7
Referências Bibliográficas

AKSIT, M.; BERGMANS, L.; VURAL, S. An object-oriented language-database integration model: The composition-Filters approach. In: PROC. OF THE 7th EUROPEAN CONF. OBJECT-ORIENTED PROGRAMMING, Springer-Verlag Lecture Notes in Computer Science, 1992. p. 372-395.

ALEXANDER, I.; STEVENS, R. **Writing Better Requirements** Addison-Wesley, 2002. 176 p. ISBN: 0-932633-55-2.

AspectJ Project. Disponível em: http://www.eclipse.org/aspectj/. Acesso em: 7 março 2006.

BAKKER J.; TEKINERDOGAN, B.; AKSIT, M. Characterization of early aspects approaches. In: THE EARLY ASPECTS WORKSHOP, 2005.

BANIASSAD, E.; CLARKE, S. Theme: An approach for aspect-oriented analysis and design. In: PROC. OF THE 7th INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING (ICSE'04), Scotland, 2004. p. 158-167.

BREITMAN, K. K. **Evolução de Software**. Rio de Janeiro, 2000. 133p. Tese de Doutorado em Engenharia de Software - PUC-Rio.

BRICHAU, J.; HAUPT, M. Survey of aspect-oriented languages and execution models. Technical Report AOSD-Europe-VUB-01, 2005. 260 p. Disponível em: http://www.aosd-europe.net/documents/index.htm. Acesso em: em 7 março 2006.

BRITO, I.; MOREIRA, A. Integrating the NFR framework in a RE model. In: THE EARLY ASPECTS WORKSHOP AT AOSD, England, 2004.

BOOCH, G.; RUMBAUGH, J.; JAVOBSON, I. **The Unified Modeling Language User Guide**. Addison-Wesley, 1999.

CARROLL, J. et al. d'etre: capturing design history and rationale in multimedia narratives. In: HUMAN FACTORS IN COMPUTING SYSTEMS (CHI94), Boston-USA, ACM Press, 1994, p. 192-197.

C&L. Disponível em: http://sl.les.inf.puc-rio.br/cel. Acesso em: 7 março 2006.

CHAVEZ, C. **A Model-Driven approach to aspect-oriented design**. Rio de Janeiro, 2004. 305 p. Tese de Doutorado em Engenharia de Software - PUC-Rio.

CHAVEZ, C.; LUCENA, C. A Theory of Aspects for Aspect-Oriented Software Development. In: PROC. OF THE XVII BRAZILIAN SIMPOSIUM ON SOFTWARE ENGINEERING, 2003. p. 130-145.

CHEN, P. P. The Entity Relationship Model - Toward a Unified View of Data. **ACM Transactions Database Systems**, Vol.1, No. 1, p. 9-36, 1976.

CHITCHYAN, R. et al. Survey of aspect-oriented analysis and design approaches. Technical Report AOSD-Europe-ULANC-9, AOSDEurope, 2005. Disponível em: http://www.aosd-europe.net/documents/index.htm. Acesso em: 7 março 2006.

CHITCHYAN, R.; RASHID, A.; SAWYER, P. Comparing Requirement Engineering Approaches for Handling Crosscutting Concerns. In: PROC. OF THE WORKSHOP EM ENGENHARIA DE REQUISITOS (WER2005), Porto-Portugal, 2005. p 1-12.

CHUNG, L. et al. (2000). **Non-Functional Requirements in Software Engineering**, Boston: Kluwer Academic Publishers. ISBN 0-7923-8666-3.

CZARNECKI, K; EISENECKER, U. **Generative Programming: Methods, Tools, and Applications**, Addison-Wesley, 2000.

CLARKE, S. et al. Subject-oriented design: Towards improved alignment of requirements, design and code. In: PROC. OBJECT-ORIENTED PROGRAMMING, SYSTEMS, LANGUAGES AND APPLICATIONS (OOPSLA), 1999. p. 325–339.

CYSNEIROS, L. M.; LEITE, J. C. S. P.; NETO, J. M. S. A Framework for Integrating Non-Functional Requirements into Conceptual Models. **Requirements Engineering Journal**, Vol. 6, No. 2, p. 97-115, 2001, Springer-Verlag London Limited.

CYSNEIROS, L. M.; YU, E.; LEITE, J. C. S. P. Cataloguing Non-Functional Requirements as Softgoal Networks. In: PROC. OF REQUIREMENTS ENGINEERING FOR ADAPTABLE ARCHITECTURES at 11th International Requirements Engineering Conference, 2003. p.13-20.

DAVIS, A. M. **Software Requirements: Analysis and Specification**, Prentice-Hall Inc, 1990.

DIJKSTRA, E. **A Discipline of Programming**. Prentice-Hall, 1976.

DUTOIT, A. H.; PAECH, B. "Rationale Management in Software Engineering. In: S.K. Chang (Ed.), **Handbook of Software Engineering and Knowledge Engineering**. World Scientific, 2001.

EASTERBROOK, S.; NUSEIBEH, B. (1996) Using Viewpoints for Inconsistency Management. **BCS/IEE Software Engineering Journal**, Vol. 11, No.1. p. 31-43.

FELICÍSSIMO, C. H. et al. C&L: Um Ambiente para Edicao e Visualizao de Cenarios e Léxicos. In: WORKSHOP DE FERRAMENTAS DO XVIII SIMPOSIO BRASILEIRO DE ENGENHARIA DE SOFTWARE, Publicantions in CD, Brasilia, 2004. p. 43-48. ISBN 85-7669-004-7.

FERREIRA, A. B. H. **Mini Aurélio: o dicionário da língua portuguesa**, ed. 6, Editora Positivo, 2004. ISBN 85-7472-416-5.

Filman, R. E. et al. **Aspect-Oriented Software Development**, Addison-Wesley, 2005.

FINKELSTEIN, A. et al. Viewpoints: A framework for integrating multiple perspectives in system development. **International Journal of Software Engineering and Knowledge Engineering**, No. 2, 1992. p. 31-58.

FINKELSTEIN, A.; SOMMERVILLE, I. The Viewpoints FAQ. **BCS/IEE Software Engineering Journal**, Vol. 11, No. 1, 1996.

GIORGINI P. et al. Reasoning with goal models. In: PROC. OF THE 21ST INTERNATIONAL CONFERENCE ON CONCEPTUAL MODELING, 2002. p. 167-181.

GOGUEN, J. A.; LINDE, C. Techiniques for Requirements Elicitation. In: PROC. OF THE FIRST IEEE INTERNATIONAL SYMPOSIUM ON REQUIREMENTS ENGINEERING, San Diego-Ca, IEEE Computer Society Press, 1994. p 152-164.

GONZÁLES, B.; LAGUNA, M.; LEITE, J. C. S. P. Visual variability analysis with goal models. IN: PROC. OF IEEE INTERNATIONAL SYMPOSIUM ON REQUIREMENTS ENGINEERING (RE'04), Japan, 2004. p. 38-47.

GOTEL, O. C.; FINKELSTEIN, A. C. An analysis of the requirements traceability problem. In: PROC. OF THE FIRST INTERNATIONAL CONFERENCE ON REQUIREMENTS ENGINEERING (ICRE'94), IEEE Conputer Society Press., 1994. p. 94-101.

GRAPHVIZ. Disponível em: http://www.graphviz.org/. Acesso em: 7 março 2006

HARRISON, W.; OSSHER, H. Subject-Oriented Programming: A Critique of Pure Objects. In: PROC. OF THE 7th CONF. ON OBJECT- ORIENTED PROGRAMMING, SYSTEMS, LANGUAGES AND APPLICATIONS (OOPSLA93), 1993. p. 411-428.

HARRISON, W.; OSSHER, H; TARR, P. Asymmetrically vc. Symmetrically Organized Paradigms for Software Composition, Technical Report RC22685, IBM Research, 2002.

HSIA, P. et al. Formal Approach to Scenario Analysis. **IEEE Software**, vol. 11, No. 2, 1994. p. 33-41.

HYPER/J Web Page. Disponível em: http://www.research.ibm.com/hyperspace/HyperJ/HyperJ.htm. Acesso em: 7 março 2006.

IEEE Std. 830-1998; IEEE Recommended Practice for Software Requirements Specifications. Institute of Electrical and Electronics Engineers, 1998, 38p. ISBN 0-7381-0332-2.

IEEE Std 1471-2000; IEEE Recommended Practice for Architectural Description of Software-Intensive Systems; Software Engineering Standards Committee of the IEEE Computer Society; Approved 21 September 2000.

KAPLAN, M. et al. Subject-Oriented Design and the Watson Subject Compiler. In: Subjectivity Workshop at OOPSLA'96, 1996. Disponível em: http://www.research.ibm.com/sop/. Acesso em: 7 março 2006.

KICZALES, G. et al. Aspect-Oriented Programming. In: PROC. OF THE 11th EUROPEAN CONFERENCE ON OBJECT-ORIENTED PROGRAMMING, Vol. 1241 de LNCS, Springer-Verlag, 1997. p. 220-242.

KICZALES, G. et al. An Overview of AspectJ. In: EUROPEAN CONF. ON OBJECT-ORIENTED PROGRAMMING, Budapest, Hungary, 2001. p. 18-22.

KOTONYA, G.; SOMMERVILLE, I. Requirements Engineering with Viewpoints. **BCS/IEE Software Engineering Journal**, Vol. 11, No.1, 1996. p. 5-18.

KOTONYA, G.; SOMMERVILLE, I. **Requirements Engineering: Process and Techniques**. Chichester, UK: John Wiley & Sons Inc., 1998.

JACOBSON, I. et al. **Object-Oriented Software Development- A Use Case Driven Approach**, Addison Wesley, 1992.

JACOBSON, I.; BOOCH, G.; RUMBAUGH, J. **The Unified Software Development Process**, Addison-Wesley, 1999. ISBN 0-201-57169-2.

LAMSWEERDE, A.; LETIER, E. Handling obstacles in goal-oriented requirements engineering. **IEEE Transaction Software Engineering**, Vol. 26, No.10, 2000. p.978–1005.

LAMSWEERDE, A. Goal-Oriented Requirements Engineering: A Guided Tour. In: PROC. OF THE 5TH IEEE INTERNATIONAL SYMPOSIUM ON REQUIREMENTS ENGINEERING (RE'01), Toronto, 2001. p. 249-263.

LEHMAN, M. M. Laws of software evolution revisited. **Lecture Notes in Computer Science**, Vol. 1149, 1996. p.108-120.

LEITE, J. C. S. P. Viewpoint Resolution in Requirements Elicitation. Irvine, 1988. PhD Thesis, Dept. of Computer Science, University of California.

LEITE, J. C. S. P. Viewpoints analysis: a case study. **ACM Soflware Engineering Notes**, Vol. 14, No. 3, 1989. p. 111-1.

LEITE, J. C. S. P.; FRANCO, A. P. M. O Uso de Hipertexto na Elicitação de Linguagens da Aplicação. In: ANAIS DE IV SIMPÓSIO BRASILEIRO DE ENGENHARIA DE SOFTWARE, 1990. p. 134–149.

LEITE, J. C. S. P.; FREEMAN, P. Requirements Validation Through Viewpoint Resolution. **IEEE Transactions on Software Engineering**, Vol. 17, No 12, 1991.

LEITE, J. C. S. P.; OLIVEIRA, A. P. A Client Oriented Requirements Baseline. In: PROC. OF THE SECOND IEEE INTERNATIONAL SYMPOSIUM ON REQUIREMENTS ENGINEERING (RE'95), 1995. Los Alamitos: IEEE Computer Society Press, 1995. p.108-115.

LEITE, J. C. S. P. Viewpoints on Viewpoints. In: ACM JOINT PROCEEDINGS OF THE SIGSOFT'96 WORKSHOPS, ACM Press, 1996. p. 285-288.

LEITE, J. C. S. P. et al. Enhancing a requirements baseline with scenarios. In: PROC. OF THE THIRD IEEE INTERNATIONAL SYMPOSIUM ON REQUIREMENTS ENGINEERING (RE'97), IEEE Computer Society Press, 1997. p. 44-53.

LEITE, J. C. S. P. et al. Scenario Construction Process. **Requirements Engineering Journal**, Vol. 5, No. 1, 2000, Springer-Verlag London Limited. p. 38-61.

LEITE, J. C. S. P. et al. Quality-Based Software Reuse. In: PROC. OF THE CAISE 2005-LNCS 3520, 2005. p. 535-550.

LIEBERHERR, K. J.; SILVA-LEPE, I.; XIAO, C. Adaptive object-oriented programming using graph-based customization. **Communications of the ACM**, Vol. 37, No. 5, 1994. p. 94-101.

LIEBERHERR, K. J. 1996. Adaptative Object-Oriented Software: The Demeter Method with Propagation Patterns. **PWS Publishing Company**, Boston, 1996.

LOPES, C. V. AOP: A Historical Perspective. In: R. Filman et al. (eds.) **Aspect-Oriented Software Development**, Addison-Wesley, 2004. ISBN 0321219767.

MARCO, T. **Structured Analysis and Structured Specifications**. Prentice Hall, 1979.

MOREIRA, A.; ARAÚJO, J.; BRITO, I. Crosscutting quality attributes for requirements engineering. In: PROC. OF THE 14TH INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING AND KNOWLEDGE ENGINEERING (SEKE 2002), ACM Press, 2002.

MOREIRA, A.; ARAÚJO, J.; RASHID, A. A Model for Multi-Dimensional Separation of Concerns in Requirements Engineering. In: THE 17TH CONFERENCE ON ADVANCED INFORMATION SYSTEMS ENGINEERING (CAISE'05), Lecture Notes in Computer Science, Vol. 3520, Porto-Portugal, 2005.

MOREIRA, A.; RASHID, A.; ARAÚJO, J. Multi-Dimensional Separation of Concerns in Requirements Engineering. In: THE 13TH INTERNATIONAL CONFERENCE ON REQUIREMENTS ENGINEERING (RE'05), IEEE Computer Society, Paris-France, 2005. p.285-296

MYLOPOULOS, J.; CHUNG, L.; NIXON, B. Representing and using nonfunctional requirements: A process-oriented approach. **IEEE Transactions on Software Engineering**, Vol. 18, No. 6, 1992, p. 483–497.

MYLOPOULOS, J. et al. Exploring Alternatives during Requirements Analysis. **IEEE Software**, Jan/Feb 2001. p. 2-6.

NUSEIBEH, B.; KRAMER, J.; FINKELSTEIN, A. A Framework for Expressing the Relationships Between Multiple Views in Requirements Specification. **IEEE Transactions on Software Engineering,** Vol. 20, No. 10, 1994. p. 760-773.

NUSEIBEH, B. Crosscutting requirements. In: PROC. OF THE 3RD INTERNATIONAL CONF. ON ASPECT-ORIENTED SOFTWARE DEVELOPMENT (AOSD 2004), Lancaster-UK, 2004. p. 3-4. ISBN:1-58113-842-3.

OSSHER, H.; TARR, P. Using Multi-dimensional Separation of Concerns to (Re)Shape Evolving Software. **Communications of the ACM**, Vol. 44, No.10, 2001. p. 43-50.

PARK, S.; KIM, M.; SUGUMARAN, V. A scenario, goal, and feature oriented domain analysis approach for developing software product lines. **Journal on Industrial Management & Data Systems**, Vol. 104, No. 4, 2004. p. 296-308.

PARNAS, D. On the Criteria To Be Used in Decomposing Systems into Modules. **Communications of the ACM**, Vol. 15, No. 12, 1972. p. 1053-1058.

PIVETA E. K. et al. Termos em português para Desenvolvimento de Software Orientado a Aspectos. In: 1º WORKSHOP BRASILEIRO DE DESENVOLVIMENTO DE SOFTWARE ORIENTADO A ASPECTOS (WASP'04), 2004.

RASHID, A. et al. Early aspects: a model for aspect-oriented requirements engineering. In: PROC. OF IEEE JOINT CONFERENCE ON REQUIREMENTS ENGINEERING, Germany, 2002. p. 199-202.

RASHID, A.; MOREIRA, A.; ARAÚJO, J. Modularization and composition of aspectual requirements. In: PROC. OF THE 2ND INTERNATIONAL

CONFERENCE ON ASPECT-ORIENTED SOFTWARE DEVELOPMENT, ACM, 2003. p. 11-20.

ROBINSON, W. N.; PAWLOWSKI, S. D.; VOLKOV, V. Requirements Interaction Management. **ACM Computing Surveys**, Vol. 35, No. 2, 2003. p. 132-190.

ROLLAND, C. et al. A proposal for a scenario classification framework. **Journal of Requirements Engineering**, Vol. 3, Springer Verlag, 1998. p. 23-47.

ROSS, D.; SCHOMAN, A. Structured analysis for requirements definition. **IEEE Transactions on Software Engineering**, Vol. 3, No. 1, 1977. p. 6-15.

SAYÃO, M.; LEITE, J. C. S. P. Rastreabilidade de Requisitos. Relatório Técnico 20/05, série Monografias em Ciência da Computação, DI/PUC-Rio, 2005.

SILVA, L. F.; LEITE, J. C. S. P.; BREITMAN, K. K. C&L uma Ferramenta de Apoio à Engenharia de Requisitos. **Revista de Informática Teórica e Aplicada (RITA)**, Vol. 12, Número 1, 2005. p. 23-46. ISSN 0103-4308.

SILVA, L. F.; LEITE, J. C. S. P. Uma linguagem de modelagem de requisitos orientada a aspectos. In: PROC. OF THE REQUIREMENT ENGINEERING WORKSHOP, junto ao CAISE 2005, Porto-Portugal, 2005. p. 13-25.

SILVA, L. F.; LEITE, J. C. S. P. Integração de Características Transversais Durante a Modelagem de Requisitos. In: ANAIS DO SIMPÓSIO BRASILEIRO DE ENGENHARIA DE SOFTWARE (SBES-2005), Uberlândia-MG, 2005.

SILVA, L. F.; LEITE, J. C. S. P. An Aspect-Oriented Approach to Model Requirements. In: RE'05 DOCTORAL CONSORTIUM, junto ao 13th IEEE International Requirements Engineering Conference, Paris-France, 2005.

SOMMERVILLE, I. **Software Engineering**, Ed. 6th, Addison- Wesley, 2000.

SOUSA, G.; SILVA, G.; CASTRO, J. Adapting the NFR framework to aspect-oriented requirements engineering, In: PROC. OF THE 17TH BRAZILIAN SYMPOSIUM ON SOFTWARE ENGINEERING (SBES'2003), Brazil, 2003.

SOUSA, G. et al. Separation of crosscutting concerns from requirements to design: Adapting the use case driven approach. In: PROC. OF THE EARLY ASPECTS WORKSHOP AT AOSD, England, 2004.

SOUSA, G. **Uma Abordagem Direcionada a Casos de Uso para o Desenvolvimento de Software Orientado a Aspectos**. Recife, 2004, 193 p. Dissertação de Mestrado - Centro de Informática, Universidade Federal de Pernambuco.

STAA, A. V. **Programação Modular: Desenvolvendo programas complexos de forma organizada e segura.** Publicação: 05/2000, Campus. p. 128-133. ISBN: 8535206086

TARR, P. et al. N Degrees of Separation: Multi-Dimensional Separation of Concerns. In: PROC. OF THE 21ST INT'L CONF. ON SOFTWARE ENGINEERING (ICSE'99), 1999. p. 107-119.

TARR, P.; OSSHER, H. Hyper/J User and Installation Manual; 2000. Disponível em: http://www.alphaworks.ibm.com/tech/hyperj. Acesso em: 7 março 2006.

TEKINERDOĐAN, B. et al. Early aspects: aspect-oriented requirements engineering and architecture design. Report. In: EARLY ASPECTS WORKSHOP AT AOSD, England, 2004.

UNICAL, Requirements Document of Unical. University of California at Irvine. Disponível em: http://www.ics.uci.edu/~taylor/ICS_52_FQ04/unical-req.html. Acesso em: 7 março 2006.

VOLERE. Disponível em: http://www.volere.co.uk/. Acesso em: 7 março 2006.

WEIDENHAUPT, K. et al. Scenario Usage in system development: current practice. **IEEE Software**, Vol. 15, No. 2, 1998. p. 34-45.

YOU Z. **Using meta-model-driven views to address scalability in istar models.** Toronto, 2004, 225 p. Master Thesis, Graduate Department of Computer Science, University of Toronto.

YU, E. **Modelling Strategic Relationships for Process Reengineering**. Toronto, 1995, 124 p. PhD Thesis, Graduate Department of Computer Science, University of Toronto.

YU, E. Towards Modelling and Reasoning Support for Early-Phase Requirements Engineering. In: PROC. OF THE 3RD IEEE INT. SYMP. ON REQUIREMENTS ENGINEERING (RE'97), Washington D.C., USA, 1997. p. 226-235.

YU, Y.; LEITE, J.; MYLOPOULOS, J. From goals to aspects: discovering aspects from requirements goal models. In: PROC. OF IEEE INTERNATIONAL SYMPOSIUM ON REQUIREMENTS ENGINEERING (RE'04), Japan, 2004. pp. 38-47.

ZAVE, P.; JACKSON, M. Four dark corners of requirements engineering. **ACM Transaction Software Engineering Methodologics**, Vol. 6, No.1, 1997. p.1–30.

ZORMAN, L. **Requirements Envisaging through utilizing scenarios – REBUS**. 1995. Ph.D. Dissertation, University of Southern California.

Apêndice A
Gramática da Linguagem Orientada a Aspectos para Modelagem de Requisitos

Neste Apêndice apresentamos a especificação da linguagem de modelagem de requisitos orientada a aspectos, em BNF e DTD.

BNF de AOV-graph

```
<aspect_oriented_model> := <goal_model>
<goal_model> := goal_model (<name>; <id>){<component> <relationship>} | <goal_model> <goal_model>
<component> := <softgoal> |  <goal> | <task> |  < component > < component >
<softgoal>:=softgoal(<name>;<softgoal_id>;<decomposition_label>;(<type>);(<topic>))
                        {<component><relationship>}
<goal> := goal (<name>; <goal_id>; <decomposition_label> ; (<type>) ; (<topic>))
                        {<component><relationship>}
<task> := task (<name>; <task_id>; <decomposition_label> ; (<type>) (<topic>))
                        {<component><relationship>}
<topic> := <name> | <topic>; <topic>
<type> := <name> | <type>; <type>
<relationship> := <correlation_relationship> | <crosscuttingRel> |  <relationship> <relationship>
<correlation_relationship> :=
        correlation <correlation_label> {source=<goal_ref> target=<softgoal_ref>} |
        correlation <correlation_label> {source=<softgoal_ref> target=<softgoal_ref>} |
        correlation <correlation_label> {source=<task_ref> target=<task_ref>}
<correlation_label> := break | hurt | unknown | help | make
<decomposition_label>:=and | or | xor | break | hurt | unknown | help | make
<goal_ref> := <goal_id> <decomposition_label>
<softgoal_ref> := <softgoal_id> <decomposition_label>
<task_ref> := <task_id> <decomposition_label>

<joinpoint> := <softgoal_ref> | <goal_ref> | <task_ref>

<crosscuttingRel> := crosscutting <crosscutting_label>{source=<joinpoint>
                                <pointcut><advice><intertype_declaration>}
<crosscutting_label:= cross
<pointcut> := pointcut (<name>; <pointcut_id>) : <pointcut_expression> | <pointcut> <pointcut>
<pointcut_expression> := <operand> |
                not <operand> |
                <pointcut_expression> and <pointcut_expression> |
                <pointcut_expression> or <pointcut_expression>
<operand> := <primitive> (<joinpoint>) | <primitive> (<regular_expression>)
<primitive> := include | substitute
<regular_expression> := <value>; "<joinpoint>"; <atribute_type>; <path>
<atribute_type> := id | name
<advice> := advice <advice_type>: <a_it_expression> {<advice_body>} |  <advice> <advice>
<advice_type> := after | before | around
<a_it_expression> := <pointcut_id> |
            not <pointcut_id> |
            <a_it_expression> and <a_it_expression> |
            <a_it_expression> or <a_it_expression>
<intertype_declaration> := intertype <intertype_type>: <a_it_expression>
                                {<intertype_declaration_body>} |
             <intertype_declaration> <intertype_declaration>
<intertype_type> := attribute | element
<advice_body> := <joinpoint> | <advice_body>; <advice_body>
<intertype_declaration_body> := new_element_type {<new_element_type>} |
                            <joinpoint> |
                            <intertype_declaration_body> ; <intertype_declaration_body>
```

```
<new_element_type> := <name>=<value> | <new_element_type>; <new_element_type>
<text> := <letter> | <digit>
<name> := <text>
<path> :=<text>
<task_id> := <text>
<softgoal_id> := <text>
<goal_id> := <text>
<pointcut_id> := <text>
```

DTD de AOV-graph

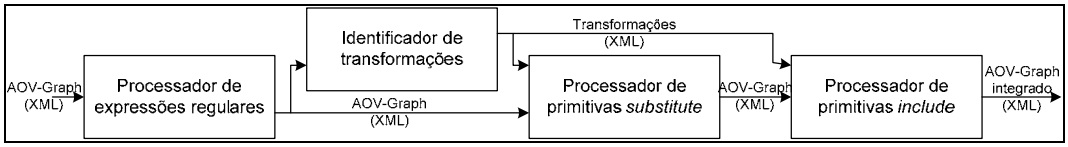
```
<!ELEMENT aspect_oriented_model (goal_model)*>
<!ELEMENT goal_model ((softgoal | goal | task | softgoal_ref | goal_ref | task_ref)*, (correlation_relationship |
crosscutting_relationship)*)>
<!ATTLIST goal_model
    id ID #REQUIRED
    name CDATA #REQUIRED
>

<!ELEMENT softgoal ( type*, topic*, (softgoal | task | softgoal_ref | task_ref | new_element_type)*,
(correlation_relationship | crosscutting_relationship)*)>
<!ATTLIST softgoal
    id ID #REQUIRED
    name CDATA #REQUIRED
    decomposition_label (and | or | break | hurt | unknown | help | make) "and"
>
<!ELEMENT goal ( type*, topic*, (goal | task | goal_ref | task_ref | new_element_type)*, (correlation_relationship |
crosscutting_relationship)*)>
<!ATTLIST goal
    id ID #REQUIRED
    name CDATA #REQUIRED
    decomposition_label (and | or | break | hurt | unknown | help | make) "and"
>
<!ELEMENT task (type*, topic*, ( task | task_ref | new_element_type)*, (correlation_relationship |
crosscutting_relationship)*)>
<!ATTLIST task
    id ID #REQUIRED
    name CDATA #REQUIRED
    decomposition_label (and | or | break | hurt | unknown | help | make) "and"
>
<!ELEMENT topic (#PCDATA)>
<!ELEMENT type (#PCDATA)>
<!ELEMENT correlation_relationship (source, target)>
<!ATTLIST correlation_relationship
    label (break | hurt | unknown | help | make) "help"
>

<!ELEMENT target (softgoal_ref | goal_ref | task_ref)>
<!ELEMENT source (softgoal_ref | goal_ref | task_ref)>
<!ELEMENT softgoal_ref EMPTY>
<!ELEMENT goal_ref EMPTY>
<!ELEMENT task_ref EMPTY>
<!ATTLIST softgoal_ref
    id IDREF #REQUIRED
    decomposition_label (and | or | break | hurt | unknown | help | make) "and"
>
<!ATTLIST goal_ref
    id IDREF #REQUIRED
    decomposition_label (and | or | break | hurt | unknown | help | make) "and"
>
<!ATTLIST task_ref
    id IDREF #REQUIRED
    decomposition_label (and | or | break | hurt | unknown | help | make) "and"
>

<!ELEMENT crosscutting_relationship (source, (pointcut | advice | intertype_declaration)*)>
<!ATTLIST crosscutting_relationship
    label (cross) “cross”
>
<!ELEMENT pointcut (pointcut_expression)>
<!ATTLIST pointcut
    id ID #REQUIRED
```

```
      name CDATA #REQUIRED
>
<!ELEMENT pointcut_expression (operand | ((operand | pointcut_expression), (and | or), (operand | pointcut_expression)) |
(not, (operand | pointcut_expression)))>
<!ELEMENT and EMPTY>
<!ELEMENT or EMPTY>
<!ELEMENT not EMPTY>
<!ELEMENT operand (softgoal_ref | goal_ref | task_ref | regular_expression)>
<!ATTLIST operand
      primitive (include | substitute) "include"
>

<!ELEMENT regular_expression (target*)>
<!ATTLIST regular_expression
      text CDATA #REQUIRED
      type (goal | sotgoal | task | any) "any"
      param (id | name) "name"
            path CDATA
>

<!ELEMENT advice (a_it_expression, advice_body)>
<!ATTLIST advice
      type (after | before | around) "after"
>
<!ELEMENT a_it_expression (pointcut_ref | ((pointcut_ref | a_it_expression), (and | or), (pointcut_ref | a_it_expression)) |
(not, (pointcut_ref | a_it_expression)))>
<!ELEMENT pointcut_ref EMPTY>
<!ATTLIST pointcut_ref
      id IDREF #REQUIRED
>
<!ELEMENT intertype_declaration (a_it_expression, intertype_declaration_body)>
<!ATTLIST intertype_declaration
      type (attribute | element) "attribute"
>
<!ELEMENT advice_body (softgoal_ref | goal_ref | task_ref)*>
<!ELEMENT intertype_declaration_body (new_element_type | softgoal | goal | task | softgoal_ref | goal_ref | task_ref)*>
<!ELEMENT new_element_type (new_element_type)*>
<!ATTLIST new_element_type
      name CDATA #REQUIRED
      value CDATA #REQUIRED
>
```

Apêndice B
Implementação dos Mecanismos de Composição e Visualização

Mecanismo de Composição

Como apresentado no Capítulo 4, o mecanismo de composição é composto por quatro componentes: processamento de expressões regulares, identificação de transformações, composição da primitiva substitute e composição da primitiva include, veja a seguir.



Componentes do mecanismo de composição

O processador de expressões regulares é responsável por criar os tipos e tópicos de todas as metas, *softmetas* e tarefas e por criar elementos target em *pointcuts* definidos por expressões regulares. Este processamento resulta em um arquivo XML que também está de acordo com a DTD definida anteriormente. Este componente exige a versão 2.0 de XSLT, pois esta inclui o tratamento de expressões regulares.

O identificador de transformações é responsável por ler os relacionamentos transversais e gerar um arquivo XML contendo onde e o que deve ser adicionado em cada *pointcut* segundo a ordem em que estas transformações devem ser aplicadas.

O processador dos elementos a serem substituídos recebe os dois arquivos XMLs anteriores e apaga todos os elementos (e seus filhos) indicados pelos *pointcuts* com primitiva substitute.

O processador dos elementos a serem incluídos também recebe os dois arquivos XMLs anteriores e inclui todos os elementos definidos em *advice* e *intertype declarations* no local indicado pelos *pointcuts* com ambas as primitivas. A implementação destes componentes está a seguir.

Composição parte 1

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="2.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
    <xsl:output method="xml" indent="yes"/>
    <xsl:template match="/">
        <xsl:apply-templates select="aspect_oriented_model"/>
    </xsl:template>

    <xsl:template match="aspect_oriented_model">
        <aspect_oriented_model>
            <xsl:apply-templates select="goal_model"/>
        </aspect_oriented_model>
    </xsl:template>

    <xsl:template match="goal_model">
        <xsl:element name="{name()}">
            <xsl:for-each select="@*">
                <xsl:attribute name="{name()}">  <xsl:value-of select="."/></xsl:attribute>
            </xsl:for-each>
            <xsl:apply-templates select="goal | softgoal | task | correlation_relationship | crosscutting_relationship"/>
        </xsl:element>
    </xsl:template>

    <xsl:template match="goal | task | softgoal">
        <xsl:element name="{name()}">
            <xsl:for-each select="@*">
                <xsl:attribute name="{name()}">  <xsl:value-of select="."/></xsl:attribute>
            </xsl:for-each>
            <xsl:call-template name="make_type"><xsl:with-param name="element_name" select="@name"/>
</xsl:call-template>
            <xsl:call-template name="make_topic"><xsl:with-param name="element_name"
select="@name"/></xsl:call-template>
            <xsl:apply-templates select="goal | softgoal | task | goal_ref | softgoal_ref | task_ref"/>
            <xsl:apply-templates select="correlation_relationship | crosscutting_relationship | new_element_type"/>
        </xsl:element>
    </xsl:template>

    <xsl:template name="make_topic">
        <xsl:param name="element_name"/>
        <xsl:variable name="regular_exp" select="$element_name"/>
        <xsl:analyze-string select="$regular_exp" regex=".*\[([^\[\]]*)\].*">
            <xsl:matching-substring>
                <xsl:element name="topic"><xsl:value-of select="regex-group(1)"/></xsl:element>
            </xsl:matching-substring>
        </xsl:analyze-string>
    </xsl:template>

    <xsl:template name="make_type">
        <xsl:param name="element_name"/>
        <xsl:if test="substring-before($element_name,' [')!=""">
            <xsl:element name="type"><xsl:value-of select="substring-before($element_name,' [')"/></xsl:element>
        </xsl:if>
    </xsl:template>

    <xsl:template match="name | or | and | not | new_element_type | correlation_relationship | advice |
intertype_declaration | source | goal_ref | softgoal_ref | task_ref">
        <xsl:copy-of select="."/>
    </xsl:template>

    <xsl:template match="name" mode="regular_expression">
        <xsl:value-of select="text()"/>
    </xsl:template>

    <xsl:template match="crosscutting_relationship">
        <xsl:element name="{name()}">
            <xsl:for-each select="@*">
                <xsl:attribute name="{name()}">  <xsl:value-of select="."/></xsl:attribute>
            </xsl:for-each>
            <xsl:apply-templates select="source | pointcut | advice | intertype_declaration"/>
        </xsl:element>
    </xsl:template>
```

```
    <xsl:template name="make_target">
        <xsl:variable name="operands" select=".//operand"/>
        <xsl:for-each select="$operands">
            <xsl:choose>
                <xsl:when test="count(goal_ref | softgoal_ref | task_ref)>0">
                    <xsl:element name="target"><xsl:copy-of select="goal_ref | softgoal_ref | task_ref"/></xsl:element>
                </xsl:when>
                <xsl:otherwise>
                    <xsl:for-each select=".//target"><xsl:element name="target"><xsl:copy-of
select="."/></xsl:element></xsl:for-each>
                </xsl:otherwise>
            </xsl:choose>
        </xsl:for-each>
    </xsl:template>

    <xsl:template match="pointcut">
        <xsl:element name="{name()}">
            <xsl:for-each select="@*">
                <xsl:attribute name="{name()}">  <xsl:value-of select="."/></xsl:attribute>
            </xsl:for-each>
            <xsl:apply-templates select="pointcut_expression"/>
        </xsl:element>
    </xsl:template>

    <xsl:template match="pointcut_expression">
        <xsl:element name="{name()}">
            <xsl:for-each select="@*">
                <xsl:attribute name="{name()}">  <xsl:value-of select="."/></xsl:attribute>
            </xsl:for-each>
            <xsl:apply-templates select="operand | pointcut_expression | or | and | not"/>
        </xsl:element>
    </xsl:template>

    <xsl:template match="operand">
        <xsl:element name="{name()}">
            <xsl:for-each select="@*">
                <xsl:attribute name="{name()}">  <xsl:value-of select="."/></xsl:attribute>
            </xsl:for-each>
            <xsl:apply-templates select="goal_ref | softgoal_ref | task_ref | regular_expression"/>
        </xsl:element>
    </xsl:template>

    <xsl:template match="regular_expression">
        <xsl:element name="{name()}">
            <xsl:for-each select="@*">
                <xsl:attribute name="{name()}">  <xsl:value-of select="."/></xsl:attribute>
            </xsl:for-each>
            <xsl:call-template name="choose_search_in"/>
        </xsl:element>
    </xsl:template>

    <xsl:template name="make_target_from_regular_expression">
        <xsl:param name="search_in"/>
    <xsl:variable name="regular_exp" select="@text"/>
    <xsl:for-each select="$search_in">
        <xsl:variable name="father" select="../."/>
        <xsl:variable name="id_father" select="$father/@id"/>
        <xsl:variable name="type" select="concat($father/name(),'_ref')"/>
        <xsl:choose>
            <xsl:when test="matches(.,$regular_exp)">
                <xsl:element name="target"><xsl:element name="{$type}">
                    <xsl:attribute name="id"><xsl:value-of select="$id_father"/></xsl:attribute>
                </xsl:element></xsl:element>
            </xsl:when>
        </xsl:choose>
    </xsl:for-each>
    </xsl:template>

    <xsl:template name="choose_search_in">
        <xsl:variable name="search_type" select="@type"/>
        <xsl:variable name="search_param" select="@param"/>
        <xsl:choose>
            <xsl:when test="$search_type='goal' and $search_param='id'"><xsl:call-template
name="make_target_from_regular_expression">
                <xsl:with-param name="search_in" select="//goal/@id"/></xsl:call-template></xsl:when>
```

```
            <xsl:when test="$search_type='goal' and $search_param='name'"><xsl:call-template
name="make_target_from_regular_expression">
                <xsl:with-param name="search_in" select="//goal/@name"/></xsl:call-template></xsl:when>
            <xsl:when test="$search_type='softgoal' and $search_param='id'"><xsl:call-template
name="make_target_from_regular_expression">
                <xsl:with-param name="search_in" select="//softgoal/@id"/></xsl:call-template></xsl:when>
            <xsl:when test="$search_type='softgoal' and $search_param='name'"><xsl:call-template
name="make_target_from_regular_expression">
                <xsl:with-param name="search_in" select="//softgoal/@name"/></xsl:call-template></xsl:when>
            <xsl:when test="$search_type='task' and $search_param='id'"><xsl:call-template
name="make_target_from_regular_expression">
                <xsl:with-param name="search_in" select="//task/@id"/></xsl:call-template></xsl:when>
            <xsl:when test="$search_type='task' and $search_param='name'"><xsl:call-template
name="make_target_from_regular_expression">
                <xsl:with-param name="search_in" select="//task/@name"/></xsl:call-template></xsl:when>
            <xsl:when test="$search_type='any' and $search_param='id'"><xsl:call-template
name="make_target_from_regular_expression">
                <xsl:with-param name="search_in" select="//goal/@id | //task/@id | //softgoal/@id"/></xsl:call-
template></xsl:when>
            <xsl:otherwise><xsl:call-template name="make_target_from_regular_expression">
                <xsl:with-param name="search_in" select="//goal/@name | //task/@name |
//softgoal/@name"/></xsl:call-template></xsl:otherwise>
        </xsl:choose>
    </xsl:template>

</xsl:stylesheet>
```

Composição parte 2

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
    <xsl:output method="xml" indent="yes" />
    <xsl:template match="/">
        <xsl:apply-templates select="aspect_oriented_model"/>
    </xsl:template>

<!-- Não estou considerando os operadores AND, OR e NOT-->
    <xsl:template match="aspect_oriented_model">
        <xsl:element name="transformations">
            <xsl:apply-templates select="goal_model"/>
        </xsl:element>
    </xsl:template>

    <xsl:template match="goal_model">
        <xsl:apply-templates select="goal | softgoal | task | crosscutting_relationship"/>
    </xsl:template>

    <xsl:template match="goal | softgoal | task">
        <xsl:apply-templates select="goal | softgoal | task | crosscutting_relationship"/>
    </xsl:template>

    <xsl:template match="crosscutting_relationship">
        <xsl:apply-templates select="advice | intertype_declaration"/>
    </xsl:template>

    <xsl:template match="advice">
        <xsl:apply-templates select="a_it_expression" mode="advice"/>
    </xsl:template>

    <xsl:template match="a_it_expression" mode="advice">

        <xsl:variable name="pointcut_ref" select=".//pointcut_ref"/>
        <xsl:variable name="father" select=".."/>
        <xsl:variable name="grandfather" select="../.."/>
        <xsl:variable name="source" select="$grandfather/source/goal_ref | $grandfather/source/softgoal_ref |
$grandfather/source/task_ref"/>

        <xsl:for-each select="$pointcut_ref">
            <xsl:variable name="pointcut_ref_actual" select="."/>
            <xsl:variable name="pointcut" select="$grandfather/pointcut[@id=$pointcut_ref_actual/@id]"/>
            <xsl:variable name="operand" select="$pointcut//operand"/>
            <xsl:for-each select="$operand">
                <xsl:variable name="operand_primitive" select="./@primitive"/>
                <xsl:variable name="where" select=".//goal_ref | .//task_ref | .//softgoal_ref"/>
```

```
                    <xsl:choose>
                        <xsl:when test="$father/@type='after' or $father/@type='before'">
                            <xsl:for-each select="$where">
                                <xsl:call-template name="make_transformation_table">
                                    <xsl:with-param name="type" select="'advice'"/>
                                    <xsl:with-param name="source" select="$source"/>
                                    <xsl:with-param name="how" select="$operand_primitive"/>
                                    <xsl:with-param name="when" select="$father/@type"/>
                                    <xsl:with-param name="where_inicial" select="."/>
                                    <xsl:with-param name="where_final" select="."/>
                                    <xsl:with-param name="what" select="$father//goal_ref | $father//task_ref |
$father//softgoal_ref"/>
                                </xsl:call-template>
                            </xsl:for-each>
                        </xsl:when>
                        <xsl:otherwise>
                            <xsl:call-template name="make_transformation_table">
                                <xsl:with-param name="type" select="'advice'"/>
                                <xsl:with-param name="source" select="$source"/>
                                <xsl:with-param name="how" select="$operand_primitive"/>
                                <xsl:with-param name="when" select="$father/@type"/>
                                <xsl:with-param name="where_inicial" select="$where"/>
                                <xsl:with-param name="where_final" select="$where"/>
                                <xsl:with-param name="what" select="$father//goal_ref | $father//task_ref |
$father//softgoal_ref"/>
                            </xsl:call-template>
                        </xsl:otherwise>
                    </xsl:choose>
                </xsl:for-each>
            </xsl:for-each>
        </xsl:template>


        <xsl:template match="a_it_expression" mode="intertype_declaration">
            <xsl:variable name="father" select=".."/>
            <xsl:variable name="pointcut_ref" select=".//pointcut_ref"/>
            <xsl:variable name="grandfather" select="../.."/>
            <xsl:variable name="source" select="$grandfather/source/goal_ref | $grandfather/source/softgoal_ref |
$grandfather/source/task_ref"/>

            <xsl:for-each select="$pointcut_ref">
                <xsl:variable name="pointcut_ref_actual" select="."/>
                <xsl:variable name="pointcut" select="$grandfather/pointcut[@id=$pointcut_ref_actual/@id]"/>
                <xsl:variable name="operand" select="$pointcut//operand"/>
                <xsl:for-each select="$operand">
                    <xsl:variable name="operand_primitive" select="./@primitive"/>
                    <xsl:call-template name="make_transformation_table">
                        <xsl:with-param name="type" select="'intertype_declaration'"/>
                        <xsl:with-param name="source" select="$source"/>
                        <xsl:with-param name="how" select="$operand_primitive"/>
                        <xsl:with-param name="when" select="$father/@type"/>
                        <xsl:with-param name="where_inicial" select=".//goal_ref | .//task_ref | .//softgoal_ref"/>
                        <xsl:with-param name="where_final" select=".//goal_ref | .//task_ref | .//softgoal_ref"/>
                        <xsl:with-param name="what" select="$father//goal | $father//task | $father//softgoal |
$father//new_element_type"/>
                    </xsl:call-template>
                </xsl:for-each>
            </xsl:for-each>
        </xsl:template>

        <xsl:template match="intertype_declaration">
            <xsl:apply-templates select="a_it_expression" mode="intertype_declaration"/>
        </xsl:template>

        <xsl:template name="make_transformation_table">
            <xsl:param name="type"/>
            <xsl:param name="source"/>
            <xsl:param name="how"/>
            <xsl:param name="when"/>
            <xsl:param name="where_inicial"/>
            <xsl:param name="where_final"/>
            <xsl:param name="what"/>

            <xsl:choose>
                <xsl:when test="$when='element' and $how='include'">
                    <xsl:element name="transformation">
```

```
                    <xsl:attribute name="type"><xsl:value-of select="$type"/></xsl:attribute>
                    <xsl:element name="how"><xsl:value-of select="$how"/></xsl:element>
                    <xsl:element name="when"><xsl:value-of select="$when"/></xsl:element>
                    <xsl:element name="where_inicial"><xsl:copy-of select="$where_inicial"/></xsl:element>
                    <xsl:element name="where_final"><xsl:copy-of select="$source"/></xsl:element>
                    <xsl:element name="what"><xsl:copy-of select="$what"/></xsl:element>
                </xsl:element>
            </xsl:when>
            <xsl:when test="$when='element' and $how='substitute'">
                <xsl:element name="transformation">
                    <xsl:attribute name="type"><xsl:value-of select="$type"/></xsl:attribute>
                    <xsl:element name="how"><xsl:value-of select="'include'"/></xsl:element>
                    <xsl:element name="when"><xsl:value-of select="$when"/></xsl:element>
                    <xsl:element name="where_inicial"><xsl:copy-of select="$where_inicial"/></xsl:element>
                    <xsl:element name="where_final"><xsl:copy-of select="$source"/></xsl:element>
                    <xsl:element name="what"><xsl:copy-of select="$what"/></xsl:element>
                </xsl:element>
            </xsl:when>
        </xsl:choose>
        <xsl:if test="$how='substitute'"> <!-- este elemento sevirá para composition_part3.xsl apagar todos os elementos
substituídos -->
                <xsl:element name="transformation">
                    <xsl:attribute name="type"><xsl:value-of select="$type"/></xsl:attribute>
                    <xsl:element name="how"><xsl:value-of select="'delete'"/></xsl:element>
                    <xsl:element name="when"><xsl:value-of select="$when"/></xsl:element>
                    <xsl:element name="where_inicial"><xsl:copy-of select="$where_inicial"/></xsl:element>
                    <xsl:element name="where_final"><xsl:copy-of select="$where_final"/></xsl:element>
                    <xsl:element name="what"><xsl:copy-of select="$what"/></xsl:element>
                </xsl:element>
        </xsl:if>

        <xsl:element name="transformation">
            <xsl:attribute name="type"><xsl:value-of select="$type"/></xsl:attribute>
            <xsl:element name="how"><xsl:value-of select="$how"/></xsl:element>
            <xsl:element name="when"><xsl:value-of select="$when"/></xsl:element>
            <xsl:element name="where_inicial"><xsl:copy-of select="$where_inicial"/></xsl:element>

            <xsl:element name="where_final">
                <xsl:choose>

                    <xsl:when test="($when='before' or $when='after') and $how='include'">
                        <xsl:variable name="Y" select="//goal[@id=$where_inicial/@id] |
//softgoal[@id=$where_inicial/@id] | //task[@id=$where_inicial/@id]"/>
                        <xsl:for-each select="$Y">
                            <xsl:variable name="X" select=".."/>
                            <xsl:for-each select="$X">
                                <xsl:variable name="element_name" select="concat(name(),'_ref')"/>
                                <xsl:element name="{$element_name}">
                                    <xsl:attribute name="id"><xsl:value-of select="@id"/></xsl:attribute>
                                    <xsl:attribute name="decomposition_label"><xsl:value-of
select="@decomposition_label"/></xsl:attribute>
                                </xsl:element>
                            </xsl:for-each></xsl:for-each>
                    </xsl:when>

                    <xsl:when test="$when='around' and $how='include'">
                        <xsl:copy-of select="$where_final"/>
                    </xsl:when>

                    <xsl:when test="$type='advice' and $how='substitute'">
                        <xsl:variable name="Y" select="//goal[@id=$where_inicial/@id] |
//softgoal[@id=$where_inicial/@id] | //task[@id=$where_inicial/@id]"/>
                        <xsl:for-each select="$Y">
                            <xsl:variable name="X" select=".."/>
                            <xsl:for-each select="$X">
                                <xsl:variable name="element_name" select="concat(name(),'_ref')"/>
                                <xsl:element name="{$element_name}">
                                    <xsl:attribute name="id"><xsl:value-of select="@id"/></xsl:attribute>
                                    <xsl:attribute name="decomposition_label"><xsl:value-of
select="@decomposition_label"/></xsl:attribute>
                                </xsl:element>
                            </xsl:for-each></xsl:for-each>
                    </xsl:when>

                    <xsl:when test="$when='attribute' and $how='include'">
                        <xsl:copy-of select="$where_final"/>
```

```
                    </xsl:when>

                    <xsl:when test="$when='element' and $how='include'">
                        <xsl:copy-of select="$where_final"/>
                    </xsl:when>

                    <xsl:when test="$when='attribute' and $how='substitute'">
                        <xsl:copy-of select="$where_final"/>
                    </xsl:when>

                    <xsl:when test="$when='element' and $how='substitute'">
                    <!-- estes for-eachs servem apenas para eu conseguir levar o ponteiro para o elemento que eu quero--
>
                        <xsl:variable name="Y" select="//goal[@id=$where_inicial/@id] |
//softgoal[@id=$where_inicial/@id] | //task[@id=$where_inicial/@id]"/>
                        <xsl:for-each select="$Y">
                            <xsl:variable name="X" select=".."/>
                            <xsl:for-each select="$X">
                                <xsl:variable name="element_name" select="concat(name(),'_ref')"/>
                                <xsl:element name="{$element_name}">
                                    <xsl:attribute name="id"><xsl:value-of select="@id"/></xsl:attribute>
                                    <xsl:attribute name="decomposition_label"><xsl:value-of
select="@decomposition_label"/></xsl:attribute>
                                </xsl:element>
                            </xsl:for-each></xsl:for-each>
                    </xsl:when>

                </xsl:choose>
            </xsl:element>

            <xsl:choose>
                <xsl:when test="$when='element'">
                    <xsl:element name="what">
                        <xsl:for-each select="$what">
                            <xsl:variable name="element_name" select="concat(name(),'_ref')"/>
                            <xsl:element name="{$element_name}">
                                <xsl:attribute name="id"><xsl:value-of select="@id"/></xsl:attribute>
                                <xsl:attribute name="decomposition_label"><xsl:value-of
select="@decomposition_label"/></xsl:attribute>
                            </xsl:element>
                        </xsl:for-each>
                    </xsl:element>
                </xsl:when>
                <xsl:otherwise><xsl:element name="what"><xsl:copy-of
select="$what"/></xsl:element></xsl:otherwise>
            </xsl:choose>
        </xsl:element>
    </xsl:template>

</xsl:stylesheet>
```

Composição parte 3

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
    <xsl:output method="xml" indent="yes"/>
    <xsl:template match="/">
        <xsl:apply-templates select="aspect_oriented_model"/>
    </xsl:template>

    <xsl:template match="aspect_oriented_model">
        <aspect_oriented_model>
            <xsl:apply-templates select="goal_model"/>
        </aspect_oriented_model>
    </xsl:template>

    <xsl:template match="goal_model">
        <xsl:element name="{name()}">
            <xsl:for-each select="@*">
                <xsl:attribute name="{name()}">   <xsl:value-of select="."/></xsl:attribute>
            </xsl:for-each>
            <xsl:apply-templates select="goal | softgoal | task | softgoal_ref | goal_ref | task_ref | correlation_relationship |
crosscutting_relationship"/>
        </xsl:element>
```

```
    </xsl:template>

    <xsl:template match="goal | task | softgoal | softgoal_ref | goal_ref | task_ref">

        <xsl:variable name="actual_element" select="./@id"/>
        <xsl:variable name="filename" select="'output_part2.xml'"/>
        <xsl:variable name="all_transformation" select="document($filename)//transformation"/>
        <xsl:variable name="transformations" select="$all_transformation[where_final/goal_ref/@id=$actual_element] |
$all_transformation[where_final/softgoal_ref/@id=$actual_element] |
$all_transformation[where_final/task_ref/@id=$actual_element]"/>

        <xsl:variable name="transformations_delete" select="$transformations/how[text()='delete']"/>

            <xsl:if test="count($transformations_delete)=0 or
count($transformations_delete)=count($transformations_delete/when[text()='attribute'])">
                <xsl:element name="{name()}">
                    <xsl:for-each select="@*">
                        <xsl:attribute name="{name()}">  <xsl:value-of select="."/></xsl:attribute>
                    </xsl:for-each>
                    <xsl:apply-templates select=" topic | type | goal | softgoal | task | softgoal_ref | goal_ref | task_ref |
correlation_relationship | crosscutting_relationship | new_element_type"/>
                </xsl:element>
            </xsl:if>
    </xsl:template>

    <xsl:template match="topic | type | crosscutting_relationship">
        <xsl:copy-of select="."/>
    </xsl:template>

    <xsl:template match="correlation_relationship">
        <xsl:variable name="source" select=".//@id"/>
        <xsl:variable name="target" select=".//@id"/>
        <xsl:variable name="filename" select="'output_part2.xml'"/>
        <xsl:variable name="all_transformation" select="document($filename)//transformation"/>
        <xsl:variable name="transformations" select="$all_transformation[where_final//@id=$source] |
$all_transformation[where_final//@id=$target]"/>

        <xsl:variable name="transformations_delete" select="$transformations/how[text()='delete']"/>
        <xsl:if test="count($transformations_delete)=0">
            <xsl:copy-of select="."/>
        </xsl:if>
    </xsl:template>

    <xsl:template match="new_element_type">
        <xsl:variable name="name" select="./@name"/>
        <xsl:variable name="value" select="./@value"/>
        <xsl:variable name="filename" select="'output_part2.xml'"/>
        <xsl:variable name="all_transformation" select="document($filename)//transformation"/>
        <xsl:variable name="transformations_delete" select="$all_transformation/how[text()='delete']"/>

        <xsl:variable name="transformations" select="$transformations_delete/what[@name=$name and
@value=$value]"/>
        <xsl:if test="count($transformations)=0">
            <xsl:copy-of select="."/>
        </xsl:if>
    </xsl:template>
</xsl:stylesheet>
```

Composição parte 4

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
    <xsl:output method="xml" indent="yes" />
    <xsl:template match="/">
        <xsl:apply-templates select="aspect_oriented_model"/>
    </xsl:template>

    <xsl:template match="aspect_oriented_model">
        <aspect_oriented_model>
            <xsl:apply-templates select="goal_model"/>
        </aspect_oriented_model>
    </xsl:template>

    <xsl:template match="goal_model">
```

```
            <xsl:element name="{name()}">
                <xsl:for-each select="@*">
                    <xsl:attribute name="{name()}">   <xsl:value-of select="."/></xsl:attribute>
                </xsl:for-each>
                <xsl:apply-templates select="goal | softgoal | task | softgoal_ref | goal_ref | task_ref | correlation_relationship |
crosscutting_relationship"/>
            </xsl:element>
        </xsl:template>

        <xsl:template match="goal | task | softgoal">
            <xsl:variable name="actual_element" select="./@id"/>
            <xsl:variable name="filename" select="'output_part2.xml'"/>
            <xsl:variable name="all_transformation" select="document($filename)//transformation"/>
            <xsl:variable name="transformations_inicial"
select="$all_transformation[where_inicial//@id=$actual_element]"/>
            <xsl:variable name="transformations_final" select="$all_transformation[where_final//@id=$actual_element]"/>
            <xsl:variable name="transformations_include_inicial" select="$transformations_inicial[how/text()='include']"/>

            <xsl:variable name="transformations_advice" select="$transformations_include_inicial[@type='advice']"/>
            <xsl:variable name="transformations_before" select="$transformations_advice[when/text()='before']"/>
            <xsl:variable name="transformations_after" select="$transformations_advice[when/text()='after']"/>

            <xsl:for-each select="$transformations_before"> <!-- include advice before -->
                <xsl:copy-of select="./what/goal_ref | ./what/softgoal_ref | ./what/task_ref"/>
            </xsl:for-each>

            <xsl:element name="{name()}">
                <xsl:for-each select="@*">
                    <xsl:attribute name="{name()}">   <xsl:value-of select="."/></xsl:attribute>
                </xsl:for-each>
                <xsl:apply-templates select="topic | type"/>
                <xsl:apply-templates select="goal | softgoal | task | softgoal_ref | goal_ref | task_ref"/>
                <xsl:for-each select="$transformations_final">
                    <xsl:copy-of select="./what/goal_ref | ./what/softgoal_ref | ./what/task_ref | ./what/goal | ./what/softgoal |
./what/task | ./what/new_element_type"/>
                </xsl:for-each>
                <xsl:apply-templates select="correlation_relationship | crosscutting_relationship | new_element_type"/>
            </xsl:element>

            <xsl:for-each select="$transformations_after">
                <xsl:copy-of select="./what/goal_ref | ./what/softgoal_ref | ./what/task_ref"/>
            </xsl:for-each>

        </xsl:template>

<!-- esta função não está sendo utilizada pq eu troquei pelo FOR anterior ao FOR do Include_advice_after-->
        <xsl:template name="process_transformation_table">
            <xsl:variable name="actual_element" select="./@id"/>
            <xsl:variable name="filename" select="'../xml/Output_part2.xml'"/>
            <xsl:for-each select="document($filename)//transformation/where_final/goal_ref[@id=$actual_element] |
document($filename)//transformation/where_final/softgoal_ref[@id=$actual_element] |
document($filename)//transformation/where_final/task_ref[@id=$actual_element]">
                <xsl:variable name="grandfather" select="../../."/>
                <xsl:variable name="element_type" select="$grandfather/@type"/>
                <xsl:variable name="primitive" select="$grandfather/how"/>
                <xsl:variable name="type" select="$grandfather/when"/>
                <xsl:variable name="body" select="$grandfather/what/goal_ref | $grandfather/what/softgoal_ref
|$grandfather/what/task_ref | $grandfather/what/new_element_type | $grandfather/what/goal | $grandfather/what/softgoal
|$grandfather/what/task"/>
                <xsl:choose> <!-- Deixei separado pq pode surgir alguma diferenciação entre as ações a seguir -->
                    <xsl:when test="$element_type='intertype_declaration' and $primitive='include'">  <!-- OK -->
                        <xsl:copy-of select="$body"/>
                    </xsl:when>
                    <xsl:when test="$type='element' and $primitive='substitute'">      <!-- o Delete é feito em
Composition_part3.xsl -->
                        <xsl:copy-of select="$body"/>
                    </xsl:when>
                    <xsl:when test="$type='attribute' and $primitive='substitute'">      <!-- o Delete é feito em
Composition_part3.xsl -->
                        <xsl:copy-of select="$body"/>
                    </xsl:when>
                    <xsl:when test="$type='around' and $primitive='include'">    <!-- OK -->
                        <xsl:copy-of select="$body"/>
                    </xsl:when>
                    <xsl:when test="$type='lili' and $primitive='include'">    <!-- OK  -->
                        <xsl:copy-of select="$body"/>
```

```
                </xsl:when>
                <xsl:when test="$type='lili' and $primitive='include'">    <!-- OK-->
                    <xsl:copy-of select="$body"/>
                </xsl:when>
                <xsl:when test="$element_type='advice' and $primitive='substitute'"> <!-- o Delete é feito em
Composition_part3.xsl -->
                    <xsl:copy-of select="$body"/>
                </xsl:when>
            </xsl:choose>
        </xsl:for-each>
    </xsl:template>
    <xsl:template match="topic | type | softgoal_ref | goal_ref | task_ref | crosscutting_relationship | new_element_type |
correlation_relationship">
        <xsl:copy-of select="."/>
    </xsl:template>
</xsl:stylesheet>
```

Mecanismo de Visualização

O mecanismo de visualização gera visões de AOV-graph. Cada visão é gerada por um arquivo XSLT que implementa separadamente as transformações definidas no Capítulo 4. O formato DOT é utilizado juntamente com o software Visigraph (Visigraph, 2006) para gerar figuras (JPG).

AOV-Graph com relacionamentos transversais resumidos

```
<?xml version="1.0" encoding="UTF-8"?> <!-- este aquivo deve ser aplicado ao XML inicial (antes da composiÃ§Ã£o)
para representar o diagrama com os crosscutting relationships resumidos -->
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
    <xsl:output method="text" indent="yes"/>
    <xsl:template match="/">
        <xsl:apply-templates select="aspect_oriented_model"/>
    </xsl:template>

    <xsl:template match="aspect_oriented_model">
digraph aspect_oriented_goal_model{
        rankdir=RL;
        fontsize=12;
        ranksep=.5;
        nodesep=.10;

        <xsl:apply-templates select="goal_model"/>
        <xsl:call-template name="decomposition_relationship_references"/>
        <xsl:call-template name="correlation_relationship"/>
        <xsl:call-template name="crosscutting_relationship"/>
    }
    </xsl:template>

    <xsl:template match="goal_model">
        subgraph <xsl:value-of select="concat('cluster_',./@id)"/> {
            label = "<xsl:value-of select="./@name"/>";
            color = tomato;

            <xsl:apply-templates select="goal | softgoal | task "/>
        }
    </xsl:template>

    <xsl:template match="goal">
        "<xsl:value-of select="@id"/>" [heigth=.3, label="<xsl:value-of select="@name"/>", shape=octagon];
        <xsl:apply-templates select="goal | softgoal | task"/>
        <xsl:call-template name="decomposition_relationship"/>
    </xsl:template>
    <xsl:template match="task">
        "<xsl:value-of select="@id"/>" [heigth=.3, label="<xsl:value-of select="@name"/>", shape=hexagon];
        <xsl:apply-templates select="goal | softgoal | task "/>
        <xsl:call-template name="decomposition_relationship"/>
    </xsl:template>
    <xsl:template match="softgoal">
```

```
        "<xsl:value-of select="@id"/>" [heigth=.3, label="<xsl:value-of select="@name"/>",
shapefile="NFRSoftgoal.gif"];
        <xsl:apply-templates select="goal | softgoal | task "/>
        <xsl:call-template name="decomposition_relationship"/>
    </xsl:template>

    <xsl:template name="crosscutting_relationship">
        <xsl:for-each select="//crosscutting_relationship">
        <xsl:variable name="target" select=".//operand/goal_ref/@id | .//operand/softgoal_ref/@id |
.//operand/task_ref/@id"/>
        <xsl:variable name="target2" select=".//operand/regular_expression//@id"/>
        <xsl:variable name="source" select=".//source//@id"/>
        <xsl:for-each select="$target">
            <xsl:variable name="x" select="."/>
             "<xsl:value-of select="$source"/>" -&gt; "<xsl:value-of select="$x"/>" [arrowhead=normal, label="cross",
style=bold];
        </xsl:for-each>

<!--    <xsl:for-each select="$target2"> Ã© necessÃ¡rio retirar apenas os que se referem a inter-types
            <xsl:variable name="x" select="."/>
             "<xsl:value-of select="$source"/>" -&gt; "<xsl:value-of select="$x"/>" [arrowhead=normal, label="cross",
color=lightgray];
        </xsl:for-each> -->
        </xsl:for-each>
    </xsl:template>

    <xsl:template name="correlation_relationship">
        <xsl:for-each select="//correlation_relationship">
             "<xsl:value-of select="./source//@id"/>" -&gt; "<xsl:value-of select="./target//@id"/>" [arrowhead=normal,
label="<xsl:value-of select="@label"/>", style=dotted];
        </xsl:for-each>
    </xsl:template>

    <xsl:template name="decomposition_relationship">
        <xsl:for-each select="./goal | ./task | ./softgoal">
             "<xsl:value-of select="@id"/>" -&gt; "<xsl:value-of select="../@id"/>" [arrowhead=normal,
label="<xsl:value-of select="@decomposition_label"/>"];
         </xsl:for-each>
    </xsl:template>

    <xsl:template name="decomposition_relationship_references">
    <xsl:for-each select="//goal/goal_ref | //goal/task_ref | //goal/softgoal_ref | //task/goal_ref | //task/task_ref |
//task/softgoal_ref | //softgoal/goal_ref | //softgoal/task_ref | //softgoal/softgoal_ref">
             "<xsl:value-of select="@id"/>" -&gt; "<xsl:value-of select="../@id"/>" [arrowhead=normal,
label="<xsl:value-of select="@decomposition_label"/>"];
         </xsl:for-each>
    </xsl:template>
</xsl:stylesheet>
```

AOV-Graph com relacionamentos transversais expandidos

```
<?xml version="1.0" encoding="UTF-8"?> <!-- este aquivo deve ser aplicado ao XML final (depois da composicao) para
representar o diagrama com os crosscutting relationships expandidos -->
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
    <xsl:output method="text" indent="yes"/>
    <xsl:template match="/">
        <xsl:apply-templates select="aspect_oriented_model"/>
    </xsl:template>

    <xsl:template match="aspect_oriented_model">
digraph aspect_oriented_goal_model{
        rankdir=RL;
        ranksep=.5;
        nodesep=.10;
        fontsize=12;

        <xsl:apply-templates select="goal_model"/>

        <xsl:call-template name="decomposition_relationship_references"/>
        <xsl:call-template name="correlation_relationship"/>
    }
    </xsl:template>

    <xsl:template match="goal_model">
```

```
        subgraph <xsl:value-of select="concat('cluster_',./@id)"/> {
            label = "<xsl:value-of select="./@name"/>";
            color = tomato;
            <xsl:for-each select=".//softgoal">"<xsl:value-of select="@id"/>" [heigth=.3, label="<xsl:value-of
select="@name"/>", shapefile="NFRSoftgoal.gif"];
            </xsl:for-each>
            <xsl:for-each select=".//goal">"<xsl:value-of select="@id"/>" [heigth=.3, label="<xsl:value-of
select="@name"/>", shape=octagon];
            </xsl:for-each>
            <xsl:for-each select=".//task">"<xsl:value-of select="@id"/>" [heigth=.3, label="<xsl:value-of
select="@name"/>", shape=hexagon];
            </xsl:for-each>
            <xsl:apply-templates select="goal | softgoal | task "/>
        }
    </xsl:template>

    <xsl:template match="goal">
        <xsl:apply-templates select="goal | softgoal | task"/>
        <xsl:call-template name="decomposition_relationship"/>
    </xsl:template>
    <xsl:template match="task">
        <xsl:apply-templates select="goal | softgoal | task "/>
        <xsl:call-template name="decomposition_relationship"/>
    </xsl:template>
    <xsl:template match="softgoal">
        <xsl:apply-templates select="goal | softgoal | task "/>
        <xsl:call-template name="decomposition_relationship"/>
    </xsl:template>

    <xsl:template name="crosscutting_relationship">
        <xsl:for-each select="//crosscutting_relationship">
        <xsl:variable name="target" select=".//operand/goal_ref/@id | .//operand/softgoal_ref/@id |
.//operand/task_ref/@id"/>
        <xsl:variable name="target2" select=".//operand/regular_expression//@id"/>
        <xsl:variable name="source" select=".//source//@id"/>
        <xsl:for-each select="$target">
            <xsl:variable name="x" select="."/>
             "<xsl:value-of select="$source"/>" -&gt; "<xsl:value-of select="$x"/>" [arrowhead=normal, label="cross",
style=bold];
        </xsl:for-each>

        <xsl:for-each select="$target2">
            <xsl:variable name="x" select="."/>
             "<xsl:value-of select="$source"/>" -&gt; "<xsl:value-of select="$x"/>" [arrowhead=normal, label="cross",
color=lightgray];
        </xsl:for-each>
        </xsl:for-each>
    </xsl:template>

    <xsl:template name="correlation_relationship">
        <xsl:for-each select="//correlation_relationship">
             "<xsl:value-of select="./source//@id"/>" -&gt; "<xsl:value-of select="./target//@id"/>" [arrowhead=normal,
label="<xsl:value-of select="@label"/>", style=dotted];
        </xsl:for-each>
    </xsl:template>

    <xsl:template name="decomposition_relationship">
        <xsl:for-each select="./goal | ./task | ./softgoal">
             "<xsl:value-of select="@id"/>" -&gt; "<xsl:value-of select="../@id"/>" [arrowhead=normal,
label="<xsl:value-of select="@decomposition_label"/>"];
         </xsl:for-each>
    </xsl:template>

    <xsl:template name="decomposition_relationship_references">
    <xsl:for-each select="//goal/goal_ref | //goal/task_ref | //goal/softgoal_ref | //task/goal_ref | //task/task_ref |
//task/softgoal_ref | //softgoal/goal_ref | //softgoal/task_ref | //softgoal/softgoal_ref">
             "<xsl:value-of select="@id"/>" -&gt; "<xsl:value-of select="../@id"/>" [arrowhead=normal, style=bold,
label="<xsl:value-of select="@decomposition_label"/>"];
         </xsl:for-each>
    </xsl:template>
</xsl:stylesheet>
```

Diagrama de classes

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
    <xsl:output method="xml" indent="yes" />
    <xsl:template match="/">
        <xsl:apply-templates select="aspect_oriented_model"/>
    </xsl:template>

    <xsl:template match="aspect_oriented_model">
        <xsl:element name="entities">
            <xsl:apply-templates select="goal_model"/>
        </xsl:element>
    </xsl:template>

    <xsl:key name="topics" match="topic" use="text()"/>

    <xsl:template name="is_class">
        <xsl:param name="topic"/>
        <xsl:variable name="son" select="//goal[topic=$topic]/goal | //softgoal[topic=$topic]/goal |
//task[topic=$topic]/goal | //goal[topic=$topic]/softgoal | //softgoal[topic=$topic]/softgoal | //task[topic=$topic]/softgoal |
//goal[topic=$topic]/task | //softgoal[topic=$topic]/task | //task[topic=$topic]/task | //task[topic=$topic]/new_element_type |
//goal[topic=$topic]/new_element_type | //softgoal[topic=$topic]/new_element_type"/>
            <xsl:choose>
                <xsl:when test="count($son)>0"><xsl:value-of select="'true'"/></xsl:when>
                <xsl:otherwise><xsl:value-of select="'false'"/></xsl:otherwise>
            </xsl:choose>
    </xsl:template>

    <xsl:template name="is_class_ref">
        <xsl:param name="topic"/>
        <xsl:variable name="son" select="//goal[topic=$topic]/goal | //softgoal[topic=$topic]/goal |
//task[topic=$topic]/goal | //goal[topic=$topic]/softgoal | //softgoal[topic=$topic]/softgoal | //task[topic=$topic]/softgoal |
//goal[topic=$topic]/task | //softgoal[topic=$topic]/task | //task[topic=$topic]/task |        //goal[topic=$topic]/goal_ref |
//softgoal[topic=$topic]/goal_ref | //task[topic=$topic]/goal_ref | //goal[topic=$topic]/softgoal_ref |
//softgoal[topic=$topic]/softgoal_ref | //task[topic=$topic]/softgoal_ref | //goal[topic=$topic]/task_ref |
//softgoal[topic=$topic]/task_ref | //task[topic=$topic]/task_ref | //task[topic=$topic]/new_element_type |
//goal[topic=$topic]/new_element_type | //softgoal[topic=$topic]/new_element_type"/>
            <xsl:choose>
                <xsl:when test="count($son)>0"><xsl:value-of select="'true'"/></xsl:when>
                <xsl:otherwise><xsl:value-of select="'false'"/></xsl:otherwise>
            </xsl:choose>
    </xsl:template>

    <xsl:template match="goal_model">
    <xsl:element name="goal_model">
        <xsl:attribute name="name"><xsl:value-of select="@name"/></xsl:attribute>
        <xsl:apply-templates select="goal | softgoal | task | goal_ref | softgoal_ref | task_ref | crosscutting_relationship"/>
    </xsl:element>
    </xsl:template>

    <xsl:template match="goal | softgoal | task">
    <xsl:variable name="id_element_actual" select="./@id"/>
    <xsl:for-each select="./topic">
        <xsl:variable name="is_class"><xsl:call-template name="is_class"><xsl:with-param name="topic"
select="."/></xsl:call-template></xsl:variable>
        <xsl:if test="$is_class='true'">
            <xsl:element name="entity">
                <xsl:attribute name="name"><xsl:value-of select="."/></xsl:attribute>
                <xsl:element name="method"><xsl:call-template name="make_element"><xsl:with-param
name="element" select="../."/></xsl:call-template></xsl:element>
                <xsl:apply-templates select="../goal | ../softgoal | ../task | ../goal_ref | ../softgoal_ref | ../task_ref |
../crosscutting_relationship  | ../new_element_type"/>
                <xsl:call-template name="correlation_relationship"><xsl:with-param name="element"
select="$id_element_actual"/></xsl:call-template>
            </xsl:element>
        </xsl:if>
        <xsl:if test="$is_class!='true'">
            <xsl:element name="attribute">
                <xsl:attribute name="name"><xsl:value-of select="."/></xsl:attribute>
                <xsl:call-template name="correlation_relationship"><xsl:with-param name="element"
select="$id_element_actual"/></xsl:call-template>
            </xsl:element>
            <xsl:element name="method"><xsl:call-template name="make_element"><xsl:with-param name="element"
select="../."/></xsl:call-template></xsl:element>
            <xsl:apply-templates select="../goal | ../softgoal | ../task | ../goal_ref | ../softgoal_ref | ../task_ref |
../crosscutting_relationship | ../new_element_type"/>
        </xsl:if>
```

```
    </xsl:for-each>
    <xsl:if test="count(./topic)=0"><xsl:apply-templates select="goal | softgoal | task | goal_ref | softgoal_ref | task_ref |
crosscutting_relationship"/></xsl:if>
    </xsl:template>

    <xsl:template match="new_element_type">
        <xsl:variable name="name" select="./@name"/>
        <xsl:variable name="value" select="./@value"/>
        <xsl:element name="attribute">
            <xsl:attribute name="name"><xsl:value-of select="concat($name, ' = ', $value)"/></xsl:attribute>
        </xsl:element>
    </xsl:template>

    <xsl:template match="goal_ref | softgoal_ref | task_ref">
    <xsl:variable name="id_reference_actual" select="./@id"/>
    <xsl:variable name="element_actual" select="//goal[@id=$id_reference_actual]"/>
    <xsl:for-each select="$element_actual">
        <xsl:for-each select="./topic">
            <xsl:variable name="is_class1"><xsl:call-template name="is_class_ref"><xsl:with-param name="topic"
select="."/></xsl:call-template></xsl:variable>
            <xsl:if test="$is_class1='true'">
                <xsl:element name="entity">
                    <xsl:attribute name="name"><xsl:value-of select="."/></xsl:attribute>
                </xsl:element>
            </xsl:if>
            <xsl:if test="$is_class1!='true'">
                <xsl:element name="attribute">
                    <xsl:attribute name="name"><xsl:value-of select="."/></xsl:attribute></xsl:element>
            </xsl:if>
    </xsl:for-each>
    </xsl:for-each>
    </xsl:template>

    <xsl:template name="make_element">
    <xsl:param name="element"/>
    <xsl:for-each select="$element">
    <xsl:element name="{name()}">
    <xsl:for-each select="@*">
        <xsl:attribute name="{name()}">   <xsl:value-of select="."/></xsl:attribute>
    </xsl:for-each>
    </xsl:element>
    </xsl:for-each>
    </xsl:template>

    <xsl:template name="correlation_relationship">
        <xsl:param name="element"/>
        <xsl:variable name="correlations" select="//correlation_relationship[source/goal_ref/@id=$element] |
//correlation_relationship[source/softgoal_ref/@id=$element] |
//correlation_relationship[source/task_ref/@id=$element]"/>

        <xsl:for-each select="$correlations">
            <xsl:variable name="label" select="./@label"/>
            <xsl:variable name="target" select="./target//@id"/>
            <xsl:for-each select="$target">
                <xsl:variable name="target_actual" select="."/>
                <xsl:variable name="element_target" select="//goal[@id=$target_actual] | //softgoal[@id=$target_actual]
| //task[@id=$target_actual]"/>
                <xsl:variable name="parent_element_target" select="//goal[goal/@id=$target_actual] |
//softgoal[goal/@id=$target_actual] | //task[goal/@id=$target_actual] | //goal[task/@id=$target_actual] |
//softgoal[task/@id=$target_actual] | //task[task/@id=$target_actual] | //goal[softgoal/@id=$target_actual] |
//softgoal[softgoal/@id=$target_actual] | //task[softgoal/@id=$target_actual]"/>
                <xsl:for-each select="$element_target">
                    <xsl:variable name="element_target_topic" select="./topic"/>
                    <xsl:for-each select="$element_target_topic">
                        <xsl:variable name="is_class"><xsl:call-template name="is_class"><xsl:with-param
name="topic" select="."/></xsl:call-template></xsl:variable>
                        <xsl:choose>
                            <xsl:when test="$is_class='true'"><xsl:call-template name="association"><xsl:with-param
name="target" select="."/><xsl:with-param name="label" select="$label"/>
                            </xsl:call-template></xsl:when>
                            <xsl:otherwise>
                                <xsl:variable name="parent_element_target_topic"
select="$parent_element_target/topic"/>
                                <xsl:for-each select="$parent_element_target/topic">
                                    <xsl:variable name="is_class1"><xsl:call-template name="is_class"><xsl:with-
param name="topic" select="."/></xsl:call-template></xsl:variable>
```

```
                                        <xsl:if test="is_class1='true'"><xsl:call-template name="association"><xsl:with-
param name="target" select="."/><xsl:with-param name="label" select="$label"/></xsl:call-template></xsl:if></xsl:for-
each>
                                    </xsl:otherwise>
                                </xsl:choose>
                                </xsl:for-each>
                                </xsl:for-each>
                                </xsl:for-each>
                                </xsl:for-each>
    </xsl:template>

    <xsl:template match="correlation_relationship1"> <!--esta nÃ£o estÃ¡ sendo usada-->
        <xsl:variable name="source" select="./source//@id"/>
        <xsl:variable name="target" select="./target//@id"/>
        <xsl:variable name="element_source" select="//goal[@id=$source] | //softgoal[@id=$source] |
//task[@id=$source]"/>
        <xsl:variable name="element_target" select="//goal[@id=$target] | //softgoal[@id=$target] |
//task[@id=$target]"/>
        <xsl:variable name="topic_source" select="$element_source/topic"/>
        <xsl:variable name="topic_target" select="$element_target/topic"/>

        <xsl:variable name="father_source" select="//goal/goal[@id=$source] | //softgoal/goal[@id=$source] |
//task/goal[@id=$source] | //goal/task[@id=$source] | //softgoal/task[@id=$source] | //task/task[@id=$source] |
//goal/softgoal[@id=$source] | //softgoal/softgoal[@id=$source] | //task/softgoal[@id=$source]"/>
        <xsl:variable name="father_target" select="//goal/goal[@id=$target] | //softgoal/goal[@id=$target] |
//task/goal[@id=$target] | //goal/task[@id=$target] | //softgoal/task[@id=$target] | //task/task[@id=$target] |
//goal/softgoal[@id=$target] | //softgoal/softgoal[@id=$target] | //task/softgoal[@id=$target]"/>
        <xsl:variable name="is_class1"><xsl:call-template name="is_class"><xsl:with-param name="topic"
select="$topic_source"/></xsl:call-template></xsl:variable>
        <xsl:variable name="is_class2"><xsl:call-template name="is_class"><xsl:with-param name="topic"
select="$topic_target"/></xsl:call-template></xsl:variable>

        <xsl:choose>
            <xsl:when test="$is_class1='true' and $is_class2='true'"><xsl:call-template name="association"><xsl:with-
param name="source" select="$topic_source"/><xsl:with-param name="target" select="$topic_target"/></xsl:call-
template></xsl:when>
            <xsl:when test="$is_class1='true' and $is_class2!='true'"><xsl:call-template name="association"><xsl:with-
param name="source" select="$topic_source"/><xsl:with-param name="target" select="$father_target/topic"/></xsl:call-
template> </xsl:when>
            <xsl:when test="$is_class1!='true' and $is_class2='true'"><xsl:call-template name="association"><xsl:with-
param name="source" select="$father_source/topic"/><xsl:with-param name="target" select="$topic_target"/></xsl:call-
template></xsl:when>
        </xsl:choose>
    </xsl:template>

    <xsl:template name="association"> <!-- Ã© necesÃ¡rio ver se ambos sÃ£o classes-->
        <xsl:param name="target"/>
        <xsl:param name="label"/>
        <xsl:element name="association">
            <xsl:attribute name="target"><xsl:value-of select="$target"/></xsl:attribute>
            <xsl:attribute name="label"><xsl:value-of select="$label"/></xsl:attribute>
        </xsl:element>
    </xsl:template>

    <xsl:template match="crosscutting_relationship">
    </xsl:template>
</xsl:stylesheet>

<?xml version="1.0" encoding="UTF-8"?><!-- este arquivo tem que ser aplicado ao XML de saÃ-de de
dataIntermediate.xsl-->
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
    <xsl:output method="text" indent="yes"/>

    <xsl:template match="/">
        <xsl:apply-templates select="entities"/>
    </xsl:template>

    <xsl:template match="entities">
    <xsl:text>digraph class_diagram{
    rankdir=LR;
    node [shape = record,height=.1];
    ranksep=.5;

    </xsl:text>
        <xsl:apply-templates select="goal_model"/>
<xsl:text>}</xsl:text>
```

```
    </xsl:template>

    <xsl:key name="topics" match="topic" use="text()"/>
    <xsl:key name="all_entities" match="entity" use="@name"/>
    <xsl:key name="all_attribute" match="attribute" use="@name"/>
    <xsl:key name="all_method" match="method/goal | method/softgoal | method/task" use="@name"/>

    <xsl:template match="goal_model">
<!--        <xsl:text>
        subgraph </xsl:text><xsl:value-of select="concat('cluster_',./@name)"/><xsl:text> {
            label = "</xsl:text><xsl:value-of select="./@name"/>"<xsl:text>;
            color = tomato;</xsl:text>-->

        <xsl:variable name="all_entities_teste" select='./entity[generate-id(.)=generate-
id(key("all_entities",@name)[1])]'/>
        <xsl:variable name="all_entities" select='.//entity[generate-id(.)=generate-id(key("all_entities",@name)[1])]'/>
        <xsl:for-each select="$all_entities">
            <xsl:call-template name="create_class"><xsl:with-param name="topic" select="."/></xsl:call-template>
        </xsl:for-each>
        <xsl:apply-templates select="entity"/>
        <xsl:text>
<!--        }-->
        </xsl:text>
    </xsl:template>

    <xsl:template match="entity">
        <xsl:variable name="entity_actual" select="."/>
        <xsl:variable name="children" select="./entity"/>
        <xsl:variable name="children_teste" select='.//entity[generate-id(.)=generate-id(key("all_entities",@name)[1])]'/>
        <xsl:variable name="children_teste2" select='$entity_actual[generate-id(.)=generate-
id(key("all_entities",@name)[1])]'/>
        <xsl:variable name="children1" select='./entity[generate-id(.)=generate-id(key("all_entities",@name)[1])]'/> <!--
nÃ£o consigo fazer com que ele nÃ£o repita relaionamentos-->
        <xsl:for-each select="$children1">
            <xsl:call-template name="decomposition"><xsl:with-param name="source"
select="$entity_actual"/><xsl:with-param name="target" select="."/></xsl:call-template>
        </xsl:for-each>
        <xsl:apply-templates select="association | entity"/>

    </xsl:template>

    <xsl:template match="association">
        <xsl:variable name="entity_actual" select="."/>
        <xsl:variable name="parent" select="../."/>
        <xsl:for-each select="$parent">
            <xsl:text>
            "</xsl:text><xsl:value-of select="./@name"/><xsl:text>" -&gt; "</xsl:text><xsl:value-of
select="$entity_actual/@target"/>
            <xsl:text>" [arrowhead=normal, style=normal, label="</xsl:text><xsl:value-of
select="$entity_actual/@label"/><xsl:text>"];</xsl:text>
        </xsl:for-each>
    </xsl:template>

    <xsl:template name="decomposition">
        <xsl:param name="source"/>
        <xsl:param name="target"/>
        <xsl:text>
        "</xsl:text><xsl:value-of select="$source/@name"/><xsl:text>" -&gt; "</xsl:text><xsl:value-of
select="$target/@name"/>
        <xsl:text>" [arrowhead=odot, style=normal];</xsl:text>
    </xsl:template>

    <xsl:template name="create_class">
        <xsl:param name="topic"/>
        <xsl:text>
            "</xsl:text>
        <xsl:value-of select="$topic/@name"/>
        <xsl:text>" [label="</xsl:text>
        <xsl:value-of select="$topic/@name"/><xsl:text> | </xsl:text>
        <xsl:call-template name="create_attribute"><xsl:with-param name="topic" select="$topic"/></xsl:call-
template><xsl:text> | </xsl:text>
        <xsl:call-template name="create_method"><xsl:with-param name="topic" select="$topic"/></xsl:call-
template><xsl:text> "];</xsl:text>
    </xsl:template>
```

```
    <xsl:template name="create_attribute">
        <xsl:param name="topic"/>
        <xsl:variable name="son" select="//entity[@name=$topic/@name]/attribute"/>
        <xsl:variable name="all_attributes" select='$son[generate-id(.)=generate-id(key("all_attribute",@name)[1])]'/>

        <xsl:for-each select="$all_attributes">
            <xsl:value-of select="./@name"/><xsl:text> \n </xsl:text>
        </xsl:for-each>
    </xsl:template>

    <xsl:template name="create_method">
        <xsl:param name="topic"/>
        <xsl:variable name="son" select="//entity[@name=$topic/@name]/method/goal |
//entity[@name=$topic/@name]/method/task | //entity[@name=$topic/@name]/method/softgoal"/>
        <xsl:variable name="all_methods" select='$son[generate-id(.)=generate-id(key("all_method",@name)[1])]'/>
        <xsl:for-each select="$all_methods">
            <xsl:choose>
                <xsl:when test="name()='goal'">G-<xsl:value-of select="./@name"/> \n </xsl:when>
                <xsl:when test="name()='softgoal'">SG-<xsl:value-of select="./@name"/> \n </xsl:when>
                <xsl:when test="name()='task'">T-<xsl:value-of select="./@name"/> \n </xsl:when>
            </xsl:choose>
        </xsl:for-each>
    </xsl:template>

</xsl:stylesheet>
```

Modelo entidade-relacionamento

```
<?xml version="1.0" encoding="UTF-8"?><!-- este arquivo tem que ser aplicado ao XML de saÃ-de de
dataIntermediate.xsl-->
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
    <xsl:output method="text" indent="yes"/>

    <xsl:template match="/">
        <xsl:apply-templates select="entities"/>
    </xsl:template>

    <xsl:template match="entities">
    <xsl:text>digraph class_diagram{
    rankdir=LR;
    node [shape = record,height=.1];

    </xsl:text>
        <xsl:apply-templates select="goal_model"/>
<xsl:text>}</xsl:text>
    </xsl:template>

    <xsl:key name="topics" match="topic" use="text()"/>
    <xsl:key name="all_entities" match="entity" use="@name"/>
    <xsl:key name="all_attribute" match="attribute" use="@name"/>

    <xsl:template match="goal_model">
        <xsl:text>
        subgraph </xsl:text><xsl:value-of select="concat('cluster_',./@name)"/><xsl:text> {
            label = "</xsl:text><xsl:value-of select="./@name"/>"<xsl:text>;
            color = tomato;</xsl:text>

        <xsl:variable name="all_entities" select='.//entity[generate-id(.)=generate-id(key("all_entities",@name)[1])]'/>
        <xsl:for-each select="$all_entities">
            <xsl:call-template name="create_class"><xsl:with-param name="topic" select="."/></xsl:call-template>
        </xsl:for-each>
        <xsl:apply-templates select="entity"/>
        <xsl:text>
        }
        </xsl:text>
    </xsl:template>

    <xsl:template match="entity">
        <xsl:variable name="entity_actual" select="."/>
        <xsl:variable name="children" select="./entity"/>
        <xsl:variable name="children1" select='./entity[generate-id(.)=generate-id(key("all_entities",@name)[1])]'/> <!--
nÃ£o consigo fazer com que ele nÃ£o repita relaionamentos-->
        <xsl:for-each select="$children1">
```

```
            <xsl:call-template name="decomposition"><xsl:with-param name="source"
select="$entity_actual"/><xsl:with-param name="target" select="."/></xsl:call-template>
        </xsl:for-each>
        <xsl:apply-templates select="association"/>
        <xsl:apply-templates select="entity"/>
    </xsl:template>

    <xsl:template match="association">
        <xsl:variable name="entity_actual" select="."/>
        <xsl:variable name="parent" select="../."/>
        <xsl:for-each select="$parent">
            <xsl:text>
            "</xsl:text><xsl:value-of select="./@name"/><xsl:text>" -&gt; "</xsl:text><xsl:value-of
select="$entity_actual/@target"/>
            <xsl:text>" [arrowhead=normal, style=normal, label="</xsl:text><xsl:value-of
select="$entity_actual/@label"/><xsl:text>"];</xsl:text>
        </xsl:for-each>
    </xsl:template>

    <xsl:template name="decomposition">
        <xsl:param name="source"/>
        <xsl:param name="target"/>
        <xsl:text>
        "</xsl:text><xsl:value-of select="$source/@name"/><xsl:text>" -&gt; "</xsl:text><xsl:value-of
select="$target/@name"/>
        <xsl:text>" [arrowhead=odot, style=normal];</xsl:text>
    </xsl:template>

    <xsl:template name="create_class">
        <xsl:param name="topic"/>
        <xsl:text>
            "</xsl:text>
        <xsl:value-of select="$topic/@name"/>
        <xsl:text>" [label="</xsl:text>
        <xsl:value-of select="$topic/@name"/><xsl:text> | </xsl:text>
        <xsl:call-template name="create_attribute"><xsl:with-param name="topic" select="$topic"/></xsl:call-
template><xsl:text> "] </xsl:text>
    </xsl:template>


    <xsl:template name="create_attribute">
        <xsl:param name="topic"/>
        <xsl:variable name="son" select="//entity[@name=$topic/@name]/attribute"/>
        <xsl:variable name="all_attributes" select='$son[generate-id(.)=generate-id(key("all_attribute",@name)[1])]'/>

        <xsl:for-each select="$all_attributes">
            <xsl:value-of select="./@name"/><xsl:text> \n </xsl:text>
        </xsl:for-each>
    </xsl:template>
</xsl:stylesheet>
```

Cenários

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
    <xsl:output method="html" indent="yes"/>
    <xsl:template match="/">
        <html>
            <head>
                <title>Aspect-oriented goal model - documentation</title>
            </head>
            <body>
                <font face="verdana">
                    <h1> <p align="center">Scenarios</p> </h1>
                    <xsl:apply-templates select="aspect_oriented_model"/>
                </font>
            </body>
        </html>
    </xsl:template>

    <xsl:template match="aspect_oriented_model">
        <xsl:apply-templates select="goal_model"/>
    </xsl:template>
```

```
    <xsl:template match="goal_model">
        <xsl:apply-templates select="goal | task | softgoal"/>
    </xsl:template>

    <xsl:template match="goal | task">

        <xsl:choose>
            <xsl:when test="count(./goal | ./task | ./softgoal | ./goal_ref | ./softgoal_ref | ./task_ref)=0"></xsl:when>
            <xsl:otherwise>
                <xsl:call-template name="make_table">
                    <xsl:with-param name="title" select="."/>
                    <xsl:with-param name="actor" select="./new_element_type[@name='actor']"/>
                    <xsl:with-param name="exception" select="./new_element_type[@name='exception']"/>
                    <xsl:with-param name="constraint" select="./new_element_type[@name='constraint']"/>
                    <xsl:with-param name="context" select="./new_element_type[@name='context']"/>
                    <xsl:with-param name="resource" select="./new_element_type[@name='resource']"/>
                    <xsl:with-param name="episode" select="./goal | ./task | ./softgoal | ./goal_ref | ./softgoal_ref |
./task_ref"/>
                </xsl:call-template>
                <br/>
            </xsl:otherwise>
        </xsl:choose>
        <xsl:apply-templates select="goal | task"/>

    </xsl:template>

    <xsl:template name="make_table">
        <xsl:param name="title"/>
        <xsl:param name="actor"/>
        <xsl:param name="exception"/>
        <xsl:param name="constraint"/>
        <xsl:param name="context"/>
        <xsl:param name="resource"/>
        <xsl:param name="episode"/>
        <xsl:variable name="title_id" select="$title/@id"/>
        <xsl:variable name="context_soft" select="//correlation_relationship[source//@id=$title_id]"/>

        <table border="0">
            <tr>
                <td bgcolor="lightgreen">Title</td>
                <td bgcolor="lightgreen">
                    <a>
                        <xsl:attribute name="name"><xsl:value-of select="$title/@id"/></xsl:attribute>
                        <xsl:value-of select="$title/@name"/>
                    </a>
                </td>
            </tr>
            <xsl:if test="count($actor)>0">
            <tr>
                <td bgcolor="lightgreen">Actor</td>
                <td><xsl:for-each select="$actor"><xsl:value-of select="@value"/>;  </xsl:for-each></td>
            </tr>
            </xsl:if>
            <xsl:if test="count($resource)>0">
            <tr>
                <td bgcolor="lightgreen">Resource</td>
                <td><xsl:for-each select="$resource"><xsl:value-of select="@value"/>;  </xsl:for-each></td>
            </tr>
            </xsl:if>
            <xsl:if test="count($context)>0 or count($context_soft)>0">
            <tr>
                <td bgcolor="lightgreen">Context</td>
                <td><xsl:if test="count($context)>0"><xsl:for-each select="$context"><xsl:value-of select="@value"/>;
</xsl:for-each></xsl:if>
                    <xsl:if test="count($context_soft)>0"><xsl:for-each select="$context_soft"><xsl:call-template
name="get_name"><xsl:with-param name="id" select="./target//@id"/></xsl:call-template>;  </xsl:for-each></xsl:if>
                </td>
            </tr>
            </xsl:if>
            <xsl:if test="count($constraint)>0">
            <tr>
                <td bgcolor="lightgreen">Constraint</td>
                <td><xsl:for-each select="$constraint"><xsl:value-of select="@value"/><br/>  </xsl:for-each></td>
            </tr>
            </xsl:if>
            <xsl:if test="count($exception)>0">
```

```
                <tr>
                    <td bgcolor="lightgreen">Exception</td>
                    <td><xsl:for-each select="$exception"><xsl:value-of select="@value"/><br/>  </xsl:for-each></td>
                </tr>
                </xsl:if>

                <tr>
                    <td bgcolor="lightgreen">Episode</td>
                    <td>
                    <xsl:for-each select="$episode">
                        <xsl:choose>
                            <xsl:when test="name()='goal_ref' or name()='softgoal_ref' or name()='task_ref' or count(.//goal
| .//softgoal | .//task | .//goal_ref | .//softgoal_ref | .//task_ref)>0">
                                <xsl:choose>
                                    <xsl:when test="./@decomposition_label='or'">
                                        <xsl:text>[  </xsl:text><a><xsl:attribute name="href"><xsl:value-of
select="concat('#',@id)"/></xsl:attribute>
                                        <xsl:call-template name="get_name"><xsl:with-param name="id"
select="./@id"/></xsl:call-template>
                                        </a>
                                        <xsl:text>  ]</xsl:text>
                                    </xsl:when>
                                    <xsl:otherwise>
                                        <a><xsl:attribute name="href"><xsl:value-of
select="concat('#',@id)"/></xsl:attribute>
                                        <xsl:call-template name="get_name"><xsl:with-param name="id"
select="./@id"/></xsl:call-template>
                                        </a>
                                    </xsl:otherwise>
                                </xsl:choose>
                            </xsl:when>
                            <xsl:otherwise>
                                <xsl:choose>
                                    <xsl:when test="./@decomposition_label='or'">
                                        <xsl:text>[  </xsl:text><xsl:value-of select="./@name"/><xsl:text>  ]</xsl:text>
                                    </xsl:when>
                                    <xsl:otherwise><xsl:value-of select="./@name"/></xsl:otherwise>
                                </xsl:choose>
                            </xsl:otherwise>
                            </xsl:choose>
                    <br/></xsl:for-each></td>
                </tr>
            </table>
        </xsl:template>

        <xsl:template name="get_name">
            <xsl:param name="id"/>
            <xsl:value-of select="//goal[@id=$id]/@name | //softgoal[@id=$id]/@name | //task[@id=$id]/@name"/>
        </xsl:template>

</xsl:stylesheet>
```

Apêndice C
Especificação do Estudo de Caso – C&L

Neste apêndice inluimos a modelagem realizada par o sistema C&L. Os documentos gerados são: lista de softmetas, metas e tarefas, cenários; diagrama de classes; e modelo entidade relacionamento. Estes documentos foram gerados a partir da descrição em XML com o apoio da implementação desenvolvida. No final deste apêndice incluímos a descrição do sistema C&L formatada de acordo com a BNF descrita no Apêndice A. Todos estes documentos retratam as informações contidas nos AOV-graph após a composição.

Lista de softmetas, metas e tarefas

1 Model: Model requirements (93)
1.1Provide cooperative definition of Scenarios and Lexicon
1.1.1.Use [modeling tool] in the WEB
1.1.1.1.Manage [project]
1.1.1.1.1.Create [project]
1.1.1.1.1.1.Insert [project]
1.1.1.1.1.1.1.**Verify if [data identification] is unique**
1.1.1.1.1.1.2.**Verify if [enters] are valid**
1.1.1.1.1.1.3.**Verify if every [data] are filled up**
1.1.1.1.2.Exclude [project]
1.1.1.1.3.Save [project] as
1.1.1.1.4.Link [users] with [project]
1.1.1.1.5.Unlink [user] with [project]
1.1.1.1.6.Insert [user]
1.1.1.1.6.1.**Verify if [data identification] is unique**
1.1.1.1.6.2.**Verify if [enters] are valid**
1.1.1.1.6.3.**Verify if every [data] are filled up**
1.1.1.1.7.Export [project]
1.1.1.1.8.Import [project]
1.1.1.1.9.**Persistence in [BD]**
1.1.1.1.10.**Restrict [access] to [data]**
1.1.1.1.11.**Restrict [access] to [functions]**
1.1.1.1.12.**Authentication by Login**
1.1.1.1.13.**Logout**
1.1.1.2.Edit [requirement model]
1.1.1.2.1.Edit [scenarios]
1.1.1.2.1.1.Insert [scenario]
1.1.1.2.1.1.1.Add [title]
1.1.1.2.1.1.2.Add [goal]
1.1.1.2.1.1.3.Add [context]
1.1.1.2.1.1.4.Add [actor]
1.1.1.2.1.1.5.Add [resource]
1.1.1.2.1.1.6.Add [episode]
1.1.1.2.1.1.7.Add [exception]
1.1.1.2.1.1.8.Make references to other [scenario]
1.1.1.2.1.1.9.Make references in other [scenario]
1.1.1.2.1.1.10.Make references to [symbol]
1.1.1.2.1.1.11.**Verify if [data identification] is unique**
1.1.1.2.1.1.12.**Verify if [enters] are valid**
1.1.1.2.1.1.13.**Verify if every [data] are filled up**
1.1.1.2.1.2.Exclude [scenario]
1.1.1.2.1.2.1.Unmake references in other [scenario]
1.1.1.2.1.3.Change [scenario]
1.1.1.2.1.4.Search [scenario]
1.1.1.2.1.5.Search [scenario] with reference to current [scenario]
1.1.1.2.1.6.**Verify [scenario]**
1.1.1.2.1.7.**Verify [lexicon] and [scenario] in conjunction**
1.1.1.2.2.Edit [lexicon]
1.1.1.2.2.1.Insert [symbol]
1.1.1.2.2.1.1.Add [name]
1.1.1.2.2.1.2.Add [synonymous]
1.1.1.2.2.1.3.Add [notion]
1.1.1.2.2.1.4.Add [behavior response]
1.1.1.2.2.1.5.Add [type]
1.1.1.2.2.1.6.Make references to other [symbol]
1.1.1.2.2.1.7.Make references in other [symbol]
1.1.1.2.2.1.8.Make references in [scenario]
1.1.1.2.2.1.9.**Verify if [data identification] is unique**
1.1.1.2.2.1.10.**Verify if [enters] are valid**
1.1.1.2.2.1.11.**Verify if every [data] are filled up**
1.1.1.2.2.2.Exclude [symbol]
1.1.1.2.2.2.1.Unmake references in [scenario]
1.1.1.2.2.2.2.Unmake references in other [symbol]
1.1.1.2.2.3.Change [symbol]
1.1.1.2.2.4.Search [symbol]
1.1.1.2.2.5.Search [scenario] with reference to current [symbol]
1.1.1.2.2.6.Search [symbol] with reference to current [symbol]
1.1.1.2.2.7.**Verify [lexicon]**
1.1.1.2.2.8.**Verify [lexicon] and [scenario] in conjunction**
1.1.1.2.3.Create [onthology]
1.1.1.2.4.Verify [requirement model]
1.1.1.2.4.1.Verify [scenario]
1.1.1.2.4.1.1.Verify [syntax of scenario]
1.1.1.2.4.2.Verify [lexicon]
1.1.1.2.4.2.1.Verify [syntax of symbol]
1.1.1.2.4.2.2.Verify [circularity property]

1.1.1.2.4.2.3.Verify [minimal vocabulary property]
1.1.1.2.4.3.Verify [lexicon] and [scenario] in conjunction
1.1.1.2.5.Generate_graph
1.1.1.2.6.**Persistence in [BD]**
1.1.1.2.7.**Restrict [access] to [data]**
1.1.1.2.8.**Restrict [access] to [functions]**
1.1.1.2.9.**Authentication by Login**
1.1.1.2.10.**Logout**
1.1.1.3.Remember [password]
1.1.1.3.1.**Persistence in [BD]**
1.1.1.4.Register [user]
1.1.1.4.1.**Persistence in [BD]**
1.1.1.4.2.**Symmetric cryptography**
1.1.1.5.Install [aplication]
1.1.1.5.1.Set [configuration file]
1.1.1.5.2.**Persistence in [file]**
1.1.1.6.Uninstall [aplication]
1.1.1.6.1.**Persistence in [file]**

2 Model: Persistence (39)
2.1Persistence
2.1.1.Persistence in [BD]
2.1.1.1.Verify if [BD] is connected
2.1.1.2.Initiate [BD]
2.1.1.3.Connect [BD]
2.1.1.3.1.**Restrict [access] to [data]**
2.1.1.3.2.**Restrict [access] to [functions]**
2.1.1.3.3.**Authentication by Login**
2.1.1.3.4.**Logout**
2.1.1.4.**Include [data]**
2.1.1.5.**Select [data]**
2.1.1.6.**Delete [data]**
2.1.1.7.**Update [data]**
2.1.1.8.Disconnect [BD]
2.1.1.9.**Detect [persistence exception]**
2.1.2.Persistence in [file]
2.1.2.1.Verify if [file] exists
2.1.2.2.Create [file]
2.1.2.3.Open [file]
2.1.2.4.**Include [data]**
2.1.2.5.**Select [data]**
2.1.2.6.**Delete [data]**
2.1.2.7.**Update [data]**
2.1.2.8.Close [file]
2.1.2.9.Delete [file]
2.1.2.10.**Detect [persistence exception]**
2.1.3.Make register operation
2.1.3.1.Include [data]
2.1.3.2.Select [data]
2.1.3.3.Delete [data]
2.1.3.4.Update [data]

3 Model: Reliability (34)
3.1Reliability
3.1.1.Robustness
3.1.1.1.Robustness [incomplete data]
3.1.1.1.1.Verify if every [data] are filled up
3.1.1.1.1.1.**Detect [robustness exception]**
3.1.1.1.1.2.**Detect [correctness exception]**
3.1.1.2.Robustness [invalid data]
3.1.1.2.1.Verify if [data identification] is unique
3.1.1.2.1.1.**Detect [robustness exception]**
3.1.1.2.1.2.**Detect [correctness exception]**
3.1.1.2.2.Verify if [enters] are valid
3.1.1.2.2.1.**Detect [robustness exception]**
3.1.1.2.2.2.**Detect [correctness exception]**
3.1.1.3.Robustness [unexpected action]
3.1.2.Correctness
3.1.2.1.Accuracy [data]
3.1.2.1.1.**Verify if [data identification] is unique**
3.1.2.1.2.**Verify if [enters] are valid**
3.1.2.2.Completeness [data]
3.1.2.2.1.**Verify if every [data] are filled up**
3.2Handling [exception]
3.2.1.Detect [exception]
3.2.1.1.Detect [persistence exception]
3.2.1.2.Detect [robustness exception]
3.2.1.3.Detect [correctness exception]
3.2.2.Apply [dealer]

4 Model: Security (30)
4.1Security
4.1.1.Confidentiality
4.1.2.Integrity
4.1.3.Availability
4.2Authentication
4.2.1.Authentication by Certification
4.2.2.Authentication by Login
4.2.2.1.**Symmetric cryptography**
4.2.3.Validate [data]
4.2.3.1.Compare [data] with [credentials]
4.2.4.Authorization
4.2.4.1.Select [permissions] to [data]
4.2.4.2.Set [permissions]
4.2.4.3.Control [access]
4.2.4.3.1.Restrict [access] to [functions]
4.2.4.3.2.Restrict [access] to [data]
4.2.5.Accounting
4.2.5.1.Measure [system time] used by [user]
4.2.5.2.Measure [data] sent by [user]
4.2.5.3.Measure [data] received by [user]
4.2.6.Logout
4.3Cryptography
4.3.1.Assymmetric cryptography
4.3.2.Symmetric cryptography
4.3.3.Make [cryptography operations]
4.3.3.1.Encrypt [data]
4.3.3.2.Decrypt [data]

Cenários

Title	Provide cooperative definition of Scenarios and Lexicon
Episode	Use [modeling tool] in the WEB

Title	Use [modeling tool] in the WEB
Episode	Manage [project] Edit [requirement model] Remember [password] Register [user] Install [aplication] Uninstall [aplication]

Title	Create [project]
Episode	Insert [project]

Title	Insert [project]
Episode	Verify if [data identification] is unique Verify if [enters] are valid Verify if every [data] are filled up

Title	Manage [project]
Episode	Create [project] Exclude [project] Save [project] as Link [users] with [project]

	Unlink [user] with [project] Insert [user] Export [project] Import [project] Persistence in [BD] Restrict [access] to [data] Restrict [access] to [functions] Authentication by Login [Logout]

Title	Insert [user]
Episode	Verify if [data identification] is unique Verify if [enters] are valid Verify if every [data] are filled up

Title	Edit [requirement model]
Episode	[Edit [scenarios]] [Edit [lexicon]] [Create [onthology]] [Verify [requirement model]] Generate_graph Persistence in [BD] Restrict [access] to [data] Restrict [access] to [functions] Authentication by Login [Logout]

Title	Edit [scenarios]
Episode	Insert [scenario] Exclude [scenario] Change [scenario] Search [scenario] Search [scenario] with reference to current [scenario] Verify [scenario] Verify [lexicon] and [scenario] in conjunction

Title	Insert [scenario]
Episode	Add [title] Add [goal] Add [context] Add [actor] Add [resource] Add [episode] Add [exception] Make references to other [scenario] Make references in other [scenario] Make references to [symbol] Verify if [data identification] is unique Verify if [enters] are valid Verify if every [data] are filled up

Title	Exclude [scenario]
Episode	Unmake references in other [scenario]

Title	Edit [lexicon]
Episode	Insert [symbol] Exclude [symbol] Change [symbol] Search [symbol] Search [scenario] with reference to current [symbol] Search [symbol] with reference to current [symbol] Verify [lexicon] Verify [lexicon] and [scenario] in conjunction

Title	Insert [symbol]
Episode	Add [name] Add [synonymous]

	Add [notion] Add [behavior response] Add [type] Make references to other [symbol] Make references in other [symbol] Make references in [scenario] Verify if [data identification] is unique Verify if [enters] are valid Verify if every [data] are filled up

Title	Exclude [symbol]
Episode	Unmake references in [scenario] Unmake references in other [symbol]

Title	Verify [requirement model]
Episode	Verify [scenario] Verify [lexicon] Verify [lexicon] and [scenario] in conjunction

Title	Verify [scenario]
Episode	Verify [syntax of scenario]

Title	Verify [lexicon]
Episode	Verify [syntax of symbol] Verify [circularity property] Verify [minimal vocabulary property]

Title	Remember [password]
Episode	Persistence in [BD]

Title	Register [user]
Episode	Persistence in [BD] Symmetric cryptography

Title	Install [aplication]
Episode	Set [configuration file] Persistence in [file]

Title	Uninstall [aplication]
Episode	Persistence in [file]

Title	Persistence
Episode	[Persistence in [BD]] [Persistence in [file]] Make register operation

Title	Persistence in [BD]
Episode	Verify if [BD] is connected Initiate [BD] Connect [BD] Include [data] Select [data] Delete [data] Update [data] Disconnect [BD] Detect [persistence exception]

Title	Connect [BD]
Episode	Restrict [access] to [data] Restrict [access] to [functions] Authentication by Login [Logout]

Title	Persistence in [file]

Episode	Verify if [file] exists Create [file] Open [file] Include [data] Select [data] Delete [data] Update [data] Close [file] Delete [file] Detect [persistence exception]

Title	Make register operation
Episode	[Include [data]] [Select [data]] [Delete [data]] [Update [data]]

Title	Verify if every [data] are filled up
Exception	Incomplete data
Episode	Detect [robustness exception] Detect [correctness exception]

Title	Verify if [data identification] is unique
Exception	Invalid data
Episode	Detect [robustness exception] Detect [correctness exception]

Title	Verify if [enters] are valid
Exception	Invalid data
Episode	Detect [robustness exception] Detect [correctness exception]

Title	Handling [exception]
Context	Robustness; Correctness;
Episode	Detect [exception] Apply [dealer]

Title	Detect [exception]
Episode	Detect [persistence exception] Detect [robustness exception] Detect [correctness exception]

Title	Authentication
Context	Confidentiality;
Constraint	confidential data
Episode	[Authentication by Certification] [Authentication by Login] Validate [data] Authorization Accounting [Logout]

Title	Authentication by Login
Episode	Symmetric cryptography

Title	Validate [data]
Episode	Compare [data] with [credentials]

Title	Authorization
Episode	Select [permissions] to [data] Set [permissions] Control [access]

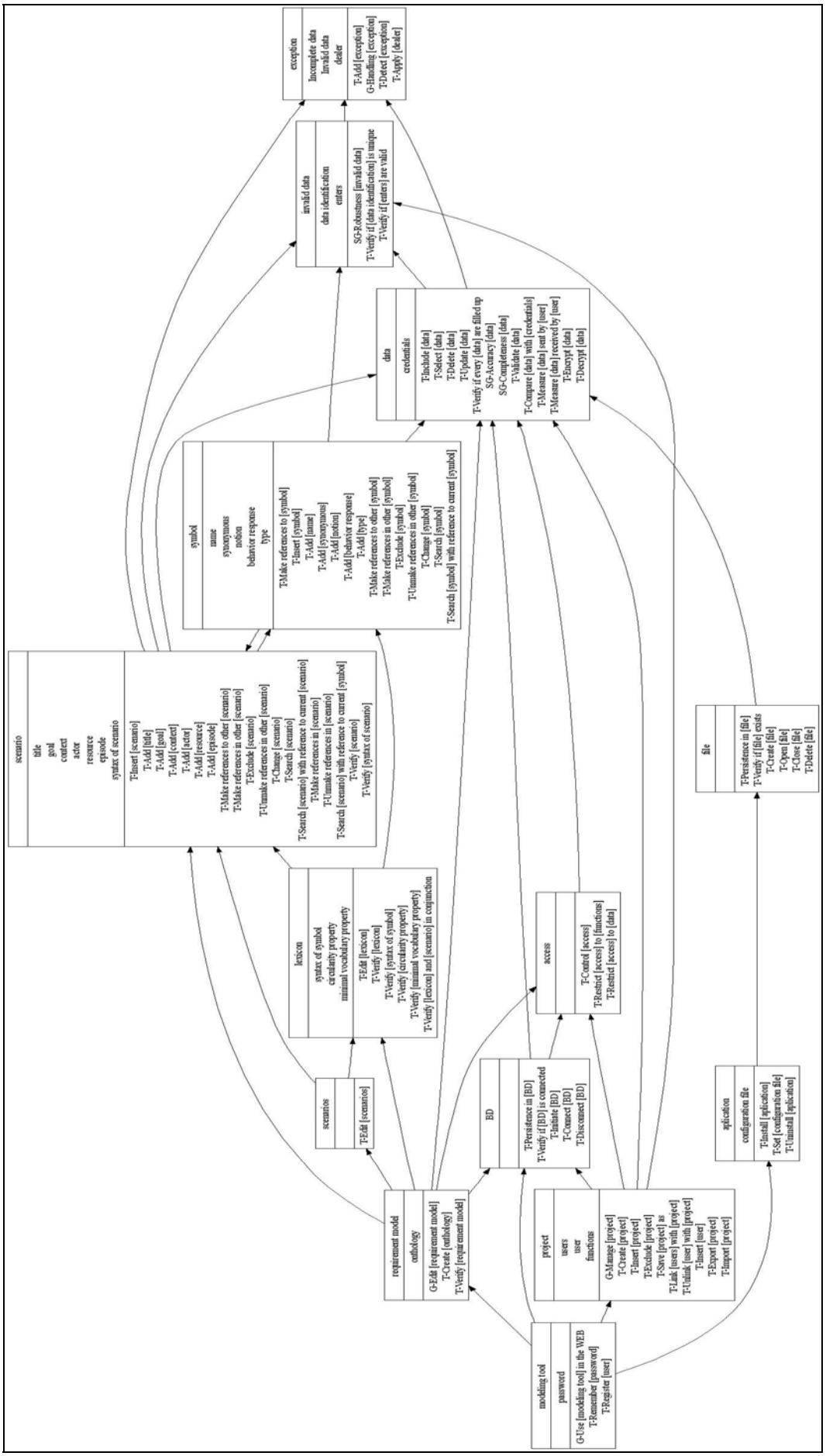
Title	Control [access]
Episode	Restrict [access] to [functions] Restrict [access] to [data]

Title	Accounting
Episode	Measure [system time] used by [user] Measure [data] sent by [user] Measure [data] received by [user]

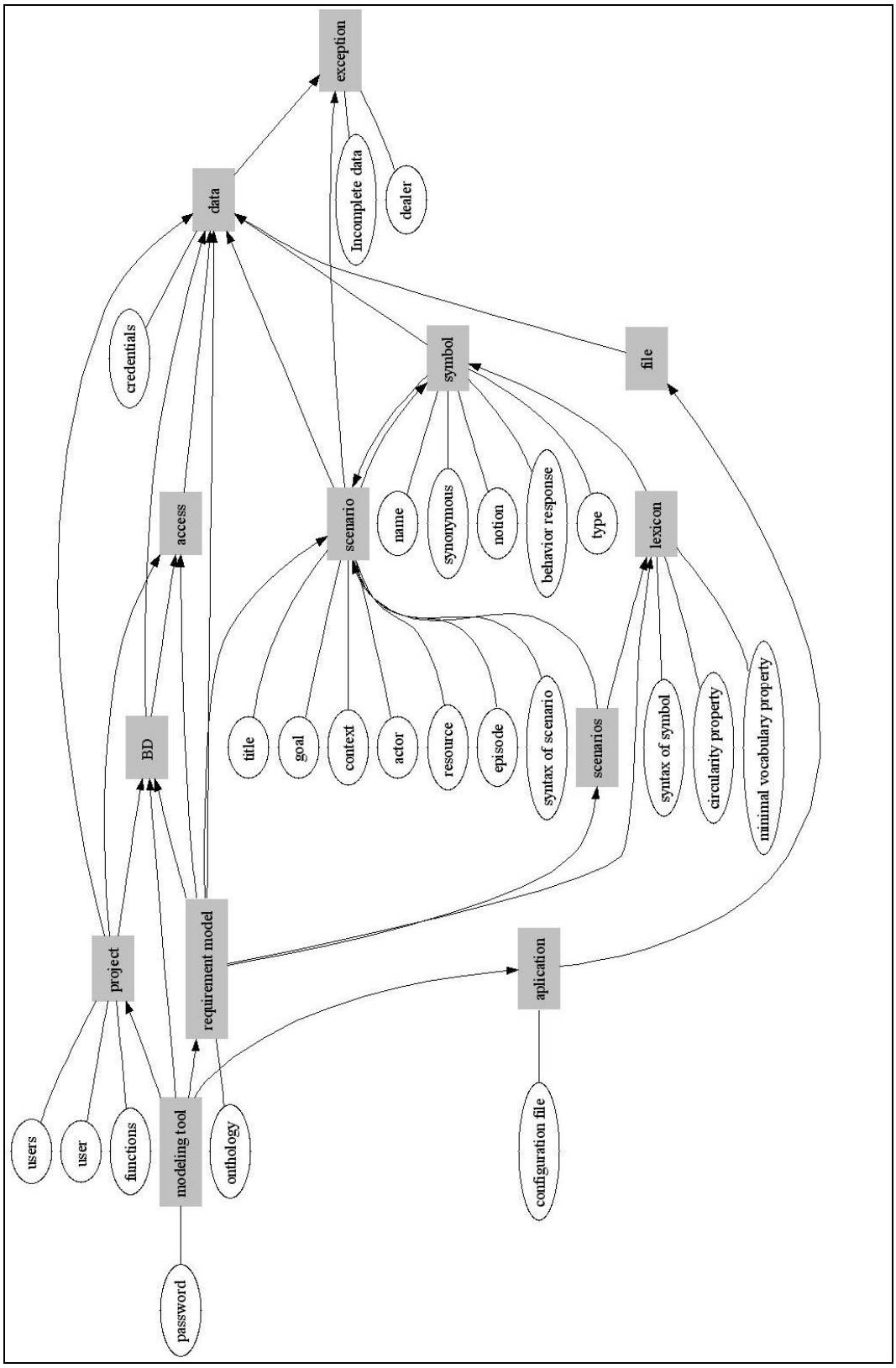
Title	Cryptography
Context	Confidentiality;
Episode	[Assymmetric cryptography] [Symmetric cryptography] Make [cryptography operations]

Title	Make [cryptography operations]
Episode	Encrypt [data] Decrypt [data]

Diagrama de classes do C&L após a composição



Modelo de entidade relacionamento do C&L após a composição



BNF

goal_model (Model requirements; GM1) {
goal = (Provide cooperative definition of Scenarios and Lexicon; G1;) {
goal = (Use [modeling tool] in the WEB; G2;) {
goal = (Manage [project]; G2.1; and) {
task = (Create [project]; T2.1.1; and) {
task = (Insert [project]; T2.1.1.2;) {
task_ref = (Verify if [data identification] is unique; T5.4;)
task_ref = (Verify if [enters] are valid; T5.5;)
task_ref = (Verify if every [data] are filled up; T5.6;)} }
task = (Exclude [project]; T2.1.2; and) {}
task = (Save [project] as; T2.1.3; and) {}
task = (Link [users] with [project]; T2.1.4; and) {}
task = (Unlink [user] with [project]; T2.1.5; and) {}
task = (Insert [user]; T2.1.6; and) {
task_ref = (Verify if [data identification] is unique; T5.4;)
task_ref = (Verify if [enters] are valid; T5.5;)
task_ref = (Verify if every [data] are filled up; T5.6;)}
task = (Export [project]; T2.1.7; and) {}
task = (Import [project]; T2.1.8; and) {}
task_ref = (Persistence in [BD]; T15.1.1;)
task_ref = (Restrict [access] to [data]; T9.3.5.3.2;)
task_ref = (Restrict [access] to [functions]; T9.3.5.3.1;)
task_ref = (Authentication by Login; T9.3.3;)
task_ref = (Logout Logout Logout Logout; T9.3.3.1; or)}
goal = (Edit [requirement model]; G2.2; and) {
task = (Edit [scenarios]; T2.2.1; or) {
task = (Insert [scenario]; T2.2.1.2;) {
task = (Add [title]; T2.2.1.2.1;) {}
task = (Add [goal]; T2.2.1.2.2;) {}
task = (Add [context]; T2.2.1.2.3;) {}
task = (Add [actor]; T2.2.1.2.4;) {}
task = (Add [resource]; T2.2.1.2.5;) {}
task = (Add [episode]; T2.2.1.2.6;) {}
task = (Add [exception]; T2.2.1.2.7;) {}
task = (Make references to other [scenario]; T2.2.1.2.8;) {}
task = (Make references in other [scenario]; T2.2.1.2.9;) {}
task = (Make references to [symbol]; T2.2.1.2.10;) {}
task_ref = (Verify if [data identification] is unique; T5.4;)
task_ref = (Verify if [enters] are valid; T5.5;)
task_ref = (Verify if every [data] are filled up; T5.6;)}
task = (Exclude [scenario]; T2.2.1.3;) {
task = (Unmake references in other [scenario]; T2.2.1.3.1;) {} }
task = (Change [scenario]; T2.2.1.4;) {}
task = (Search [scenario]; T2.2.1.5;) {}
task = (Search [scenario] with reference to current [scenario]; T2.2.1.6;) {} }
task = (Edit [lexicon]; T2.2.2; or) {
task = (Insert [symbol]; T2.2.2.2;) {
task = (Add [name]; T2.2.2.2.1;) {}
task = (Add [synonymous]; T2.2.2.2.2;) {}
task = (Add [notion]; T2.2.2.2.3;) {}
task = (Add [behavior response]; T2.2.2.2.4;) {}
task = (Add [type]; T2.2.2.2.5;) {}
task = (Make references to other [symbol]; T2.2.2.2.6;) {}
task = (Make references in other [symbol]; T2.2.2.2.7;) {}
task = (Make references in [scenario]; T2.2.2.2.8;) {}
task_ref = (Verify if [data identification] is unique; T5.4;)
task_ref = (Verify if [enters] are valid; T5.5;)
task_ref = (Verify if every [data] are filled up; T5.6;)}
task = (Exclude [symbol]; T2.2.2.3;) {
task = (Unmake references in [scenario]; T2.2.2.3.1;) {}
task = (Unmake references in other [symbol]; T2.2.2.3.2;) {} }
task = (Change [symbol]; T2.2.2.4;) {}
task = (Search [symbol]; T2.2.2.5;) {}
task = (Search [scenario] with reference to current [symbol]; T2.2.2.6;) {}
task = (Search [symbol] with reference to current [symbol]; T2.2.2.7;) {} }
task = (Create [onthology]; T2.2.3; or) {}
task = (Verify [requirement model]; T2.2.4; or) {
task = (Verify [scenario]; T2.2.4.1;) {
task = (Verify [syntax of scenario]; T2.2.4.1.1;) {} }

```
        task = (Verify [lexicon]; T2.2.4.2; ) {
        task = (Verify [syntax of symbol]; T2.2.4.2.1; ) {}
        task = (Verify [circularity property]; T2.2.4.2.2; ) {}
        task = (Verify [minimal vocabulary property]; T2.2.4.2.3; ) {} }
        task = (Verify [lexicon] and [scenario] in conjunction; T2.2.1.1; ) {}
        crosscutting {source = Verify [requirement model](T2.2.4)
            pointcut (verify_scenario; PC1.2.2.4.1): (Edit [scenarios]; T2.2.1)
            pointcut (verify_lexicon; PC1.2.2.4.2): (Edit [lexicon]; T2.2.2)
            advice (around): PC1.2.2.4.1 {
                task_ref = (Verify [scenario]; T2.2.4.1; )
                task_ref = (Verify [lexicon] and [scenario] in conjunction; T2.2.1.1; )}
            advice (around): PC1.2.2.4.2 {
                task_ref = (Verify [lexicon]; T2.2.4.2; )
                task_ref = (Verify [lexicon] and [scenario] in conjunction; T2.2.1.1; )} } }
        task = (Generate_graph; T2.2.5; and) {}
        task_ref = (Persistence in [BD]; T15.1.1; )
        task_ref = (Restrict [access] to [data]; T9.3.5.3.2; )
        task_ref = (Restrict [access] to [functions]; T9.3.5.3.1; )
        task_ref = (Authentication by Login; T9.3.3; )
        task_ref = (Logout Logout Logout Logout; T9.3.3.1; or)   }
        task = (Remember [password]; T2.3; and) {
        task_ref = (Persistence in [BD]; T15.1.1; )}
        task = (Register [user]; T2.4; ) {
        task_ref = (Persistence in [BD]; T15.1.1; )
        task_ref = (Symmetric cryptography; T9.2.4; )}
        task = (Install [aplication]; T2.5; ) {
        task = (Set [configuration file]; T2.5.1; and) {}
        task_ref = (Persistence in [file]; T15.1.2; )}
        task = (Uninstall [aplication]; T2.6; ) {
        task_ref = (Persistence in [file]; T15.1.2; )
    } } } }

    goal_model (Persistence; GM15) {
        goal =  (Persistence; G15.1; ) {
        task = (Persistence in [BD]; T15.1.1; or) {
        task = (Verify if [BD] is connected; T15.1.1.1; ) {}
        task = (Initiate [BD]; T15.1.1.2; ) {}
        task = (Connect [BD]; T15.1.1.3; ) {
        task_ref = (Restrict [access] to [data]; T9.3.5.3.2; )
        task_ref = (Restrict [access] to [functions]; T9.3.5.3.1; )
        task_ref = (Authentication by Login; T9.3.3; )
        task_ref = (Logout; T9.3.3.1; or)     }
        task_ref = (Include [data]; T15.1.3.1; )
        task_ref = (Select [data]; T15.1.3.2; )
        task_ref = (Delete [data]; T15.1.3.3; )
        task_ref = (Update [data]; T15.1.3.4; )
        task = (Disconnect [BD]; T15.1.1.4; ) {}
        task_ref = (Detect [persistence exception]; T5.2.1.3; and)}
        task = (Persistence in [file]; T15.1.2; or) {
        task = (Verify if [file] exists; T15.1.2.1; ) {}
        task = (Create [file]; T15.1.2.2; ) {}
        task = (Open [file]; T15.1.2.3; ) {}
        task_ref = (Include [data]; T15.1.3.1; )
        task_ref = (Select [data]; T15.1.3.2; )
        task_ref = (Delete [data]; T15.1.3.3; )
        task_ref = (Update [data]; T15.1.3.4; )
        task = (Close [file]; T15.1.2.4; ) {}
        task = (Delete [file]; T15.1.2.5; ) {}
        task_ref = (Detect [persistence exception]; T5.2.1.3; and)}
        task = (Make register operation; T15.1.3; and) {
        task = (Include [data]; T15.1.3.1; or) {}
        task = (Select [data]; T15.1.3.2; or) {}
        task = (Delete [data]; T15.1.3.3; or) {}
        task = (Update [data]; T15.1.3.4; or) {} }
        crosscutting {source = Persistence(G15.1)
            pointcut (persistence_BD; PC15.2): include(Manage [project]; G2.1) and
            pointcut (persistence_file; PC15.3): include(Install [aplication]; T2.5) and include(Uninstall [aplication]; T2.6)
            advice (around): PC15.2 {
                task_ref = (Persistence in [BD]; T15.1.1; )}
            advice (around): PC15.3 {
                task_ref = (Persistence in [file]; T15.1.2; )} }
        crosscutting {source = Make register operation(T15.1.3)
            pointcut (register_operation; PC15.1): include(Connect [BD]; T15.1.1.3) and include(Open [file]; T15.1.2.3)
            advice (after): PC15.1 {
                task_ref = (Include [data]; T15.1.3.1; )
                task_ref = (Select [data]; T15.1.3.2; )
```

```
                task_ref = (Delete [data]; T15.1.3.3; )
                task_ref = (Update [data]; T15.1.3.4; )} } } }

goal_model (Reliability; GM5) {
    softgoal = (Reliability; S5.1; ) {
    softgoal = (Robustness; S5.1.1; help) {
    softgoal = (Robustness [incomplete data]; S5.1.1.1; make) {
    task = (Verify if every [data] are filled up; T5.6; make) {
    task_ref = (Detect [robustness exception]; T5.2.1.1; and)
    task_ref = (Detect [correctness exception]; T5.2.1.2; and)} }
    softgoal = (Robustness [invalid data]; S5.1.1.2; make) {
    task = (Verify if [data identification] is unique; T5.4; make) {
    task_ref = (Detect [robustness exception]; T5.2.1.1; and)
    task_ref = (Detect [correctness exception]; T5.2.1.2; and)}
    task = (Verify if [enters] are valid; T5.5; make) {
    task_ref = (Detect [robustness exception]; T5.2.1.1; and)
    task_ref = (Detect [correctness exception]; T5.2.1.2; and)} }
    softgoal = (Robustness [unexpected action]; S5.1.1.3; make) {} }
    softgoal = (Correctness; S5.3; help) {
    softgoal = (Accuracy [data]; S5.3.1; ) {
    task_ref = (Verify if [data identification] is unique; T5.4; )
    task_ref = (Verify if [enters] are valid; T5.5; )}
    softgoal = (Completeness [data]; S5.3.2; ) {
    task_ref = (Verify if every [data] are filled up; T5.6; )} } }
    goal =  (Handling [exception] ; G5.2; ) {
    task = (Detect [exception]; T5.2.1; and) {
    task = (Detect [persistence exception]; T5.2.1.3; and) {}
    task = (Detect [persistence exception]; T5.2.1.3; and) {}
    task = (Detect [robustness exception]; T5.2.1.1; and) {}
    task = (Detect [correctness exception]; T5.2.1.2; and) {}
    task = (Detect [robustness exception]; T5.2.1.1; and) {}
    task = (Detect [correctness exception]; T5.2.1.2; and) {}
    task = (Detect [robustness exception]; T5.2.1.1; and) {}
    task = (Detect [correctness exception]; T5.2.1.2; and) {} }
    task = (Apply [dealer]; T5.2.4; and) {} }
    crosscutting {source = Detect [exception](T5.2.1)
        pointcut (invalid_data; PC5.1): include(Verify if [data identification] is unique; T5.4) and include(Verify if
[enters] are valid; T5.5)
        pointcut (incomplete_data; PC5.2): include(Verify if every [data] are filled up; T5.6)
        pointcut (persistence; PC5.3): include(Persistence in [BD]; T15.1.1) and include(Persistence in [file]; T15.1.2)
        pointcut (insert_data; PC5.4): include(Insert.*; task; name)
        advice (around): PC5.4 {
            task_ref = (Verify if [data identification] is unique; T5.4; )
            task_ref = (Verify if [enters] are valid; T5.5; )
            task_ref = (Verify if every [data] are filled up; T5.6; )}
        intertype declaration (element): PC5.3 {
            task = (Detect [persistence exception]; T5.2.1.3; and) {} }
        intertype declaration (attribute): PC5.1 {exception = Invalid data; }
        intertype declaration (attribute): PC5.2 {exception = Incomplete data; }
        intertype declaration (element): PC5.1 and PC5.2 {
            task = (Detect [robustness exception]; T5.2.1.1; and) {}
            task = (Detect [correctness exception]; T5.2.1.2; and) {} } }
    correlation (help) {
        source = goal_ref = (Handling [exception] ; G5.2; )
        target = softgoal_ref = (Robustness; S5.1.1; )}
    correlation (help) {
        source = goal_ref = (Handling [exception] ; G5.2; )
        target = softgoal_ref = (Correctness; S5.3; )}
    correlation (help) {
        source = softgoal_ref = (Reliability; S5.1; )
        target = goal_ref = (Use [modeling tool] in the WEB; G2; )}
    correlation (help) {
        source = softgoal_ref = (Reliability; S5.1; )
        target = softgoal_ref = (Security; S9.0; )} }
goal_model (Security; GM9) {
    softgoal = (Security; S9.0; ) {
    softgoal = (Confidentiality; S9.1; make) {}
    softgoal = (Integrity; S9.2; make) {}
    softgoal = (Availability; S9.3; make) {} }
    goal =  (Authentication; G9.3; ) {
    task = (Authentication by Certification; T9.3.2; or) {}
    task = (Authentication by Login; T9.3.3; or) {
    task_ref = (Symmetric cryptography; T9.2.4; )}
    task = (Validate [data]; T9.3.4; ) {
    task = (Compare [data] with [credentials]; T9.3.4.2; ) {} }
    goal =  (Authorization; G9.3.5; ) {
```

```
task = (Select [permissions] to [data]; T9.3.5.1; ) {}
task = (Set [permissions]; T9.3.5.2; ) {}
task = (Control [access]; T9.3.5.3; ) {
task = (Restrict [access] to [functions]; T9.3.5.3.1; ) {}
task = (Restrict [access] to [data]; T9.3.5.3.2; ) {}
crosscutting {source = Authentication(G9.3)
      pointcut (Authetication_login; PC9.3.1): include(Manage [project]; G2.1) and
      advice (around): PC9.3.1 {
            task_ref = (Restrict [access] to [data]; T9.3.5.3.2; )
            task_ref = (Restrict [access] to [functions]; T9.3.5.3.1; )
            task_ref = (Authentication by Login; T9.3.3; )}
      intertype declaration (element): PC9.3.1 {
            task = (Logout; T9.3.3.1; or) {} }
      intertype declaration (element): PC9.3.1 {constraint = confidential data; } } } }
goal =  (Accounting; G9.3.6; ) {
task = (Measure [system time] used by [user]; T9.3.6.1; ) {}
task = (Measure [data] sent by [user]; T9.3.6.2; ) {}
task = (Measure [data] received by [user]; T9.3.6.3; ) {} }
task = (Logout; T9.3.3.1; or) {}
task = (Logout; T9.3.3.1; or) {}
task = (Logout; T9.3.3.1; or) {}
correlation (make) {
      source = softgoal_ref = (Authentication; G9.3; )
      target = softgoal_ref = (Confidentiality; S9.1; )} }
task = (Cryptography; T9.2; ) {
task = (Assymmetric cryptography; T9.2.3; or) {}
task = (Symmetric cryptography; T9.2.4; or) {}
task = (Make [cryptography operations]; T9.2.5; ) {
task = (Encrypt [data]; T9.2.1; ) {}
task = (Decrypt [data]; T9.2.2; ) {} }
correlation (help) {
      source = task_ref = (Cryptography; T9.2; )
      target = softgoal_ref = (Confidentiality; S9.1; )}
crosscutting {source = Cryptography(T9.2)
      pointcut (Symetric; PC9.3.5.1): include(Authentication by Login; T9.3.3) and include(Register [user]; T2.4)
      advice (around): PC9.3.5.1 {
            task_ref = (Symmetric cryptography; T9.2.4; )} } }
correlation (help) {
      source = softgoal_ref = (Confidentiality; S9.1; )
      target = goal_ref = (Use [modeling tool] in the WEB; G2; )} }
```

Apêndice D
Especificação do Estudo de Caso – Unical

Neste apêndice inluimos a modelagem realizada par o sistema Unical. Os documentos gerados são: lista de softmetas, metas e tarefas, cenários; diagrama de classes; e modelo entidade relacionamento. Estes documentos também foram gerados a partir da descrição em XML com o apoio da implementação desenvolvida. No final deste apêndice incluímos a descrição do sistema Unical formatada de acordo com a BNF descrita no Apêndice A. Todos estes documentos retratam as informações contidas nos AOV-graph após a composição.

Lista de softmetas, metas e tarefas

1 Model: Unical (96)
1.1Adoption of system to divulge events
1.1.1.Manage [schedule]
1.1.1.1.Schedule [event]
1.1.1.1.1.Edit [event]
1.1.1.1.1.1.Add [event name]
1.1.1.1.1.2.Add [start time]
1.1.1.1.1.3.Add [end time]
1.1.1.1.1.4.Add [location]
1.1.1.1.1.4.1.Add [room]
1.1.1.1.1.4.2.Add [build]
1.1.1.1.1.4.3.Add [street]
1.1.1.1.1.4.4.Add [others]
1.1.1.1.1.5.Add [event description]
1.1.1.1.1.6.Set [recurrence]
1.1.1.1.1.6.1.Select [no recurrence]
1.1.1.1.1.6.2.Select [weekly recurrence]
1.1.1.1.1.6.3.Select [monthly recurrence]
1.1.1.1.1.6.4.Select [yearly recurrence]
1.1.1.1.1.7.Choose [category]
1.1.1.1.1.8.Set [reminder]
1.1.1.1.1.8.1.Select [0, 5, 10, 15 or 30 minutes]
1.1.1.1.1.8.2.Select [1 to 12, or 18 hours]
1.1.1.1.1.8.3.Select [1 to 6 days]
1.1.1.1.1.8.4.Select [1 to 2 weeks]
1.1.1.1.1.9.**Verify if [data identification] is unique**
1.1.1.1.1.10.**Verify if [enters] are valid**
1.1.1.1.1.11.**Verify if every [data] are filled up**
1.1.1.1.2.Set [event] to be viewed
1.1.1.1.2.1.**Mark the [category name] are visible in the [calendar view window]**
1.1.1.1.2.2.**Unmark the [category name] are invisible in the [calendar view window]**
1.1.1.1.3.**View in [calendar window]**
1.1.1.1.4.**View in [monthly calendar]**
1.1.1.1.5.**View in [weekly calendar]**
1.1.1.1.6.**Set [default recurrence]**
1.1.1.1.7.**Set [default reminder]**
1.1.1.2.Edit [category]
1.1.1.2.1.Add [category name]
1.1.1.2.2.Add [category description]
1.1.1.2.3.Set [export flag]
1.1.1.2.3.1.**Mark the [task] as completed in the [task list window]**
1.1.1.2.3.2.**Unmark the [task] as not completed in the [task list window]**
1.1.1.2.4.Import [category]
1.1.1.2.4.1.Choose [category] from a list of all user categories
1.1.1.2.4.2.Edit imported [category]
1.1.1.2.4.2.1.**Verify if [data identification] is unique**
1.1.1.2.4.2.2.**Verify if [enters] are valid**
1.1.1.2.4.2.3.**Verify if every [data] are filled up**
1.1.1.2.4.3.Import [event] that is in imported [category]
1.1.1.2.4.3.1.**Set [reminder]**
1.1.1.2.5.**View in [list window]**
1.1.1.2.6.**Set [color]**
1.1.1.2.7.**Set [default sorting]**
1.1.1.2.8.**Verify if [data identification] is unique**
1.1.1.2.9.**Verify if [enters] are valid**
1.1.1.2.10.**Verify if every [data] are filled up**
1.1.1.3.Schedule [task]
1.1.1.3.1.Edit [task]
1.1.1.3.1.1.Add [task name]
1.1.1.3.1.2.Add [task description]
1.1.1.3.1.3.Add [due]
1.1.1.3.1.3.1.Add [date]
1.1.1.3.1.3.2.Add [time]
1.1.1.3.1.4.**Set [reminder]**
1.1.1.3.1.5.**Verify if [data identification] is unique**
1.1.1.3.1.6.**Verify if [enters] are valid**
1.1.1.3.1.7.**Verify if every [data] are filled up**
1.1.1.3.2.Set [task] to be viewed
1.1.1.3.2.1.**Mark the [category name] to be exported**
1.1.1.3.2.2.**Unmark the [category name] to be not exported**
1.1.1.3.3.**View in [calendar window]**
1.1.1.3.4.**View in [monthly calendar]**
1.1.1.3.5.**View in [weekly calendar]**
1.1.1.3.6.**View in [list window]**
1.1.1.3.7.**Set [default recurrence]**
1.1.1.3.8.**Set [default reminder]**

1.1.1.4.Remind [event]
1.1.1.4.1.Verify if [time] and [date] are greater than [reminder] scheduled time
1.1.1.4.2.Play sound
1.1.1.4.3.Display [event] properties
1.1.1.4.3.1.**Use [reminder pop up window]**
1.1.1.5.Dismiss [reminder]
1.1.1.5.1.Stop sound
1.1.1.5.2.**Use [reminder pop up window]**
1.1.1.6.Snooze [reminder]
1.1.1.6.1.**Stop sound**
1.1.1.6.2.Reschedule [reminder] to 5 minutes later
1.1.1.6.3.**Use [reminder pop up window]**
1.1.1.7.Updating
1.1.1.7.1.Share [event]
1.1.1.7.1.1.Use [export flag]
1.1.1.7.2.Synchronyze [event]
1.1.1.8.**Persistence in [BD]**
1.1.1.9.**Restrict [access] to [data]**
1.1.1.10.**Restrict [access] to [functions]**
1.1.1.11.**Authentication by Login**
1.1.1.12.**Logout**

2 Model: Usability (58)
2.1Usability
2.1.1.Usability [documentation]
2.1.1.1.Usability [tutorial]
2.1.1.2.Usability [help] available on user interface
2.1.2.Usability [user interface]
2.1.2.1.Accessibility
2.1.2.1.1.Access [functionality] through [menu] itens
2.1.2.1.2.Access [functionality] with mouse
2.1.2.1.3.Access [functionality] with keyboard
2.1.2.2.Use [check box]
2.1.2.2.1.Mark the [category name] to be exported
2.1.2.2.2.Unmark the [category name] to be not exported
2.1.2.2.3.Mark the [category name] are visible in the [calendar view window]
2.1.2.2.4.Unmark the [category name] are invisible in the [calendar view window]
2.1.2.2.5.Mark the [task] as completed in the [task list window]
2.1.2.2.6.Unmark the [task] as not completed in the [task list window]
2.1.2.3.Use [pop up window]
2.1.2.3.1.Use [reminder pop up window]
2.1.2.3.2.Use [reminder pop up window]
2.1.2.3.3.Use [reminder pop up window]
2.1.2.3.4.**Resize [window]**
2.1.2.4.View in [calendar window]
2.1.2.4.1.View in [monthly calendar]
2.1.2.4.2.View in [weekly calendar]
2.1.2.5.View in [monthly calendar]
2.1.2.6.View in [weekly calendar]
2.1.2.7.View in [calendar window]
2.1.2.7.1.View in [monthly calendar]
2.1.2.7.2.View in [weekly calendar]
2.1.2.8.View in [monthly calendar]
2.1.2.9.View in [weekly calendar]
2.1.2.10.View in [list window]
2.1.2.11.View in [list window]
2.2Configurability
2.2.1.Resize [window]
2.2.1.1.**Resize [window]**
2.2.2.Set [color]
2.2.3.Set [default sorting]
2.2.4.Set [default recurrence]
2.2.5.Set [default reminder]
2.2.6.Set [default recurrence]
2.2.7.Set [default reminder]
2.2.8.**Persistence in [file]**

3 Model: Persistence (39)
3.1Persistence
3.1.1.Persistence in [BD]
3.1.1.1.Verify if [BD] is connected
3.1.1.2.Initiate [BD]
3.1.1.3.Connect [BD]
3.1.1.3.1.**Restrict [access] to [data]**
3.1.1.3.2.**Restrict [access] to [functions]**
3.1.1.3.3.**Authentication by Login**
3.1.1.3.4.**Logout**
3.1.1.4.**Include [data]**
3.1.1.5.**Select [data]**
3.1.1.6.**Delete [data]**
3.1.1.7.**Update [data]**
3.1.1.8.Disconnect [BD]
3.1.1.9.**Include [data]**
3.1.1.10.**Select [data]**
3.1.1.11.**Delete [data]**
3.1.1.12.**Update [data]**
3.1.1.13.**Detect [persistence exception]**
3.1.2.Persistence in [file]
3.1.2.1.Verify if [file] exists
3.1.2.2.Create [file]
3.1.2.3.Open [file]
3.1.2.4.**Include [data]**
3.1.2.5.**Select [data]**
3.1.2.6.**Delete [data]**
3.1.2.7.**Update [data]**
3.1.2.8.Close [file]
3.1.2.9.Delete [file]
3.1.2.10.**Include [data]**
3.1.2.11.**Select [data]**
3.1.2.12.**Delete [data]**
3.1.2.13.**Update [data]**
3.1.2.14.**Detect [persistence exception]**
3.1.3.Make register operation
3.1.3.1.Include [data]
3.1.3.2.Select [data]
3.1.3.3.Delete [data]
3.1.3.4.Update [data]

4 Model: Reliability (34)
4.1Reliability
4.1.1.Robustness
4.1.1.1.Robustness [incomplete data]
4.1.1.1.1.Verify if every [data] are filled up
4.1.1.1.1.1.**Detect [robustness exception]**
4.1.1.1.1.2.**Detect [correctness exception]**
4.1.1.2.Robustness [invalid data]
4.1.1.2.1.Verify if [data identification] is unique
4.1.1.2.1.1.**Detect [robustness exception]**
4.1.1.2.1.2.**Detect [correctness exception]**
4.1.1.2.2.Verify if [enters] are valid
4.1.1.2.2.1.**Detect [robustness exception]**
4.1.1.2.2.2.**Detect [correctness exception]**
4.1.1.3.Robustness [unexpected action]
4.1.2.Correctness
4.1.2.1.Accuracy [data]
4.1.2.1.1.**Verify if [data identification] is unique**
4.1.2.1.2.**Verify if [enters] are valid**
4.1.2.2.Completeness [data]
4.1.2.2.1.**Verify if every [data] are filled up**
4.2Handling [exception]
4.2.1.Detect [exception]
4.2.1.1.Detect [persistence exception]
4.2.1.2.Detect [persistence exception]
4.2.1.3.Detect [robustness exception]
4.2.1.4.Detect [correctness exception]
4.2.1.5.Detect [robustness exception]
4.2.1.6.Detect [correctness exception]
4.2.1.7.Detect [robustness exception]
4.2.1.8.Detect [correctness exception]
4.2.2.Apply [dealer]

5 Model: Security (53)
5.1Security
5.1.1.Confidentiality
5.1.2.Integrity
5.1.3.Availability

5.2Authentication
5.2.1.Authentication by Certification
5.2.2.Authentication by Login
5.2.3.Validate [data]
5.2.3.1.Compare [data] with [credentials]
5.2.4.Authorization
5.2.4.1.Select [permissions] to [data]
5.2.4.2.Set [permissions]
5.2.4.3.Control [access]
5.2.4.3.1.Restrict [access] to [functions]
5.2.4.3.2.Restrict [access] to [data]
5.2.5.Accounting
5.2.5.1.Measure [system time] used by [user]
5.2.5.2.Measure [data] sent by [user]
5.2.5.3.Measure [data] received by [user]
5.2.6.Manage [account]
5.2.6.1.Edit [account]
5.2.6.1.1.Edit [user account]
5.2.6.1.1.1.**Verify if [data identification] is unique**
5.2.6.1.1.2.**Verify if [enters] are valid**
5.2.6.1.1.3.**Verify if every [data] are filled up**
5.2.6.1.2.Edit [administrator account]
5.2.6.1.2.1.Add [e-mail]
5.2.6.1.2.2.**Verify if [data identification] is unique**
5.2.6.1.2.3.**Verify if [enters] are valid**
5.2.6.1.2.4.**Verify if every [data] are filled up**
5.2.6.1.3.Add [username]
5.2.6.1.4.Add [password]
5.2.6.1.4.1.**Symmetric cryptography**
5.2.6.1.5.Add [full name]
5.2.6.1.6.Reset [password]
5.2.6.1.6.1.**Symmetric cryptography**
5.2.6.1.7.Change [password]
5.2.6.1.7.1.**Symmetric cryptography**
5.2.6.2.There is always at least one [administrator account]
5.2.6.3.**Restrict [access] to [data]**
5.2.6.4.**Restrict [access] to [functions]**
5.2.6.5.**Authentication by Login**
5.2.6.6.**Logout**
5.2.7.Logout
5.2.8.Logout
5.2.9.Logout
5.3Cryptography
5.3.1.Assymmetric cryptography
5.3.2.Symmetric cryptography
5.3.3.Make [cryptography operations]
5.3.3.1.Encrypt [data]
5.3.3.2.Decrypt [data]

6 Model: Portability (7)
6.1Portability
6.1.1.Portability [plataform]
6.1.1.1.Use [Microsoft Windows]
6.1.1.2.Use [Macintosh OS]
6.1.1.3.Use [PDA]
6.1.1.4.Use [Web]
6.1.1.5.Program with [Java TM (jdk 1.4)]

7 Model: Performance (3)
7.1Performance
7.1.1.Performance [time]
7.1.2.Performance [resource]

8 Model: Reusability (2)
8.1Reusability
8.1.1.Reusability [system]

9 Model: Maintainability (6)
9.1Maintainability
9.1.1.Reparability
9.1.2.Evolvability
9.2Understandability
9.2.1.Understandability [code]
9.2.2.Understandability [documentation]

10 Model: Scalability (2)
10.1Scalability
10.1.1.Scalability [quantity of users]

11 Model: Timeliness (5)
11.1Timeliness
11.1.1.Timeliness [development time]
11.1.1.1.Timeliness [design]
11.1.1.2.Timeliness [implementation]
11.1.1.3.Timeliness [test]

12 Model: Profitable (2)
12.1Profitable
12.1.1.Profitable [development cost]

Cenários

Title	Adoption of system to divulge events
Episode	Manage [schedule]

Title	Manage [schedule]
Actor	User
Episode	Schedule [event] Edit [category] Schedule [task] Remind [event] Dismiss [reminder] Snooze [reminder] Updating Persistence in [BD] Restrict [access] to [data] Restrict [access] to [functions] Authentication by Login [Logout]

Title	Schedule [event]
Episode	Edit [event] [Set [event] to be viewed] View in [calendar window] View in [monthly calendar] View in [weekly calendar] Set [default recurrence] Set [default reminder]

Title	Add [location]
Constraint	performance: instantaneous
Episode	Add [room] Add [build] Add [street] Add [others]

Title	Edit [event]
Constraint	performance: instantaneous
Episode	Add [event name] Add [start time] Add [end time] Add [location] Add [event description] Set [recurrence] Choose [category] Set [reminder] Verify if [data identification] is unique

	Verify if [enters] are valid Verify if every [data] are filled up

Title	Set [recurrence]
Constraint	performance: instantaneous
Episode	[Select [no recurrence]] [Select [weekly recurrence]] [Select [monthly recurrence]] [Select [yearly recurrence]]

Title	Set [reminder]
Constraint	performance: instantaneous
Episode	[Select [0, 5, 10, 15 or 30 minutes]] [Select [1 to 12, or 18 hours]] [Select [1 to 6 days]] [Select [1 to 2 weeks]]

Title	Set [event] to be viewed
Constraint	performance: instantaneous
Episode	Mark the [category name] are visible in the [calendar view window] Unmark the [category name] are invisible in the [calendar view window]

Title	Edit [category]
Constraint	performance: instantaneous
Episode	Add [category name] Add [category description] Set [export flag] [Import [category]] View in [list window] Set [color] Set [default sorting] Verify if [data identification] is unique Verify if [enters] are valid Verify if every [data] are filled up

Title	Set [export flag]
Constraint	performance: instantaneous
Episode	Mark the [task] as completed in the [task list window] Unmark the [task] as not completed in the [task list window]

Title	Import [category]
Constraint	performance: instantaneous
Episode	Choose [category] from a list of all user categories [Edit imported [category]] [Import [event] that is in imported [category]]

Title	Edit imported [category]
Constraint	performance: instantaneous
Episode	Verify if [data identification] is unique Verify if [enters] are valid Verify if every [data] are filled up

Title	Import [event] that is in imported [category]
Constraint	performance: instantaneous
Episode	[Set [reminder]]

Title	Schedule [task]
Episode	Edit [task] [Set [task] to be viewed]

	View in [calendar window] View in [monthly calendar] View in [weekly calendar] View in [list window] Set [default recurrence] Set [default reminder]

Title	Edit [task]
Constraint	performance: instantaneous
Episode	Add [task name] Add [task description] Add [due] Set [reminder] Verify if [data identification] is unique Verify if [enters] are valid Verify if every [data] are filled up

Title	Add [due]
Constraint	performance: instantaneous
Episode	Add [date] Add [time]

Title	Set [task] to be viewed
Constraint	performance: instantaneous
Episode	Mark the [category name] to be exported Unmark the [category name] to be not exported

Title	Remind [event]
Constraint	performance: instantaneous
Episode	Verify if [time] and [date] are greater than [reminder] scheduled time Play sound Display [event] properties

Title	Display [event] properties
Constraint	performance: instantaneous
Episode	Use [reminder pop up window]

Title	Dismiss [reminder]
Constraint	performance: instantaneous
Episode	Stop sound Use [reminder pop up window]

Title	Snooze [reminder]
Constraint	performance: instantaneous
Episode	Stop sound Reschedule [reminder] to 5 minutes later Use [reminder pop up window]

Title	Updating
Constraint	performance: 1 second to 50 events
Episode	Share [event] Synchronyze [event]

Title	Share [event]
Episode	Use [export flag]

Title	Use [check box]
Constraint	performance: instantaneous
Episode	Mark the [category name] to be exported Unmark the [category name] to be not exported Mark the [category name] are visible in the

	[calendar view window] Unmark the [category name] are invisible in the [calendar view window] Mark the [task] as completed in the [task list window] Unmark the [task] as not completed in the [task list window]

Title	Use [pop up window]
Constraint	performance: instantaneous
Episode	Use [reminder pop up window] Resize [window]

Title	View in [calendar window]
Episode	View in [monthly calendar] View in [weekly calendar]

Title	Resize [window]
Constraint	performance: instantaneous
Episode	Resize [window]

Title	Persistence
Episode	[Persistence in [BD]] [Persistence in [file]] Make register operation

Title	Persistence in [BD]
Constraint	performance: instantaneous
Episode	Verify if [BD] is connected Initiate [BD] Connect [BD] Include [data] Select [data] Delete [data] Update [data] Disconnect [BD] Detect [persistence exception]

Title	Connect [BD]
Constraint	performance: instantaneous
Episode	Restrict [access] to [data] Restrict [access] to [functions] Authentication by Login [Logout]

Title	Persistence in [file]
Constraint	performance: instantaneous
Episode	Verify if [file] exists Create [file] Open [file] Include [data] Select [data] Delete [data] Update [data] Close [file] Delete [file] Detect [persistence exception]

Title	Make register operation
Constraint	performance: instantaneous
Episode	[Include [data]] [Select [data]] [Delete [data]] [Update [data]]

Title	Verify if every [data] are filled up
Constraint	performance: instantaneous
Exception	Incomplete data
Episode	Detect [robustness exception] Detect [correctness exception]

Title	Verify if [data identification] is unique
Constraint	performance: instantaneous
Exception	Invalid data
Episode	Detect [robustness exception] Detect [correctness exception]

Title	Verify if [enters] are valid
Constraint	performance: instantaneous
Exception	Invalid data
Episode	Detect [robustness exception] Detect [correctness exception]

Title	Handling [exception]
Context	Robustness; Correctness;
Episode	Detect [exception] Apply [dealer]

Title	Detect [exception]
Constraint	performance: instantaneous
Episode	Detect [persistence exception] Detect [robustness exception] Detect [correctness exception]

Title	Authentication
Context	Confidentiality;
Constraint	confidential data
Episode	[Authentication by Certification] [Authentication by Login] Validate [data] Authorization Accounting Manage [account] [Logout]

Title	Validate [data]
Constraint	performance: instantaneous
Episode	Compare [data] with [credentials]

Title	Authorization
Episode	Select [permissions] to [data] Set [permissions] Control [access]

Title	Control [access]
Constraint	performance: instantaneous
Episode	Restrict [access] to [functions] Restrict [access] to [data]

Title	Accounting
Episode	Measure [system time] used by [user] Measure [data] sent by [user] Measure [data] received by [user]

Title	Manage [account]
Actor	Administrator user
Episode	Edit [account] There is always at least one [administrator account]

	Restrict [access] to [data] Restrict [access] to [functions] Authentication by Login [Logout]

Title	Edit [account]
Episode	[Edit [user account]] [Edit [administrator account]] Add [username] Add [password] Add [full name] Reset [password] Change [password]

Title	Edit [user account]
Constraint	performance: instantaneous
Episode	Verify if [data identification] is unique Verify if [enters] are valid Verify if every [data] are filled up

Title	Edit [administrator account]
Constraint	performance: instantaneous
Episode	Add [e-mail] Verify if [data identification] is unique Verify if [enters] are valid Verify if every [data] are filled up

Title	Add [password]
Constraint	performance: instantaneous
Episode	Symmetric cryptography

Title	Reset [password]
Constraint	performance: instantaneous
Episode	Symmetric cryptography

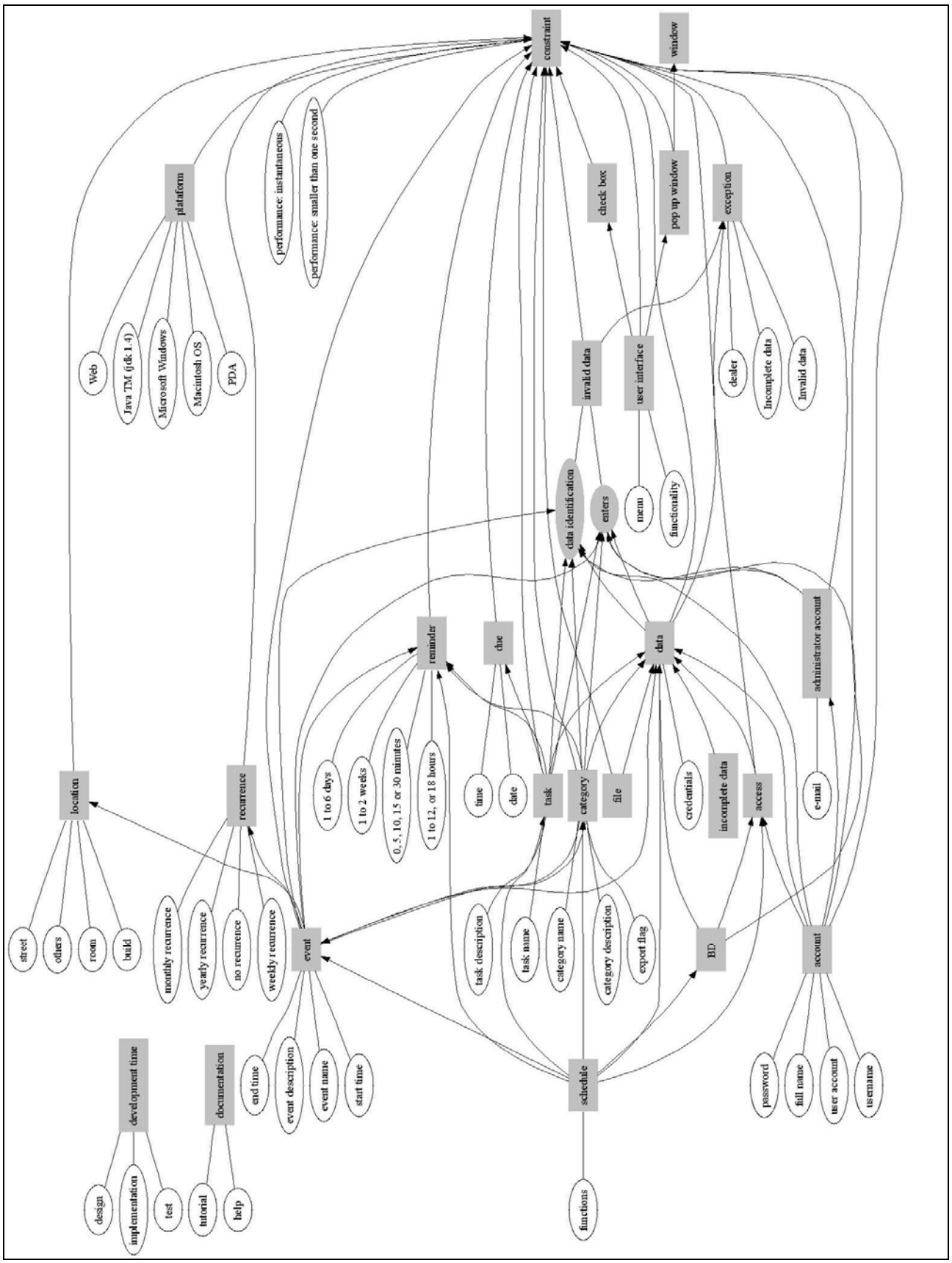
Title	Change [password]
Constraint	performance: instantaneous
Episode	Symmetric cryptography

Title	Cryptography
Context	Confidentiality;
Constraint	performance: instantaneous
Episode	[Assymmetric cryptography] [Symmetric cryptography] Make [cryptography operations]

Title	Make [cryptography operations]
Constraint	performance: instantaneous
Episode	Encrypt [data] Decrypt [data]

Diagrama de classes do Unical após a composição

Modelo de entidade-relacionamento do Unical após a composição



BNF

goal_model (Unical; GM1) {
goal = (Adoption of system to divulge events; S1;) {
goal = (Manage [schedule]; G1.10; and) {
goal = (Schedule [event]; T1.1.1;) {
task = (Edit [event]; T1.1.1.1;) {
task = (Add [event name]; T1.1.1.1.1;) {}
task = (Add [start time]; T1.1.1.1.2;) {}
task = (Add [end time]; T1.1.1.1.3;) {}
task = (Add [location]; T1.1.1.1.4;) {
task = (Add [room]; T1.1.1.1.4.1;) {}
task = (Add [build]; T1.1.1.1.4.2;) {}
task = (Add [street]; T1.1.1.1.4.3;) {}
task = (Add [others]; T1.1.1.1.4.4;) {} }
task = (Add [event description]; T1.1.1.1.5;) {}
task = (Set [recurrence]; T1.1.1.1.6;) {
task = (Select [no recurrence]; T1.1.1.1.6.1; or) {}
task = (Select [weekly recurrence]; T1.1.1.1.6.2; or) {}
task = (Select [monthly recurrence]; T1.1.1.1.6.3; or) {}
task = (Select [yearly recurrence]; T1.1.1.1.6.4; or) {} }
task = (Choose [category]; T1.1.1.1.7; and) {}
task = (Set [reminder]; T1.1.1.1.8; and) {
task = (Select [0, 5, 10, 15 or 30 minutes]; T1.1.1.1.8.1; or) {}
task = (Select [1 to 12, or 18 hours]; T1.1.1.1.8.2; or) {}
task = (Select [1 to 6 days]; T1.1.1.1.8.3; or) {}
task = (Select [1 to 2 weeks]; T1.1.1.1.8.4; or) {} }
task_ref = (Verify if [data identification] is unique; T5.4;)
task_ref = (Verify if [enters] are valid; T5.5;)
task_ref = (Verify if every [data] are filled up; T5.6;)}
task = (Set [event] to be viewed; T1.1.1.1.9; or) {
task_ref = (Mark the [category name] are visible in the [calendar view window]; T3.1.2.5.3;)
task_ref = (Unmark the [category name] are invisible in the [calendar view window]; T3.1.2.5.4;)}
task_ref = (View in [calendar window]; T3.1.2.5.7;)
task_ref = (View in [monthly calendar]; T3.1.2.5.7.1;)
task_ref = (View in [weekly calendar]; T3.1.2.5.7.2;)
task_ref = (Set [default recurrence]; T3.1.2.5.11;)
task_ref = (Set [default reminder]; T3.1.2.5.12;)}
task = (Edit [category]; T1.1.5;) {
task = (Add [category name]; T1.1.5.1; and) {}
task = (Add [category description]; T1.1.5.4; and) {}
task = (Set [export flag]; T1.1.5.5; and) {
task_ref = (Mark the [task] as completed in the [task list window]; T3.1.2.5.5;)
task_ref = (Unmark the [task] as not completed in the [task list window]; T3.1.2.5.6;)}
task = (Import [category]; T1.1.5.2; or) {
task = (Choose [category] from a list of all user categories; T1.1.1.5.2.1;) {}
task = (Edit imported [category]; T1.1.1.5.2.2; or) {
task_ref = (Verify if [data identification] is unique; T5.4;)
task_ref = (Verify if [enters] are valid; T5.5;)
task_ref = (Verify if every [data] are filled up; T5.6;)}
task = (Import [event] that is in imported [category]; T1.1.1.5.2.3; or) {
task_ref = (Set [reminder]; T1.1.1.1.8; or)} }
task_ref = (View in [list window] View in [list window] View in [list window]; T3.1.2.5.8;)
task_ref = (Set [color] Set [color]; T3.1.2.5.9;)
task_ref = (Set [default sorting] Set [default sorting]; T3.1.2.5.10;)
task_ref = (Verify if [data identification] is unique; T5.4;)
task_ref = (Verify if [enters] are valid; T5.5;)
task_ref = (Verify if every [data] are filled up; T5.6;)}
goal = (Schedule [task]; G1.2;) {
task = (Edit [task]; T1.2.1;) {
task = (Add [task name]; T1.2.1.1;) {}
task = (Add [task description]; T1.2.1.2;) {}
task = (Add [due]; T1.2.1.3;) {
task = (Add [date]; T1.2.1.3.1;) {}
task = (Add [time]; T1.2.1.3.2;) {} }
task_ref = (Set [reminder]; T1.1.1.1.8;)
task_ref = (Verify if [data identification] is unique; T5.4;)
task_ref = (Verify if [enters] are valid; T5.5;)
task_ref = (Verify if every [data] are filled up; T5.6;)}
task = (Set [task] to be viewed; T1.2.2; or) {
task_ref = (Mark the [category name] to be exported; T3.1.2.5.1;)

```
        task_ref = (Unmark the [category name] to be not exported; T3.1.2.5.2; )}
        task_ref = (View in [calendar window]; T3.1.2.5.7; )
        task_ref = (View in [monthly calendar]; T3.1.2.5.7.1; )
        task_ref = (View in [weekly calendar] ; T3.1.2.5.7.2; )
        task_ref = (View in [list window]; T3.1.2.5.8; )
        task_ref = (Set [default recurrence]; T3.1.2.5.11; )
        task_ref = (Set [default reminder]; T3.1.2.5.12; )}
        task = (Remind [event]; T1.3; ) {
        task = (Verify if [time] and [date] are greater than [reminder] scheduled time; T1.3.3; ) {}
        task = (Play sound; T1.3.1; ) {}
        task = (Display [event] properties; T1.3.2; ) {
        task_ref = (Use [reminder pop up window]; T3.1.2.6.1; )} }
        task = (Dismiss [reminder]; T1.7; ) {
        task = (Stop sound; T1.7.1; ) {}
        task_ref = (Use [reminder pop up window]; T3.1.2.6.1; )}
        task = (Snooze [reminder]; T1.8; ) {
        task_ref = (Stop sound; T1.7.1; )
        task = (Reschedule [reminder] to 5 minutes later; T1.8.1; ) {}
        task_ref = (Use [reminder pop up window]; T3.1.2.6.1; )}
        goal =  (Updating; G1.9; ) {
        goal =  (Share [event]; G1.9.1; ) {
        task = (Use [export flag]; T1.9.1.1; and) {} }
        goal =  (Synchronyze [event]; G1.9.2; ) {} }
        task_ref = (Persistence in [BD]; T15.1.1; )
        task_ref = (Restrict [access] to [data]; T9.3.5.3.2; )
        task_ref = (Restrict [access] to [functions]; T9.3.5.3.1; )
        task_ref = (Authentication by Login; T9.3.3; )
        task_ref = (Logout Logout Logout Logout; T9.3.3.1; or)} } }

    goal_model (Usability; GM3) {
        softgoal = (Usability; S3.1; ) {
        softgoal = (Usability [documentation]; S3.1.1; make) {
        softgoal = (Usability [tutorial]; S3.1.1.1; make) {}
        softgoal = (Usability [help] available on user interface; S3.1.1.2; make) {} }
        softgoal = (Usability [user interface]; S3.1.2; make) {
        softgoal = (Accessibility; S3.1.2.7; help) {
        task = (Access [functionality] through [menu] itens; T3.1.2.1; make) {}
        task = (Access [functionality] with mouse; T3.1.2.2; make) {}
        task = (Access [functionality] with keyboard; T3.1.2.3; make) {} }
        task = (Use [check box]; T3.1.2.5; make) {
        task = (Mark the [category name] to be exported; T3.1.2.5.1; ) {}
        task = (Unmark the [category name] to be not exported; T3.1.2.5.2; ) {}
        task = (Mark the [category name] are visible in the [calendar view window]; T3.1.2.5.3; ) {}
        task = (Unmark the [category name] are invisible in the [calendar view window]; T3.1.2.5.4; ) {}
        task = (Mark the [task] as completed in the [task list window]; T3.1.2.5.5; ) {}
        task = (Unmark the [task] as not completed in the [task list window]; T3.1.2.5.6; ) {} }
        task = (Use [pop up window]; T3.1.2.6; make) {
        task = (Use [reminder pop up window]; T3.1.2.6.1; ) {}
        task = (Use [reminder pop up window]; T3.1.2.6.1; ) {}
        task = (Use [reminder pop up window]; T3.1.2.6.1; ) {}
        task_ref = (Resize [window]; T3.1.2.4; )}
        task = (View in [calendar window]; T3.1.2.5.7; ) {
        task = (View in [monthly calendar]; T3.1.2.5.7.1; ) {}
        task = (View in [weekly calendar]; T3.1.2.5.7.2; ) {} }
        task = (View in [list window]; T3.1.2.5.8; ) {}
        task = (View in [list window]; T3.1.2.5.8; ) {}
        crosscutting {source = Use [pop up window](T3.1.2.6)
            pointcut (context_menu; PC3.3): include(Display [event] properties; T1.3.2) and
            intertype declaration (element): PC3.3 {
                task = (Use [reminder pop up window]; T3.1.2.6.1; ) {} } }
        crosscutting {source = Use [check box](T3.1.2.5)
            pointcut (check_box; PC3.2): include(Set [event] to be viewed; T1.1.1.1.9)
            pointcut (check_box; PC3.5): include(Set [task] to be viewed; T1.2.2)
            pointcut (check_box; PC3.4): include(Set [export flag]; T1.1.5.5)
            intertype declaration (element): PC3.5 {
                task = (Mark the [category name] to be exported; T3.1.2.5.1; ) {}
                task = (Unmark the [category name] to be not exported; T3.1.2.5.2; ) {} }
            intertype declaration (element): PC3.2 {
                task = (Mark the [category name] are visible in the [calendar view window]; T3.1.2.5.3; ) {}
                task = (Unmark the [category name] are invisible in the [calendar view window]; T3.1.2.5.4; ) {} }
            intertype declaration (element): PC3.4 {
                task = (Mark the [task] as completed in the [task list window]; T3.1.2.5.5; ) {}
                task = (Unmark the [task] as not completed in the [task list window]; T3.1.2.5.6; ) {} } }
        crosscutting {source = Usability [user interface](S3.1.2)
            pointcut (view; PC3.6): include(Schedule [event]; T1.1.1)
            pointcut (view; PC3.7): include(Edit [category]; T1.1.5)
```

```
            pointcut (view; PC3.8): include(Schedule [task]; G1.2)
            intertype declaration (element): PC3.6 and PC3.8 {
                task = (View in [calendar window]; T3.1.2.5.7; ) {
                task = (View in [monthly calendar]; T3.1.2.5.7.1; ) {}
                task = (View in [weekly calendar]; T3.1.2.5.7.2; ) {} } }
            intertype declaration (element): PC3.7 and PC3.8 {
                task = (View in [list window]; T3.1.2.5.8; ) {} } }
        crosscutting {source = Configurability(S3.2)
            pointcut (view; PC3.9): include(Schedule [event]; T1.1.1)
            pointcut (view; PC3.10): include(Edit [category]; T1.1.5)
            pointcut (view; PC3.11): include(Schedule [task]; G1.2)
            pointcut (window; PC3.12): include(.*window; task; name)
            advice (around): PC3.12 {
                task_ref = (Resize [window]; T3.1.2.4; )}
            intertype declaration (element): PC3.10 {
                task = (Set [color]; T3.1.2.5.9; ) {}
                task = (Set [default sorting]; T3.1.2.5.10; ) {} }
            intertype declaration (element): PC3.9 and PC3.11 {
                task = (Set [default recurrence]; T3.1.2.5.11; ) {}
                task = (Set [default reminder]; T3.1.2.5.12; ) {} } } } }
        softgoal = (Configurability; S3.2; make) {
        task = (Resize [window]; T3.1.2.4; make) {
        task_ref = (Resize [window]; T3.1.2.4; )}
        task = (Set [color]; T3.1.2.5.9; ) {}
        task = (Set [default sorting]; T3.1.2.5.10; ) {}
        task = (Set [default recurrence]; T3.1.2.5.11; ) {}
        task = (Set [default reminder]; T3.1.2.5.12; ) {}
        task = (Set [default recurrence]; T3.1.2.5.11; ) {}
        task = (Set [default reminder]; T3.1.2.5.12; ) {}
        task_ref = (Persistence in [file]; T15.1.2; )
        correlation (help) {
            source = softgoal_ref = (Configurability; S3.2; )
            target = softgoal_ref = (Usability; S3.1; )} } }

goal_model (Persistence; GM15) {
        goal =  (Persistence; G15.1; ) {
        task = (Persistence in [BD]; T15.1.1; or) {
        task = (Verify if [BD] is connected; T15.1.1.1; ) {}
        task = (Initiate [BD]; T15.1.1.2; ) {}
        task = (Connect [BD]; T15.1.1.3; ) {
        task_ref = (Restrict [access] to [data]; T9.3.5.3.2; )
        task_ref = (Restrict [access] to [functions]; T9.3.5.3.1; )
        task_ref = (Authentication by Login; T9.3.3; )
        task_ref = (Logout Logout Logout Logout; T9.3.3.1; or)}
        task_ref = (Include [data]; T15.1.3.1; )
        task_ref = (Select [data]; T15.1.3.2; )
        task_ref = (Delete [data]; T15.1.3.3; )
        task_ref = (Update [data]; T15.1.3.4; )
        task = (Disconnect [BD]; T15.1.1.4; ) {}
        task_ref = (Detect [persistence exception]; T5.2.1.3; and)}
        task = (Persistence in [file]; T15.1.2; or) {
        task = (Verify if [file] exists; T15.1.2.1; ) {}
        task = (Create [file]; T15.1.2.2; ) {}
        task = (Open [file]; T15.1.2.3; ) {}
        task_ref = (Include [data]; T15.1.3.1; )
        task_ref = (Select [data]; T15.1.3.2; )
        task_ref = (Delete [data]; T15.1.3.3; )
        task_ref = (Update [data]; T15.1.3.4; )
        task = (Close [file]; T15.1.2.4; ) {}
        task = (Delete [file]; T15.1.2.5; ) {}
        task_ref = (Detect [persistence exception]; T5.2.1.3; and)}
        task = (Make register operation; T15.1.3; and) {
        task = (Include [data]; T15.1.3.1; or) {}
        task = (Select [data]; T15.1.3.2; or) {}
        task = (Delete [data]; T15.1.3.3; or) {}
        task = (Update [data]; T15.1.3.4; or) {} }
        crosscutting {source = Persistence(G15.1)
            pointcut (persistence_BD; PC15.2): include(Manage [schedule]; G1.10)
            pointcut (persistence_file; PC15.3): include(Configurability; S3.2)
            advice (around): PC15.2 {
                task_ref = (Persistence in [BD]; T15.1.1; )}
            advice (around): PC15.3 {
                task_ref = (Persistence in [file]; T15.1.2; )} }
        crosscutting {source = Make register operation(T15.1.3)
            pointcut (register_operation; PC15.1): include(Connect [BD]; T15.1.1.3) and include(Open [file]; T15.1.2.3)
            advice (after): PC15.1 {
```

```
                task_ref = (Include [data]; T15.1.3.1; )
                task_ref = (Select [data]; T15.1.3.2; )
                task_ref = (Delete [data]; T15.1.3.3; )
                task_ref = (Update [data]; T15.1.3.4; )} } } }

goal_model (Reliability; GM5) {
    softgoal = (Reliability; S5.1; ) {
    softgoal = (Robustness; S5.1.1; help) {
    softgoal = (Robustness [incomplete data]; S5.1.1.1; make) {
    task = (Verify if every [data] are filled up; T5.6; make) {
    task_ref = (Detect [robustness exception]; T5.2.1.1; and)
    task_ref = (Detect [correctness exception]; T5.2.1.2; and)} }
    softgoal = (Robustness [invalid data]; S5.1.1.2; make) {
    task = (Verify if [data identification] is unique; T5.4; make) {
    task_ref = (Detect [robustness exception]; T5.2.1.1; and)
    task_ref = (Detect [correctness exception]; T5.2.1.2; and)}
    task = (Verify if [enters] are valid; T5.5; make) {
    task_ref = (Detect [robustness exception]; T5.2.1.1; and)
    task_ref = (Detect [correctness exception]; T5.2.1.2; and)} }
    softgoal = (Robustness [unexpected action]; S5.1.1.3; make) {} }
    softgoal = (Correctness; S5.3; help) {
    softgoal = (Accuracy [data]; S5.3.1; ) {
    task_ref = (Verify if [data identification] is unique; T5.4; )
    task_ref = (Verify if [enters] are valid; T5.5; )}
    softgoal = (Completeness [data]; S5.3.2; ) {
    task_ref = (Verify if every [data] are filled up; T5.6; )} } }
    goal =  (Handling [exception] ; G5.2; ) {
    task = (Detect [exception]; T5.2.1; and) {
    task = (Detect [persistence exception]; T5.2.1.3; and) {}
    task = (Detect [persistence exception]; T5.2.1.3; and) {}
    task = (Detect [robustness exception]; T5.2.1.1; and) {}
    task = (Detect [correctness exception]; T5.2.1.2; and) {}
    task = (Detect [robustness exception]; T5.2.1.1; and) {}
    task = (Detect [correctness exception]; T5.2.1.2; and) {}
    task = (Detect [robustness exception]; T5.2.1.1; and) {}
    task = (Detect [correctness exception]; T5.2.1.2; and) {} }
    task = (Apply [dealer]; T5.2.4; and) {} }
    crosscutting {source = Detect [exception](T5.2.1)
        pointcut (invalid_data; PC5.1): include(Verify if [data identification] is unique; T5.4) and include(Verify if
[enters] are valid; T5.5)
        pointcut (incomplete_data; PC5.2): include(Verify if every [data] are filled up; T5.6)
        pointcut (persistence; PC5.3): include(Persistence in [BD]; T15.1.1) and include(Persistence in [file]; T15.1.2)
        pointcut (insert_data; PC5.4): include(Edit.*; task; name)
        advice (around): PC5.4 {
            task_ref = (Verify if [data identification] is unique; T5.4; )
            task_ref = (Verify if [enters] are valid; T5.5; )
            task_ref = (Verify if every [data] are filled up; T5.6; )}
        intertype declaration (element): PC5.3 {
            task = (Detect [persistence exception]; T5.2.1.3; and) {} }
        intertype declaration (attribute): PC5.1 {exception = Invalid data; }
        intertype declaration (attribute): PC5.2 {exception = Incomplete data; }
        intertype declaration (element): PC5.1 and PC5.2 {
            task = (Detect [robustness exception]; T5.2.1.1; and) {}
            task = (Detect [correctness exception]; T5.2.1.2; and) {} } }
    correlation (help) {
        source = goal_ref = (Handling [exception] ; G5.2; )
        target = softgoal_ref = (Robustness; S5.1.1; )}
    correlation (help) {
        source = goal_ref = (Handling [exception] ; G5.2; )
        target = softgoal_ref = (Correctness; S5.3; )}
    correlation (help) {
        source = softgoal_ref = (Reliability; S5.1; )
        target = goal_ref = (Adoption of system to divulge events; S1; )}
    correlation (help) {
        source = softgoal_ref = (Reliability; S5.1; )
        target = softgoal_ref = (Security; S9.0; )} }

goal_model (Security; GM9) {
    softgoal = (Security; S9.0; ) {
    softgoal = (Confidentiality; S9.1; make) {}
    softgoal = (Integrity; S9.2; make) {}
    softgoal = (Availability; S9.3; make) {} }
    goal =  (Authentication; G9.3; ) {
    task = (Authentication by Certification; T9.3.2; or) {}
    task = (Authentication by Login; T9.3.3; or) {}
    task = (Validate [data]; T9.3.4; ) {
```

```
        task = (Compare [data] with [credentials]; T9.3.4.2; ) {} }
        goal =  (Authorization; G9.3.5; ) {
        task = (Select [permissions] to [data]; T9.3.5.1; ) {}
        task = (Set [permissions]; T9.3.5.2; ) {}
        task = (Control [access]; T9.3.5.3; ) {
        task = (Restrict [access] to [functions]; T9.3.5.3.1; ) {}
        task = (Restrict [access] to [data]; T9.3.5.3.2; ) {}
        crosscutting {source = Authentication(G9.3)
            pointcut (Authetication_login; PC9.3.1): include(Manage [account]; G9.4) and
            advice (around): PC9.3.1 {
                task_ref = (Restrict [access] to [data]; T9.3.5.3.2; )
                task_ref = (Restrict [access] to [functions]; T9.3.5.3.1; )
                task_ref = (Authentication by Login; T9.3.3; )}
            intertype declaration (element): PC9.3.1 {
                task = (Logout; T9.3.3.1; or) {} }
            intertype declaration (element): PC9.3.1 {constraint = confidential data; } } } }
        goal =  (Accounting; G9.3.6; ) {
        task = (Measure [system time] used by [user]; T9.3.6.1; ) {}
        task = (Measure [data] sent by [user]; T9.3.6.2; ) {}
        task = (Measure [data] received by [user]; T9.3.6.3; ) {} }
        goal =  (Manage [account]; G9.4; ) {
        goal =  (Edit [account]; G9.4.1; ) {
        task = (Edit [user account]; T9.4.1.1; or) {
        task_ref = (Verify if [data identification] is unique; T5.4; )
        task_ref = (Verify if [enters] are valid; T5.5; )
        task_ref = (Verify if every [data] are filled up; T5.6; )}
        task = (Edit [administrator account]; T9.4.1.2; or) {
        task = (Add [e-mail]; T9.4.1.2.1; and) {}
        task_ref = (Verify if [data identification] is unique; T5.4; )
        task_ref = (Verify if [enters] are valid; T5.5; )
        task_ref = (Verify if every [data] are filled up; T5.6; )}
        task = (Add [username]; T9.4.1.3; and) {}
        task = (Add [password]; T9.4.1.4; and) {
        task_ref = (Symmetric cryptography; T9.2.4; )}
        task = (Add [full name]; T9.4.1.5; and) {}
        task = (Reset [password]; T9.4.1.6; and) {
        task_ref = (Symmetric cryptography; T9.2.4; )}
        task = (Change [password]; T9.4.1.7; and) {
        task_ref = (Symmetric cryptography; T9.2.4; )} }
        goal =  (There is always at least one [administrator account]; G9.4.2; ) {}
        task_ref = (Restrict [access] to [data]; T9.3.5.3.2; )
        task_ref = (Restrict [access] to [functions]; T9.3.5.3.1; )
        task_ref = (Authentication by Login; T9.3.3; )
        task_ref = (Logout Logout Logout Logout; T9.3.3.1; or)}
        task = (Logout; T9.3.3.1; or) {}
        correlation (make) {
            source = softgoal_ref = (Authentication; G9.3; )
            target = softgoal_ref = (Confidentiality; S9.1; )} }
        task = (Cryptography; T9.2; ) {
        task = (Assymmetric cryptography; T9.2.3; or) {}
        task = (Symmetric cryptography; T9.2.4; or) {}
        task = (Make [cryptography operations]; T9.2.5; ) {
        task = (Encrypt [data]; T9.2.1; ) {}
        task = (Decrypt [data]; T9.2.2; ) {} }
        correlation (help) {
            source = task_ref = (Cryptography; T9.2; )
            target = softgoal_ref = (Confidentiality; S9.1; )}
        crosscutting {source = Cryptography(T9.2)
            pointcut (Symetric; PC9.3.5.1): include(.*password; task; name)
            advice (around): PC9.3.5.1 {
                task_ref = (Symmetric cryptography; T9.2.4; )} } }
        correlation (help) {
            source = softgoal_ref = (Confidentiality; S9.1; )
            target = softgoal_ref = (Adoption of system to divulge events; S1; )} }

    goal_model (Portability; GM2) {
        softgoal = (Portability; S2.1; ) {
        softgoal = (Portability [plataform]; S2.1.1; make) {
        task = (Use [Microsoft Windows]; T2.1.2; make) {}
        task = (Use [Macintosh OS]; T2.1.3; make) {}
        task = (Use [PDA]; T2.1.4; make) {}
        task = (Use [Web]; T2.1.5; make) {}
        task = (Program with [Java TM (jdk 1.4)]; T2.1.6; help) {} } }
        correlation (help) {
            source = softgoal_ref = (Portability [plataform]; S2.1.1; )
            target = softgoal_ref = (Adoption of system to divulge events; S1; )}
```

```
        correlation (hurt) {
            source = softgoal_ref = (Portability [plataform]; S2.1.1; )
            target = softgoal_ref = (Reliability; S5.1; )}
        correlation (hurt) {
            source = softgoal_ref = (Portability [plataform]; S2.1.1; )
            target = softgoal_ref = (Performance [time]; S6.1.1; )}
        correlation (hurt) {
            source = softgoal_ref = (Portability [plataform]; S2.1.1; )
            target = softgoal_ref = (Security; S9.0; )} }

    goal_model (Performance; GM6) {
        softgoal = (Performance; S6.1; ) {
        softgoal = (Performance [time]; S6.1.1; make) {}
        softgoal = (Performance [resource]; S6.1.2; make) {}
        crosscutting {source = Performance [time](S6.1.1)
            pointcut (performance_smallerthanone; PC6.1): include(Synchronyze [event]; G1.9.2)
            pointcut (performance_instantaneous; PC6.2): include(.*; task; name)
            intertype declaration (attribute): PC6.1 {constraint = performance: smaller than one second; }
            intertype declaration (attribute): PC6.2 {constraint = performance: instantaneous; } } }
        correlation (help) {
            source = softgoal_ref = (Performance [time]; S6.1.1; )
            target = softgoal_ref = (Adoption of system to divulge events; S1; )} }

    goal_model (Reusability; GM7) {
        softgoal = (Reusability; S7.1; ) {
        softgoal = (Reusability [system]; S7.1.1; make) {} }
        correlation (help) {
            source = softgoal_ref = (Reusability [system]; S7.1.1; )
            target = softgoal_ref = (Adoption of system to divulge events; S1; )} }

    goal_model (Maintainability; GM11) {
        softgoal = (Maintainability; S11.1; ) {
        softgoal = (Reparability; S11.2; make) {}
        softgoal = (Evolvability; S11.3; make) {} }
        softgoal = (Understandability; S11.4; ) {
        softgoal = (Understandability [code]; S11.4.1; make) {}
        softgoal = (Understandability [documentation]; S11.4.2; make) {} }
        correlation (help) {
            source = softgoal_ref = (Understandability; S11.4; )
            target = softgoal_ref = (Reparability; S11.2; )}
        correlation (help) {
            source = softgoal_ref = (Understandability; S11.4; )
            target = softgoal_ref = (Evolvability; S11.3; )}
        correlation (help) {
            source = softgoal_ref = (Maintainability; S11.1; )
            target = softgoal_ref = (Reusability; S7.1; )}
        correlation (help) {
            source = softgoal_ref = (Maintainability; S11.1; )
            target = softgoal_ref = (Adoption of system to divulge events; S1; )} }

    goal_model (Scalability; GM12) {
        softgoal = (Scalability; S12.1; ) {
        softgoal = (Scalability [quantity of users]; S12.1.1; make) {} }
        correlation (help) {
            source = softgoal_ref = (Scalability [quantity of users]; S12.1.1; )
            target = softgoal_ref = (Adoption of system to divulge events; S1; )} }

    goal_model (Timeliness; GM14) {
        softgoal = (Timeliness; S14.1; ) {
        softgoal = (Timeliness [development time]; S14.1.1; make) {
        softgoal = (Timeliness [design]; S14.1.1.1; make) {}
        softgoal = (Timeliness [implementation]; S14.1.1.2; make) {}
        softgoal = (Timeliness [test]; S14.1.1.3; make) {}
        correlation (help) {
            source = softgoal_ref = (Timeliness [development time]; S14.1.1; )
            target = softgoal_ref = (Adoption of system to divulge events; S1; )}
        correlation (hurt) {
            source = softgoal_ref = (Timeliness [development time]; S14.1.1; )
            target = softgoal_ref = (Reliability; S5.1; )}
        correlation (hurt) {
            source = softgoal_ref = (Timeliness [development time]; S14.1.1; )
            target = softgoal_ref = (Security; S9.0; )}
        correlation (hurt) {
            source = softgoal_ref = (Timeliness [development time]; S14.1.1; )
            target = softgoal_ref = (Usability; S3.1; )}
        correlation (hurt) {
```

```
            source = softgoal_ref = (Timeliness [development time]; S14.1.1; )
            target = softgoal_ref = (Performance; S6.1; )}
        correlation (hurt) {
            source = softgoal_ref = (Timeliness [development time]; S14.1.1; )
            target = softgoal_ref = (Reusability; S7.1; )}
        correlation (hurt) {
            source = softgoal_ref = (Timeliness [development time]; S14.1.1; )
            target = softgoal_ref = (Maintainability; S11.1; )}
        correlation (hurt) {
            source = softgoal_ref = (Timeliness [development time]; S14.1.1; )
            target = softgoal_ref = (Timeliness; S14.1; )} } } }

goal_model (Profitable; GM4) {
    softgoal = (Profitable; S4.1; ) {
    softgoal = (Profitable [development cost]; S4.1.1; ) {}
    correlation (help) {
        source = softgoal_ref = (Profitable [development cost]; S4.1.1; )
        target = softgoal_ref = (Adoption of system to divulge events; S1; )}
    correlation (hurt) {
        source = softgoal_ref = (Profitable [development cost]; S4.1.1; )
        target = softgoal_ref = (Reliability; S5.1; )}
    correlation (hurt) {
        source = softgoal_ref = (Profitable [development cost]; S4.1.1; )
        target = softgoal_ref = (Security; S9.0; )}
    correlation (hurt) {
        source = softgoal_ref = (Profitable [development cost]; S4.1.1; )
        target = softgoal_ref = (Usability; S3.1; )}
    correlation (hurt) {
        source = softgoal_ref = (Profitable [development cost]; S4.1.1; )
        target = softgoal_ref = (Performance; S6.1; )}
    correlation (hurt) {
        source = softgoal_ref = (Profitable [development cost]; S4.1.1; )
        target = softgoal_ref = (Reusability; S7.1; )}
    correlation (hurt) {
        source = softgoal_ref = (Profitable [development cost]; S4.1.1; )
        target = softgoal_ref = (Maintainability; S11.1; )}
    correlation (hurt) {
        source = softgoal_ref = (Profitable [development cost]; S4.1.1; )
        target = softgoal_ref = (Timeliness; S14.1; )} } }
```

Anexo A
Documento de Requisitos do Sistema Unical

Requirements Document for UniCal

1 Introduction

Since the opening of the campus, the University of California, Irvine (UCI) has held many academic and non-academic events for its many students, faculty, staff, and visitors. Until now, UCI has informed people of these events using a haphazard collection of e-mails, web pages, flyers, and word of mouth. However, as the campus grows, it becomes increasingly more difficult to inform people of the many different UCI events. UCI now wishes to supplement this process by creating a university calendar system, called UniCal, to inform interested parties of campus events.

Because UCI is a research oriented institution, it does not want to burden its professors or students with production oriented software engineering projects. Consequently, it has hired Retro Software Inc. to develop this requirements specification for UniCal, as well as its subsequent design and implementation. The information in this requirements specification is based on interviews conducted with the relevant parties at UCI, by the requirements engineers at Retro Software Inc. All information in this document was determined to be accurate at the time of this writing.

This document contains the detailed requirements specification for UniCal that Retro Software Inc. is developing for UCI. The document should serve as the official basis for any further development of UniCal.

This document contains the following sections:

- Executive Summary
- Application Context
- Functional Requirements
- Environmental Requirements
- Software Qualities
- Other Requirements
- Time Schedule
- Potential Risks
- Future Changes
- Glossary
- References

2. Executive Summary

UniCal aims to provide a solution for scheduling campus events and publicizing their information to UCI faculty, staff, students, and visitors. The application allows customers to schedule, share, and view events via any computer running UniCal. UniCal facilitates communication by unifying and providing easy access to event information. Changes in events will be updated to users' schedules so that everybody is well informed. By maintaining consistent and reliable event information, UniCal enables effective event arrangement and collaboration among users. Retro Software anticipates that the overall effect of adopting UniCal will result in more accurate, efficient and convenient schedule coordinating among UCI personnel. UniCal provides the following key features:

- **Scheduling and sharing events** - A user can schedule events and group them into specific categories using UniCal. These events can be seen by other users of the system.
- **Viewing events** - A user can view all of the events in their calendar in either a weekly or monthly view.
- **Event reminders** - UniCal will provide reminders for a user by playing a sound and displaying a pop-up window with details of the event at a customizable time interval before each event.
- **Task list** - Users can also create a task list with deadlines and reminders to help them meet goals.

Some of the most important risks posed by the development of UniCal are the following:

- **Lack of adoption** - Because there are many other calendaring systems currently on the market, it is possible that students, faculty, and staff will not adopt UniCal into their normal routine. This would effectively render the system as no more useful than current approaches to informing people of events.

• **Usability** – To help encourage adoption of the system, UniCal must be extremely intuitive and easy to use.

• **Rapid development** – The schedule according to which UniCal will be developed is extremely aggressive to ensure that another company will not develop a similar product first and claim the market. Other qualities cannot be sacrificed as a result of this rapid development.

Although the contents of this document were thoroughly verified, it may still occur that some requirements are inconclusive or ambiguous. If it is so determined, it is requested that Retro Software Inc. be contacted via e-mail at shendric@uci.edu to resolve the issue.

3. Application Context

UniCal will be used in the institutional environment, UC, Irvine, where scheduling and coordinating events can be difficult because of the large numbers of people involved. The introduction of UniCal should reduce the effort required to coordinate schedules among multiple people by automating much of the effort of keeping everyone's schedule up to date. Adopting UniCal will result in substantial change for UCI. It will require that the application be installed on each person's computer, and that the application be used as the primary means of scheduling and coordinating events on campus.

Users of UniCal fall into two different roles: administrators, and users: Administrators of UniCal have all of the same functionality as a user, but in addition are responsible for managing the different accounts of the system. They may create, delete, and edit user accounts, as well as other administrator accounts. Should users forget their password, it is the responsibility of an administrator to reset their password so that they may once again gain access to the system. Should an administrator forget their password, they may have the password e-mailed to the e-mail address associated with their account.

Users of the UniCal system use the system to schedule and share events. Users may be faculty, staff, students, or even offices who wish to publicize office events. Users may create events, schedule reminders for events, specify recurrences for events, and group events together into categories. The categories that users create may be shared with other users of the system and included on their calendars. Users may also maintain a private task list of tasks that need to be completed. It is the hope of UCI that UniCal will eventually be used by other campuses. Such a change would require the application to be highly scalable and likely require instances of the UniCal system to interoperate since, due to political issues, it is unlikely that all campuses will share one instance.

4. Functional Requirements

4.1. ADMINISTRATOR FUNCTIONALITY

4.1.1. Administrators have the following functionality in addition to regular users (see 4.2) of the system

4.1.2. Login

4.1.2.1. The administrator should log in to UniCal using their username and password

4.1.2.2. The administrator password must be changed the first time the administrator logs in to the system

4.1.2.3. There should be one default administrator account with the username and password of 'admin' and 'admin' respectively. This account cannot be deleted

4.1.3. Adding/deleting/editing administrator accounts

4.1.3.1. An administrator should be able to add, delete, or edit other administrator accounts through the user interface

4.1.3.2. An administrator can enter the following information for each administrator. All information must be entered for an administrator account to be added to the system

4.1.3.2.1. **Username:** must be between 5 and 20 characters and should consist only of upper case, lower case, and numeric characters

4.1.3.2.2. **Password:** must be between 5 and 20 characters and should consist only of upper case, lower case, numeric, and punctuation characters

4.1.3.2.3. **Full Name:** must be less than 100 characters

4.1.3.2.4. **E-mail:** must conform to section 3.4 of RFC 2822 (http://www.ietf.org/rfc/rfc2822.txt?number=2822).

4.1.3.3. The administrator should be able to edit and save their password, full name, and e-mail address

4.1.3.4. An administrator should be able to reset the password of administrator accounts that they created or that were directly or indirectly created by accounts that they created

4.1.3.5. An administrator should be able to delete only administrator accounts that they created or that were directly or indirectly created by accounts that they created

4.1.4. Adding/deleting/editing user accounts

4.1.4.1. An administrator should be able to add, delete, or edit all user accounts through the user interface

4.1.4.2. An administrator can enter the following information for each user. All information must be entered for a user account to be added to the system

4.1.4.2.1. **Username:** see 4.1.3.2.1

4.1.4.2.2. **Password:** see 4.1.3.2.2

4.1.4.2.3. **Full Name:** must be less than 100 characters

4.1.4.3. All usernames in the system must be unique at all times
4.1.4.4. An administrator should be able to reset a user's password when requested by a user. However, an administrator should not be able to see any users current password
4.1.5. User Interface
4.1.5.1. UCI has few restrictions on the user interface for administrator functionality except that it should be functional and user friendly. The layout of the user interface for administrator functionality is left to the designers

4.2. USER FUNCTIONALITY

4.2.1. Login
4.2.1.1. A user logs in to UniCal using a username and password (see 4.1.4.2.1, 4.1.4.2.2)
4.2.1.2. The user password must be changed the first time a user logs in to the system
4.2.2. Adding/deleting/editing categories
4.2.2.1. A user should be able to add, delete, or edit categories through the user interface
4.2.2.2. A user can enter the following information for each category. All information must be entered for a category to be added to the users account
4.2.2.2.1. **Name:** a single line description used to identify the category (50 characters max)
4.2.2.2.2. **Color:** the color in which all events in the category are displayed
4.2.2.2.3. **Description:** a multi-line description of the category
4.2.2.3. All category names for a given user must be unique at all times.
4.2.2.3.1. If a change in the system will cause a category name to be identical to another category name, then the user should be notified of this and given an opportunity to change the name before continuing with the function
4.2.2.4. Once a category is created, events (see 4.2.5) may be added to the category
4.2.3. Importing categories
4.2.3.1. A user may include another users category (and the events therein) in their calendar by selecting it from a list of all user categories stored on UniCal
4.2.3.2. A user should be able to use a local name and color for an imported category, and set a local reminder (see 4.2.5.2.8) for each event (see 4.2.5) in an imported category
4.2.4. Updating
4.2.4.1. A user should be allowed to update their calendar so that their categories are available for others to import (see 4.2.3), and a user's imported categories are updated with changes made to the events in the imported category
4.2.4.2. Users may edit their calendar while not connected to the internet
4.2.5. Adding/deleting/editing events
4.2.5.1. A user should be able to add, delete, or edit events through the user interface
4.2.5.2. A user can enter the following information for each event. All information must entered for the event to be added to a category
4.2.5.2.1. **Name:** a single line description used to identify the event (50 characters max)
4.2.5.2.2. **Start time:** the starting date and time of the event formatted as (hours)/(minutes) (am or pm) (month)/(day)/(year) using two digit numbers except for hour and month, and four digit number for the year
4.2.5.2.3. **End time:** the starting time of the event formatted as (hours)/(minutes) (am or pm) (month)/(day)/(year) using two digit numbers except for hour and month, and four digit number for the year
4.2.5.2.4. **Location:** the location of the event (using previously entered values or entering a new value)
4.2.5.2.5. **Description:** a description of the event
4.2.5.2.6. **Recurrence:** the recurrence for the event set with respect to the start time of the event (see 4.2.7)
4.2.5.2.7. **Category:** the category that contains the event (see 4.2.2)
4.2.5.2.8. **Reminder:** the reminder for the event set with respect to the start time of the event (see 4.2.8)
4.2.5.3. An event must belong to exactly one category
4.2.5.4. The end time of the event must occur after the start time of the event
4.2.5.5. The start and end time may have a minimum granularity of 1 minute
4.2.5.6. Event names do not need to be unique
4.2.6. Adding/deleting/editing tasks
4.2.6.1. A user should be able to add, delete, and edit tasks through the user interface
4.2.6.2. A user can enter the following information for each task. All information must entered for the task to be added
4.2.6.2.1. **Name:** a single line description used to identify the task (50 characters max)
4.2.6.2.2. **Description:** a description of the task
4.2.6.2.3. **Due:** the date and time the task is due formatted as (hours)/(minutes) (am or pm) (month)/(day)/(year) using two digit numbers except for hour and month, and four digit number for the year
4.2.6.2.4. **Reminder:** the reminder for the task set with respect to when the task is due (see 4.2.8)
4.2.7. Setting a recurrence
4.2.7.1. When setting a recurrence, a user should be able to select any of the following recurrence settings provided by the user interface

4.2.7.1.1. **No recurrence:** the event only occurs once at the specified time
4.2.7.1.2. **Weekly recurrence:** the user may specify which days of the week for which the event reoccurs each week
4.2.7.1.3. **Monthly recurrence:** the user may specify which day of the month for which the event reoccurs each month
4.2.7.1.4. **Yearly recurrence:** the user may specify the day of the year for which the event reoccurs each year
4.2.7.2. The following functionality is required for events with recurrences
4.2.7.2.1. The recurrences of an event should start on or after the start date of an event
4.2.7.2.2. The user must specify an end date after which the event will not occur
4.2.7.2.3. Where there are conflicts between an event's information and the event's recurrence, the recurrence settings should take precedence
4.2.8. Setting a reminder
4.2.8.1. When setting a reminder, a user may schedule the reminder at any of the following time intervals before the event: 0, 5, 10, 15, or 30 minutes, 1 to 12, or 18 hours, 1 to 6 days, or 1 to 2 weeks
4.2.8.2. If the reminder is set for something that has a recurrence, then the reminder will occur for each instance of the recurrence
4.2.8.3. The reminder will be triggered when the system's time and date are greater than the reminders scheduled time, and the reminder has not been dismissed
4.2.8.4. When a reminder is triggered, a popup window will be displayed with all event details and a sound will be played
4.2.8.5. Dismissing a reminder
4.2.8.5.1. When a reminder is triggered, a user may dismiss the reminder, in which case the reminder window closes
4.2.8.6. Snoozing
4.2.8.6.1. When a reminder is triggered, a user may press snooze, in which case the reminder window will close and the reminder will be reschedule to 5 minutes after the time of pressing snooze
4.2.9. User interface
4.2.9.1. Appropriate menus and toolbars should be available to access all functionality using the mouse or keyboard
4.2.9.2. There should be the following three parts to the user interface arranged from left to right: the category list (see 4.2.10), the calendar view (see 4.2.11), and the task list (see 4.2.14)
4.2.9.3. The category list should initially utilize 10% of the screen
4.2.9.4. The calendar view should initially utilize 80% of the screen
4.2.9.5. The task list should initially utilize 10% of the screen
4.2.9.6. A user should be able to resize the portion of the screen that each section utilizes
4.2.10. Category list
4.2.10.1. This displays a list of imported (see 4.2.3) and created (see 4.2.2) category names sorted alphabetically
4.2.10.2. The list should display each category name to the right of a checkbox (which indicates whether the events of the category are visible in the calendar view (see 4.2.11)). The user should be able to check or uncheck this box to include or exclude the events from the calendar view respectively.
4.2.10.3. The area behind each category name should be colored according to
the category color (see 4.2.2.2.2)
4.2.10.4. The user should be able to edit category properties via a context
menu for each category in the list
4.2.11. Calendar view
4.2.11.1. This view should consist of a tabbed pane which includes a tab for a monthly calendar (see 4.2.12) and a tab for a weekly calendar (see 4.2.13). The tabs are located at the top left of the view.
4.2.11.2. All calendar views should start on the same day of the week, either Sunday or Monday. The user should be able to set this in system preferences.
4.2.11.3. A user should be able to enter a date, and have any view focus in on that date.
4.2.12. Monthly calendar view
4.2.12.1. This view should display 7 days by 5 weeks using equally sized boxes to represent the scheduled events for each day.
4.2.12.2. The day of the month should be displayed in the top right corner of each box
4.2.12.3. The box representing the first of any month should contain the name of the month in the top left corner and the four digit year between the month name and the day of month
4.2.12.4. The name of each event for each day should be listed in that days box in order of the event starting time
4.2.12.5. Events that span multiple days should be included in each day that they occur
4.2.12.6. The background of each event should be colored according to the color of its respective category (see 4.2.2.2.2)
4.2.12.7. Scrollbars for a given box should be included in the box if all information cannot fit within the box

4.2.12.8. Double clicking on the number of any day should change the current view to the weekly calendar (see 4.2.13). The weekly calendar should include the day that was double clicked

4.2.12.9. The current day (according to the system clock) should be highlighted with a red border at the edge of the box representing that day

4.2.12.10. A day may be selected using the mouse or the arrow keys of the keyboard, the current day is selected by default

4.2.12.11. Page-up will select the date that is one month prior to the currently selected day, Page-down will work similarly

4.2.12.12. The selected day is highlighted with a black border (within the current day highlight (see 4.2.12.9) if necessary)

4.2.12.13. Adding an event to the calendar when in this view should use the currently selected date as the default starting and ending date for the event.

4.2.13. Weekly calendar view

4.2.13.1. This view should display a grid where each day of the week is represented by 7 columns and the time of the day is indicated along the left hand side

4.2.13.2. The date for each day should be displayed at the top of each column and formatted as (Week Day), (Month)/(Day)/(Year) using a the full week day name for the week day, a number for the month, a two digit number for the day, and a four digit number for the year

4.2.13.3. Time increments for the left hand side should be selected from 5, 10, 15, 30 minutes, and 1 hour, the default should be 30 minute increments

4.2.13.4. Each event for a given day should be displayed in the following way

4.2.13.4.1. A 25% translucent rectangle should indicate the starting and ending time and day of each event by covering the area indicated by the start date and time and end date and time of the event. The color of the rectangle should be the color indicated by the events category

4.2.13.4.2. The rectangle should span multiple days for multi-day events

4.2.13.4.3. The name of the event should appear at the top left corner of the translucent rectangle

4.2.13.4.4. Conflicting events should simply overlap

4.2.13.5. The current day (according to the system clock) should be highlighted with a red border at the edge of column representing that day

4.2.13.6. An event or time range may be selected using the mouse or keyboard, the selected event or time range is highlighted with a black border

4.2.13.7. Page-up will display the previous week, Page-down will display the next week

4.2.13.8. Adding an event to the calendar when in this view should use the currently selected date and time range as the default starting and ending date and time for the event

4.2.14. Task list

4.2.14.1. The task list should display each task in descending order of due date

4.2.14.2. A user should be able to mark tasks as completed

4.2.14.3. Tasks that are marked as completed should disappear in 24 hours

4.3. MISCELLANEOUS FUNCTIONALITY

4.3.1. Additional user interface requirements

4.3.1.1. All functionality should be accessible through menu items

4.3.1.2. All functionality should be accessible with or without the use of a mouse

4.3.2. Help

4.3.2.1. UniCal should have help documentation provided through the UI that describes each function in this document including an example of how to perform each function

4.3.2.2. UniCal help documentation should also include a short tutorial on how UniCal is used

4.3.3. Error Handling

4.3.3.1. All error handling is left to the designers of the UniCal system

4.3.3.2. All error handling should allow a user to correct the error with minimal effort, and when possible, allowing a user to continue their previous function rather than starting over

5. Environmental Requirements

Since most people on the UCI campus use either Microsoft Windows or the Macintosh OS, UCI has decided that UniCal should be able to run on both platforms. UCI has also hired a consultant regarding programming languages, and, based on her report recommends that UniCal be implemented in Java™, to ensure portability across platforms as well as easy maintainability. Use of the JDK 1.4 is expected.

6. Other Requirements

Users should interface with UniCal through a stand alone application on their machines. The cost of development for the UniCal system must not exceed $399,999.98. UCI has consulted with financial analysts to determine that this is the maximum amount the system can cost without becoming unprofitable. Retro Software Inc. should carefully document all decisions made, especially decisions pertaining to changes in this

document (and subsequent documents; always per agreement with UCI). Retro Software Inc. will deliver detailed user manuals for the UniCal system.

7. Software Qualities

• **User-friendliness** – Because many of UniCal's users are expected to be people with minimal to no time to learn a complicated new system, it is imperative that the system be as user-friendly as possible.

• **Correctness** – Because UCI wants the system to be widely used, it is important that UniCal perform all of its requirements correctly and does not result in the proliferation of inaccurate or incomplete information.

• **Reliability** – Reliability of UniCal is not essential, but nonetheless important. The accepted rate of failure is one crash per month. Any failure rate above this will create an unfavorable impression of UCI towards Retro Software.

• **Performance** – Performance is crucial to UniCal – if the system is too slow, customers may revert to sending e-mails and using word of mouth. Synchronizing events with UniCal should take no more than one second for every fifty events. All other operations must be performed nearly instantaneously.

• **Reusability** – Reusability is important because it is hoped that UniCal will eventually be adopted at all other UC campuses.

• **Extensibility** – Several future enhancements for UniCal have already been proposed, and there will undoubtedly be more forthcoming. Therefore, it is critical that UniCal can be extended with relative ease.

• **Evolvability** – UniCal is expected to undergo significant changes as UCI determines which features are most important to its population. Evolvability is central to this goal.

• **Robustness** – UniCal must not crash if a user enters incorrect or invalid data, or performs some otherwise unexpected action. A reasonable response that allows the application to resume normal operation must be given.

• **Verifiability** – Although it is preferable that the system undergo formal, thorough verification and testing before deployment, this is not feasible with the rigorous time schedule desired by UCI. However, extensive testing of the accuracy of the system's functionality should be performed before release.

• **Maintainability** – UCI anticipates that UniCal will be used over long periods of time, eventually by large numbers of users. Due to this fact, the likelihood of several future enhancements, and the high probability of an equally tight time schedule for their development, it is imperative that UniCal be easily maintainable.

• **Reparability** – Because UniCal is being developed under such a tight time schedule, it is likely that testing will not be done as thoroughly as desired, and some faults will make it into the product. Therefore, the system must be easily repairable in order to fix these defects in a timely manner so as not to disrupt the business of UCI.

• **Safety** – Since UniCal is not a safety-critical application, there are no safety concerns.

• **Portability** – Even though UniCal will be implemented entirely in Java™, a highly portable language, portability will still need to be considered in the design and implementation of the system. It is UCI's desire that no features of UniCal will violate commonly accepted standards for each platform.

• **Understandability** – To support evolvability, reparability, and maintainability, it is imperative that all aspects of UniCal be easily understood, even to future developers who are not currently involved in the creation of the system.

• **Interoperability** – For this current set of requirements, the interoperability of UniCal with any other application is not needed. However, it is of prime importance that UniCal interoperate with other calendaring systems in future versions.

• **Productivity** – Because UCI would like to release the UniCal application as quickly as possible, it is desirable that the productivity of all involved in the development of UniCal be supported and facilitated with quality processes and tools as much as possible. However, limited funding is available for this requirement, so it must be foregone.

• **Size** – Hard drive space and memory are abundant nowadays – size is not an issue.

• **Scalability** – It is important that UniCal be able to scale to support the entire population of UCI, and perhaps, the UC system. Therefore, scalability is an important issue to consider in the development of UniCal.

• **Timeliness** – It is imperative that all artifacts of the UniCal system be delivered on time.

• **Visibility** – Due to the rigorous time schedule, extra time and effort to make the project status visible should be forfeited.

8. Time Schedule

The development schedule for UniCal is as follows:

• Design must be completed by **February 26th, 2004 at 2:00 PM**.

• Implementation and module testing must be completed by **March 9th, 2004 at 2:00 PM**.

• System testing must be completed by **March 18th, 2004 at 2:00 PM**, however this date may be changed.

9. Potential Risks

• **Difficult to use** – UniCal has a lot of features and addresses every kind of user. Furthermore, a significant portion of its expected user base is not computer literate. Hence, it is possible that some users might find it too difficult to use UniCal.

• **Limited schedule** – The schedule according to which UniCal is being developed is extremely aggressive and may result in a product that does not adequately satisfy the needs of UCI.

• **Lack of adequate support processes and tools** – A project of this size would ideally be supported by a number of quality software engineering processes and tools. Unfortunately, UCI lacks the funds necessary to obtain these tools.

• **Privacy issues** – As all categories will be viewable by every user, some users may feel uncomfortable letting other people see their (private) events. This could cause users to refrain from using UniCal.

10. Future Changes

• **Programmatic API for administrator functionality** – In future versions, UCI may want to be able to programmatically access administrator functionality so that the creation and removal of user accounts can be done automatically by computers processing incoming and outgoing students, staff, and faculty.

• **Web & PDA interface** – In future versions, UCI may want to create Web interfaces and PDA interfaces to supplement UniCal.

• **Permissions** – In future versions, UCI may want to introduce permissions to allow users of UniCal to keep certain events private.

• **Interface with other calendaring applications** – In future versions, UCI may want UniCal to interface with other calendaring applications. Minimally this will involve sharing event information such as through importing and exporting .vcal files (see http://www.imc.org/pdi/vcal-10.txt). However, support of more automated techniques such as scheduling meetings may also be desired.

• **Scheduling meetings** – In future versions, UCI may want to have UniCal facilitate the process of scheduling meetings, reserving rooms, and request RSVP information.

11. Glossary

• **Administrator** – One of the two rules in the UniCal System; administrators have the ability to add, delete, and modify user accounts.

• **Category** – A grouping of events into a common theme

• **Event** – A social gathering or activity that the system will help to coordinate.

• **Java™** – A highly portable, general purpose programming language developed by Sun Microsystems in the early- to mid-1990's.

• **UCI** – University of California, Irvine. The client company contracting the development of the UniCal system.

• **User** – One of the two roles in the UniCal system; users are ordinary people who wish to schedule, share, and view different university events.

12. References

• http://www.uci.edu

• http://www.java.sun.com

• http://www.dictionary.com

• http://www.imc.org/pdi/vcal-10.txt

• http://www.ietf.org/rfc/rfc2822.txt?number=2822