTED-BOOKSTORE)) ((?ORG GENERAL-STORE)))
(RETRIEVE (?ROLE-NO-OWNED) (AND (?ROLE-NO-OWNED ROLE-CLASS) (NOT (?ROLE-NO-OWNED (SOME IS-END2
OWNERSHIP))))) --> (((?ROLE-NO-OWNED OFFER)) ((?ROLE-NO-OWNED MEDIATOR)) ((?ROLE-NO-OWNED
SELLER)))
(RETRIEVE (?PATH-WITHOUT-MODEL) (AND (?PATH-WITHOUT-MODEL PATH) (NIL ?PATH-WITHOUT-MODEL HAS-
PATH))) --> (((?PATH-WITHOUT-MODEL INTL-BOOK-BUYER-PATH)) ((?PATH-WITHOUT-MODEL INTL-BOOK-
SELLER-PATH)))
(RETRIEVE (?BAD-OBJ-MSG) (AND (?BAD-OBJ-MSG OBJECT-MSG) ($?SENDER ?BAD-OBJ-MSG IS-MSG-SENDER)
($?RECEIVER ?BAD-OBJ-MSG IS-MSG-RECEIVER) (NOT (AND ($?SENDER (OR CITIZEN-CLASS-PATH CITIZEN-
INSTANCE-PATH ENVIRONMENT-CLASS-PATH ENVIRONMENT-INSTANCE-PATH)) ($?RECEIVER (OR OBJECT-CLASS-
PATH OBJECT-INSTANCE-PATH PASSIVE-ENVIRONMENT-CLASS-PATH P-ENVIRONMENT-INSTANCE-PATH)))))) -->
(((?BAD-OBJ-MSG GETAUTHOR)))
(RETRIEVE (?BAD-OBJ-MSG) (AND (?BAD-OBJ-MSG OBJECT-MSG) (?SENDER-RECEIVER ?BAD-OBJ-MSG IS-MSG-
SENDER) (?SENDER-RECEIVER ?BAD-OBJ-MSG IS-MSG-RECEIVER) (?SENDER-RECEIVER (OR OBJECT-CLASS-
PATH OBJECT-INSTANCE-PATH PASSIVE-ENVIRONMENT-CLASS-PATH P-ENVIRONMENT-INSTANCE-PATH)))) -->
(((?BAD-OBJ-MSG GETABSTRACT)))
(RETRIEVE ($?SENDER $?RECEIVER ?AG-MSG) (AND (?AG-MSG AGENT-MSG) ($?SENDER ?AG-MSG IS-MSG-
SENDER) ($?RECEIVER ?AG-MSG IS-MSG-RECEIVER) (OR (AND ($?SENDER (OR AGENT-ROLE-CLASS-PATH
AGENT-ROLE-INSTANCE-PATH)) ($?RECEIVER (OR AGENT-CLASS-PATH AGENT-INSTANCE-PATH OBJECT-CLASS-
PATH OBJECT-INSTANCE-PATH SUB-ORGANIZATION-CLASS-PATH SUB-ORGANIZATION-INSTANCE-PATH PASSIVE-
ENVIRONMENT-CLASS-PATH P-ENVIRONMENT-INSTANCE-PATH OBJECT-ROLE-CLASS-PATH OBJECT-ROLE-
INSTANCE-PATH))) (AND ($?SENDER (OR ACTIVE-ENVIRONMENT-CLASS-PATH A-ENVIRONMENT-INSTANCE-PATH
MAIN-ORGANIZATION-CLASS-PATH MAIN-ORGANIZATION-INSTANCE-PATH)) ($?RECEIVER (OR AGENT-CLASS-
PATH AGENT-INSTANCE-PATH SUB-ORGANIZATION-CLASS-PATH SUB-ORGANIZATION-INSTANCE-PATH OBJECT-
CLASS-PATH OBJECT-INSTANCE-PATH PASSIVE-ENVIRONMENT-CLASS-PATH P-ENVIRONMENT-INSTANCE-PATH
OBJECT-ROLE-CLASS-PATH OBJECT-ROLE-INSTANCE-PATH))) (NOT ($?SENDER (OR AGENT-ROLE-CLASS-PATH
AGENT-ROLE-INSTANCE-PATH ACTIVE-ENVIRONMENT-CLASS-PATH A-ENVIRONMENT-INSTANCE-PATH MAIN-
ORGANIZATION-CLASS-PATH MAIN-ORGANIZATION-INSTANCE-PATH))) (NOT ($?RECEIVER (OR AGENT-ROLE-
CLASS-PATH AGENT-ROLE-INSTANCE-PATH ACTIVE-ENVIRONMENT-CLASS-PATH A-ENVIRONMENT-INSTANCE-PATH
MAIN-ORGANIZATION-CLASS-PATH MAIN-ORGANIZATION-INSTANCE-PATH)))))) --> ((($?SENDER BOOK-PATH)
($?RECEIVER USER-AGENT-PATH) (?AG-MSG ANSWERPROPOSAL)) (($?SENDER USER-AGENT-PATH) ($?RECEIVER
STORE-AGENT-PATH) (?AG-MSG INFORM)) (($?SENDER USER-AGENT-PATH) ($?RECEIVER BOOK-PATH) (?AG-
MSG PROPOSAL)))
(RETRIEVE (?SENDER ?RECEIVER ?CREATE) (AND (?CREATE CREATE-MSG) (?SENDER ?CREATE IS-MSG-
SENDER) (?RECEIVER ?CREATE IS-MSG-RECEIVER) (OR (AND (?SENDER (OR MAIN-ORGANIZATION-CLASS-PATH
MAIN-ORGANIZATION-INSTANCE-PATH AGENT-CLASS-PATH AGENT-INSTANCE-PATH)) (?RECEIVER (OR ROLE-
CLASS-PATH ROLE-INSTANCE-PATH))) (AND (?SENDER (OR SUB-ORGANIZATION-CLASS-PATH SUB-
ORGANIZATION-INSTANCE-PATH)) (?RECEIVER (OR MAIN-ORGANIZATION-CLASS-PATH MAIN-ORGANIZATION-
INSTANCE-PATH ROLE-CLASS-PATH ROLE-INSTANCE-PATH))) (AND (?SENDER (OR ENVIRONMENT-CLASS-PATH
ENVIRONMENT-INSTANCE-PATH)) (?RECEIVER (OR ROLE-CLASS-PATH ROLE-INSTANCE-PATH))) (AND (?SENDER
(OR OBJECT-CLASS-PATH OBJECT-INSTANCE-PATH)) (?RECEIVER (OR AGENT-CLASS-PATH AGENT-INSTANCE-
PATH ORGANIZATION-CLASS-PATH ORGANIZATION-INSTANCE-PATH ENVIRONMENT-CLASS-PATH ENVIRONMENT-
INSTANCE-PATH ROLE-CLASS-PATH ROLE-INSTANCE-PATH))) (?SENDER (OR ROLE-CLASS-PATH ROLE-
INSTANCE-PATH))))) --> (((?SENDER AMERICANAS-BUYER-PATH) (?RECEIVER JOHN-PATH) (?CREATE
CREATE-AGENT-2)))
(RETRIEVE (?SENDER ?CREATE) (AND (?CREATE CREATE-MSG) (?SENDER ?CREATE IS-MSG-SENDER) (?SENDER
?CREATE IS-MSG-RECEIVER))) --> (((?SENDER WEBSHOPPING-PATH) (?CREATE CREATE-ENV-1)))
(RETRIEVE (?SENDER ?RECEIVER ?DESTROY) (AND (?DESTROY DESTROY-MSG) (?SENDER ?DESTROY IS-MSG-
SENDER) (?RECEIVER ?DESTROY IS-MSG-RECEIVER) (OR (AND (?SENDER (OR MAIN-ORGANIZATION-CLASS-
PATH MAIN-ORGANIZATION-INSTANCE-PATH AGENT-CLASS-PATH AGENT-INSTANCE-PATH)) (?RECEIVER (OR
ROLE-CLASS-PATH ROLE-INSTANCE-PATH))) (AND (?SENDER (OR SUB-ORGANIZATION-CLASS-PATH SUB-
ORGANIZATION-INSTANCE-PATH)) (?RECEIVER (OR MAIN-ORGANIZATION-CLASS-PATH MAIN-ORGANIZATION-
INSTANCE-PATH ROLE-CLASS-PATH ROLE-INSTANCE-PATH))) (AND (?SENDER (OR ENVIRONMENT-CLASS-PATH
ENVIRONMENT-INSTANCE-PATH)) (?RECEIVER (OR ROLE-CLASS-PATH ROLE-INSTANCE-PATH))) (AND (?SENDER
(OR OBJECT-CLASS-PATH OBJECT-INSTANCE-PATH)) (?RECEIVER (OR AGENT-CLASS-PATH AGENT-INSTANCE-
PATH ORGANIZATION-CLASS-PATH ORGANIZATION-INSTANCE-PATH ENVIRONMENT-CLASS-PATH ENVIRONMENT-
INSTANCE-PATH ROLE-CLASS-PATH ROLE-INSTANCE-PATH))) (?SENDER (OR ROLE-CLASS-PATH ROLE-
INSTANCE-PATH))))) --> (((?SENDER JOHN-PATH) (?RECEIVER AMERICANAS-BUYER-PATH) (?DESTROY
DESTROY-ORG-1)))
(RETRIEVE (?BAD-COMMIT-AG-ROLE ?WRONG-SENDER) (AND (?BAD-COMMIT-AG-ROLE ROLE-COMMIT-MSG)
(?BAD-COMMIT-AG-ROLE ?WRONG-SENDER HAS-MSG-SENDER) (NOT (?WRONG-SENDER (OR AGENT-CLASS-PATH
AGENT-INSTANCE-PATH))))) --> (((?BAD-COMMIT-AG-ROLE COMMIT-OB-ROLE-1) (?WRONG-SENDER INTL-
BOOK-SELLER)) ((?BAD-COMMIT-AG-ROLE COMMIT-ROLE-3) (?WRONG-SENDER BOOK-PATH)))
(RETRIEVE (?BAD-COMMIT-AG-ROLE ?SENDER ?WRONG-RECEIVER) (AND (?BAD-COMMIT-AG-ROLE ROLE-COMMIT-
MSG) (?BAD-COMMIT-AG-ROLE ?SENDER HAS-MSG-SENDER) (?BAD-COMMIT-AG-ROLE ?WRONG-RECEIVER HAS-
MSG-RECEIVER) (?SENDER (OR AGENT-CLASS-PATH AGENT-INSTANCE-PATH)) (NOT (?WRONG-RECEIVER (OR
AGENT-CLASS-PATH AGENT-INSTANCE-PATH))))) --> (((?BAD-COMMIT-AG-ROLE COMMIT-ROLE-2) (?SENDER
USER-AGENT-PATH) (?WRONG-RECEIVER SELLER-PATH)) ((?BAD-COMMIT-AG-ROLE COMMIT-ROLE-1) (?SENDER
JOHN-PATH) (?WRONG-RECEIVER OFFER-PATH)))
(RETRIEVE (?BAD-COMMIT-AG-ROLE ?SENDER ?WRONG-RECEIVER) (AND (?BAD-COMMIT-AG-ROLE ROLE-COMMIT-
MSG) (?BAD-COMMIT-AG-ROLE ?SENDER HAS-MSG-SENDER) (?BAD-COMMIT-AG-ROLE ?WRONG-RECEIVER HAS-
MSG-RECEIVER) (?SENDER (OR SUB-ORGANIZATION-CLASS-PATH SUB-ORGANIZATION-INSTANCE-PATH)) (NOT
(?WRONG-RECEIVER (OR SUB-ORGANIZATION-CLASS-PATH SUB-ORGANIZATION-INSTANCE-PATH))))) --> NIL
(RETRIEVE (?BAD-COMMIT-AG-ROLE ?SENDER ?RECEIVER) (AND (?BAD-COMMIT-AG-ROLE ROLE-COMMIT-MSG)
(?BAD-COMMIT-AG-ROLE ?SENDER HAS-MSG-SENDER) (?BAD-COMMIT-AG-ROLE ?RECEIVER HAS-MSG-RECEIVER)
(?SENDER (OR AGENT-CLASS-PATH AGENT-INSTANCE-PATH)) (?RECEIVER (OR AGENT-CLASS-PATH AGENT-
INSTANCE-PATH)))) --> (((?BAD-COMMIT-AG-ROLE COMMIT-ROLE-4) (?SENDER STORE-AGENT-PATH)
(?RECEIVER USER-AGENT-PATH)))
(RETRIEVE (?BAD-COMMIT-AG-ROLE ?SENDER ?RECEIVER) (AND (?BAD-COMMIT-AG-ROLE ROLE-COMMIT-MSG)
(?BAD-COMMIT-AG-ROLE ?SENDER HAS-MSG-SENDER) (?BAD-COMMIT-AG-ROLE ?RECEIVER HAS-MSG-RECEIVER)
(?SENDER (OR SUB-ORGANIZATION-CLASS-PATH SUB-ORGANIZATION-INSTANCE-PATH)) (?RECEIVER (OR SUB-
ORGANIZATION-CLASS-PATH SUB-ORGANIZATION-INSTANCE-PATH)))) --> NIL
(RETRIEVE (?BAD-COMMIT-AG-ROLE $?SENDER $?RECEIVER) (AND (?BAD-COMMIT-AG-ROLE ROLE-COMMIT-MSG)
(?BAD-COMMIT-AG-ROLE $?SENDER HAS-MSG-SENDER) (?BAD-COMMIT-AG-ROLE $?RECEIVER HAS-MSG-
RECEIVER) ($?SENDER AGENT-INSTANCE-PATH) ($?RECEIVER AGENT-INSTANCE-PATH) ($?SENDER ?HEAD HAS-
HEAD) ($?RECEIVER ?HEAD HAS-HEAD) (?ROLE AGENT-ROLE-INSTANCE-PATH) ($?SENDER ?ROLE HAS-TAIL)
($?RECEIVER ?ROLE HAS-TAIL))) --> (((?BAD-COMMIT-AG-ROLE COMMIT-ROLE-5) ($?SENDER JOHN-PATH)
($?RECEIVER JOHN-PATH)))
(RETRIEVE (?BAD-COMMIT-AG-ROLE $?SENDER $?RECEIVER) (AND (?BAD-COMMIT-AG-ROLE ROLE-COMMIT-MSG)
(?BAD-COMMIT-AG-ROLE $?SENDER HAS-MSG-SENDER) (?BAD-COMMIT-AG-ROLE $?RECEIVER HAS-MSG-
RECEIVER) ($?SENDER SUB-ORGANIZATION-INSTANCE-PATH) ($?RECEIVER SUB-ORGANIZATION-INSTANCE-
PATH) ($?SENDER ?HEAD HAS-HEAD) ($?RECEIVER ?HEAD HAS-HEAD) (?ROLE AGENT-ROLE-INSTANCE-PATH)
($?SENDER ?ROLE HAS-TAIL) ($?RECEIVER ?ROLE HAS-TAIL))) --> NIL
(RETRIEVE (?BAD-COMMIT-OBJ-ROLE ?SENDER ?RECEIVER) (AND (?BAD-COMMIT-OBJ-ROLE ROLE-COMMIT-MSG)
(?BAD-COMMIT-OBJ-ROLE ?SENDER HAS-MSG-SENDER) (?BAD-COMMIT-OBJ-ROLE ?RECEIVER HAS-MSG-
RECEIVER) (?SENDER (OR OBJECT-CLASS-PATH OBJECT-INSTANCE-PATH ENVIRONMENT-CLASS-PATH
ENVIRONMENT-INSTANCE-PATH ROLE-CLASS-PATH ROLE-INSTANCE-PATH)))) --> (((?BAD-COMMIT-OBJ-ROLE
COMMIT-ROLE-3) (?SENDER BOOK-PATH) (?RECEIVER USER-AGENT-PATH)))
(RETRIEVE (?BAD-COMMIT-OBJ-ROLE ?SENDER ?RECEIVER) (AND (?BAD-COMMIT-OBJ-ROLE ROLE-COMMIT-MSG)
(?BAD-COMMIT-OBJ-ROLE ?SENDER HAS-MSG-SENDER) (?BAD-COMMIT-OBJ-ROLE ?RECEIVER HAS-MSG-
RECEIVER) (?SENDER (OR AGENT-CLASS-PATH AGENT-INSTANCE-PATH ORGANIZATION-CLASS-PATH
ORGANIZATION-INSTANCE-PATH)) (NOT (?RECEIVER (OR OBJECT-CLASS-PATH OBJECT-INSTANCE-PATH))))) -
-> (((?BAD-COMMIT-OBJ-ROLE COMMIT-ROLE-4) (?SENDER STORE-AGENT-PATH) (?RECEIVER USER-AGENT-
PATH)) ((?BAD-COMMIT-OBJ-ROLE COMMIT-ROLE-2) (?SENDER USER-AGENT-PATH) (?RECEIVER SELLER-
PATH)) ((?BAD-COMMIT-OBJ-ROLE COMMIT-ROLE-1) (?SENDER JOHN-PATH) (?RECEIVER OFFER-PATH)))
(RETRIEVE (?BAD-CANCEL-AG-ROLE ?SENDER ?RECEIVER) (AND (?BAD-CANCEL-AG-ROLE ROLE-CANCEL-MSG)
(?BAD-CANCEL-AG-ROLE ?SENDER HAS-MSG-SENDER) (?BAD-CANCEL-AG-ROLE ?RECEIVER HAS-MSG-RECEIVER)
(OR (?SENDER (OR OBJECT-CLASS-PATH OBJECT-INSTANCE-PATH ENVIRONMENT-CLASS-PATH ENVIRONMENT-
INSTANCE-PATH ROLE-CLASS-PATH ROLE-INSTANCE-PATH MAIN-ORGANIZATION-CLASS-PATH MAIN-
ORGANIZATION-INSTANCE-PATH)) (AND (?SENDER (OR AGENT-CLASS-PATH AGENT-INSTANCE-PATH))
(?RECEIVER (OR AGENT-CLASS-PATH AGENT-INSTANCE-PATH)) (?SENDER ?HEAD-S HAS-HEAD) (?RECEIVER
?HEAD-R HAS-HEAD)) (AND (?SENDER (OR SUB-ORGANIZATION-CLASS-PATH SUB-ORGANIZATION-INSTANCE-
PATH)) (?RECEIVER (OR SUB-ORGANIZATION-CLASS-PATH SUB-ORGANIZATION-INSTANCE-PATH)) (?SENDER
?HEAD-S HAS-HEAD) (?RECEIVER ?HEAD-R HAS-HEAD))))) --> (((?BAD-CANCEL-AG-ROLE CANCEL-OB-ROLE-
3) (?SENDER SPECIAL-OFFER-PATH) (?RECEIVER JOHN-PATH)))
(RETRIEVE (?BAD-CANCEL-OBJ-ROLE ?SENDER ?RECEIVER) (AND (?BAD-CANCEL-OBJ-ROLE ROLE-CANCEL-MSG)
(?BAD-CANCEL-OBJ-ROLE ?SENDER HAS-MSG-SENDER) (?BAD-CANCEL-OBJ-ROLE ?RECEIVER HAS-MSG-
RECEIVER) (OR (?SENDER (OR OBJECT-CLASS-PATH OBJECT-INSTANCE-PATH ENVIRONMENT-CLASS-PATH
ENVIRONMENT-INSTANCE-PATH ROLE-CLASS-PATH ROLE-INSTANCE-PATH MAIN-ORGANIZATION-CLASS-PATH
MAIN-ORGANIZATION-INSTANCE-PATH)) (AND (?SENDER (OR AGENT-CLASS-PATH AGENT-INSTANCE-PATH
ORGANIZATION-CLASS-PATH ORGANIZATION-INSTANCE-PATH)) (?RECEIVER (NOT (OR OBJECT-CLASS-PATH
OBJECT-INSTANCE-PATH))))))) --> (((?BAD-CANCEL-OBJ-ROLE CANCEL-OB-ROLE-3) (?SENDER SPECIAL-
OFFER-PATH) (?RECEIVER JOHN-PATH)))
(RETRIEVE (?SENDER ?RECEIVER ?ACTIVATE) (AND (?ACTIVATE ROLE-ACTIVATE-MSG) (?SENDER ?ACTIVATE
IS-MSG-SENDER) (?RECEIVER ?ACTIVATE IS-MSG-RECEIVER) (OR (?SENDER (OR MAIN-ORGANIZATION-CLASS-
PATH MAIN-ORGANIZATION-INSTANCE-PATH ROLE-CLASS-PATH ROLE-INSTANCE-PATH ENVIRONMENT-CLASS-PATH
ENVIRONMENT-INSTANCE-PATH OBJECT-CLASS-PATH OBJECT-INSTANCE-PATH)) (AND (?SENDER (OR AGENT-
CLASS-PATH AGENT-INSTANCE-PATH SUB-ORGANIZATION-CLASS-PATH SUB-ORGANIZATION-INSTANCE-PATH))
(?RECEIVER (OR CITIZEN-CLASS-PATH CITIZEN-INSTANCE-PATH ENVIRONMENT-CLASS-PATH ENVIRONMENT-
INSTANCE-PATH OBJECT-ROLE-CLASS-PATH OBJECT-ROLE-INSTANCE-PATH)))))) --> (((?SENDER JOHN-PATH)
(?RECEIVER MARY-PATH) (?ACTIVATE ACTIVATE-ROLE-3)) ((?SENDER JOHN-PATH) (?RECEIVER SPECIAL-
OFFER-PATH) (?ACTIVATE ACTIVATE-ROLE-2)))
(RETRIEVE (?SENDER ?RECEIVER ?DEACTIVATE) (AND (?DEACTIVATE ROLE-DEACTIVATE-MSG) (?SENDER
?DEACTIVATE IS-MSG-SENDER) (?RECEIVER ?DEACTIVATE IS-MSG-RECEIVER) (OR (?SENDER (OR MAIN-
ORGANIZATION-CLASS-PATH MAIN-ORGANIZATION-INSTANCE-PATH ROLE-CLASS-PATH ROLE-INSTANCE-PATH
ENVIRONMENT-CLASS-PATH ENVIRONMENT-INSTANCE-PATH OBJECT-CLASS-PATH OBJECT-INSTANCE-PATH)) (AND
(?SENDER (OR AGENT-CLASS-PATH AGENT-INSTANCE-PATH SUB-ORGANIZATION-CLASS-PATH SUB-
ORGANIZATION-INSTANCE-PATH)) (?RECEIVER (OR CITIZEN-CLASS-PATH CITIZEN-INSTANCE-PATH
ENVIRONMENT-CLASS-PATH ENVIRONMENT-INSTANCE-PATH OBJECT-ROLE-CLASS-PATH OBJECT-ROLE-INSTANCE-
PATH)))))) --> (((?SENDER JOHN-PATH) (?RECEIVER SPECIAL-OFFER-PATH) (?DEACTIVATE DEACTIVATE-
ROLE-2)))
(RETRIEVE (?SENDER ?RECEIVER ?CHANGE) (AND (?CHANGE ROLE-CHANGE-MSG) (?SENDER ?CHANGE IS-MSG-
SENDER) (?RECEIVER ?CHANGE IS-MSG-RECEIVER) (OR (?SENDER (OR MAIN-ORGANIZATION-CLASS-PATH
MAIN-ORGANIZATION-INSTANCE-PATH ROLE-CLASS-PATH ROLE-INSTANCE-PATH ENVIRONMENT-CLASS-PATH
ENVIRONMENT-INSTANCE-PATH OBJECT-CLASS-PATH OBJECT-INSTANCE-PATH)) (AND (?SENDER (OR AGENT-
CLASS-PATH AGENT-INSTANCE-PATH SUB-ORGANIZATION-CLASS-PATH SUB-ORGANIZATION-INSTANCE-PATH))
(?RECEIVER (OR CITIZEN-CLASS-PATH CITIZEN-INSTANCE-PATH ENVIRONMENT-CLASS-PATH ENVIRONMENT-
INSTANCE-PATH OBJECT-ROLE-CLASS-PATH OBJECT-ROLE-INSTANCE-PATH)))))) --> (((?SENDER JOHN-PATH)
(?RECEIVER SPECIAL-OFFER-PATH) (?CHANGE CHANGE-ROLE-1)))
(RETRIEVE (?PATH-WITHOUT-SOURCE) (AND (?PATH-WITHOUT-SOURCE INSTANCE-PATH) (?PATH-WITHOUT-
SOURCE ?HEAD HAS-HEAD) (?HEAD NIL IS-INSTANCEOF))) --> (((?PATH-WITHOUT-SOURCE MARY-PATH)))
(RETRIEVE (?MSG-BAD-DEF ?SENDER ?RECEIVER) (AND (?MSG-BAD-DEF MSG) (?MSG-BAD-DEF ?SENDER HAS-
MSG-SENDER) (?MSG-BAD-DEF ?RECEIVER HAS-MSG-RECEIVER) (OR (NOT (?SENDER PATH)) (NOT (?RECEIVER
PATH))))) --> (((?MSG-BAD-DEF CANCEL-ROLE-1) (?SENDER INTL-BOOK-SELLER) (?RECEIVER SPECIAL-
OFFER-PATH)) ((?MSG-BAD-DEF COMMIT-OB-ROLE-1) (?SENDER INTL-BOOK-SELLER) (?RECEIVER SPECIAL-
OFFER-PATH)))
(RETRIEVE (?AG-MSG ?AG-ROLE-CLASS) (AND (?AG-MSG AGENT-MSG) (?AG-ROLE-CLASS AGENT-ROLE-CLASS)
(?AG-ROLE-CLASS ?AG-MSG HAS-MSG) (NOT (?AG-MSG (SOME (INV HAS-MSG) PROTOCOL))))) --> (((?AG-
MSG AGENT-MSG-4) (?AG-ROLE-CLASS BUYER)))
(RETRIEVE (?PRTCL-ALONE) (AND (?PRTCL-ALONE PROTOCOL) (NOT (?PRTCL-ALONE (SOME (INV HAS-
PROTOCOL) AGENT-ROLE-CLASS))))) --> (((?PRTCL-ALONE PROTOCOL-1)))
(RETRIEVE (?PLAN-ALONE) (AND (?PLAN-ALONE PLAN) (NOT (?PLAN-ALONE (SOME (INV HAS-PLAN) (OR
AGENT-CLASS ORGANIZATION-CLASS ACTIVE-ENVIRONMENT-CLASS)))))) --> (((?PLAN-ALONE PLAN-3))
((?PLAN-ALONE PLAN-2)))
(RETRIEVE (?ACTION-ALONE) (AND (?ACTION-ALONE ACTION) (NOT (?ACTION-ALONE (SOME (INV HAS-
ACTION) (OR AGENT-CLASS ORGANIZATION-CLASS ACTIVE-ENVIRONMENT-CLASS)))))) --> (((?ACTION-ALONE
ACTION-3)) ((?ACTION-ALONE ACTION-2)))
```

```
(RETRIEVE (?AG-MSG-ALONE) (AND (?AG-MSG-ALONE AGENT-MSG) (NOT (?AG-MSG-ALONE (SOME (INV HAS-
MSG) (OR AGENT-ROLE-CLASS MAIN-ORGANIZATION-CLASS ACTIVE-ENVIRONMENT-CLASS)))))) --> (((?AG-
MSG-ALONE ANSWERPROPOSAL)) ((?AG-MSG-ALONE INFORM)) ((?AG-MSG-ALONE PROPOSAL)))
(RETRIEVE (?PATH-WITHOUT-HEAD) (AND (?PATH-WITHOUT-HEAD PATH) (?PATH-WITHOUT-HEAD NIL HAS-
HEAD))) --> (((?PATH-WITHOUT-HEAD STORE-AGENT-PATH)) ((?PATH-WITHOUT-HEAD OFFER-PATH))
((?PATH-WITHOUT-HEAD SPECIAL-OFFER-PATH)) ((?PATH-WITHOUT-HEAD INTL-BOOK-BUYER-PATH)) ((?PATH-
WITHOUT-HEAD INTL-BOOK-SELLER-PATH)) ((?PATH-WITHOUT-HEAD BUYER-PATH)) ((?PATH-WITHOUT-HEAD
AMERICANAS-BUYER-PATH)) ((?PATH-WITHOUT-HEAD SELLER-PATH)) ((?PATH-WITHOUT-HEAD AMERICANAS-
SELLER-PATH)) ((?PATH-WITHOUT-HEAD MAGAZINE-PATH)) ((?PATH-WITHOUT-HEAD COMM-ACM-PATH))
((?PATH-WITHOUT-HEAD BOOK-PATH)) ((?PATH-WITHOUT-HEAD SOMMERVILLE2001-PATH)) ((?PATH-WITHOUT-
HEAD VIRTUAL-MARKET-PATH)) ((?PATH-WITHOUT-HEAD WEBSHOPPING-PATH)) ((?PATH-WITHOUT-HEAD
IMPORTED-BOOKSTORE-PATH)) ((?PATH-WITHOUT-HEAD INTL-BOOKSTORE-DOTCOM-PATH)) ((?PATH-WITHOUT-
HEAD AMERICANAS-DOTCOM-PATH)) ((?PATH-WITHOUT-HEAD VENDOR-AMERICANAS-PATH)) ((?PATH-WITHOUT-
HEAD GENERAL-STORE-PATH)))
(RETRIEVE (?PRTCL-WITHOUT-STRUCT) (AND (?PRTCL-WITHOUT-STRUCT PROTOCOL) (?PRTCL-WITHOUT-STRUCT
NIL HAS-STRUCTURE))) --> (((?PRTCL-WITHOUT-STRUCT PROTOCOL-3)))
(RETRIEVE (?PLAN-WITHOUT-STRUCT) (AND (?PLAN-WITHOUT-STRUCT PLAN) (?PLAN-WITHOUT-STRUCT NIL
HAS-STRUCTURE))) --> (((?PLAN-WITHOUT-STRUCT PLAN-3)) ((?PLAN-WITHOUT-STRUCT PLAN-2)))
(RETRIEVE (?BAD-STRUCT ?PRTCL ?ELEMENT) (AND (?BAD-STRUCT SEQUENCE) (?PRTCL PROTOCOL) (?PRTCL
?BAD-STRUCT HAS-STRUCTURE) (OR (AND (?BAD-STRUCT ?ELEMENT HAS-ELEMENT) (NOT (?PRTCL ?ELEMENT
HAS-MSG))) (AND (?PRTCL ?ELEMENT HAS-MSG) (NOT (?BAD-STRUCT ?ELEMENT HAS-ELEMENT)))))) -->
(((?BAD-STRUCT SEQ-MODEL-STRUCT) (?PRTCL PROTOCOL-1) (?ELEMENT AGENT-MSG-2)) ((?BAD-STRUCT
SEQ-MODEL-STRUCT-2) (?PRTCL PROTOCOL-2) (?ELEMENT AGENT-MSG-1)) ((?BAD-STRUCT SEQ-MODEL-
STRUCT-2) (?PRTCL PROTOCOL-2) (?ELEMENT AGENT-MSG-2)))

(RETRIEVE (?BAD-STRUCT) (AND (?BAD-STRUCT SEQUENCE) (?BAD-STRUCT ?ELEMENT1 HAS-FIRST) (?BAD-
STRUCT ?ELEMENT2 HAS-LAST) (?ELEMENT2 ?ELEMENT1 IS-BEFORE))) --> (((?BAD-STRUCT SEQ-MODEL-
STRUCT-2)))
(RETRIEVE (?AG-ROLE-NO-PRTCL) (AND (?AG-ROLE-NO-PRTCL AGENT-ROLE-CLASS) (?AG-ROLE-NO-PRTCL NIL
HAS-PROTOCOL))) --> (((?AG-ROLE-NO-PRTCL MEDIATOR)))
(RETRIEVE (?AGENT ?ROLE) (AND (?AGENT AGENT-CLASS) (?PL PLAY) (?ROLE AGENT-ROLE-CLASS) (?AGENT
?PL IS-END1) (?ROLE ?PL IS-END2) (NOT (?ROLE (SOME IS-END2 (AND OWNERSHIP (SOME HAS-END1
ORGANIZATION-CLASS))))))) --> (((?AGENT STORE-AGENT) (?ROLE SELLER)))
(RETRIEVE (?SUB-ORG ?ROLE) (AND (?SUB-ORG SUB-ORGANIZATION-CLASS) (?PL PLAY) (?ROLE AGENT-
ROLE-CLASS) (?SUB-ORG ?PL IS-END1) (?ROLE ?PL IS-END2) (NOT (?ROLE (SOME IS-END2 (AND
OWNERSHIP (SOME HAS-END1 ORGANIZATION-CLASS))))))) --> (((?SUB-ORG IMPORTED-BOOKSTORE) (?ROLE
SELLER)))
(RETRIEVE (?BAD-PRTCL ?FIRST-OR-LAST) (AND (?BAD-PRTCL PROTOCOL) (?FIRST-OR-LAST MSG) (OR
(?BAD-PRTCL ?FIRST-OR-LAST HAS-FIRST) (?BAD-PRTCL ?FIRST-OR-LAST HAS-LAST)))) --> NIL

;;;;;;;;;;;;;;;;;;;;;;;;
(IN-KNOWLEDGE-BASE MASML-TBOX PHASE1-ABOX) --> (MASML-TBOX PHASE1-ABOX)
(IN-TBOX MASML-TBOX) --> MASML-TBOX
(TBOX-CLASSIFIED-P) --> T
(TBOX-COHERENT-P) --> T
(TBOX-CYCLIC-P) --> NIL
(IN-ABOX PHASE1-ABOX MASML-TBOX) --> PHASE1-ABOX
(CHECK-ABOX-COHERENCE) --> T

(RETRIEVE (?ROLECL ?CLMD) (AND (?ROLECL ROLE-CLASS) (?CLMD CLASS-MODEL) (?CLMD ?ROLECL HAS-
CLASS))) --> (((?ROLECL SELLER) (?CLMD CLASS-DIAGRAM)) ((?ROLECL BUYER) (?CLMD CLASS-DIAGRAM))
((?ROLECL MEDIATOR) (?CLMD CLASS-DIAGRAM)))
(RETRIEVE (?PL ?CLMD) (AND (?PL PLAY) (?CLMD CLASS-MODEL) (?CLMD ?PL HAS-RELATIONSHIP))) -->
(((?PL PLAY-1) (?CLMD CLASS-DIAGRAM)))
(RETRIEVE (?OWN ?CLMD) (AND (?OWN OWNERSHIP) (?CLMD CLASS-MODEL) (?CLMD ?OWNERSHIP HAS-
RELATIONSHIP))) --> (((?OWN OWN-1) (?CLMD CLASS-DIAGRAM)) ((?OWN OWN-2) (?CLMD CLASS-DIAGRAM))
((?OWN OWN-3) (?CLMD CLASS-DIAGRAM)))
(RETRIEVE (?CTRL ?CLMD) (AND (?CTRL CONTROL) (?CLMD CLASS-MODEL) (?CLMD ?CTRL HAS-
RELATIONSHIP))) --> (((?CTRL CTRL-3) (?CLMD CLASS-DIAGRAM)))
(RETRIEVE (?INH ?CLMD) (AND (?INH INHABIT) (?CLMD CLASS-MODEL) (?CLMD ?INH HAS-RELATIONSHIP)
(?END1 ?INH IS-END1) (?END2 ?INH IS-END2) (OR (?END1 (OR AGENT-CLASS ORGANIZATION-CLASS ROLE-
CLASS ENVIRONMENT-CLASS)) (?END2 (OR CITIZEN-CLASS ROLE-CLASS))))) --> (((?INH INHABIT-2)
(?CLMD CLASS-DIAGRAM)) ((?INH INHABIT-3) (?CLMD CLASS-DIAGRAM)) ((?INH INHABIT-1) (?CLMD
CLASS-DIAGRAM)))
(RETRIEVE (?ASS ?CLMD) (AND (?ASS ASSOCIATION) (?CLMD CLASS-MODEL) (?CLMD ?ASS HAS-
RELATIONSHIP) (?END1 ?ASS IS-END1) (?END2 ?ASS IS-END2) (OR (AND (?END1 (OR AGENT-CLASS
ORGANIZATION-CLASS)) (?END2 (OR AGENT-CLASS ORGANIZATION-CLASS ENVIRONMENT-CLASS))) (AND
(?END1 ENVIRONMENT-CLASS) (?END2 CITIZEN-CLASS)) (?END1 ROLE-CLASS) (?END2 ROLE-CLASS)))) -->
(((?ASS ASSOC-1) (?CLMD CLASS-DIAGRAM)))
(RETRIEVE (?SPEC ?CLMD) (AND (?SPEC SPECIALIZATION) (?CLMD CLASS-MODEL) (?CLMD ?SPEC HAS-
RELATIONSHIP) (?END1 ?SPEC IS-END1) (?END2 ?SPEC IS-END2) (OR (?END1 ROLE-CLASS) (?END2 ROLE-
CLASS) (AND (?END1 AGENT-CLASS) (?END2 (OR OBJECT-CLASS ORGANIZATION-CLASS ENVIRONMENT-
CLASS))) (AND (?END1 MAIN-ORGANIZATION-CLASS) (?END2 (OR OBJECT-CLASS SUB-ORGANIZATION-CLASS
AGENT-CLASS ENVIRONMENT-CLASS))) (AND (?END1 SUB-ORGANIZATION-CLASS) (?END2 (OR OBJECT-CLASS
MAIN-ORGANIZATION-CLASS AGENT-CLASS ENVIRONMENT-CLASS))) (AND (?END1 ACTIVE-ENVIRONMENT-CLASS)
(?END2 (OR CITIZEN-CLASS PASSIVE-ENVIRONMENT-CLASS))) (AND (?END1 PASSIVE-ENVIRONMENT-CLASS)
(?END2 (OR CITIZEN-CLASS ACTIVE-ENVIRONMENT-CLASS)))))) --> (((?SPEC SPEC-3) (?CLMD CLASS-
DIAGRAM)))
(RETRIEVE (?AGG ?CLMD) (AND (?AGG AGGREGATION) (?CLMD CLASS-MODEL) (?CLMD ?AGG HAS-
RELATIONSHIP) (?END1 ?AGG IS-END1) (?END2 ?AGG IS-END2) (OR (?END1 (OR ROLE-CLASS AGENT-CLASS
ORGANIZATION-CLASS ENVIRONMENT-CLASS)) (?END2 (OR ROLE-CLASS AGENT-CLASS ORGANIZATION-CLASS
ENVIRONMENT-CLASS))))) --> (((?AGG AGGR-2) (?CLMD CLASS-DIAGRAM)))
(RETRIEVE (?DEP ?CLMD) (AND (?DEP DEPENDENCY) (?CLMD CLASS-MODEL) (?CLMD ?DEP HAS-
RELATIONSHIP) (?END1 ?DEP IS-END1) (?END2 ?DEP IS-END2) (OR (?END1 (OR ROLE-CLASS AGENT-CLASS
ORGANIZATION-CLASS ENVIRONMENT-CLASS)) (?END2 (OR ROLE-CLASS AGENT-CLASS ORGANIZATION-CLASS
ENVIRONMENT-CLASS))))) --> (((?DEP DEP-4) (?CLMD CLASS-DIAGRAM)))

;;;;;;;;;;;;;;;;;;;;;
```

```
(IN-KNOWLEDGE-BASE MASML-TBOX PHASE1-ABOX) --> (MASML-TBOX PHASE1-ABOX)
(IN-TBOX MASML-TBOX) --> MASML-TBOX
(TBOX-CLASSIFIED-P) --> T
(TBOX-COHERENT-P) --> T
(TBOX-CYCLIC-P) --> NIL
(IN-ABOX PHASE1-ABOX MASML-TBOX) --> PHASE1-ABOX
(CHECK-ABOX-COHERENCE) --> T
(RETRIEVE (?SPEC ?ORGMD) (AND (?SPEC SPECIALIZATION) (?ORGMD ORGANIZATION-MODEL) (?ORGMD ?SPEC
HAS-RELATIONSHIP))) --> (((?SPEC SPEC-2) (?ORGMD ORG1-DIAGRAM)))
(RETRIEVE (?ASSOC ?ORGMD) (AND (?ASSOC ASSOCIATION) (?ORGMD ORGANIZATION-MODEL) (?ORGMD ?ASSOC
HAS-RELATIONSHIP))) --> (((?ASSOC ASSOC-4) (?ORGMD ORG1-DIAGRAM)) ((?ASSOC ASSOC-2) (?ORGMD
ORG1-DIAGRAM)))
(RETRIEVE (?AGG ?ORGMD) (AND (?AGG AGGREGATION) (?ORGMD ORGANIZATION-MODEL) (?ORGMD ?AGG HAS-
RELATIONSHIP))) --> (((?AGG AGGR-1) (?ORGMD ORG1-DIAGRAM)))
(RETRIEVE (?CTRL ?ORGMD) (AND (?CTRL CONTROL) (?ORGMD ORGANIZATION-MODEL) (?ORGMD ?CTRL HAS-
RELATIONSHIP))) --> (((?CTRL CTRL-3) (?ORGMD ORG1-DIAGRAM)))
(RETRIEVE (?DEP ?ORGMD) (AND (?DEP DEPENDENCY) (?ORGMD ORGANIZATION-MODEL) (?ORGMD ?DEP HAS-
RELATIONSHIP))) --> (((?DEP DEP-1) (?ORGMD ORG1-DIAGRAM)))
(RETRIEVE (?REL ?ORGMD) (AND (?REL RELATIONSHIP) (?ORGMD ORGANIZATION-MODEL) (?ORGMD ?REL HAS-
RELATIONSHIP) (?END1 ?REL IS-END1) (?END2 ?REL IS-END2) (?END1 ROLE-CLASS) (?END2 ROLE-
CLASS))) --> (((?REL CTRL-3) (?ORGMD ORG1-DIAGRAM)))
(RETRIEVE (?REL ?ORGMD) (AND (?REL RELATIONSHIP) (?ORGMD ORGANIZATION-MODEL) (?ORGMD ?REL HAS-
RELATIONSHIP) (?END1 ?REL IS-END1) (?END2 ?REL IS-END2) (?END1 AGENT-CLASS) (?END2 AGENT-
CLASS))) --> (((?REL SPEC-2) (?ORGMD ORG1-DIAGRAM)))
(RETRIEVE (?REL ?ORGMD) (AND (?REL RELATIONSHIP) (?ORGMD ORGANIZATION-MODEL) (?ORGMD ?REL HAS-
RELATIONSHIP) (?END1 ?REL IS-END1) (?END2 ?REL IS-END2) (?END1 ENVIRONMENT-CLASS) (?END2
ENVIRONMENT-CLASS))) --> (((?REL ASSOC-4) (?ORGMD ORG1-DIAGRAM)))
(RETRIEVE (?REL ?ORGMD-BAD) (AND (?REL RELATIONSHIP) (?ORGMD-BAD ORGANIZATION-MODEL) (?ORGMD-
BAD ?REL HAS-RELATIONSHIP) (?REL (OR ASSOCIATION AGGREGATION SPECIALIZATION CONTROL
DEPENDENCY)))) --> (((?REL DEP-1) (?ORGMD-BAD ORG1-DIAGRAM)) ((?REL CTRL-3) (?ORGMD-BAD ORG1-
DIAGRAM)) ((?REL AGGR-1) (?ORGMD-BAD ORG1-DIAGRAM)) ((?REL ASSOC-4) (?ORGMD-BAD ORG1-DIAGRAM))
((?REL ASSOC-2) (?ORGMD-BAD ORG1-DIAGRAM)) ((?REL SPEC-2) (?ORGMD-BAD ORG1-DIAGRAM)))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(IN-KNOWLEDGE-BASE MASML-TBOX PHASE1-ABOX) --> (MASML-TBOX PHASE1-ABOX)
(IN-TBOX MASML-TBOX) --> MASML-TBOX
(TBOX-CLASSIFIED-P) --> T
(TBOX-COHERENT-P) --> T
(TBOX-CYCLIC-P) --> NIL
(IN-ABOX PHASE1-ABOX MASML-TBOX) --> PHASE1-ABOX
(CHECK-ABOX-COHERENCE) --> T
(RETRIEVE (?MSG-WITHOUT-MODEL) (AND (?MSG-WITHOUT-MODEL (OR STEREOTYPED-MSG OBJECT-MSG)) (NOT
(?MSG-WITHOUT-MODEL (SOME (INV HAS-MSG) SEQUENCE-MODEL))))) --> (((?MSG-WITHOUT-MODEL
GETTITLE)) ((?MSG-WITHOUT-MODEL GETABSTRACT)) ((?MSG-WITHOUT-MODEL GETAUTHOR)) ((?MSG-WITHOUT-
MODEL CHANGE-ROLE-1)) ((?MSG-WITHOUT-MODEL DEACTIVATE-ROLE-1)) ((?MSG-WITHOUT-MODEL
DEACTIVATE-ROLE-2)) ((?MSG-WITHOUT-MODEL ACTIVATE-ROLE-2)) ((?MSG-WITHOUT-MODEL ACTIVATE-ROLE-
3)) ((?MSG-WITHOUT-MODEL ACTIVATE-ROLE-1)) ((?MSG-WITHOUT-MODEL CANCEL-ROLE-1)) ((?MSG-
WITHOUT-MODEL CANCEL-OB-ROLE-2)) ((?MSG-WITHOUT-MODEL CANCEL-OB-ROLE-3)) ((?MSG-WITHOUT-MODEL
COMMIT-ROLE-3)))
(RETRIEVE (?MSG-WITHOUT-OWNER ?SEQ-MD ?SENDER) (AND (?MSG-WITHOUT-OWNER AGENT-MSG) (?SEQ-MD
SEQUENCE-MODEL) (?SEQ-MD ?MSG-WITHOUT-OWNER HAS-MSG) (?MSG-WITHOUT-OWNER ?SENDER HAS-MSG-
SENDER) (NOT (?MSG-WITHOUT-OWNER (SOME (INV HAS-MSG) (OR AGENT-ROLE-CLASS MAIN-ORGANIZATION-
CLASS ACTIVE-ENVIRONMENT-CLASS)))))) --> (((?MSG-WITHOUT-OWNER AGENT-MSG-6) (?SEQ-MD SEQ-
MODEL) (?SENDER BUYER-PATH)) ((?MSG-WITHOUT-OWNER AG-MSG-7) (?SEQ-MD SEQ-MODEL) (?SENDER
BUYER-PATH)))
(RETRIEVE (?MSG-WITHOUT-PRTCL ?AG-ROLE-SENDER ?SEQ-MD) (AND (?MSG-WITHOUT-PRTCL AGENT-MSG)
(?AG-ROLE-SENDER AGENT-ROLE-CLASS-PATH) (?SEQ-MD SEQUENCE-MODEL) (?SEQ-MD ?MSG-WITHOUT-PRTCL
HAS-MSG) (?AG-ROLE-SENDER ?MSG-WITHOUT-PRTCL IS-MSG-SENDER) (NOT (?MSG-WITHOUT-PRTCL (SOME
(INV HAS-MSG) PROTOCOL))))) --> (((?MSG-WITHOUT-PRTCL AGENT-MSG-6) (?AG-ROLE-SENDER BUYER-
PATH) (?SEQ-MD SEQ-MODEL)))
(RETRIEVE (?MSG-WITHOUT-PRTCL ?AG-ROLE-SENDER ?SEQ-MD) (AND (?MSG-WITHOUT-PRTCL AGENT-MSG)
(?AG-ROLE-SENDER AGENT-ROLE-CLASS-PATH) (?SEQ-MD SEQUENCE-MODEL) (?PRTCL PROTOCOL) (?SEQ-MD
?MSG-WITHOUT-PRTCL HAS-MSG) (?SEQ-MD ?AG-ROLE-SENDER HAS-PATH) (?AG-ROLE-SENDER ?MSG-WITHOUT-
PRTCL IS-MSG-SENDER) (?AG-ROLE-SENDER ?AG-ROLE HAS-HEAD) (?AG-ROLE ?PRTCL HAS-PROTOCOL) (NOT
(?PRTCL ?MSG-WITHOUT-PRTCL HAS-MSG)))) --> (((?MSG-WITHOUT-PRTCL AGENT-MSG-6) (?AG-ROLE-SENDER
BUYER-PATH) (?SEQ-MD SEQ-MODEL)) ((?MSG-WITHOUT-PRTCL AG-MSG-7) (?AG-ROLE-SENDER BUYER-PATH)
(?SEQ-MD SEQ-MODEL)))
(RETRIEVE (?SEQ-MD ?PLAN-WITHOUT-OWNER) (AND (?SEQ-MD SEQUENCE-MODEL) (?PLAN-WITHOUT-OWNER
PLAN) (?SEQ-MD ?PLAN-WITHOUT-OWNER HAS-PLAN) (NOT (?SEQ-MODEL (SOME (INV HAS-PATH) (OR AGENT-
CLASS-PATH ORGANIZATION-CLASS-PATH ACTIVE-ENVIRONMENT-CLASS-PATH)))))) --> (((?SEQ-MD SEQ-
MODEL) (?PLAN-WITHOUT-OWNER PLAN-2)))
(RETRIEVE (?PATH-WITHOUT-MODEL) (AND (?PATH-WITHOUT-MODEL PATH) (NOT (?PATH-WITHOUT-MODEL
(SOME (INV HAS-PATH) SEQUENCE-MODEL))))) --> (((?PATH-WITHOUT-MODEL STORE-AGENT-PATH))
((?PATH-WITHOUT-MODEL MAGAZINE-PATH)) ((?PATH-WITHOUT-MODEL IMPORTED-BOOKSTORE-PATH)) ((?PATH-
WITHOUT-MODEL SELLER-PATH)) ((?PATH-WITHOUT-MODEL VIRTUAL-MARKET-PATH)) ((?PATH-WITHOUT-MODEL
SOMMERVILLE2001-PATH)) ((?PATH-WITHOUT-MODEL COMM-ACM-PATH)) ((?PATH-WITHOUT-MODEL VENDOR-
AMERICANAS-PATH)) ((?PATH-WITHOUT-MODEL INTL-BOOKSTORE-DOTCOM-PATH)) ((?PATH-WITHOUT-MODEL
AMERICANAS-DOTCOM-PATH)) ((?PATH-WITHOUT-MODEL SPECIAL-OFFER-PATH)) ((?PATH-WITHOUT-MODEL
INTL-BOOK-SELLER-PATH)) ((?PATH-WITHOUT-MODEL INTL-BOOK-BUYER-PATH)) ((?PATH-WITHOUT-MODEL
AMERICANAS-SELLER-PATH)))
(RETRIEVE (?SEQ-WITHOUT-STRUCT) (AND (?SEQ-WITHOUT-STRUCT SEQUENCE-MODEL) (?SEQ-WITHOUT-STRUCT
NIL HAS-STRUCTURE))) --> (((?SEQ-WITHOUT-STRUCT SEQ-MODEL)))
(RETRIEVE (?SEQ-MD-LOOP ?ELEMENT1 ?ELEMENT2) (AND (?SEQ-MD SEQUENCE-MODEL) (?SEQ SEQUENCE)
(?SEQ-MD-LOOP ?SEQ HAS-STRUCTURE) (?SEQ ?ELEMENT1 HAS-FIRST) (?SEQ ?ELEMENT2 HAS-LAST)
(?ELEMENT2 ?ELEMENT1 IS-BEFORE))) --> (((?SEQ-MD-LOOP PROTOCOL-1) (?ELEMENT1 AGENT-MSG-3)
(?ELEMENT2 AGENT-MSG-1)))
```

Código da base de conhecimento usada na fase F2

```
(in-abox phase2-ABox MASML-2-TBox)
;; class-model
(instance class-diagram class-model)
;; organization-model
(instance org1-diagram organization-model)
;; sequence-model
(instance seq-model sequence-model)
(instance seq-model-struct sequence)
(instance seq-model-struct-2 sequence)
;; virtual-market is an environment-class
(instance virtual-market passive-environment-class)
;; general-store, imported-bookstore, 2nd-hand-bookstore are organization-classes
(instance general-store main-organization-class)
(instance imported-bookstore sub-organization-class)
(instance 2nd-hand-bookstore sub-organization-class)
;; user-agent and store-agent are agent-classes
(instance user-agent agent-class)
(instance store-agent agent-class)
(instance super-user agent-class)
;; item and book are object-classes
(instance item object-class)
(instance book object-class)
(instance magazine object-class)
;; role-model
(instance role-diagram role-model)
;; roles
(instance seller agent-role-class)
(instance buyer agent-role-class)
(instance offer object-role-class)
(instance desire object-role-class)
(instance mediator agent-role-class)
(instance market-of-used-goods agent-role-class)
(instance market-of-special-goods agent-role-class)
;;;;;;; teste protocol
(related market-of-used-goods negotiate-prtcl has-protocol)
(related market-of-special-goods negotiate-prtcl has-protocol)
;; association
(instance assoc-2 association)
(related assoc-2 book has-end1)
(related assoc-2 magazine has-end2)
(instance assoc-3 association)
(related assoc-3 book has-end1)
(related assoc-3 desire has-end2)
(instance assoc-5 association)
(related assoc-5 buyer has-end1)
(related assoc-5 desire has-end2)
(instance assoc-6 association)
(related assoc-6 seller has-end1)
(related assoc-6 offer has-end2)
(instance org1-diagram organization-model)
(instance general-store main-organization-class)
(instance item object-class)
(instance book object-class)
(instance spec-1 specialization)
(related spec-1 book has-end1)
(related spec-1 item has-end2)
(related org1-diagram general-store has-class)
(related org1-diagram item has-class)
(related org1-diagram book has-class)
(related org1-diagram spec-1 has-relationship)
(instance inhabit-1 inhabit)
(related inhabit-1 user-agent has-end1)
(related inhabit-1 virtual-market has-end2)
(instance inhabit-2 inhabit)
(related inhabit-2 store-agent has-end1)
(related inhabit-2 virtual-market has-end2)
(instance inhabit-3 inhabit)
(related inhabit-3 general-store has-end1)
(related inhabit-3 virtual-market has-end2)
(instance inhabit-4 inhabit)
(related inhabit-4 2nd-hand-bookstore has-end1)
(related inhabit-4 virtual-market has-end2)
(instance inhabit-4 inhabit)
(related inhabit-4 imported-bookstore has-end1)
(related inhabit-4 virtual-market has-end2)
(instance inhabit-5 inhabit)
(related inhabit-5 book has-end1)
(related inhabit-5 virtual-market has-end2)
(instance inhabit-6 inhabit)
(related inhabit-6 item has-end1)
(related inhabit-6 virtual-market has-end2)
(instance spec-1 specialization)
(related spec-1 book has-end1)
(related spec-1 item has-end2)
(instance own-1 ownership)
(related own-1 general-store has-end1)
(related own-1 buyer has-end2)
(instance own-2 ownership)
(related own-2 general-store has-end1)
(related own-2 seller has-end2)
(instance own-3 ownership)
(related own-3 general-store has-end1)
(related own-3 desire has-end2)
(instance own-4 ownership)
```

```
(related own-4 general-store has-end1)
(related own-4 offer has-end2)
(instance own-5 ownership)
(related own-5 general-store has-end1)
(related own-5 market-special-goods has-end2)
(instance own-6 ownership)
(related own-6 general-store has-end1)
(related own-6 market-special-goods has-end2)
(instance play-1 play)
(related play-1 user-agent has-end1)
(related play-1 buyer has-end2)
(instance play-2 play)
(related play-2 store-agent has-end1)
(related play-2 seller has-end2)
(instance play-3 play)
(related play-3 book has-end1)
(related play-3 desire has-end2)
(instance play-4 play)
(related play-4 book has-end1)
(related play-4 offer has-end2)
(instance play-5 play)
(related play-5 imported-bookstore has-end1)
(related play-5 market-special-goods has-end2)
(instance play-6 play)
(related play-6 2nd-hand-bookstore has-end1)
(related play-6 market-used-goods has-end2)
(related org1-diagram general-store has-class)
(related org1-diagram user-agent has-class)
(related org1-diagram store-agent has-class)
(related org1-diagram imported-bookstore has-class)
(related org1-diagram 2nd-hand-bookstore has-class)
(related org1-diagram virtual-market has-class)
(related org1-diagram item has-class)
(related org1-diagram book has-class)
(related org1-diagram seller has-class)
(related org1-diagram buyer has-class)
(related org1-diagram offer has-class)
(related org1-diagram desire has-class)
(related org1-diagram market-used-goods has-class)
(related org1-diagram market-special-goods has-class)
(related org1-diagram inhabit-1 has-relationship)
(related org1-diagram inhabit-2 has-relationship)
(related org1-diagram inhabit-3 has-relationship)
(related org1-diagram inhabit-4 has-relationship)
(related org1-diagram inhabit-5 has-relationship)
(related org1-diagram inhabit-6 has-relationship)
(related org1-diagram play-1 has-relationship)
(related org1-diagram play-2 has-relationship)
(related org1-diagram play-3 has-relationship)
(related org1-diagram play-4 has-relationship)
(related org1-diagram play-5 has-relationship)
(related org1-diagram play-6 has-relationship)
(related org1-diagram own-1 has-relationship)
(related org1-diagram own-2 has-relationship)
(related org1-diagram own-3 has-relationship)
(related org1-diagram own-4 has-relationship)
(related org1-diagram own-5 has-relationship)
(related org1-diagram own-6 has-relationship)
;;;
(instance negotiate-prtcl protocol)
(instance struct-negotiate sequence)
(instance seq-md-1 sequence-model)
(related seller negotiate-prtcl has-protocol)
(related buyer negotiate-prtcl has-protocol)
(instance buyer-path agent-role-class-path)
(related buyer-path buyer has-head)
(instance seller-path agent-role-class-path)
(related seller-path seller has-head)
(related seq-md-1 struct-negotiate has-structure)
(related negotiate-prtcl struct-negotiate has-structure)
(related seq-md-1 buyer-path has-path)
(related seq-md-1 seller-path has-path)
(instance assoc-5 association)
(related assoc-5 buyer has-end1)
(related assoc-5 desire has-end2)
(instance assoc-6 association)
(related assoc-6 seller has-end1)
(related assoc-6 offer has-end2)
(instance assoc-7 association)
;(related assoc-7 buyer has-end1)
;(related assoc-7 seller has-end2)
;; ROLE-MODEL-RELATIONSHIPS
(related role-diagram assoc-5 has-relationship)
;(related role-diagram assoc-7 has-relationship)
(related role-diagram assoc-6 has-relationship)
(related role-diagram ctrl-4 has-relationship)
(related role-diagram ctrl-5 has-relationship)
;; ROLE-MODEL-CLASS
(related role-diagram offer has-class)
(related role-diagram desire has-class)
(related role-diagram seller has-class)
(related role-diagram buyer has-class)
(related role-diagram market-of-special-goods has-class)
(related role-diagram market-of-used-goods has-class)
(check-abox-coherence)
```

Respostas às consultas QV2

```
(IN-TBOX MASML-2-TBOX) --> MASML-2-TBOX
(TBOX-COHERENT-P) --> T
(TBOX-CYCLIC-P) --> NIL
(IN-ABOX PHASE2-ABOX MASML-2-TBOX) --> PHASE2-ABOX
(CHECK-ABOX-COHERENCE) --> T
(RETRIEVE (?AG-CLASS ?CLASS-DIAGRAM ?ORGMD) (AND (?AG-CLASS AGENT-CLASS) (?CLASS-DIAGRAM
CLASS-MODEL) (?ORGMD ORGANIZATION-MODEL) (?CLASS-DIAGRAM ?AG-CLASS HAS-CLASS) (NOT (?ORG-MD
?AG-CLASS HAS-CLASS)))) --> (((?AG-CLASS USER-AGENT) (?CLASS-DIAGRAM EC-CLASS-MODEL) (?ORGMD
THE-ORGANIZING-MODEL)))
(RETRIEVE (?ORG-CLASS ?CLASS-DIAGRAM ?ORGMD) (AND (?ORG-CLASS ORGANIZATION-CLASS) (?CLASS-
DIAGRAM CLASS-MODEL) (?ORGMD ORGANIZATION-MODEL) (?CLASS-DIAGRAM ?ORG-CLASS HAS-CLASS) (NOT
(?ORGMD ?ORG-CLASS HAS-CLASS)))) --> (((?ORG-CLASS REVIEWERS-BOARD) (?CLASS-DIAGRAM EC-CLASS-
MODEL) (?ORGMD THE-ORGANIZING-MODEL)))
(RETRIEVE (?OBJ-CLASS ?ORG-DIAGRAM ?CLMD) (AND (?OBJ-CLASS OBJECT-CLASS) (?ORG-DIAGRAM
ORGANIZATION-MODEL) (?ORG-DIAGRAM ?OBJ-CLASS HAS-CLASS) (?CLMD CLASS-MODEL) (NOT (?CLMD ?OBJ-
CLASS HAS-CLASS)))) --> (((?OBJ-CLASS EVAL-FORM) (?ORG-DIAGRAM THE-ORGANIZING-MODEL) (?CLMD
EC-CLASS-MODEL)))
(RETRIEVE (?RCLASS ?ROLE-DIAGRAM ?ORGMD) (AND (?RCLASS ROLE-CLASS) (?ROLE-DIAGRAM ROLE-MODEL)
(?ROLE-DIAGRAM ?RCLASS HAS-CLASS) (?ORGMD ORGANIZATION-MODEL) (NOT (?ORGMD ?RCLASS HAS-
CLASS)))) --> (((?RCLASS CO-CHAIR) (?ROLE-DIAGRAM THE-ROLE-MODEL) (?ORGMD THE-ORGANIZING-
MODEL)))
(RETRIEVE (?OBJ-CLASS ?ROLE-DIAGRAM ?CLMD) (AND (?OBJ-CLASS OBJECT-CLASS) (?ROLE-DIAGRAM ROLE-
MODEL) (?ROLE-DIAGRAM ?OBJ-CLASS HAS-CLASS) (?CLMD CLASS-MODEL) (NOT (?CLMD ?OBJ-CLASS HAS-
CLASS)))) --> (((?OBJ-CLASS EVAL-FORM) (?ROLE-DIAGRAM THE-ROLE-MODEL) (?CLMD EC-CLASS-MODEL)))
(RETRIEVE (?IPATH ?SEQMD ?HEAD) (AND (?IPATH INSTANCE-PATH) (?SEQMD SEQUENCE-MODEL) (?SEQMD
?IPATH HAS-PATH) (?IPATH ?HEAD HAS-HEAD) (?HEAD NIL IS-INSTANCEOF))) --> (((?IPATH PAPER01-
PATH) (?SEQMD SEQ-1) (?HEAD PAPER-01)))
```

Apêndice G
Resultados do estudo de caso Expert Committee usando o método aplicado a MAS-ML

Código da base de conhecimento usada na fase F1

Nesta seção encontra-se o código da base de conhecimento usada no estudo de caso sobre o Expert Committee, usando a aplicação do método para a linguagem MAS-ML. Esta base refere-se à execução da fase F1 do Observed-MAS e possui erros de modelagem, cuja introdução foi deliberada.

```
(in-abox phase1-ABox MASML-TBox)

; creating the environment of the EC system
(instance expert-commitee passive-environment-class)
(instance expert-commitee-path passive-environment-class-path)
(related expert-commitee-path expert-commitee has-head)

; creating the main-organization of the EC System
(instance organizing-institution main-organization-class)
(instance organizing-institution-path main-organization-class-path)
(related organizing-institution-path organizing-institution has-head)
(related organizing-institution-path expert-commitee-path has-tail)

;;;;;;;;;; internal state of the main organization

; creating the main-organization goals
(instance start-event goal)
(related organizing-institution start-event has-goal)

; creating the main-organization axioms
(instance dont-play-CA axiom)
(instance dont-play-CHCM axiom)
(instance dont-play-CR axiom)
(instance dont-play-CCH axiom)
(instance reviewer-not-author axiom)
(instance pcmember-not-author axiom)
(instance dont-play-CMC axiom)

(related organizing-institution dont-play-CA has-axiom)
(related organizing-institution dont-play-CHCM has-axiom)
(related organizing-institution dont-play-CR has-axiom)
(related organizing-institution dont-play-CCH has-axiom)
(related organizing-institution reviewer-not-author has-axiom)
(related organizing-institution pcmember-not-author has-axiom)
(related organizing-institution dont-play-CMC has-axiom)


; creating the main-organization actions
(instance to-create-roles action)
(instance to-create-reviewer action)
(instance to-create-PCmember action)
(instance to-create-PCchair action)
(instance to-create-PCmanager action)
(instance to-create-agenda action)

(related organizing-institution to-create-roles has-action)
(related organizing-institution to-create-reviewer has-action)
(related organizing-institution to-create-PCmember has-action)
(related organizing-institution to-create-PCchair has-action)
(related organizing-institution to-create-PCmanager has-action)
(related organizing-institution to-create-agenda has-action)

; creating the main-organization plans
```

```
(instance start-event-plan plan)
(related organizing-institution start-event-plan has-plan)
(instance start-event-plan-struct sequence)
(related start-event-plan start-event-plan-struct has-structure)

(related start-event-plan to-create-PCmanager has-action)
(related start-event-plan to-create-PCmember has-action)
(related start-event-plan to-create-reviewer has-action)
(related start-event-plan to-create-roles has-action)
(related start-event-plan to-create-PCchair has-action)

(related start-event-plan-struct to-create-PCmanager has-first)
(related start-event-plan-struct to-create-PCchair has-last)

(related start-event-plan-struct to-create-PCmanager has-element)
(related start-event-plan-struct to-create-PCmember has-element)
(related start-event-plan-struct to-create-reviewer has-element)
(related start-event-plan-struct to-create-roles has-element)
(related start-event-plan-struct to-create-PCchair has-element)

(related to-create-PCmanager to-create-PCmember is-before)
(related to-create-PCmember to-create-reviewer is-before)
(related to-create-reviewer to-create-roles is-before)
(related to-create-roles to-create-PCchair is-before)

;;;;;;;;;;;;;;;;;;;;;;;;;end internal state main-org

; creating roles that are owned by the main-organization
(instance reviewer agent-role-class)
(instance reviewer-path agent-role-class-path)
(related reviewer-path reviewer has-head)
(related reviewer-path organizing-institution-path has-tail)

(instance chair agent-role-class)
(instance chair-path agent-role-class-path)
(related chair-path chair has-head)
(related organizing-institution-path chair-path has-tail)

(instance manager agent-role-class)
(instance manager-path agent-role-class-path)
(related manager-path manager has-head)
(related manager-path organizing-institution-path has-tail)

(instance pc-member agent-role-class)
(instance pc-member-path agent-role-class-path)
(related pc-member-path pc-member has-head)
(related pc-member-path organizing-institution-path has-tail)

(instance author agent-role-class)
(instance author-path agent-role-class-path)
(related author-path author has-head)
(related author-path organizing-institution-path has-tail)

; creating the main-organization's ownership relationships
(instance own-1 ownership)
(instance own-2 ownership)
(instance own-3 ownership)
(instance own-4 ownership)
(instance own-5 ownership)

; resources of the system
(instance paper object-class)
(instance presentation object-class)
(instance proposition object-class)
(instance result object-class)
(instance vote object-class)
(instance conflict object-class)
(instance camera-ready object-class)

; creating the relationships between objects
(instance assoc-1-has association)
(instance assoc-2-has association)
(instance assoc-3-has association)
(instance assoc-4-has association)
(instance assoc-conflict association)

(instance dep-1 dependency)

(instance inhabit-1 inhabit)
(instance inhabit-2 inhabit)
(instance inhabit-3 inhabit)
(instance inhabit-4 inhabit)
(instance inhabit-5 inhabit)
(instance inhabit-6 inhabit)
(instance inhabit-7 inhabit)
(instance inhabit-8 inhabit)

; agents of the system
```

```
(instance researcher-agent agent-class)
(instance researcher-agent-path agent-class-path)
(related researcher-agent-path researcher-agent has-head)
(related researcher-agent-path reviewer-path has-tail)
(related researcher-agent-path chair-path has-tail)
(related researcher-agent-path manager-path has-tail)
(related researcher-agent-path pc-member-path has-tail)
(related researcher-agent-path author-path has-tail)


; creating the researcher goals
(instance receive-camera-ready goal)
(instance submit-paper goal)
(instance distribute-paper goal)
(instance receive-final-results goal)
(instance review-paper goal)
(instance receive-paper goal)
(instance solve-conflict goal)
(instance accept-paper goal)
(instance present-final-results goal)
(instance vote-to-solve goal)
(instance make-review goal)
(instance send-camera-ready goal)
(instance engage-reviewer goal)
(instance provide-reviewer goal)

(related researcher-agent receive-camera-ready has-goal)
(related researcher-agent submit-paper has-goal)
(related researcher-agent distribute-paper has-goal)
(related researcher-agent receive-final-results has-goal)
(related researcher-agent review-paper has-goal)
(related researcher-agent receive-paper has-goal)
(related researcher-agent solve-conflict has-goal)
(related researcher-agent accept-paper has-goal)
(related researcher-agent present-final-results has-goal)
(related researcher-agent vote-to-solve has-goal)
(related researcher-agent make-review has-goal)
(related researcher-agent send-camera-ready has-goal)
(related researcher-agent engage-reviewer has-goal)
(related researcher-agent provide-reviewer has-goal)

; creating the researcher actions
(instance to-make-additional-relation action)
(instance to-make-presentation action)
(instance to-select-new-reviewer action)
(instance to-register-vote action)
(instance to-search-for-conflict action)
(instance to-vote action)
(instance to-register-result action)
(instance to-make-final-result action)
(instance to-review action)
(instance to-prepare-paper-pack action)
(instance to-check-less-then-3-reviewer action)
(instance to-make-camera-ready action)
(instance to-make-paper action)
(instance to-register-paper-and-presentation action)
(instance to-select-reviewer action)
(instance to-make-review-proposition action)
(instance to-refresh-reviewer action)
(instance to-register-camera-ready action)
(instance to-assess-review-proposition action)

(related researcher-agent to-make-additional-relation has-action)
(related researcher-agent to-make-presentation has-action)
(related researcher-agent to-select-new-reviewer has-action)
(related researcher-agent to-register-vote has-action)
(related researcher-agent to-search-for-conflict has-action)
(related researcher-agent to-vote has-action)
(related researcher-agent to-register-result has-action)
(related researcher-agent to-make-final-result has-action)
(related researcher-agent to-review has-action)
(related researcher-agent to-prepare-paper-pack has-action)
(related researcher-agent to-check-less-then-3-reviewer has-action)
(related researcher-agent to-make-camera-ready has-action)
(related researcher-agent to-register-paper-and-presentation has-action)
(related researcher-agent to-select-reviewer has-action)
(related researcher-agent to-make-review-proposition has-action)
(related researcher-agent to-make-paper has-action)
(related researcher-agent to-refresh-reviewer has-action)
(related researcher-agent to-register-camera-ready has-action)
(related researcher-agent to-assess-review-proposition has-action)

; creating the researcher plans
(instance receive-camera-ready-plan plan)
(instance send-camera-ready-plan plan)
(instance submit-paper-plan plan)
(instance receive-paper-plan plan)
(instance assign-paper-plan plan)
```

```
(instance accept-paper-plan plan)
(instance provide-reviewer-plan plan)
(instance engage-reviewer-plan plan)
(instance make-review-plan plan)
(instance review-plan plan)
(instance solve-conflict-plan plan)
(instance vote-conflict-plan plan)
(instance show-final-result-plan plan)

; assigning the researcher's plans to it
(related researcher-agent receive-camera-ready-plan has-plan)
(related researcher-agent send-camera-ready-plan has-plan)
(related researcher-agent submit-paper-plan has-plan)
(related researcher-agent assign-paper-plan has-plan)
(related researcher-agent receive-paper-plan has-plan)
(related researcher-agent accept-paper-plan has-plan)
(related researcher-agent provide-reviewer-plan has-plan)
(related researcher-agent engage-reviewer-plan has-plan)
(related researcher-agent make-review-plan has-plan)
(related researcher-agent review-plan has-plan)
(related researcher-agent solve-conflict-plan has-plan)
(related researcher-agent vote-conflict-plan has-plan)
(related researcher-agent show-final-result-plan has-plan)

;; turning the researcher into a citizen
(instance inhabit-9 inhabit)
(related researcher inhabit-9 is-end1)
(related expert-commitee  inhabit-9 is-end2)

; structuring the researcher plans
(instance accept-paper-plan-struct sequence)
(related accept-paper-plan accept-paper-plan-struct has-structure)

(instance submit-paper-plan-struct sequence)
(related submit-paper-plan submit-paper-plan-struct has-structure)

(instance review-plan-struct sequence)
(related review-plan review-plan-struct has-structure)

(instance receive-paper-plan-struct sequence)
(related receive-paper-plan receive-paper-plan-struct has-structure)

(instance make-review-plan-struct sequence)
(related make-review-plan make-review-plan-struct has-structure)

(instance engage-reviewer-plan-struct sequence)
(related engage-reviewer-plan engage-reviewer-plan-struct has-structure)

(instance provide-reviewer-plan-struct sequence)
(related provide-reviewer-plan provide-reviewer-plan-struct has-structure)

(instance receive-camera-ready-plan-struct sequence)
(related receive-camera-ready-plan receive-camera-ready-plan-struct has-structure)

(instance send-camera-ready-plan-struct sequence)
(related send-camera-ready-plan send-camera-ready-plan-struct has-structure)

(instance show-final-result-plan-struct sequence)
(related show-final-result-plan show-final-result-plan-struct has-structure)

(instance solve-conflict-plan-struct sequence)
(related solve-conflict-plan solve-conflict-plan-struct has-structure)

(instance vote-conflict-plan-struct sequence)
(related vote-conflict-plan vote-conflict-plan-struct has-structure)

; creating the researcher beliefs (no beliefs assigned)

; creating the play relationships the researcher plays
(instance play-1 play)
(instance play-2 play)
(instance play-3 play)
(instance play-4 play)
(instance play-5 play)

; creating the internal structure of the agent-roles


; agent-role beliefs and goals (agent-role goals are the
; same of the researcher agent goals)

(instance review-proposal-belief belief)
(instance package-list-belief belief)
(instance paper-belief belief)
(instance paper-list-belief belief)
(instance result-belief belief)
(instance review-belief belief)
(instance assessment-belief belief)
```

```
(related reviewer review-proposal-belief has-belief)
(related reviewer package-list-belief has-belief)
(related reviewer paper-list-belief has-belief)
(related reviewer result-belief has-belief)
(related reviewer review-belief has-belief)
(related reviewer assessment-belief has-belief)

(related reviewer accept-paper has-goal)
(related reviewer review-paper has-goal)

(instance presentation-belief belief)
(instance final-result-belief belief)
(instance final-result-list-belief belief)
(instance camera-ready-belief belief)
(instance reviewers-list-belief belief)
(instance chair-reviewers-list-belief belief)
(instance presentation-list-belief belief)
(instance deadline-reviewer-belief belief)
(instance review-proposal-list-belief belief)
(instance result-list-belief belief)
(instance deadline-paper-belief belief)
(instance deadline-camera-ready-belief belief)
(instance review-acceptance-belief belief)
(instance voting-list-belief belief)
(instance camera-ready-list-belief belief)
(instance additional-relation-belief belief)

(related author presentation-belief has-belief)
(related author paper-belief has-belief)
(related author review-belief has-belief)
(related author final-result-belief has-belief)
(related author camera-ready-belief has-belief)

(related author submit-paper has-goal)
(related author receive-final-results has-goal)
(related author send-camera-ready has-goal)

(related chair chair-reviewers-list-belief has-belief)
(related chair paper-list-belief has-belief)
(related chair presentation-list-belief has-belief)
(related chair deadline-reviewer-belief has-belief)
(related chair review-proposal-list-belief has-belief)
(related chair review-acceptance-belief has-belief)
(related chair package-list-belief has-belief)
(related chair result-list-belief has-belief)
(related chair voting-list-belief has-belief)
(related chair camera-ready-list-belief has-belief)
(related chair final-result-list-belief has-belief)
(related chair additional-relation-belief has-belief)
(related chair deadline-paper-belief has-belief)
(related chair deadline-camera-ready-belief has-belief)

(related chair receive-paper has-goal)
(related chair distribute-paper has-goal)
(related chair engage-reviewer has-goal)
(related chair make-review has-goal)
(related chair solve-conflict has-goal)
(related chair present-final-results has-goal)
(related chair receive-camera-ready has-goal)

(instance reviewers-list-belief belief)

(related manager reviewers-list-belief  has-belief)

(related manager provide-reviewer has-goal)

(instance voting-belief belief)
(instance conflict-belief belief)

(related pc-member voting-belief has-belief)
(related pc-member conflict-belief has-belief)

(related pc-member solve-conflict has-goal)

; agent-role duties
(instance duty-make-additional-relation duty)
(instance duty-make-presentation duty)
(instance duty-select-new-reviewer duty)
(instance duty-register-vote duty)
(instance duty-search-for-conflict duty)
(instance duty-vote duty)
(instance duty-register-result duty)
(instance duty-make-final-result duty)
(instance duty-review duty)
(instance duty-prepare-paper-pack duty)
(instance duty-check-less-then-3-reviewer duty)
(instance duty-make-camera-ready duty)
```

```
(instance duty-make-paper duty)
(instance duty-register-paper-and-presentation duty)
(instance duty-select-reviewer duty)
(instance duty-make-review-proposition duty)
(instance duty-refresh-reviewer duty)
(instance duty-register-camera-ready duty)
(instance duty-assess-review-proposition duty)
(instance duty-provide-reviewer duty)

(related chair duty-register-paper-and-presentation has-duty)
(related chair duty-select-reviewer has-duty)
(related chair duty-make-review-proposition has-duty)
(related chair duty-refresh-reviewer has-duty)
(related chair duty-check-less-then-3-reviewer has-duty)
(related chair duty-prepare-paper-pack has-duty)
(related chair duty-register-result has-duty)
(related chair duty-search-for-conflict has-duty)
(related chair duty-register-vote has-duty)
(related chair duty-register-camera-ready has-duty)
(related chair duty-make-final-result has-duty)

(related manager duty-select-new-reviewer has-duty)
(related manager duty-make-additional-relation has-duty)

(related pc-member duty-vote has-duty)

(related author duty-make-paper has-duty)
(related author duty-make-presentation has-duty)
(related author duty-make-camera-ready has-duty)

(related reviewer duty-assess-review-proposition has-duty)
(related reviewer duty-review has-duty)

; agent-role protocols
(instance send-review-proposal-prtcl protocol)
(instance inform-review-acceptance-prtcl protocol)
(instance send-pack-to-reviewer-prtcl protocol)
(instance send-result-prtcl protocol)
(instance submit-paper-and-presentation-prtcl protocol)
(instance send-final-result-and-review-prtcl protocol)
(instance send-camera-ready-prtcl protocol)
(instance engage-more-reviewer-prtcl protocol)
(instance solve-conflict-with-pc-prtcl protocol)

(related reviewer send-review-proposal-prtcl has-protocol)
(related reviewer inform-review-acceptance-prtcl has-protocol)
(related reviewer send-pack-to-reviewer-prtcl has-protocol)
(related reviewer send-result-prtcl has-protocol)

(related author submit-paper-and-presentation-prtcl has-protocol)
(related author send-final-result-and-review-prtcl has-protocol)
(related author send-camera-ready-prtcl has-protocol)

(related manager engage-more-reviewer-prtcl has-protocol)

(related chair submit-paper-and-presentation-prtcl has-protocol)
(related chair send-review-proposal-prtcl has-protocol)
(related chair inform-review-acceptance-prtcl has-protocol)
(related chair send-pack-to-reviewer-prtcl has-protocol)
(related chair send-result-prtcl has-protocol)
(related chair solve-conflict-with-pc-prtcl has-protocol)
(related chair send-final-result-and-review-prtcl has-protocol)
(related chair send-camera-ready-prtcl has-protocol)
(related chair engage-more-reviewer-prtcl has-protocol)

(related pc-member solve-conflict-with-pc-prtcl has-protocol)

;; agent-role msgs
(instance send-proposal-msg agent-msg)
(instance send-final-result-msg agent-msg)
(instance send-pack-msg agent-msg)
(instance send-result-msg agent-msg)
(instance send-request-msg agent-msg)
(instance send-relation-msg agent-msg)
(instance send-conflict-msg agent-msg)
(instance send-vote-msg agent-msg)
(instance send-msg agent-msg)
(instance inform-msg agent-msg)
(instance inform-deadline-msg agent-msg)
(instance submit-msg agent-msg)

(related chair send-conflict-msg has-msg)
(related chair send-proposal-msg has-msg)
(related chair send-final-result-msg has-msg)
(related chair send-pack-msg has-msg)
(related chair send-result-msg has-msg)
(related chair send-request-msg has-msg)
(related chair send-relation-msg has-msg)
```

```
(related chair submit-msg has-msg)
(related chair send-vote-msg has-msg)
(related chair send-msg has-msg)
(related chair inform-msg has-msg)
(related chair inform-deadline-msg has-msg)

(related manager send-request-msg has-msg)
(related manager send-relation-msg has-msg)

(related pc-member send-conflict-msg has-msg)
(related pc-member send-vote-msg has-msg)

(related reviewer send-proposal-msg has-msg)
(related reviewer send-pack-msg has-msg)
(related reviewer send-result-msg has-msg)
(related reviewer inform-msg has-msg)

(related author submit-msg has-msg)
(related author send-final-result-msg has-msg)
(related author inform-deadline-msg has-msg)
(related author send-msg has-msg)

;; structuring the agent-roles protocols
(instance submit-struct sequence)
(related submit-paper-and-presentation-prtcl submit-struct has-structure)
(related submit-paper-and-presentation-prtcl submit-msg has-msg)
(related submit-struct submit-msg has-element)
(related submit-msg author-path has-msg-sender)
(related submit-msg chair-path has-msg-receiver)
(related submit-struct submit-msg has-first)
(related submit-struct submit-msg has-last)

(instance send-proposal-struct sequence)
(related send-review-proposal-prtcl send-proposal-struct has-structure)
(related send-review-proposal-prtcl send-proposal-msg has-msg)
(related send-proposal-struct send-proposal-msg has-element)
(related send-proposal-msg chair-path has-msg-sender)
(related send-proposal-msg reviewer-path has-msg-receiver)
(related send-proposal-struct send-proposal-msg has-first)
(related send-proposal-struct send-proposal-msg has-last)

(instance inform-acceptance-struct sequence)
(related inform-review-acceptance-prtcl inform-acceptance-struct has-structure)
(related inform-review-acceptance-prtcl inform-msg has-msg)
(related inform-acceptance-struct inform-msg has-element)
(related inform-msg reviewer-path has-msg-sender)
(related inform-msg chair-path has-msg-receiver)
(related inform-acceptance-struct inform-msg has-first)
(related inform-acceptance-struct inform-msg has-last)

(instance send-result-struct sequence)
(related send-result-prtcl send-result-struct has-structure)
(related send-result-prtcl send-result-msg has-msg)
(related send-result-struct send-result-msg has-element)
(related send-result-msg reviewer-path has-msg-sender)
(related send-result-msg chair-path has-msg-receiver)
(related send-result-struct send-result-msg has-first)
(related send-result-struct send-result-msg has-last)

(instance send-pack-struct sequence)
(related send-pack-to-reviewer-prtcl send-pack-struct has-structure)
(related send-pack-to-reviewer-prtcl send-pack-msg has-msg)
(related send-pack-struct send-pack-msg has-element)
(related send-pack-msg chair-path has-msg-sender)
(related send-pack-msg reviewer-path has-msg-receiver)
(related send-pack-struct send-pack-msg has-first)
(related send-pack-struct send-pack-msg has-last)

(instance send-final-result-struct sequence)
(related send-final-result-and-review-prtcl send-final-result-struct has-structure)
(related send-final-result-and-review-prtcl send-final-result-msg has-msg)
(related send-final-result-struct send-final-result-msg has-element)
(related send-final-result-msg chair-path has-msg-sender)
(related send-final-result-msg author-path has-msg-receiver)
(related send-final-result-struct send-final-result-msg has-first)
(related send-final-result-struct send-final-result-msg has-last)

(instance send-camera-ready-struct sequence)
(related send-camera-ready-prtcl send-camera-ready-struct has-structure)
(related send-camera-ready-prtcl inform-deadline-msg has-msg)
(related send-camera-ready-prtcl send-msg has-msg)
(related send-camera-ready-struct inform-deadline-msg has-element)
(related send-camera-ready-struct send-msg has-element)
(related inform-deadline-msg chair-path has-msg-sender)
(related inform-deadline-msg author-path has-msg-receiver)
(related send-msg author-path has-msg-sender)
(related send-msg chair-path has-msg-receiver)
(related send-camera-ready-struct inform-deadline-msg has-first)
```

```
(related send-camera-ready-struct send-msg has-last)
(related inform-deadline-msg send-msg is-before)

(instance engage-reviewer-struct sequence)
(related engage-more-reviewer-prtcl engage-reviewer-struct has-structure)
(related engage-more-reviewer-prtcl send-request-msg has-msg)
(related engage-more-reviewer-prtcl send-relation-msg has-msg)
(related engage-reviewer-struct send-request-msg has-element)
(related engage-reviewer-struct send-relation-msg has-element)
(related send-request-msg chair-path has-msg-sender)
(related send-request-msg manager-path has-msg-receiver)
(related send-relation-msg manager-path has-msg-sender)
(related send-relation-msg chair-path has-msg-receiver)
(related engage-reviewer-struct send-request-msg has-first)
(related engage-reviewer-struct send-relation-msg has-last)
(related send-request-msg send-relation-msg is-before)

(instance solve-conflict-struct sequence)
(related solve-conflict-with-pc-prtcl solve-conflict-struct has-structure)
(related solve-conflict-with-pc-prtcl send-conflict-msg has-msg)
(related solve-conflict-with-pc-prtcl send-vote-msg has-msg)
(related solve-conflict-struct send-conflict-msg has-element)
(related solve-conflict-struct send-vote-msg has-element)
(related send-conflict-msg chair-path has-msg-sender)
(related send-conflict-msg pc-member-path has-msg-receiver)
(related send-vote-msg pc-member-path has-msg-sender)
(related send-vote-msg chair-path has-msg-receiver)
(related solve-conflict-struct send-conflict-msg has-first)
(related solve-conflict-struct send-vote-msg has-last)

; creating a class-model
(instance ec-class-model class-model)

; populating the ec-class-model
;; INTRODUCING INTRA-MODEL ERRORS
;; ERROR 1 : AN AGENT-ROLE-CLASS MODELED IN A CLASS DIAGRAM
(related ec-class-model chair has-class)

(related ec-class-model expert-commitee has-class)
(related ec-class-model organizing-institution has-class)
(related ec-class-model paper has-class)
(related ec-class-model presentation has-class)
(related ec-class-model camera-ready has-class)
(related ec-class-model conflict has-class)
(related ec-class-model result has-class)
(related ec-class-model proposition has-class)
(related ec-class-model vote has-class)

(related ec-class-model assoc-1-has has-relationship)
(related ec-class-model assoc-2-has has-relationship)
(related ec-class-model assoc-3-has has-relationship)
(related ec-class-model assoc-4-has has-relationship)
(related ec-class-model assoc-conflict has-relationship)

(related ec-class-model dep-1 has-relationship)

(related ec-class-model inhabit-1 has-relationship)
(related ec-class-model inhabit-2 has-relationship)
(related ec-class-model inhabit-3 has-relationship)
(related ec-class-model inhabit-4 has-relationship)
(related ec-class-model inhabit-5 has-relationship)
(related ec-class-model inhabit-6 has-relationship)
(related ec-class-model inhabit-7 has-relationship)
;(related ec-class-model inhabit-8 has-relationship)
(related assoc-1-has paper has-end1)
(related assoc-1-has camera-ready  has-end2)

(related assoc-2-has paper has-end1)
(related assoc-2-has proposition has-end2)

(related assoc-3-has paper has-end1)
(related assoc-3-has presentation has-end2)

(related assoc-4-has paper has-end1)
(related assoc-4-has result has-end2)

(related assoc-conflict vote has-end1)
(related assoc-conflict conflict has-end2)

(related dep-1 result has-end1)
(related dep-1 conflict has-end2)

(related inhabit-1 paper has-end1)
(related inhabit-1 expert-commitee has-end2)

(related inhabit-2 camera-ready has-end1)
(related inhabit-2 expert-commitee has-end2)
```

```
(related inhabit-3 presentation has-end1)
(related inhabit-3 expert-commitee has-end2)

(related inhabit-4 result has-end1)
(related inhabit-4 expert-commitee has-end2)

(related inhabit-5 proposition has-end1)
(related inhabit-5 expert-commitee has-end2)

(related inhabit-6 vote has-end1)
(related inhabit-6 expert-commitee has-end2)

(related inhabit-7 conflict has-end1)
(related inhabit-7 expert-commitee has-end2)

(related inhabit-8 organizing-institution has-end1)
(related inhabit-8 expert-commitee has-end2)

; creating an organization-model
(instance the-organizing-model organization-model)

(related the-organizing-model inhabit-8 has-relationship)

; populating the the-organizing-model
(related the-organizing-model reviewer has-class)
(related the-organizing-model chair has-class)
(related the-organizing-model manager has-class)
(related the-organizing-model pc-member has-class)
(related the-organizing-model author has-class)
(related the-organizing-model organizing-institution has-class)
(related the-organizing-model researcher-agent has-class)

(related the-organizing-model own-1 has-relationship)
(related the-organizing-model own-2 has-relationship)
(related the-organizing-model own-3 has-relationship)
(related the-organizing-model own-4 has-relationship)
(related the-organizing-model own-5 has-relationship)

(related the-organizing-model assoc-1 has-relationship)

(related the-organizing-model play-1 has-relationship)
(related the-organizing-model play-2 has-relationship)
(related the-organizing-model play-3 has-relationship)
(related the-organizing-model play-4 has-relationship)
(related the-organizing-model play-5 has-relationship)

(related own-1 organizing-institution has-end1)
(related own-1 reviewer has-end2)

(related own-2 organizing-institution has-end1)
(related own-2 chair has-end2)

(related own-3 organizing-institution has-end1)
(related own-3 manager has-end2)

(related own-4 organizing-institution has-end1)
(related own-4 pc-member has-end2)

(related own-5 organizing-institution has-end1)
(related own-5 author has-end2)

(related play-1 researcher-agent has-end1)
(related play-1 reviewer has-end2)

(related play-2 researcher-agent has-end1)
(related play-2 chair has-end2)

(related play-3 researcher-agent has-end1)
(related play-3 manager has-end2)

(related play-4 researcher-agent has-end1)
(related play-4 pc-member has-end2)

(related play-5 researcher-agent has-end1)
(related play-5 author has-end2)

; creating a role-model
(instance the-role-model role-model)

; populating the the-role-model
;; ERROR 2: AN AGENT-CLASS MODELED IN A ROLE-MODEL
(related the-role-model researcher-agent has-class)

;(related the-role-model reviewer has-class)
(related the-role-model chair has-class)
(related the-role-model manager has-class)
(related the-role-model pc-member has-class)
(related the-role-model author has-class)
```

```
(instance assoc-5 association)
(instance assoc-6 association)
(instance ctrl-1 control)
(instance ctrl-2 control)

(related the-role-model assoc-5 has-relationship)
(related the-role-model assoc-6 has-relationship)
(related the-role-model ctrl-1 has-relationship)
(related the-role-model ctrl-2 has-relationship)

(related assoc-5 chair has-end1)
(related assoc-5 manager has-end2)

(related assoc-6 chair has-end1)
(related assoc-6 author has-end2)

(related ctrl-1 chair has-end1)
(related ctrl-1 pc-member has-end2)

(related ctrl-2 chair has-end1)
(related ctrl-2 reviewer has-end2)

; creating sequence-models
(instance seq-1 sequence-model)
(related seq-1 submit-struct has-structure)
(related seq-1 author-path has-path)
(related seq-1 chair-path has-path)

(instance seq-2 sequence-model)
(related seq-2 send-proposal-struct has-structure)
(related seq-2 chair-path has-path)
(related seq-2 reviewer-path has-path)

(instance seq-3 sequence-model)
(related seq-3 inform-acceptance-struct has-structure)
(related seq-3 reviewer-path has-path)
(related seq-3 chair-path has-path)

(instance seq-4 sequence-model)
(related seq-4 send-result-struct has-structure)
(related seq-4 reviewer-path has-path)
(related seq-4 chair-path has-path)

(instance seq-5 sequence-model)
(related seq-5 send-pack-struct has-structure)
(related seq-5 chair-path has-path)
(related seq-5 reviewer-path has-path)

(instance seq-6 sequence-model)
(related seq-6 send-final-result-struct has-structure)
(related seq-6 chair-path has-path)
(related seq-6 author-path has-path)

(instance seq-7 sequence-model)
(related seq-7 send-camera-ready-struct has-structure)
(related seq-7 chair-path has-path)
(related seq-7 author-path has-path)

(instance seq-8 sequence-model)
(related seq-8 engage-reviewer-struct has-structure)
(related seq-8 chair-path has-path)
(related seq-8 manager-path has-path)

(instance seq-9 sequence-model)
(related seq-9 solve-conflict-struct has-structure)
(related seq-9 chair-path has-path)
(related seq-9 pc-member-path has-path)

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(check-abox-coherence)
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;
```

Respostas às consultas QV1

Respostas obtidas a partir das consultas QV1 aplicadas à base de conhecimento descrita na seção anterior.

```
(IN-KNOWLEDGE-BASE MASML-TBOX PHASE1-ABOX) --> (MASML-TBOX PHASE1-ABOX)
(IN-TBOX MASML-TBOX) --> MASML-TBOX
(TBOX-CLASSIFIED-P) --> T
(TBOX-COHERENT-P) --> T
(TBOX-CYCLIC-P) --> NIL
(IN-ABOX PHASE1-ABOX MASML-TBOX) --> PHASE1-ABOX
(CHECK-ABOX-COHERENCE) --> T
(RETRIEVE (?ROLECL ?CLMD) (AND (?ROLECL ROLE-CLASS) (?CLMD CLASS-MODEL) (?CLMD ?ROLECL HAS-
CLASS))) --> (((?ROLECL CHAIR) (?CLMD EC-CLASS-MODEL)))
(RETRIEVE (?PL ?CLMD) (AND (?PL PLAY) (?CLMD CLASS-MODEL) (?CLMD ?PL HAS-RELATIONSHIP))) -->
NIL
(RETRIEVE (?OWN ?CLMD) (AND (?OWN OWNERSHIP) (?CLMD CLASS-MODEL) (?CLMD ?OWN HAS-
RELATIONSHIP))) --> NIL
(RETRIEVE (?CTRL ?CLMD) (AND (?CTRL CONTROL) (?CLMD CLASS-MODEL) (?CLMD ?CTRL HAS-
RELATIONSHIP))) --> NIL
(RETRIEVE (?INH ?CLMD) (AND (?INH INHABIT) (?CLMD CLASS-MODEL) (?CLMD ?INH HAS-RELATIONSHIP)
(?END1 ?INH IS-END1) (?END2 ?INH IS-END2) (OR (?END1 (OR AGENT-CLASS ORGANIZATION-CLASS ROLE-
CLASS ENVIRONMENT-CLASS)) (?END2 (OR CITIZEN-CLASS ROLE-CLASS))))) --> (((?INH INHABIT-8)
(?CLMD EC-CLASS-MODEL)))
(RETRIEVE (?ASS ?CLMD) (AND (?ASS ASSOCIATION) (?CLMD CLASS-MODEL) (?CLMD ?ASS HAS-
RELATIONSHIP) (?END1 ?ASS IS-END1) (?END2 ?ASS IS-END2) (OR (AND (?END1 (OR AGENT-CLASS
ORGANIZATION-CLASS)) (?END2 (OR AGENT-CLASS ORGANIZATION-CLASS ENVIRONMENT-CLASS))) (AND
(?END1 ENVIRONMENT-CLASS) (?END2 CITIZEN-CLASS)) (?END1 ROLE-CLASS) (?END2 ROLE-CLASS)))) -->
NIL
(RETRIEVE (?SPEC ?CLMD) (AND (?SPEC SPECIALIZATION) (?CLMD CLASS-MODEL) (?CLMD ?SPEC HAS-
RELATIONSHIP) (?END1 ?SPEC IS-END1) (?END2 ?SPEC IS-END2) (OR (?END1 ROLE-CLASS) (?END2 ROLE-
CLASS) (AND (?END1 AGENT-CLASS) (?END2 (OR OBJECT-CLASS ORGANIZATION-CLASS ENVIRONMENT-
CLASS))) (AND (?END1 MAIN-ORGANIZATION-CLASS) (?END2 (OR OBJECT-CLASS SUB-ORGANIZATION-CLASS
AGENT-CLASS ENVIRONMENT-CLASS))) (AND (?END1 SUB-ORGANIZATION-CLASS) (?END2 (OR OBJECT-CLASS
MAIN-ORGANIZATION-CLASS AGENT-CLASS ENVIRONMENT-CLASS))) (AND (?END1 ACTIVE-ENVIRONMENT-CLASS)
(?END2 (OR CITIZEN-CLASS PASSIVE-ENVIRONMENT-CLASS))) (AND (?END1 PASSIVE-ENVIRONMENT-CLASS)
(?END2 (OR CITIZEN-CLASS ACTIVE-ENVIRONMENT-CLASS)))))) --> NIL
(RETRIEVE (?AGG ?CLMD) (AND (?AGG AGGREGATION) (?CLMD CLASS-MODEL) (?CLMD ?AGG HAS-
RELATIONSHIP) (?END1 ?AGG IS-END1) (?END2 ?AGG IS-END2) (OR (?END1 (OR ROLE-CLASS AGENT-CLASS
ORGANIZATION-CLASS ENVIRONMENT-CLASS)) (?END2 (OR ROLE-CLASS AGENT-CLASS ORGANIZATION-CLASS
ENVIRONMENT-CLASS))))) --> NIL
(RETRIEVE (?DEP ?CLMD) (AND (?DEP DEPENDENCY) (?CLMD CLASS-MODEL) (?CLMD ?DEP HAS-
RELATIONSHIP) (?END1 ?DEP IS-END1) (?END2 ?DEP IS-END2) (OR (?END1 (OR ROLE-CLASS AGENT-CLASS
ORGANIZATION-CLASS ENVIRONMENT-CLASS)) (?END2 (OR ROLE-CLASS AGENT-CLASS ORGANIZATION-CLASS
ENVIRONMENT-CLASS))))) --> NIL

;;;;;;;;;;;;;;;;;;

(IN-KNOWLEDGE-BASE MASML-TBOX PHASE1-ABOX) --> (MASML-TBOX PHASE1-ABOX)
(IN-TBOX MASML-TBOX) --> MASML-TBOX
(TBOX-CLASSIFIED-P) --> T
(TBOX-COHERENT-P) --> T
(TBOX-CYCLIC-P) --> NIL
(IN-ABOX PHASE1-ABOX MASML-TBOX) --> PHASE1-ABOX
(CHECK-ABOX-COHERENCE) --> T
(RETRIEVE (?ORG-MD ?ENV ?NO-CITIZEN) (AND (?ORG-MD ORGANIZATION-MODEL) (?ENV ENVIRONMENT-
CLASS) (?NO-CITIZEN CITIZEN-CLASS) (?ORG-MD ?NO-CITIZEN HAS-CLASS) (NOT (?ORG-MD (SOME HAS-
RELATIONSHIP INHABIT))))) --> (((?ORG-MD THE-ORGANIZING-MODEL) (?ENV EXPERT-COMMITEE) (?NO-
CITIZEN ORGANIZING-INSTITUTION)) ((?ORG-MD THE-ORGANIZING-MODEL) (?ENV EXPERT-COMMITEE) (?NO-
CITIZEN RESEARCHER-AGENT)))
(RETRIEVE (?SPEC ?ORGMD) (AND (?SPEC SPECIALIZATION) (?ORGMD ORGANIZATION-MODEL) (?ORGMD ?SPEC
HAS-RELATIONSHIP))) --> NIL
(RETRIEVE (?ASSOC ?ORGMD) (AND (?ASSOC ASSOCIATION) (?ORGMD ORGANIZATION-MODEL) (?ORGMD ?ASSOC
HAS-RELATIONSHIP))) --> (((?ASSOC ASSOC-1) (?ORGMD THE-ORGANIZING-MODEL)))
(RETRIEVE (?AGG ?ORGMD) (AND (?AGG AGGREGATION) (?ORGMD ORGANIZATION-MODEL) (?ORGMD ?AGG HAS-
RELATIONSHIP))) --> NIL
(RETRIEVE (?CTRL ?ORGMD) (AND (?CTRL CONTROL) (?ORGMD ORGANIZATION-MODEL) (?ORGMD ?CTRL HAS-
RELATIONSHIP))) --> NIL
(RETRIEVE (?DEP ?ORGMD) (AND (?DEP DEPENDENCY) (?ORGMD ORGANIZATION-MODEL) (?ORGMD ?DEP HAS-
RELATIONSHIP))) --> NIL
(RETRIEVE (?REL ?ORGMD) (AND (?REL RELATIONSHIP) (?ORGMD ORGANIZATION-MODEL) (?ORGMD ?REL HAS-
RELATIONSHIP) (?END1 ?REL IS-END1) (?END2 ?REL IS-END2) (?END1 ROLE-CLASS) (?END2 ROLE-
CLASS))) --> NIL
(RETRIEVE (?REL ?ORGMD) (AND (?REL RELATIONSHIP) (?ORGMD ORGANIZATION-MODEL) (?ORGMD ?REL HAS-
RELATIONSHIP) (?END1 ?REL IS-END1) (?END2 ?REL IS-END2) (?END1 AGENT-CLASS) (?END2 AGENT-
CLASS))) --> NIL
(RETRIEVE (?REL ?ORGMD) (AND (?REL RELATIONSHIP) (?ORGMD ORGANIZATION-MODEL) (?ORGMD ?REL HAS-
RELATIONSHIP) (?END1 ?REL IS-END1) (?END2 ?REL IS-END2) (?END1 ENVIRONMENT-CLASS) (?END2
ENVIRONMENT-CLASS))) --> NIL
```

```
(RETRIEVE (?REL ?ORGMD-BAD) (AND (?REL RELATIONSHIP) (?ORGMD-BAD ORGANIZATION-MODEL) (?ORGMD-
BAD ?REL HAS-RELATIONSHIP) (?REL (OR ASSOCIATION AGGREGATION SPECIALIZATION CONTROL
DEPENDENCY)))) --> (((?REL ASSOC-1) (?ORGMD-BAD THE-ORGANIZING-MODEL)))
```

Código da base de conehecimento usado na fase F2

Nesta seção encontra-se o código da base de conhecimento usada no estudo de caso sobre o Expert Committee, usando a aplicação do método para a linguagem MAS-ML. Esta base refere-se à execução da fase F2 do Observed-MAS e possui erros de modelagem, cuja introdução foi deliberada.

```
(in-abox phase2-ABox MASML-2-TBox)

; creating the environment of the EC system
(instance expert-commitee passive-environment-class)
(instance expert-commitee-path passive-environment-class-path)
(related expert-commitee-path expert-commitee has-head)

; creating the main-organization of the EC System
(instance organizing-institution main-organization-class)
(instance organizing-institution-path main-organization-class-path)
(related organizing-institution-path organizing-institution has-head)
(related organizing-institution-path expert-commitee-path has-tail)

;;;;;;;;;; internal state of the main organization

; creating the main-organization goals
(instance start-event goal)
(related organizing-institution start-event has-goal)

; creating the main-organization axioms
(instance dont-play-CA axiom)
(instance dont-play-CHCM axiom)
(instance dont-play-CR axiom)
(instance dont-play-CCH axiom)
(instance reviewer-not-author axiom)
(instance pcmember-not-author axiom)
(instance dont-play-CMC axiom)

(related organizing-institution dont-play-CA has-axiom)
(related organizing-institution dont-play-CHCM has-axiom)
(related organizing-institution dont-play-CR has-axiom)
(related organizing-institution dont-play-CCH has-axiom)
(related organizing-institution reviewer-not-author has-axiom)
(related organizing-institution pcmember-not-author has-axiom)
(related organizing-institution dont-play-CMC has-axiom)


; creating the main-organization actions
(instance to-create-roles action)
(instance to-create-reviewer action)
(instance to-create-PCmember action)
(instance to-create-PCchair action)
(instance to-create-PCmanager action)
(instance to-create-agenda action)

(related organizing-institution to-create-roles has-action)
(related organizing-institution to-create-reviewer has-action)
(related organizing-institution to-create-PCmember has-action)
(related organizing-institution to-create-PCchair has-action)
(related organizing-institution to-create-PCmanager has-action)
(related organizing-institution to-create-agenda has-action)

; creating the main-organization plans
(instance start-event-plan plan)
(related organizing-institution start-event-plan has-plan)
(instance start-event-plan-struct sequence)
(related start-event-plan start-event-plan-struct has-structure)

(related start-event-plan to-create-PCmanager has-action)
(related start-event-plan to-create-PCmember has-action)
(related start-event-plan to-create-reviewer has-action)
(related start-event-plan to-create-roles has-action)
(related start-event-plan to-create-PCchair has-action)

(related start-event-plan-struct to-create-PCmanager has-first)
(related start-event-plan-struct to-create-PCchair has-last)
```

```
(related start-event-plan-struct to-create-PCmanager has-element)
(related start-event-plan-struct to-create-PCmember has-element)
(related start-event-plan-struct to-create-reviewer has-element)
(related start-event-plan-struct to-create-roles has-element)
(related start-event-plan-struct to-create-PCchair has-element)

(related to-create-PCmanager to-create-PCmember is-before)
(related to-create-PCmember to-create-reviewer is-before)
(related to-create-reviewer to-create-roles is-before)
(related to-create-roles to-create-PCchair is-before)

;;;;;;;;;;;;;;;;;;;;;;;;;end internal state main-org

; creating roles that are owned by the main-organization
(instance reviewer agent-role-class)
(instance reviewer-path agent-role-class-path)
(related reviewer-path reviewer has-head)
(related reviewer-path organizing-institution-path has-tail)

(instance chair agent-role-class)
(instance chair-path agent-role-class-path)
(related chair-path chair has-head)
(related organizing-institution-path chair-path has-tail)

(instance manager agent-role-class)
(instance manager-path agent-role-class-path)
(related manager-path manager has-head)
(related manager-path organizing-institution-path has-tail)

(instance pc-member agent-role-class)
(instance pc-member-path agent-role-class-path)
(related pc-member-path pc-member has-head)
(related pc-member-path organizing-institution-path has-tail)

(instance author agent-role-class)
(instance author-path agent-role-class-path)
(related author-path author has-head)
(related author-path organizing-institution-path has-tail)

; creating the main-organization's ownership relationships
(instance own-1 ownership)
(instance own-2 ownership)
(instance own-3 ownership)
(instance own-4 ownership)
(instance own-5 ownership)

; resources of the system
(instance paper object-class)
(instance presentation object-class)
(instance proposition object-class)
(instance result object-class)
(instance vote object-class)
(instance conflict object-class)
(instance camera-ready object-class)

; creating the relationships between objects
(instance assoc-1-has association)
(instance assoc-2-has association)
(instance assoc-3-has association)
(instance assoc-4-has association)
(instance assoc-conflict association)

(instance dep-1 dependency)

(instance inhabit-1 inhabit)
(instance inhabit-2 inhabit)
(instance inhabit-3 inhabit)
(instance inhabit-4 inhabit)
(instance inhabit-5 inhabit)
(instance inhabit-6 inhabit)
(instance inhabit-7 inhabit)
(instance inhabit-8 inhabit)

; agents of the system
(instance researcher-agent agent-class)
(instance researcher-agent-path agent-class-path)
(related researcher-agent-path researcher-agent has-head)
(related researcher-agent-path reviewer-path has-tail)
(related researcher-agent-path chair-path has-tail)
(related researcher-agent-path manager-path has-tail)
(related researcher-agent-path pc-member-path has-tail)
(related researcher-agent-path author-path has-tail)


; creating the researcher goals
(instance receive-camera-ready goal)
(instance submit-paper goal)
(instance distribute-paper goal)
```

```
(instance receive-final-results goal)
(instance review-paper goal)
(instance receive-paper goal)
(instance solve-conflict goal)
(instance accept-paper goal)
(instance present-final-results goal)
(instance vote-to-solve goal)
(instance make-review goal)
(instance send-camera-ready goal)
(instance engage-reviewer goal)
(instance provide-reviewer goal)

(related researcher-agent receive-camera-ready has-goal)
(related researcher-agent submit-paper has-goal)
(related researcher-agent distribute-paper has-goal)
(related researcher-agent receive-final-results has-goal)
(related researcher-agent review-paper has-goal)
(related researcher-agent receive-paper has-goal)
(related researcher-agent solve-conflict has-goal)
(related researcher-agent accept-paper has-goal)
(related researcher-agent present-final-results has-goal)
(related researcher-agent vote-to-solve has-goal)
(related researcher-agent make-review has-goal)
(related researcher-agent send-camera-ready has-goal)
(related researcher-agent engage-reviewer has-goal)
(related researcher-agent provide-reviewer has-goal)

; creating the researcher actions
(instance to-make-additional-relation action)
(instance to-make-presentation action)
(instance to-select-new-reviewer action)
(instance to-register-vote action)
(instance to-search-for-conflict action)
(instance to-vote action)
(instance to-register-result action)
(instance to-make-final-result action)
(instance to-review action)
(instance to-prepare-paper-pack action)
(instance to-check-less-then-3-reviewer action)
(instance to-make-camera-ready action)
(instance to-make-paper action)
(instance to-register-paper-and-presentation action)
(instance to-select-reviewer action)
(instance to-make-review-proposition action)
(instance to-refresh-reviewer action)
(instance to-register-camera-ready action)
(instance to-assess-review-proposition action)

(related researcher-agent to-make-additional-relation has-action)
(related researcher-agent to-make-presentation has-action)
(related researcher-agent to-select-new-reviewer has-action)
(related researcher-agent to-register-vote has-action)
(related researcher-agent to-search-for-conflict has-action)
(related researcher-agent to-vote has-action)
(related researcher-agent to-register-result has-action)
(related researcher-agent to-make-final-result has-action)
(related researcher-agent to-review has-action)
(related researcher-agent to-prepare-paper-pack has-action)
(related researcher-agent to-check-less-then-3-reviewer has-action)
(related researcher-agent to-make-camera-ready has-action)
(related researcher-agent to-register-paper-and-presentation has-action)
(related researcher-agent to-select-reviewer has-action)
(related researcher-agent to-make-review-proposition has-action)
(related researcher-agent to-make-paper has-action)
(related researcher-agent to-refresh-reviewer has-action)
(related researcher-agent to-register-camera-ready has-action)
(related researcher-agent to-assess-review-proposition has-action)

; creating the researcher plans
(instance receive-camera-ready-plan plan)
(instance send-camera-ready-plan plan)
(instance submit-paper-plan plan)
(instance receive-paper-plan plan)
(instance assign-paper-plan plan)
(instance accept-paper-plan plan)
(instance provide-reviewer-plan plan)
(instance engage-reviewer-plan plan)
(instance make-review-plan plan)
(instance review-plan plan)
(instance solve-conflict-plan plan)
(instance vote-conflict-plan plan)
(instance show-final-result-plan plan)

; assigning the researcher's plans to it
(related researcher-agent receive-camera-ready-plan has-plan)
(related researcher-agent send-camera-ready-plan has-plan)
(related researcher-agent submit-paper-plan has-plan)
(related researcher-agent assign-paper-plan has-plan)
```

```
(related researcher-agent receive-paper-plan has-plan)
(related researcher-agent accept-paper-plan has-plan)
(related researcher-agent provide-reviewer-plan has-plan)
(related researcher-agent engage-reviewer-plan has-plan)
(related researcher-agent make-review-plan has-plan)
(related researcher-agent review-plan has-plan)
(related researcher-agent solve-conflict-plan has-plan)
(related researcher-agent vote-conflict-plan has-plan)
(related researcher-agent show-final-result-plan has-plan)

;; turning the researcher into a citizen
(instance inhabit-9 inhabit)
(related researcher inhabit-9 is-end1)
(related expert-commitee  inhabit-9 is-end2)

; structuring the researcher plans
(instance accept-paper-plan-struct sequence)
(related accept-paper-plan accept-paper-plan-struct has-structure)

(instance submit-paper-plan-struct sequence)
(related submit-paper-plan submit-paper-plan-struct has-structure)

(instance review-plan-struct sequence)
(related review-plan review-plan-struct has-structure)

(instance receive-paper-plan-struct sequence)
(related receive-paper-plan receive-paper-plan-struct has-structure)

(instance make-review-plan-struct sequence)
(related make-review-plan make-review-plan-struct has-structure)

(instance engage-reviewer-plan-struct sequence)
(related engage-reviewer-plan engage-reviewer-plan-struct has-structure)

(instance provide-reviewer-plan-struct sequence)
(related provide-reviewer-plan provide-reviewer-plan-struct has-structure)

(instance receive-camera-ready-plan-struct sequence)
(related receive-camera-ready-plan receive-camera-ready-plan-struct has-structure)

(instance send-camera-ready-plan-struct sequence)
(related send-camera-ready-plan send-camera-ready-plan-struct has-structure)

(instance show-final-result-plan-struct sequence)
(related show-final-result-plan show-final-result-plan-struct has-structure)

(instance solve-conflict-plan-struct sequence)
(related solve-conflict-plan solve-conflict-plan-struct has-structure)

(instance vote-conflict-plan-struct sequence)
(related vote-conflict-plan vote-conflict-plan-struct has-structure)

; creating the researcher beliefs (no beliefs assigned)

; creating the play relationships the researcher plays
(instance play-1 play)
(instance play-2 play)
(instance play-3 play)
(instance play-4 play)
(instance play-5 play)

; creating the internal structure of the agent-roles


; agent-role beliefs and goals (agent-role goals are the
; same of the researcher agent goals)

(instance review-proposal-belief belief)
(instance package-list-belief belief)
(instance paper-belief belief)
(instance paper-list-belief belief)
(instance result-belief belief)
(instance review-belief belief)
(instance assessment-belief belief)

(related reviewer review-proposal-belief has-belief)
(related reviewer package-list-belief has-belief)
(related reviewer paper-list-belief has-belief)
(related reviewer result-belief has-belief)
(related reviewer review-belief has-belief)
(related reviewer assessment-belief has-belief)

(related reviewer accept-paper has-goal)
(related reviewer review-paper has-goal)

(instance presentation-belief belief)
(instance final-result-belief belief)
(instance final-result-list-belief belief)
```

```
(instance camera-ready-belief belief)
(instance reviewers-list-belief belief)
(instance chair-reviewers-list-belief belief)
(instance presentation-list-belief belief)
(instance deadline-reviewer-belief belief)
(instance review-proposal-list-belief belief)
(instance result-list-belief belief)
(instance deadline-paper-belief belief)
(instance deadline-camera-ready-belief belief)
(instance review-acceptance-belief belief)
(instance voting-list-belief belief)
(instance camera-ready-list-belief belief)
(instance additional-relation-belief belief)

(related author presentation-belief has-belief)
(related author paper-belief has-belief)
(related author review-belief has-belief)
(related author final-result-belief has-belief)
(related author camera-ready-belief has-belief)

(related author submit-paper has-goal)
(related author receive-final-results has-goal)
(related author send-camera-ready has-goal)

(related chair chair-reviewers-list-belief has-belief)
(related chair paper-list-belief has-belief)
(related chair presentation-list-belief has-belief)
(related chair deadline-reviewer-belief has-belief)
(related chair review-proposal-list-belief has-belief)
(related chair review-acceptance-belief has-belief)
(related chair package-list-belief has-belief)
(related chair result-list-belief has-belief)
(related chair voting-list-belief has-belief)
(related chair camera-ready-list-belief has-belief)
(related chair final-result-list-belief has-belief)
(related chair additional-relation-belief has-belief)
(related chair deadline-paper-belief has-belief)
(related chair deadline-camera-ready-belief has-belief)

(related chair receive-paper has-goal)
(related chair distribute-paper has-goal)
(related chair engage-reviewer has-goal)
(related chair make-review has-goal)
(related chair solve-conflict has-goal)
(related chair present-final-results has-goal)
(related chair receive-camera-ready has-goal)

(instance reviewers-list-belief belief)

(related manager reviewers-list-belief  has-belief)

(related manager provide-reviewer has-goal)

(instance voting-belief belief)
(instance conflict-belief belief)

(related pc-member voting-belief has-belief)
(related pc-member conflict-belief has-belief)

(related pc-member solve-conflict has-goal)

; agent-role duties
(instance duty-make-additional-relation duty)
(instance duty-make-presentation duty)
(instance duty-select-new-reviewer duty)
(instance duty-register-vote duty)
(instance duty-search-for-conflict duty)
(instance duty-vote duty)
(instance duty-register-result duty)
(instance duty-make-final-result duty)
(instance duty-review duty)
(instance duty-prepare-paper-pack duty)
(instance duty-check-less-then-3-reviewer duty)
(instance duty-make-camera-ready duty)
(instance duty-make-paper duty)
(instance duty-register-paper-and-presentation duty)
(instance duty-select-reviewer duty)
(instance duty-make-review-proposition duty)
(instance duty-refresh-reviewer duty)
(instance duty-register-camera-ready duty)
(instance duty-assess-review-proposition duty)
(instance duty-provide-reviewer duty)

(related chair duty-register-paper-and-presentation has-duty)
(related chair duty-select-reviewer has-duty)
(related chair duty-make-review-proposition has-duty)
(related chair duty-refresh-reviewer has-duty)
(related chair duty-check-less-then-3-reviewer has-duty)
```

```
(related chair duty-prepare-paper-pack has-duty)
(related chair duty-register-result has-duty)
(related chair duty-search-for-conflict has-duty)
(related chair duty-register-vote has-duty)
(related chair duty-register-camera-ready has-duty)
(related chair duty-make-final-result has-duty)

(related manager duty-select-new-reviewer has-duty)
(related manager duty-make-additional-relation has-duty)

(related pc-member duty-vote has-duty)

(related author duty-make-paper has-duty)
(related author duty-make-presentation has-duty)
(related author duty-make-camera-ready has-duty)

(related reviewer duty-assess-review-proposition has-duty)
(related reviewer duty-review has-duty)

; agent-role protocols
(instance send-review-proposal-prtcl protocol)
(instance inform-review-acceptance-prtcl protocol)
(instance send-pack-to-reviewer-prtcl protocol)
(instance send-result-prtcl protocol)
(instance submit-paper-and-presentation-prtcl protocol)
(instance send-final-result-and-review-prtcl protocol)
(instance send-camera-ready-prtcl protocol)
(instance engage-more-reviewer-prtcl protocol)
(instance solve-conflict-with-pc-prtcl protocol)

(related reviewer send-review-proposal-prtcl has-protocol)
(related reviewer inform-review-acceptance-prtcl has-protocol)
(related reviewer send-pack-to-reviewer-prtcl has-protocol)
(related reviewer send-result-prtcl has-protocol)

(related author submit-paper-and-presentation-prtcl has-protocol)
(related author send-final-result-and-review-prtcl has-protocol)
(related author send-camera-ready-prtcl has-protocol)

(related manager engage-more-reviewer-prtcl has-protocol)

(related chair submit-paper-and-presentation-prtcl has-protocol)
(related chair send-review-proposal-prtcl has-protocol)
(related chair inform-review-acceptance-prtcl has-protocol)
(related chair send-pack-to-reviewer-prtcl has-protocol)
(related chair send-result-prtcl has-protocol)
(related chair solve-conflict-with-pc-prtcl has-protocol)
(related chair send-final-result-and-review-prtcl has-protocol)
(related chair send-camera-ready-prtcl has-protocol)
(related chair engage-more-reviewer-prtcl has-protocol)

(related pc-member solve-conflict-with-pc-prtcl has-protocol)

;; agent-role msgs
(instance send-proposal-msg agent-msg)
(instance send-final-result-msg agent-msg)
(instance send-pack-msg agent-msg)
(instance send-result-msg agent-msg)
(instance send-request-msg agent-msg)
(instance send-relation-msg agent-msg)
(instance send-conflict-msg agent-msg)
(instance send-vote-msg agent-msg)
(instance send-msg agent-msg)
(instance inform-msg agent-msg)
(instance inform-deadline-msg agent-msg)
(instance submit-msg agent-msg)

(related chair send-conflict-msg has-msg)
(related chair send-proposal-msg has-msg)
(related chair send-final-result-msg has-msg)
(related chair send-pack-msg has-msg)
(related chair send-result-msg has-msg)
(related chair send-request-msg has-msg)
(related chair send-relation-msg has-msg)
(related chair submit-msg has-msg)
(related chair send-vote-msg has-msg)
(related chair send-msg has-msg)
(related chair inform-msg has-msg)
(related chair inform-deadline-msg has-msg)

(related manager send-request-msg has-msg)
(related manager send-relation-msg has-msg)

(related pc-member send-conflict-msg has-msg)
(related pc-member send-vote-msg has-msg)

(related reviewer send-proposal-msg has-msg)
(related reviewer send-pack-msg has-msg)
```

```
(related reviewer send-result-msg has-msg)
(related reviewer inform-msg has-msg)

(related author submit-msg has-msg)
(related author send-final-result-msg has-msg)
(related author inform-deadline-msg has-msg)
(related author send-msg has-msg)

;; structuring the agent-roles protocols
(instance submit-struct sequence)
(related submit-paper-and-presentation-prtcl submit-struct has-structure)
(related submit-paper-and-presentation-prtcl submit-msg has-msg)
(related submit-struct submit-msg has-element)
(related submit-msg author-path has-msg-sender)
(related submit-msg chair-path has-msg-receiver)
(related submit-struct submit-msg has-first)
(related submit-struct submit-msg has-last)

(instance send-proposal-struct sequence)
(related send-review-proposal-prtcl send-proposal-struct has-structure)
(related send-review-proposal-prtcl send-proposal-msg has-msg)
(related send-proposal-struct send-proposal-msg has-element)
(related send-proposal-msg chair-path has-msg-sender)
(related send-proposal-msg reviewer-path has-msg-receiver)
(related send-proposal-struct send-proposal-msg has-first)
(related send-proposal-struct send-proposal-msg has-last)

(instance inform-acceptance-struct sequence)
(related inform-review-acceptance-prtcl inform-acceptance-struct has-structure)
(related inform-review-acceptance-prtcl inform-msg has-msg)
(related inform-acceptance-struct inform-msg has-element)
(related inform-msg reviewer-path has-msg-sender)
(related inform-msg chair-path has-msg-receiver)
(related inform-acceptance-struct inform-msg has-first)
(related inform-acceptance-struct inform-msg has-last)

(instance send-result-struct sequence)
(related send-result-prtcl send-result-struct has-structure)
(related send-result-prtcl send-result-msg has-msg)
(related send-result-struct send-result-msg has-element)
(related send-result-msg reviewer-path has-msg-sender)
(related send-result-msg chair-path has-msg-receiver)
(related send-result-struct send-result-msg has-first)
(related send-result-struct send-result-msg has-last)

(instance send-pack-struct sequence)
(related send-pack-to-reviewer-prtcl send-pack-struct has-structure)
(related send-pack-to-reviewer-prtcl send-pack-msg has-msg)
(related send-pack-struct send-pack-msg has-element)
(related send-pack-msg chair-path has-msg-sender)
(related send-pack-msg reviewer-path has-msg-receiver)
(related send-pack-struct send-pack-msg has-first)
(related send-pack-struct send-pack-msg has-last)

(instance send-final-result-struct sequence)
(related send-final-result-and-review-prtcl send-final-result-struct has-structure)
(related send-final-result-and-review-prtcl send-final-result-msg has-msg)
(related send-final-result-struct send-final-result-msg has-element)
(related send-final-result-msg chair-path has-msg-sender)
(related send-final-result-msg author-path has-msg-receiver)
(related send-final-result-struct send-final-result-msg has-first)
(related send-final-result-struct send-final-result-msg has-last)

(instance send-camera-ready-struct sequence)
(related send-camera-ready-prtcl send-camera-ready-struct has-structure)
(related send-camera-ready-prtcl inform-deadline-msg has-msg)
(related send-camera-ready-prtcl send-msg has-msg)
(related send-camera-ready-struct inform-deadline-msg has-element)
(related send-camera-ready-struct send-msg has-element)
(related inform-deadline-msg chair-path has-msg-sender)
(related inform-deadline-msg author-path has-msg-receiver)
(related send-msg author-path has-msg-sender)
(related send-msg chair-path has-msg-receiver)
(related send-camera-ready-struct inform-deadline-msg has-first)
(related send-camera-ready-struct send-msg has-last)
(related inform-deadline-msg send-msg is-before)

(instance engage-reviewer-struct sequence)
(related engage-more-reviewer-prtcl engage-reviewer-struct has-structure)
(related engage-more-reviewer-prtcl send-request-msg has-msg)
(related engage-more-reviewer-prtcl send-relation-msg has-msg)
(related engage-reviewer-struct send-request-msg has-element)
(related engage-reviewer-struct send-relation-msg has-element)
(related send-request-msg chair-path has-msg-sender)
(related send-request-msg manager-path has-msg-receiver)
(related send-relation-msg manager-path has-msg-sender)
(related send-relation-msg chair-path has-msg-receiver)
(related engage-reviewer-struct send-request-msg has-first)
```

```
(related engage-reviewer-struct send-relation-msg has-last)
(related send-request-msg send-relation-msg is-before)

(instance solve-conflict-struct sequence)
(related solve-conflict-with-pc-prtcl solve-conflict-struct has-structure)
(related solve-conflict-with-pc-prtcl send-conflict-msg has-msg)
(related solve-conflict-with-pc-prtcl send-vote-msg has-msg)
(related solve-conflict-struct send-conflict-msg has-element)
(related solve-conflict-struct send-vote-msg has-element)
(related send-conflict-msg chair-path has-msg-sender)
(related send-conflict-msg pc-member-path has-msg-receiver)
(related send-vote-msg pc-member-path has-msg-sender)
(related send-vote-msg chair-path has-msg-receiver)
(related solve-conflict-struct send-conflict-msg has-first)
(related solve-conflict-struct send-vote-msg has-last)

; creating a class-model
(instance ec-class-model class-model)

; populating the ec-class-model
(related ec-class-model expert-commitee has-class)
(related ec-class-model organizing-institution has-class)
(related ec-class-model paper has-class)
(related ec-class-model presentation has-class)
(related ec-class-model camera-ready has-class)
(related ec-class-model conflict has-class)
(related ec-class-model result has-class)
(related ec-class-model proposition has-class)
(related ec-class-model vote has-class)

(related ec-class-model assoc-1-has has-relationship)
(related ec-class-model assoc-2-has has-relationship)
(related ec-class-model assoc-3-has has-relationship)
(related ec-class-model assoc-4-has has-relationship)
(related ec-class-model assoc-conflict has-relationship)

(related ec-class-model dep-1 has-relationship)

(related ec-class-model inhabit-1 has-relationship)
(related ec-class-model inhabit-2 has-relationship)
(related ec-class-model inhabit-3 has-relationship)
(related ec-class-model inhabit-4 has-relationship)
(related ec-class-model inhabit-5 has-relationship)
(related ec-class-model inhabit-6 has-relationship)
(related ec-class-model inhabit-7 has-relationship)
;(related ec-class-model inhabit-8 has-relationship)
(related assoc-1-has paper has-end1)
(related assoc-1-has camera-ready  has-end2)

(related assoc-2-has paper has-end1)
(related assoc-2-has proposition has-end2)

(related assoc-3-has paper has-end1)
(related assoc-3-has presentation has-end2)

(related assoc-4-has paper has-end1)
(related assoc-4-has result has-end2)

(related assoc-conflict vote has-end1)
(related assoc-conflict conflict has-end2)

(related dep-1 result has-end1)
(related dep-1 conflict has-end2)

(related inhabit-1 paper has-end1)
(related inhabit-1 expert-commitee has-end2)

(related inhabit-2 camera-ready has-end1)
(related inhabit-2 expert-commitee has-end2)

(related inhabit-3 presentation has-end1)
(related inhabit-3 expert-commitee has-end2)

(related inhabit-4 result has-end1)
(related inhabit-4 expert-commitee has-end2)

(related inhabit-5 proposition has-end1)
(related inhabit-5 expert-commitee has-end2)

(related inhabit-6 vote has-end1)
(related inhabit-6 expert-commitee has-end2)

(related inhabit-7 conflict has-end1)
(related inhabit-7 expert-commitee has-end2)

(related inhabit-8 organizing-institution has-end1)
(related inhabit-8 expert-commitee has-end2)
```

```
; creating an organization-model
(instance the-organizing-model organization-model)

(related the-organizing-model inhabit-8 has-relationship)

; populating the the-organizing-model
(related the-organizing-model reviewer has-class)
(related the-organizing-model chair has-class)
(related the-organizing-model manager has-class)
(related the-organizing-model pc-member has-class)
(related the-organizing-model author has-class)
(related the-organizing-model organizing-institution has-class)
(related the-organizing-model researcher-agent has-class)

(related the-organizing-model own-1 has-relationship)
(related the-organizing-model own-2 has-relationship)
(related the-organizing-model own-3 has-relationship)
(related the-organizing-model own-4 has-relationship)
(related the-organizing-model own-5 has-relationship)

(related the-organizing-model play-1 has-relationship)
(related the-organizing-model play-2 has-relationship)
(related the-organizing-model play-3 has-relationship)
(related the-organizing-model play-4 has-relationship)
(related the-organizing-model play-5 has-relationship)

(related own-1 organizing-institution has-end1)
(related own-1 reviewer has-end2)

(related own-2 organizing-institution has-end1)
(related own-2 chair has-end2)

(related own-3 organizing-institution has-end1)
(related own-3 manager has-end2)

(related own-4 organizing-institution has-end1)
(related own-4 pc-member has-end2)

(related own-5 organizing-institution has-end1)
(related own-5 author has-end2)

(related play-1 researcher-agent has-end1)
(related play-1 reviewer has-end2)

(related play-2 researcher-agent has-end1)
(related play-2 chair has-end2)

(related play-3 researcher-agent has-end1)
(related play-3 manager has-end2)

(related play-4 researcher-agent has-end1)
(related play-4 pc-member has-end2)

(related play-5 researcher-agent has-end1)
(related play-5 author has-end2)

; creating a role-model
(instance the-role-model role-model)

; populating the the-role-model
(related the-role-model reviewer has-class)
(related the-role-model chair has-class)
;(related the-role-model manager has-class)
(related the-role-model pc-member has-class)
(related the-role-model author has-class)

;; introducing an object to the role-model to
;; create an interdependency between the role
;; and the class-diagram
(related the-role-model paper has-class)

(instance assoc-5 association)
(instance assoc-6 association)
(instance ctrl-1 control)
(instance ctrl-2 control)

(related the-role-model assoc-5 has-relationship)
(related the-role-model assoc-6 has-relationship)
(related the-role-model ctrl-1 has-relationship)
(related the-role-model ctrl-2 has-relationship)

(related assoc-5 chair has-end1)
(related assoc-5 manager has-end2)

(related assoc-6 chair has-end1)
(related assoc-6 author has-end2)

(related ctrl-1 chair has-end1)
```

```
(related ctrl-1 pc-member has-end2)

(related ctrl-2 chair has-end1)
(related ctrl-2 reviewer has-end2)

; creating sequence-models
(instance seq-1 sequence-model)
(related seq-1 submit-struct has-structure)
(related seq-1 author-path has-path)
(related seq-1 chair-path has-path)

(instance seq-2 sequence-model)
(related seq-2 send-proposal-struct has-structure)
(related seq-2 chair-path has-path)
(related seq-2 reviewer-path has-path)

(instance seq-3 sequence-model)
(related seq-3 inform-acceptance-struct has-structure)
(related seq-3 reviewer-path has-path)
(related seq-3 chair-path has-path)

(instance seq-4 sequence-model)
(related seq-4 send-result-struct has-structure)
(related seq-4 reviewer-path has-path)
(related seq-4 chair-path has-path)

(instance seq-5 sequence-model)
(related seq-5 send-pack-struct has-structure)
(related seq-5 chair-path has-path)
(related seq-5 reviewer-path has-path)

(instance seq-6 sequence-model)
(related seq-6 send-final-result-struct has-structure)
(related seq-6 chair-path has-path)
(related seq-6 author-path has-path)

(instance seq-7 sequence-model)
(related seq-7 send-camera-ready-struct has-structure)
(related seq-7 chair-path has-path)
(related seq-7 author-path has-path)

(instance seq-8 sequence-model)
(related seq-8 engage-reviewer-struct has-structure)
(related seq-8 chair-path has-path)
(related seq-8 manager-path has-path)

(instance seq-9 sequence-model)
(related seq-9 solve-conflict-struct has-structure)
(related seq-9 chair-path has-path)
(related seq-9 pc-member-path has-path)

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;INTRODUCING INTER-MODEL ERRORS
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;; Q1 agent-class in a class-model and not in an org-model
(instance user-agent agent-class)
(related ec-class-model user-agent has-class)

;;;;;; Q2 org-class in a class-model and not in an org-model
(instance reviewers-board sub-organization-class)
(related ec-class-model reviewers-board has-class)

;;;;;; Q3 role-class in a role-model and not in an org-model
(instance co-chair agent-role-class)
(instance the-role-model role-model)
(related the-role-model co-chair has-class)

;;;;;; Q4 object in an org-model and not in a class-model
;;;;;; Q5 object in a role-model and ot in a class-model
(instance eval-form object-class)
(related the-organizing-model eval-form has-class)
(related the-role-model eval-form has-class)

;;;;;; Q6 classless (ipath without an original class)
(instance paper01-path object-instance-path)
(instance paper01 object)
(related paper01-path paper-01 has-head)
(related seq-1 paper01-path has-path)

;;;;;; Q7 classless special case
(instance LES-PUC-Rio main-organization)
(related LES-PUC-Rio organizing-institution is-instanceOf)
(instance LES-PUC-Rio-path main-organization-instance-path)
(related LES-PUC-Rio LES-PUC-Rio-path has-path)
(related LES-PUC-Rio-path LES-PUC-Rio has-head)

(instance lucena agent)
(related lucena researcher-agent is-instanceOf)
```

```
(instance lucena-path agent-instance-path)
(related lucena lucena-path has-path)
(related lucena-path lucena has-head)
(instance selmas-chair agent-role)
;(related selmas-chair chair is-instanceOf)
(instance selmas-chair-path agent-role-instance-path)
(related selmas-chair selmas-chair-path has-path)
(related selmas-chair-path selmas-chair has-head)
(related lucena-path selmas-chair-path has-tail)

(instance create-agent-1 create-msg)
(related LES-PUC-Rio-path create-agent-1 is-msg-sender)
(related lucena-path create-agent-1 is-msg-receiver)

(instance seq-create-agent-1 sequence-model)
(related seq-create-agent-1 LES-PUC-Rio-path has-path)
(related seq-create-agent-1 lucena-path has-path)
(related seq-create-agent-1 create-agent-1 has-msg)
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(check-abox-coherence)
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
```

Resposta às consultas QV2

Nesta seção encontram-se as respostas às consultas QV2 aplicadas à base de conhecimento descrita na seção anterior.

```
(IN-TBOX MASML-2-TBOX) --> MASML-2-TBOX
(TBOX-COHERENT-P) --> T
(TBOX-CYCLIC-P) --> NIL
(IN-ABOX PHASE2-ABOX MASML-2-TBOX) --> PHASE2-ABOX
(CHECK-ABOX-COHERENCE) --> T
(RETRIEVE (?AG-CLASS ?CLASS-DIAGRAM)(AND (?AG-CLASS AGENT-CLASS) (?CLASS-DIAGRAM CLASS-
MODEL)(?ORGMD ORGANIZATION-MODEL)(?CLASS-DIAGRAM ?AG-CLASS HAS-CLASS)(NOT (?ORG-MD ?AG-CLASS
HAS-CLASS)))) --> (((?AG-CLASS USER-AGENT) (?CLASS-DIAGRAM EC-CLASS-MODEL)))
(RETRIEVE (?ORG-CLASS ?CLASS-DIAGRAM)(AND (?ORG-CLASS ORGANIZATION-CLASS) (?CLASS-DIAGRAM
CLASS-MODEL)(?ORGMD ORGANIZATION-MODEL)(?CLASS-DIAGRAM ?ORG-CLASS HAS-CLASS)(NOT (?ORGMD ?ORG-
CLASS HAS-CLASS)))) --> (((?ORG-CLASS REVIEWERS-BOARD) (?CLASS-DIAGRAM EC-CLASS-MODEL)))
(RETRIEVE (?OBJ-CLASS ?ORG-DIAGRAM) (AND (?OBJ-CLASS OBJECT-CLASS) (?ORG-DIAGRAM ORGANIZATION-
MODEL) (?ORG-DIAGRAM ?OBJ-CLASS HAS-CLASS) (?CLMD CLASS-MODEL) (NOT (?CLMD ?OBJ-CLASS HAS-
CLASS)))) --> (((?OBJ-CLASS EVAL-FORM) (?ORG-DIAGRAM THE-ORGANIZING-MODEL)))
(RETRIEVE (?RCLASS ?ROLE-DIAGRAM) (AND (?RCLASS ROLE-CLASS) (?ROLE-DIAGRAM ROLE-MODEL) (?ROLE-
DIAGRAM ?RCLASS HAS-CLASS) (?ORGMD ORGANIZATION-MODEL) (NOT (?ORGMD ?RCLASS HAS-CLASS)))) -->
(((?RCLASS CO-CHAIR) (?ROLE-DIAGRAM THE-ROLE-MODEL)))
(RETRIEVE (?OBJ-CLASS ?ROLE-DIAGRAM) (AND (?OBJ-CLASS OBJECT-CLASS) (?ROLE-DIAGRAM ROLE-MODEL)
(?ROLE-DIAGRAM ?OBJ-CLASS HAS-CLASS) (?CLMD CLASS-MODEL) (NOT (?CLMD ?OBJ-CLASS HAS-CLASS))))
--> (((?OBJ-CLASS EVAL-FORM) (?ROLE-DIAGRAM THE-ROLE-MODEL)))
(RETRIEVE (?IPATH ?SEQMD) (AND (?IPATH INSTANCE-PATH) (?SEQMD SEQUENCE-MODEL) (?SEQMD ?IPATH
HAS-PATH) (?IPATH ?HEAD HAS-HEAD)(?HEAD NIL IS-INSTANCEOF))) --> (((?IPATH PAPER01-PATH)
(?SEQMD SEQ-1)))
```

Apêndice H

Dados sobre a aplicação do método para AUML

A ontologia Ont1 para AUML

```
(in-knowledge-base AUML-TBox phase1-ABox)
(in-tbox AUML-TBox)
(signature
        :atomic-concepts (;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                            ;;;;;;;  TAO concepts
                            ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                           entity
                            class
                              agent-class
                              organization-class
                              object-class
                              environment-class
                              agent-role-class
                             class-instance
                            agent
                              object
                             features
                            goal
                              belief
                              action
                              protocol
                              attribute
                              method
                              condition
                            relationship
                               specialization
                               association
                             aggregation
                               dependency
                            msg
                             agent-msg
                             object-msg
                        ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                          ;;;;;;  AUML concepts
                          ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                          model
                             static-model
                            class-model
                             dynamic-model
                              sequence-model
                              activity-model
                              collaboration-model
                           path
                          agent-path
                            object-path
                           sequence
                          seq-element
                         )
        :roles ( ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                 ;;;;;;;;;;; TAO properties
                  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                 (has-goal :range goal)
                  (has-belief :range belief)
                  (has-action :range action
                             :parent has-element)
                 (has-precond :domain action
                               :range condition)
                  (has-poscond :domain action
                               :range condition)
                  (has-protocol :range protocol)
                  (has-end1 :domain relationship
                           :range class
                             :inverse is-end1)
                 (has-end2 :domain relationship
                           :range class
                             :inverse is-end2)
                  (is-end1 :domain class
                            :range relationship
                            :inverse has-end1)
                   (is-end2 :domain class
                            :range relationship
                            :inverse has-end2)
                  (has-msg :range msg
                           :parent has-element)
                  (has-msg-end :domain msg
                               :inverse is-msg-end)
                  (is-msg-end :inverse has-msg-end)
                  (has-msg-sender :parent has-msg-end)
                  (has-msg-receiver :parent has-msg-end)
                  (is-msg-sender :inverse has-msg-sender)
                  (is-msg-receiver :inverse has-msg-receiver)
                  (is-instanceOf :domain class-instance
                                  :range class)
                   (has-attribute :range attribute)
                   (has-method :range method)
                  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                   ;;;;;;;;;; AUML properties
                   ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                  (plays-role ;:domain agent-class
                               :range agent-role-class)
                   (has-class :domain static-model
                             :range (or class class-instance path) ;class
                             :inverse is-in-static-model)
                  (is-in-static-model :inverse has-class)
                  (has-relationship :domain static-model
                                    :range relationship
                                      :inverse is-relationship-of)
                   (is-relationship-of :inverse has-relationship)
                   (has-path :range path)
                  (has-head :domain path
                             :range (or class-instance class))
                  (has-tail :domain path
                             :range path)
                  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                   ;;;;;;;;;; ordering properties
                   ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                   (has-structure :range sequence)
                   (has-element :range seq-element)
                   (has-first :domain sequence
                              :range seq-element
                              :parent has-element
                              :inverse is-first)
                   (is-first :inverse has-first)
                   (has-last :domain sequence
                             :range seq-element
                             :parent has-element
                             :inverse is-last)
                   (is-last :inverse has-last)
                   (is-before :domain seq-element
                              :range seq-element
                              :transitive t
                              :inverse is-after)
                   (is-after  :domain seq-element
                              :range seq-element
                              :transitive t
                              :inverse is-before)
                  )
)
;; concepts restrictions
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;; TAO concepts taxonomy
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(implies class entity)
(implies class-instance entity)
;(implies path entity)

(implies environment-class class)
(implies agent-role-class class)
(implies agent-class class)
(implies organization-class class)
(implies object-class class)

(implies specialization relationship)
(implies association relationship)
(implies aggregation relationship)
(implies dependency relationship)

(implies goal features)
(implies belief features)
(implies action features)
;(implies plan features)

(implies protocol features)
(implies attribute features)
```

```
(implies method features)
(implies condition features)

(implies agent class-instance)
(implies object class-instance)

(implies agent-msg msg)
(implies object-msg msg)

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;; AUML concepts taxonomy (it reuses some TAO concepts)
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(implies static-model model)
(implies dynamic-model model)

(implies sequence-model dynamic-model)
(implies activity-model dynamic-model)
(implies collaboration-model dynamic-model)

(implies class-model static-model)

(equivalent path (or agent-path object-path))

;;; axiom that guarantee the possibility of the dynamic structure definition
(implies msg seq-element)
(implies action seq-element)

;;;;;;;;;;;;;;;;;;;;;;;;;;
;; ontology axioms
;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;               TAO disjoint restrictions
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(disjoint class class-instance features msg relationship)
(disjoint agent-class organization-class object-class environment-class agent-role-class)

(disjoint agent object)
(disjoint agent-msg object-msg)

(disjoint goal belief protocol attribute)
(disjoint specialization association aggregation dependency)
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;             AUML disjoint restrictions
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(disjoint static-model dynamic-model)
(disjoint sequence-model activity-model collaboration-model)
(disjoint object-path agent-path)
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;                                              ;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;      Instances-related axioms                ;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;                                              ;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(implies agent (all is-instanceOf agent-class))
(implies object (all is-instanceOf object-class))

(implies agent-path
 (or agent-role-class (and agent (some is-instanceOf
                                 (and agent-class (some plays-role agent-role-class))))
      (and agent-class (some plays-role agent-role-class))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;                                order axioms                                ;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; semantics of the first element of a sequence
(implies (and (some is-before *top*) (some is-after *bottom*)) (some is-first *top*))

;; semantics of the last element of a sequence
(implies (and (some is-after *top*) (some is-before *bottom*)) (some is-last *top*))

(implies (some has-first msg) (some has-msg msg))
(implies (some has-last msg) (some has-msg msg))

(implies (some has-first action) (some has-action action))
(implies (some has-last action) (some has-action action))
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;; end of tbox ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(classify-tbox)
(tbox-classified-p)
(tbox-coherent-p)
(tbox-cyclic-p)
```

Consultas QV1 para a o método aplicado a AUML

```
;; QV1 para AUML
;; Usa a especificação de Bauer

;;Q1 busca relacionamentos specialization mal estruturados
(retrieve (?end1 ?end2 ?spec)
        (and (?spec specialization)
             (?end1 ?spec is-end1)
             (?end2 ?spec is-end2)
             (or (and (?end1 object-class) (not (?end2 object-class)))
                 (and (?end1 agent-class) (not (?end2 agent-class)))
                  (and (not (?end1 object-class)) (not (?end1 agent-class)))
             )
         )
)

;;Q2 busca relacionamentos aggregation mal estruturados
(retrieve (?end1 ?end2 ?agg)
        (and (?agg aggregation)
             (?end1 ?agg is-end1)
             (?end2 ?agg is-end2)
             (or (and (?end1 object-class) (not (?end2 (or agent-class object-class))))
                 (and (?end1 agent-class) (not (?end2 agent-class)))
                  (and (?end1 agent-role-class) (not (?end2 organization-class)))
                  (and (not (?end1 agent-class)) (not (?end1 object-class))
                       (not (?end1 agent-role-class)))
             )
         )
)

;; Q3 busca relacionamentos dependency mal estruturados
(retrieve (?end1 ?end2 ?dep)
        (and (?dep dependency)
             (?end1 ?dep is-end1)
             (?end2 ?dep is-end2)
            (or (and (?end1 object-class) (not (?end2 object-class)))
                 (not (?end1 object-class))
             )
           )
)

;; Em AUML organizações incluem agentes desempenhando papéis. Estes agentes interagem
;; usando o ambiente. Assim, tanto papéis como agentes e ambientes são agregados de
;; organizações. "Representing Social Structures in UML"
;; As questões 4, 5 e 6 referem-se a esta propriedade

;; Q4
(retrieve (?org-without-roles)
  (and  (?org-without-roles organization-class)
        (not (?org-without-roles
               (some is-end2 (and aggregation (some has-end1 agent-role-class))))))
)

;; Q5 não deveria ser guideline????
(retrieve (?org-without-agents)
  (and  (?org-without-agents organization-class)
        (not (?org-without-agents
               (some is-end2 (and aggregation (some has-end1 agent-class))))))
)

;; Q6 não deveria ser guideline?????
(retrieve (?org-without-env)
  (and  (?org-without-env organization-class)
        (not (?org-without-env
               (some is-end2 (and aggregation (some has-end1 environment-class))))))
)

;; Q7 busca mensagens do tipo object-msg mal estruturadas
(retrieve (?bad-obj-msg)
     (and (?bad-obj-msg object-msg)
          ($?sender ?bad-obj-msg is-msg-sender)
           ($?receiver ?bad-obj-msg is-msg-receiver)
          (or (not ($?sender (or object agent)))
               ($?receiver agent))
     )
)

;; Q8 busca por agent-msgs mal estruturadas
(retrieve ($?sender $?receiver ?ag-msg)
        (and (?ag-msg agent-msg)
             ($?sender ?ag-msg is-msg-sender)
             ($?receiver ?ag-msg is-msg-receiver)
             (not (and ($?sender (and agent (some is-instanceOf
                                          (and agent-class
                                                 (some plays-role agent-role-class)))))
                         ($?receiver (and agent (some is-instanceOf
                                          (and agent-class
                                                 (some plays-role agent-role-class)))))
                  )
              )
             (not (and ($?sender (and agent-class
                                (some plays-role agent-role-class)))
                        ($?receiver (and agent-class
                                (some plays-role agent-role-class)))
                    )
             )
              (not (and ($?sender agent-role-class) ($?receiver agent-role-class)))
           )
)

;; Q9 busca instâncias que não possuem classes asscociadas
(retrieve (?instance)
   (and (?instance class-instance)
        (?instance nil is-instanceOf)
   )
)

;; Q10 busca protocolos sem templates definidos
(retrieve (?prtcl-without-struct)
   (and (?prtcl-without-struct protocol)
        (?prtcl-without-struct nil has-structure)
   )
)

;; Q11 busca protocolos mal estruturados
(retrieve (?prtcl-no-msgs)
  (and (?prtcl-no-msgs protocol)
       (?prtcl-no-msg nil has-msg)
  )
)

;; Q12 structures definem sequências ordenadas
;; busca structures mal estruturadas.
(retrieve (?bad-struct)
   (and (?bad-struct sequence)
        (?bad-struct ?element1 has-first)
        (?bad-struct ?element2 has-last)
        (?element2 ?element1 is-before)
   )
)

;; Q13 buscam diagramas de sequência sem estrutura
(retrieve (?seq-md-without-struct)
   (and (?seq-md-without-struct sequence-model)
        (?seq-md-without-struct nil has-structure)
   )
)
```

Consultas QD1 para o método aplicado a AUML

```
;; QD1 - boas práticas usando AUML

;; Q1 AUML não especifica se agentes devem sempre
;; possuir metas, assim vale indicar as agent-classes
;; que não tiveram metas associadas
(retrieve (?agentwithoutgoal)
            (and (?agentwithoutgoal agent-class)
                  (?agentwithoutgoal NIL has-goal)
            )
)

;; Q2 AUML não especifica se agentes devem desempenhar papéis
;; logo, vale indicar as agent-classes que não estão associadas
;; a nenhum papel de agente.
(retrieve (?agentwithoutrole)
            (and (?agentwithoutrole agent-class)
                  (?agentwithoutrole NIL plays-role)
            )
)

;; Q3 Agentes interagem através de protocolos de interação
;; logo, agent-classes que não possuem protocolos associados
;; precisam ser indicadas para o desenvolvedor
(retrieve (?agentwithoutprtcl)
            (and (?agentwithoutprtcl agent-class)
                  (?agentwithoutprtcl NIL has-protocol)
            ))
```

Base de conhecimento usada na fase F1

```
(in-abox phase1-ABox AUML-TBox)

(instance user-agent agent-class)
(instance store-agent agent-class)
(instance user-agent-path agent-path)
(instance store-agent-path agent-path)
(related user-agent user-agent-path has-path)
(related store-agent store-agent-path has-path)

(instance book object-class)
(instance imported-book object-class)
(instance used-book object-class)

(instance general-store organization-class)
(instance imported-bookstore organization-class)
(instance 2nd-hand-bookstore organization-class)

(instance buyer agent-role-class)
(instance seller agent-role-class)
(instance imported-book-buyer agent-role-class)
(instance imported-book-seller agent-role-class)
(instance 2nd-hand-book-buyer agent-role-class)
(instance 2nd-hand-book-seller agent-role-class)

(instance buy-a-book goal)
(instance sell-a-book goal)

(instance assoc-1 association)
(related assoc-1 buyer has-end1)
(related assoc-1 seller has-end2)

(instance assoc-2 association)
(related assoc-2 imported-book-buyer has-end1)
(related assoc-2 imported-book-seller has-end2)

(instance assoc-3 association)
(related assoc-3 2nd-hand-book-buyer has-end1)
(related assoc-3 2nd-hand-book-seller has-end2)

(instance aggr-1 aggregation)
(related aggr-1 buyer has-end1)
(related aggr-1 general-store has-end2)

(instance aggr-2 aggregation)
(related aggr-2 seller has-end1)
(related aggr-2 general-store has-end2)

(instance aggr-3 aggregation)
(related aggr-3 imported-book-buyer has-end1)
(related aggr-3 imported-bookstore has-end2)

(instance aggr-4 aggregation)
(related aggr-4 imported-book-seller has-end1)
(related aggr-4 imported-bookstore has-end2)

(instance aggr-5 aggregation)
;(related aggr-5 2nd-hand-book-buyer has-end1)
;(related aggr-5 2nd-hand-bookstore has-end2)

(instance aggr-6 aggregation)
;(related aggr-6 2nd-hand-book-seller has-end1)
;(related aggr-6 2nd-hand-bookstore has-end2)

(instance class-diag-1 class-model)

(related class-diag-1 store-agent-path has-entity)
(related class-diag-1 user-agent-path has-entity)
(related class-diag-1 general-store has-entity)
(related class-diag-1 imported-bookstore has-entity)
(related class-diag-1 2nd-hand-bookstore has-entity)
(related class-diag-1 buyer has-entity)
(related class-diag-1 seller has-entity)
(related class-diag-1 imported-book-buyer has-entity)
(related class-diag-1 imported-book-seller has-entity)
(related class-diag-1 2nd-hand-buyer has-entity)
(related class-diag-1 2nd-hand-seller has-entity)

(related class-diag-1 assoc-1 has-relationship)
(related class-diag-1 assoc-2 has-relationship)
(related class-diag-1 assoc-3 has-relationship)
```

```
(related class-diag-1 aggr-1 has-relationship)
(related class-diag-1 aggr-2 has-relationship)
(related class-diag-1 aggr-3 has-relationship)
(related class-diag-1 aggr-4 has-relationship)
;(related class-diag-1 aggr-5 has-relationship)
;(related class-diag-1 aggr-6 has-relationship)

(related user-agent buyer plays-role)
(related user-agent imported-book-buyer plays-role)
(related user-agent 2nd-hand-book-buyer plays-role)

(related store-agent seller plays-role)
(related store-agent imported-book-seller plays-role)
(related store-agent 2nd-hand-book-seller plays-role)

(instance class-diag-2 class-model)

(instance assoc-4 association)
(related assoc-4 user-agent has-end1)
(related assoc-4 store-agent has-end2)

(instance assoc-5 association)
(related assoc-5 user-agent has-end1)
(related assoc-5 book has-end2)

(instance assoc-6 association)
(related assoc-6 book has-end1)
(related assoc-6 store-agent has-end2)

(instance spec-1 specialization)
(related spec-1 imported-book has-end1)
(related spec-1 book has-end2)

(instance spec-2 specialization)
(related spec-2 used-book has-end1)
(related spec-2 book has-end2)

(related class-diag-2 user-agent has-entity)
(related class-diag-2 store-agent has-entity)
(related class-diag-2 book has-entity)
(related class-diag-2 imported-book has-entity)
(related class-diag-2 used-book has-entity)

(related class-diag-2 assoc-4 has-relationship)
(related class-diag-2 assoc-5 has-relationship)
(related class-diag-2 assoc-6 has-relationship)

(instance seq-diag-1 sequence-model)
(instance negotiate-prtcl protocol)
(instance request agent-msg)
(instance proposal agent-msg)
(instance reject-proposal agent-msg)
(instance accept-proposal agent-msg)
(instance inform-charge agent-msg)
(instance inform-payment agent-msg)
(instance negotiate-prtcl-struct sequence)

(related negotiate-protocol request has-msg)
(related negotiate-protocol proposal has-msg)
(related negotiate-protocol reject-proposal has-msg)
(related negotiate-protocol accept-proposal has-msg)
(related negotiate-protocol inform-charge has-msg)
(related negotiate-protocol inform-payment has-msg)

(related seq-model-1 request has-first)
(related request proposal is-before)
(related proposal reject-proposal is-before)
(related proposal accept-proposal is-before)
(related accept-proposal inform-charge is-before)
(related inform-charge inform-payment is-before)
(related seq-model-2 inform-payment has-last)

;;; introduzindo ERROS
;; Q1
(instance spec-3 specialization)
(related spec-3 user-agent has-end1)
(related spec-3 book has-end2)

;; Q2
(instance aggr-7 aggregation)
(related aggr-7 general-store has-end1)
(related aggr-7 2nd-hand-bookstore has-end2)

;; Q3
(instance dep-1 dependency)
(related dep-1 user-agent has-end1)
(related dep-1 book has-end2)
```

```
;; Q4
;; comentados roles de 2nd-hand-bookstore

;; Q5
;; user-agent e store-agent não foram aggregados às orgs

(instance getTitle object-msg)
(related getTitle user-agent-path has-msg-sender)
(related getTitle book-path has-msg-receiver)

(instance getAuthor object-msg)
(related getAuthor user-agent-path has-msg-sender)
(related getAuthor user-agent-path has-msg-receiver)

(instance getAbstract object-msg)
(related getAbstract book-path has-msg-sender)
(related getAbstract book-path has-msg-receiver)

(related inform-charge user-agent-path has-msg-sender)
(related inform-charge store-agent-path has-msg-receiver)

(related proposal user-agent-path has-msg-sender)
(related proposal book-path has-msg-receiver)

(instance answerProposal agent-msg)
(related answerProposal book-path has-msg-sender)
(related answerProposal user-agent-path has-msg-receiver)
```

Respostas às consultas QV1

```
(IN-KNOWLEDGE-BASE AUML-TBOX PHASE1-ABOX) --> (AUML-TBOX PHASE1-ABOX)
(IN-TBOX AUML-TBOX) --> AUML-TBOX
(TBOX-CLASSIFIED-P) --> T
(TBOX-COHERENT-P) --> T
(TBOX-CYCLIC-P) --> NIL
(RETRIEVE (?END1 ?END2 ?SPEC) (AND (?SPEC SPECIALIZATION) (?END1 ?SPEC IS-END1) (?END2 ?SPEC
IS-END2) (OR (AND (?END1 OBJECT-CLASS) (NOT (?END2 OBJECT-CLASS))) (AND (?END1 AGENT-CLASS)
(NOT (?END2 AGENT-CLASS))) (AND (NOT (?END1 OBJECT-CLASS)) (NOT (?END1 AGENT-CLASS)))))) -->
NIL
(RETRIEVE (?END1 ?END2 ?AGG) (AND (?AGG AGGREGATION) (?END1 ?AGG IS-END1) (?END2 ?AGG IS-END2)
(OR (AND (?END1 OBJECT-CLASS) (NOT (?END2 (OR AGENT-CLASS OBJECT-CLASS)))) (AND (?END1 AGENT-
CLASS) (NOT (?END2 AGENT-CLASS))) (AND (?END1 AGENT-ROLE-CLASS) (NOT (?END2 ORGANIZATION-
CLASS))) (AND (NOT (?END1 AGENT-CLASS)) (NOT (?END1 OBJECT-CLASS)) (NOT (?END1 AGENT-ROLE-
CLASS)))))) --> NIL
(RETRIEVE (?END1 ?END2 ?DEP) (AND (?DEP DEPENDENCY) (?END1 ?DEP IS-END1) (?END2 ?DEP IS-END2)
(OR (AND (?END1 OBJECT-CLASS) (NOT (?END2 OBJECT-CLASS))) (NOT (?END1 OBJECT-CLASS))))) -->
NIL
(RETRIEVE (?ORG-WITHOUT-ROLES) (AND (?ORG-WITHOUT-ROLES ORGANIZATION-CLASS) (NOT (?ORG-
WITHOUT-ROLES (SOME IS-END2 (AND AGGREGATION (SOME HAS-END1 AGENT-ROLE-CLASS))))))) --> NIL
(RETRIEVE (?ORG-WITHOUT-AGENTS) (AND (?ORG-WITHOUT-AGENTS ORGANIZATION-CLASS) (NOT (?ORG-
WITHOUT-AGENTS (SOME IS-END2 (AND AGGREGATION (SOME HAS-END1 AGENT-CLASS))))))) --> NIL
(RETRIEVE (?ORG-WITHOUT-ENV) (AND (?ORG-WITHOUT-ENV ORGANIZATION-CLASS) (NOT (?ORG-WITHOUT-ENV
(SOME IS-END2 (AND AGGREGATION (SOME HAS-END1 ENVIRONMENT-CLASS))))))) --> NIL
(RETRIEVE (?BAD-OBJ-MSG) (AND (?BAD-OBJ-MSG OBJECT-MSG) ($?SENDER ?BAD-OBJ-MSG IS-MSG-SENDER)
($?RECEIVER ?BAD-OBJ-MSG IS-MSG-RECEIVER) (OR (NOT ($?SENDER (OR OBJECT AGENT))) ($?RECEIVER
AGENT)))) --> NIL
(RETRIEVE ($?SENDER $?RECEIVER ?AG-MSG) (AND (?AG-MSG AGENT-MSG) ($?SENDER ?AG-MSG IS-MSG-
SENDER) ($?RECEIVER ?AG-MSG IS-MSG-RECEIVER) (NOT (AND ($?SENDER (AND AGENT (SOME IS-
INSTANCEOF (AND AGENT-CLASS (SOME PLAYS-ROLE AGENT-ROLE-CLASS))))) ($?RECEIVER (AND AGENT
(SOME IS-INSTANCEOF (AND AGENT-CLASS (SOME PLAYS-ROLE AGENT-ROLE-CLASS)))))) (NOT (AND
($?SENDER (AND AGENT-CLASS (SOME PLAYS-ROLE AGENT-ROLE-CLASS))) ($?RECEIVER (AND AGENT-CLASS
(SOME PLAYS-ROLE AGENT-ROLE-CLASS))))) (NOT (AND ($?SENDER AGENT-ROLE-CLASS) ($?RECEIVER
AGENT-ROLE-CLASS))))) --> NIL
(RETRIEVE (?INSTANCE) (AND (?INSTANCE CLASS-INSTANCE) (?INSTANCE NIL IS-INSTANCEOF))) --> NIL
(RETRIEVE (?PRTCL-WITHOUT-STRUCT) (AND (?PRTCL-WITHOUT-STRUCT PROTOCOL) (?PRTCL-WITHOUT-STRUCT
NIL HAS-STRUCTURE))) --> NIL
(RETRIEVE (?PRTCL-NO-MSGS) (AND (?PRTCL-NO-MSGS PROTOCOL) (?PRTCL-NO-MSG NIL HAS-MSG))) -->
NIL
(RETRIEVE (?BAD-STRUCT) (AND (?BAD-STRUCT SEQUENCE) (?BAD-STRUCT ?ELEMENT1 HAS-FIRST) (?BAD-
STRUCT ?ELEMENT2 HAS-LAST) (?ELEMENT2 ?ELEMENT1 IS-BEFORE))) --> NIL
(RETRIEVE (?SEQ-MD-WITHOUT-STRUCT) (AND (?SEQ-MD-WITHOUT-STRUCT SEQUENCE-MODEL) (?SEQ-MD-
WITHOUT-STRUCT NIL HAS-STRUCTURE))) --> NIL
```

A ontologia Ont2 para AUML

```
(in-knowledge-base AUML-phase2-TBox phase2-ABox)
(in-tbox AUML-phase2-TBox)
(signature
:atomic-concepts (;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                   ;;;;;;;  TAO concepts
                   ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                  entity
                   class
                     citizen-class
                      agent-class
                      organization-class
                      object-class
                     environment-class
                     agent-role-class
                    class-instance
                   agent
                     object
                    features
                   goal
                     belief
                     action
                     plan
                     protocol
                     attribute
                     method
                     condition
                   relationship
                      specialization
                      association
                    aggregation
                      dependency
                   msg
                    agent-msg
                    object-msg
               ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                 ;;;;;;  AUML concepts
                 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                 model
                    static-model
                   class-model
                    dynamic-model
                     sequence-model
                     activity-model
                     collaboration-model
                  path
                 agent-path
                   object-path
                  sequence
                 seq-element
                )
:roles ( ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
         ;;;;;;;;;; TAO properties
         ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
         (has-goal :range goal)
          (has-belief :range belief)
          (has-action :range action
                    :parent has-element)
         (has-precond :domain action
                      :range condition)
          (has-poscond :domain action
                      :range condition)
          (has-plan :range plan)
         (has-protocol :range protocol)
          (has-end1 :domain relationship
                  :range class
                    :inverse is-end1)
         (has-end2 :domain relationship
                  :range class
                    :inverse is-end2)
         (is-end1 :domain class
                  :range relationship
                   :inverse has-end1)
          (is-end2 :domain class
                  :range relationship
                   :inverse has-end2)
         (has-msg :range msg
                 :parent has-element)
         (has-msg-end :domain msg
                     :inverse is-msg-end)
         (is-msg-end :inverse has-msg-end)
         (has-msg-sender :parent has-msg-end)
         (has-msg-receiver :parent has-msg-end)
         (is-msg-sender :inverse has-msg-sender)
         (is-msg-receiver :inverse has-msg-receiver)
         (is-instanceOf :domain class-instance
                         :range class)
          (has-attribute :range attribute)
          (has-method :range method)
        ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
         ;;;;;;;;;; AUML properties
         ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
```

```
                  (plays-role :domain agent-class
                              :range agent-role-class)
                  (has-class :domain static-model
                            :range (or class class-instance path)
                            :inverse is-in-static-model)
                 (is-in-static-model :inverse has-class)
                 (has-relationship :domain static-model
                                   :range relationship
                                     :inverse is-relationship-of)
                  (is-relationship-of :inverse has-relationship)
                  (has-path :range path)
                 (has-head :domain path
                            :range (or class-instance class))
                  (has-tail :domain path
                            :range path)
                 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                  ;;;;;;;;;; ordering properties
                  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                  (has-structure :range sequence)
                  (has-element :range seq-element)
                  (has-first :domain sequence
                             :range seq-element
                             :parent has-element
                             :inverse is-first)
                  (is-first :inverse has-first)
                  (has-last :domain sequence
                            :range seq-element
                            :parent has-element
                            :inverse is-last)
                  (is-last :inverse has-last)
                  (is-before :domain seq-element
                             :range seq-element
                             :transitive t
                             :inverse is-after)
                  (is-after  :domain seq-element
                             :range seq-element
                             :transitive t
                             :inverse is-before)
                 )
)
;; concepts restrictions
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;; TAO concepts taxonomy
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(implies class entity)
(implies class-instance entity)

(implies environment-class class)
(implies agent-role-class class)
(implies agent-class class)
(implies organization-class class)
(implies object-class class)

(implies specialization relationship)
(implies association relationship)
(implies aggregation relationship)
(implies dependency relationship)

(implies goal features)
(implies belief features)
(implies action features)
(implies plan features)

(implies protocol features)
(implies attribute features)
(implies method features)
(implies condition features)

(implies agent class-instance)
(implies object class-instance)

(implies agent-msg msg)
(implies object-msg msg)

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;; AUML concepts taxonomy
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(implies static-model model)
(implies dynamic-model model)

(implies sequence-model dynamic-model)
(implies activity-model dynamic-model)
(implies collaboration-model dynamic-model)

(implies class-model static-model)

(equivalent path (or agent-path object-path))
```

```
;;; axiom that guarantee the possibility of the dynamic structure definition
(implies msg seq-element)
(implies action seq-element)

;;;;;;;;;;;;;;;;;;;;;;;;;;
;; ontology axioms
;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;               TAO disjoint restrictions
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(disjoint class class-instance features msg relationship)
(disjoint agent-class organization-class object-class environment-class agent-role-class)

(disjoint agent object)
(disjoint agent-msg object-msg)

(disjoint goal belief protocol attribute)
(disjoint specialization association aggregation dependency)
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;             AUML disjoint restrictions
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(disjoint static-model dynamic-model)
(disjoint sequence-model activity-model collaboration-model)
(disjoint object-path agent-path)
;;;;;;;;;;;;;;;;;;;;;;                                              ;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;      Instances-related axioms                ;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;                                              ;;;;;;;;;;;;;;;;;;;;
(implies agent (all is-instanceOf agent-class))
(implies object (all is-instanceOf object-class))

(implies agent-path
 (or agent-role-class (and agent (some is-instanceOf
                                 (and agent-class (some plays-role agent-role-class))))
      (and agent-class (some plays-role agent-role-class))))

;;;;;;                                order axioms                                ;;;;;;;
;; semantics of the first element of a sequence
(implies (and (some is-before *top*) (some is-after *bottom*)) (some is-first *top*))

;; semantics of the last element of a sequence
(implies (and (some is-after *top*) (some is-before *bottom*)) (some is-last *top*))

(implies (some has-first msg) (some has-msg msg))
(implies (some has-last msg) (some has-msg msg))

(implies (some has-first action) (some has-action action))
(implies (some has-last action) (some has-action action))

;;;;;;;;;;;;;;;;               relationships semantics

;;;;;; specialization
(implies specialization
  (or (and (some has-end1 agent-class) (all has-end2 agent-class))
      (and (some has-end1 object-class) (all has-end2 object-class))
  )
)
;;;;;; end specialization

;; association is commutative ;;;;;;;;;;;;;;;;;;;;;;
(implies association
         (or
           (and (some has-end1 object-class) (some has-end2 (or object-class))
            (and (some has-end1 citizen-class) (some has-end1 object-class))
            (and (some has-end1 object-class) (some has-end2 role-class))
           (and (some has-end1 role-class) (some has-end1 object-class))
           (and (some has-end1 role-class) (some has-end2 role-class))
          )
)
;; end association ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;; aggregation ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(implies aggregation
            (or
              (and (some has-end1 object-class) (all has-end2 object-class))
              (and (some has-end1 agent-role-class) (all has-end2 agent-role-class))
              (and (some has-end1 object-role-class) (all has-end2 object-role-class))
            )
)
;; end aggregation ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;; end of tbox ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(classify-tbox)
(tbox-classified-p)
(tbox-coherent-p)
(tbox-cyclic-p)
```