



MINISTÉRIO DA CIÊNCIA E TECNOLOGIA
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS



SYSML PARA ENGENHARIA SIMULTÂNEA DE SISTEMAS ESPACIAIS

**RELATÓRIO FINAL DE PROJETO DE INICIAÇÃO CIENTÍFICA
(PIBIC/CNPq/INPE)**

Maiara Guimarães Flausino (UFSCar, Bolsista PIBIC/CNPq)

E-mail: maiara.flausino@lit.inpe.br

Geilson Loureiro (LIT/INPE, Orientador)

E-mail: geilson@lit.inpe.br

São José dos Campos – SP

Julho de 2012

*Põe quanto és
No mínimo que fazes*

Fernando Pessoa

SUMÁRIO

Pág.

1.	INTRODUÇÃO.....	1
2.	OBJETIVOS.....	3
3.	MOTIVAÇÃO.....	4
4.	METODOLOGIA.....	5
5.	CRONOGRAMA DE ATIVIDADES.....	6
6.	ENGENHARIA SIMULTÂNEA DE SISTEMAS.....	8
6.1.	Sistemas.....	9
6.2.	Engenharia de Sistemas.....	10
6.3.	Engenharia Simultânea.....	12
6.4.	Método do Professor Dr. Geilson Loureiro.....	13
6.4.1.	Principais etapas do método.....	16
6.4.1.1.	Gestão de Projetos.....	17
6.4.1.2.	Análise de requisitos e Measure of Effectiveness (MoEs).....	20
6.4.1.3.	Contexto funcional.....	22
6.4.1.4.	Ciclo de vida.....	23
6.4.1.5.	Análise funcional.....	24
6.4.1.6.	Análise de perigo.....	26
6.4.1.7.	Contexto físico.....	29
6.4.1.8.	Arquitetura de sistemas.....	30
7.	MODELAGEM GRÁFICA.....	32
7.1.	As quatro bases da modelagem.....	35
8.	ANÁLISE ESTRUTURADA.....	36
9.	UNIFIED MODELING LANGUAGE (UML).....	38
10.	SYSTEMS MODELING LANGUAGE (SYSML).....	40
10.1.	Diagramas da linguagem SysML.....	44
10.1.1.	Diagrama de atividade.....	45
10.1.2.	Diagrama de sequencia.....	46
10.1.3.	Diagrama de transcrição de estados.....	47
10.1.4.	Diagrama de caso de uso.....	48
10.1.5.	Diagrama de definição de blocos.....	49

10.1.6. Diagrama de blocos internos.....	50
10.1.7. Diagrama de pacotes.....	51
10.1.8. Diagrama de parâmetros.....	52
10.1.9. Diagrama de requisitos.....	53
11. FERRAMENTAS.....	54
11.1. Cradle.....	55
11.2. Enterprise Architect.....	56
11.3. IBM Rational Doors.....	57
11.4. IBM Rational Rhapsody.....	58
11.5. Magic Draw – SysML Plug-in.....	59
11.6. Modelio System Architecture.....	60
11.7. Papyrus UML for SysML.....	61
11.8. Top Cased – SysML.....	62
12. RESULTADO.....	63
13. TRABALHO FUTURO.....	76
14. CONSIDERAÇÕES FINAIS.....	77
15. REFERÊNCIAS BIBLIOGRÁFICAS.....	81

LISTA DE FIGURAS

	<u>Pág.</u>
1. Método de engenharia simultânea de sistemas.....	13
2. Método evoluído de engenharia simultânea de sistemas.....	14
3. Principais etapas do método.....	16
4. Gestão de projetos.....	17
5. Definição da missão.....	18
6. SEMP.....	19
7. Análise de stakeholders.....	21
8. Ciclo de vida.....	22
9. Processos do ciclo de vida e seus cenários.....	23
10. Análise funcional do sistema.....	25
11. Análise de perigo.....	26
12. Contexto físico.....	29
13. AIT.....	30
14. Especificação de componentes.....	31
15. Modelagem de um sistema.....	33
16. Fluxograma.....	34
17. As quatro bases da modelagem.....	35
18. Relação entre as linguagens UML e SysML.....	40
19. Diagramas da linguagem SysML.....	44
20. Tela do software Enterprise Architect.....	56
21. Tela do software DOORS.....	57
22. Tela do software Rhapsody.....	58
23. Tela do software MagicDraw.....	59
24. Tela do software Modelio System Architecture.....	60
25. Tela do software Papyrus.....	61
26. Tela do software TopCased.....	62
27. Sistema CANSAT.....	64
28. Subsistemas da estação terrena.....	65
29. Estação terrena.....	66
30. Detalhamento dos subsistemas da estação terrena.....	67
31. Identificação de stakeholders.....	69

32. Ciclo de vida do sistema.....	70
33. Fluxo de material, energia e informação.....	71
34. Cenário AIT.....	72
35. Cenário Manutenção.....	73
36. Cenário Descarte.....	74
37. Cenário Produção.....	75

LISTA DE TABELAS

	<u>Pág.</u>
1. Cronograma.....	6
2. Lista de Eventos.....	24
3. Falhas e análise crítica FMECA.....	28
4. SysML Partners.....	41
5. Comparação entre as linguagens UML e SysML.....	42

AGRADECIMENTOS

Agradeço ao CNPq pela bolsa de Iniciação Científica, a qual me permitiu dar o primeiro passo como futura pesquisadora.

Ao INPE pelo Programa Institucional oferecido, o qual me apoiou no desenvolvimento deste trabalho.

Ao LIT pela estrutura oferecida para o desenvolvimento deste trabalho.

Ao Professor Dr. Geilson Loureiro pela oportunidade de desenvolvimento deste trabalho, com o qual aprendi muito sobre a dinâmica do mundo acadêmico.

A minha família, pela dedicação e amor constante.

A Deus!

RESUMO

Este trabalho tem por objetivo implementar o método de engenharia simultânea de sistemas usando a linguagem de modelagem de sistemas, Systems Modeling Language (SysML). O método de engenharia simultânea de sistemas foi inicialmente desenvolvido pelo Professor Dr. Geilson Loureiro e evoluído desde 1999. O método consiste no desenvolvimento simultâneo de produto e das organizações que implementam o processo do ciclo de vida do produto ao longo dos processos de engenharia de sistemas, quais sejam: análise de stakeholders, análise de requisitos, análise funcional e arquitetura de sistemas. O método deve ser aplicado em todos os níveis de abstração de um produto complexo, como um satélite. O método foi originalmente desenvolvido usando técnicas de modelagem baseadas em Análise Estruturada e Unified Modeling Language (UML). A partir de 2007, ferramentas computacionais que implementam a linguagem SysML passaram a estar disponíveis. A SysML é uma linguagem de modelagem descritiva de sistemas, construída a partir de estereótipos da linguagem SysML. Um dos pilares da engenharia simultânea de sistemas é a modelagem, especialmente a modelagem gráfica, que permite ao profissional ter uma visão do todo, bem como identificar cada relacionamento existente num dado sistema. Este trabalho aplicou o método, usando a linguagem SysML, no desenvolvimento de um picossatélite chamado CANSAT. Como conclusão tem-se que a linguagem de SysML ainda precisa de adaptações. No entanto, ela já possui as construções suficientes para uma implementação do método.

1. INTRODUÇÃO

Este trabalho, desenvolvido no Laboratório de Integração e Testes (LIT) do Instituto Nacional de Pesquisas Espaciais (INPE), refere-se à compreensão do método de engenharia simultânea de sistemas, desenvolvido pelo Professor Dr. Geilson Loureiro e evoluído desde 1999, e ao conhecimento da linguagem de modelagem descritiva de sistemas Systems Modeling Language (SysML).

Neste trabalho realizou-se pesquisas sobre a linguagem SysML, a qual pode ser usada para modelar produtos e organizações, os quais fazem parte do método de engenharia simultânea de sistemas do Professor Dr. Geilson Loureiro.

O método consiste no desenvolvimento simultâneo de produto e de organizações que implementam as fases do ciclo de vida do produto ao longo dos processos de engenharia de sistemas, quais sejam: análise de stakeholders, análise de requisitos, análise funcional e arquitetura de sistemas. O método deve ser aplicado em todos os níveis de abstração de um produto complexo, como um satélite.

A engenharia de sistemas requer uma visão do todo, de todas as fases do ciclo de vida do produto. Evidência disto é a evolução dos conceitos de engenharia de sistemas adotados pelas indústrias automobilística e aeroespacial no sentido de obter uma solução balanceada que considera variáveis, como o tempo de desenvolvimento, custo dos processos do ciclo de vida, gerenciamento de risco, desempenho do produto e, ainda, atende aos requisitos. (LOUREIRO, 1999)

A evolução citada anteriormente consiste em analisar os stakeholders, os requisitos, o conceito funcional e a arquitetura de implementação, simultaneamente, para o produto, seus processos de ciclo de vida e suas organizações atuando em todas as camadas da estrutura de divisão de sistemas. (LOUREIRO, 2010)

Um dos pilares da engenharia simultânea de sistemas é a modelagem, especialmente a modelagem gráfica, a qual permite ao profissional ter uma visão do todo, bem como identificar cada relacionamento existente num dado sistema. A modelagem de um sistema pode ajudar na rastreabilidade dos requisitos, os quais futuramente serão validados com os stakeholders.

Para o desenvolvimento desta pesquisa, foram utilizadas três bibliografias específicas sobre o tema em estudo, como também a análise de ferramentas computacionais para diagramação dos modelos em SysML, as quais a partir de 2007, passaram a estar disponíveis no mercado, quais sejam Magic Draw - SysML Plug-in, Enterprise Architecture (EA), Modelio System Architecture, IBM Rational Rhapsody, Papyrus UML for SysML e TopCased - SysML, as duas últimas ferramentas são open source.

O software utilizado para modelar o template foi o IBM Rational Rhapsody, por ser uma ferramenta com diversas funções, que ajudam na modelagem e análise de um sistema, entre elas, rastreabilidade de requisitos, interface gráfica amigável, integração com o IBM Rational Doors, geração de códigos e relatórios automaticamente e também por estar disponível no Laboratório onde a estudante realiza a Iniciação Científica. Adicionalmente, foram realizadas discussões técnicas com o orientador sobre o assunto.

O produto espacial escolhido para ser modelado foi o Sistema CANSAT, o qual foi utilizado como exemplo em situação de sala de aula. Este sistema é composto por uma estação terrena, um foguete e um picossatélite chamado CANSAT, artesanalmente construídos para a competição 7th Intercollegiate Rocket Engineering Competition (7th IREC).

Com a duração de um ano de atividades, esta pesquisa teve como resultado a primeira versão do template da modelagem do Sistema CANSAT, a qual abrange a modelagem do segmento da estação terrena do Sistema CANSAT. Muitas adaptações foram e serão necessárias para a realização deste template, pois o método utilizado é baseado em Análise Estruturada, ou seja, os modelos refletem diretamente a linguagem estruturada, separando os registros (dados) das funcionalidades (procedimentos e funções).

2. OBJETIVOS

Diante do exposto, este trabalho, de caráter de pesquisa de iniciação científica, tem como objetivo principal desenvolver um template para aplicar o método de engenharia simultânea de sistemas, desenvolvido pelo Professor Dr. Geilson Loureiro, usando a Systems Modeling Language (SysML), através da realização de modelos experimentais utilizando o software IBM Rational Rhapsody, disponível no Laboratório de Engenharia de Sistemas (LSIS) do Laboratório de Integração de Testes (LIT).

Este trabalho tem também como objetivo consolidar os conhecimentos adquiridos durante um ano de pesquisa e estudo, tanto da linguagem de modelagem gráfica SysML como da Engenharia de Sistemas sob orientação do Professor Dr. Geilson Loureiro.

Para atingir o objetivo principal proposto, foram estabelecidos os seguintes objetivos específicos.

- ✓ Realizar leitura de referências bibliográficas sobre o tema em análise.
- ✓ Realizar leituras de notas de aulas da disciplina Engenharia de Sistemas e Modelagem de Sistemas do Professor Dr. Geilson Loureiro.
- ✓ Pesquisar e analisar documentos sobre a linguagem SysML e Engenharia de Sistemas.
- ✓ Aprofundar o conhecimento do sistema CANSAT, para que seja possível a modelagem do produto espacial.
- ✓ Unificar as informações colhidas, analisar o sistema e preparar o texto final do trabalho propondo a primeira versão do template do sistema CANSAT.

3. MOTIVAÇÃO

A motivação para o desenvolvimento deste trabalho foi o aumento da necessidade do uso da engenharia de sistemas na construção de produtos de vários setores da sociedade, essa abordagem tem sido cada vez mais requisitada devido à evolução das tecnologias e ao crescimento da complexidade dos próprios produtos, como por exemplo, aumento de riscos e tempo gasto nas etapas de modelagem do projeto.

O método de engenharia simultânea de sistemas é aquele utilizado na pós-graduação do Instituto Nacional de Pesquisas Espaciais (INPE) para ensinar engenharia de sistemas espaciais. Esse método modela simultaneamente o produto e a organização ao longo dos processos de análise de stakeholders, de análise de requisitos, de análise funcional e de análise de implementação. A organização se refere às organizações que implementam os processos do ciclo de vida do produto. (LOUREIRO, 2010)

Atualmente, os projetos são mais complexos, o que dificulta a visão sistêmica do produto. Por isso foi necessário o estudo da Systems Modeling Language (SysML), a qual tem se tornado padrão de modelagem de sistemas no mundo. Esta linguagem ainda está em fase de desenvolvimento, a qual se dá em conjunto com o Object Management Group (OMG) e o International Council on Systems Engineering (INCOSE).

Diante de tal cenário mundial, o Laboratório de Integração e Testes (LIT) tem um projeto aprovado, já em andamento, junto ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) e a Agência Espacial Brasileira (AEB) para a implantação do Laboratório de Engenharia de Sistemas (LSIS). Esse laboratório deverá usar diversas ferramentas computacionais modernas de engenharia simultânea de sistemas tais como Cradle (ferramenta de engenharia de sistemas), IBM Rational Rhapsody (ferramenta de modelagem na linguagem SysML), IBM Rational DOORS (ferramenta de engenharia e gestão de requisitos) e MATLAB/Simulink (ferramenta de modelagem matemática).

4. METODOLOGIA

A metodologia utilizada neste estudo envolveu diferentes momentos. No início, com base na análise crítica da literatura científica pesquisada foi feita a extração de conceitos do referencial teórico sobre a Systems Modeling Language (SysML), e a Engenharia de Sistemas. Estes ofereceram os fundamentos a partir do qual o sistema CANSAT foi modelado.

A realização deste trabalho ocorreu de agosto de 2011 a julho de 2012. Durante esse um ano de pesquisas e estudo contínuo, através da revisão bibliográfica, leitura de artigos e revistas científicos da área, sobre a linguagem SysML foi investigado as características da mesma, os diagramas dessa linguagem, as utilidades e as perspectivas desse novo padrão de modelagem para a descrição de sistemas complexos.

Para colocar em prática o aprendizado da linguagem SysML foi necessário o treinamento da ferramenta IBM Rational Rhapsody, a qual deu-se através de manuais de usuários, tutoriais virtuais e também da participação em fóruns online.

Durante o processo foi realizada uma vasta pesquisa sobre as ferramentas que modelam na linguagem SysML e o uso de todas as ferramentas deu-se com a licença TRIAL para testes. E como resultado apresentado nesse trabalho, o software utilizado foi o IBM Rational Rhapsody versão FULL, disponível no Laboratório de Engenharia de Sistemas (LSIS) do Laboratório de Integração e Testes (LIT).

Foi preciso também realizar a identificação das interfaces da ferramenta IBM Rational Rhapsody com outras ferramentas, que dão suporte ao processo ao processo de Engenharia de Sistemas, como, IBM Rational DOORS, Cradle e MATLAB/Simulink.

O desenvolvimento da primeira versão do template baseado no método de engenharia simultânea de sistemas, realizado pelo Professor Dr. Geilson Loureiro, usando a linguagem SysML foi o resultado deste aprendizado da Engenharia de Sistemas e a linguagem SysML. Para a aplicação do template foi utilizado um produto da área espacial, o sistema CANSAT.

5. CRONOGRAMA DE ATIVIDADES

	2011					2012							
Atividades	Ago.	Set.	Out.	Nov.	Dez.	Jan.	Fev.	Mar.	Abril	Mai	Jun.	Jul.	Ago.
Método	■	■											
SysML			■										
Rhapsody				■	■								
Interfaces						■	■						
Template								■	■	■			
Exemplo											■	■	
Relatório												■	
Apresentação													■

Tabela 1 - Cronograma.

Atividade 1: MÉTODO: Obtenção e leitura da bibliografia específica e do material de treinamento relativo à engenharia simultânea de sistemas, disponível no site do curso CSE201-4 da pós-graduação do INPE.

Atividade 2: SYSML: Leitura das três referências básicas sobre a linguagem SysML.

- HOLT, J.; PERRY, S. **SysML for systems engineering**. IET. Stevenage. 2008.
- WEILKIENS, T. **Systems engineering with SysML/UML: modeling, analysis, design**. The MK/OMG Press. Elsevier. Amsterdam, 2006.
- FRIEDENTHAL, S.; MOORE, A. STEINER, R. **A practical guide to SysML: the systems modeling language**. The MK/OMG Press. Elsevier. Amsterdam, 2009.

Atividade 3: RHAPSODY: Treinamento da ferramenta IBM Rational Rhapsody.

Atividade 4: INTERFACES: Identificação de interfaces com as ferramentas IBM Rational DOORS, Cradle e MATLAB/Simulink.

Atividade 5: TEMPLATE: Desenvolvimento do template de engenharia simultânea de sistemas com a linguagem SysML, utilizando como referência a dissertação de mestrado do CSE/INPE de Jonas Bianchini Fulindi e o trabalho de iniciação científica de Bráulio Fernandes.

Atividade 6: EXEMPLO: Aplicação do template a um exemplo da área espacial.

Atividade 7: RELATÓRIO: Elaboração do relatório sobre as atividades da iniciação científica e apresentações deste trabalho.

6. ENGENHARIA SIMULTÂNEA DE SISTEMAS

Defini-se que Engenharia Simultânea de Sistemas é a antecipação dos requisitos para fase inicial do projeto, de uma forma relacional, ou seja, a visão do todo é levada em consideração para que as antecipações não causem nenhum prejuízo ao produto, depois que os elementos sejam integrados.

6.1. Sistemas

De acordo com o International Council of Systems Engineering (INCOSE), um sistema é uma coleção de componentes os quais são organizados a fim de operarem integradamente, para atingir os requisitos estabelecidos.

Para ser entendido, um sistema deve ser analisado e estudado de forma holística, a qual visa compreender toda a extensão do sistema e também de maneira sistêmica, a qual tem como objetivo entender e identificar cada ligação e conexão do sistema, ou seja, as relações internas do sistema.

6.2. Engenharia de Sistemas

Engenharia de Sistemas é uma abordagem interdisciplinar e colaborativa para solucionar problemas complexos, que torna possível a concretização de sistemas de elevada complexidade. (INCOSE, 2012)

O seu foco encontra-se em definir, de maneira precoce, no ciclo de desenvolvimento de um sistema, as necessidades do usuário, bem como as funcionalidades requeridas, realizando a documentação sistemática dos requisitos, e abordando a síntese de projeto e a etapa de validação de forma a considerar o problema por completo.

A Engenharia de Sistemas integra diferentes especialidades do conhecimento, formando um processo de desenvolvimento estruturado que se estende da fase conceitual à fase operacional do sistema. Logo, considera tanto as questões de ordem econômica quanto técnica, com o objetivo de gerar produtos de qualidade que atendam às necessidades dos stakeholders.

Os sistemas complexos, abordados por este trabalho, referem-se àqueles que resultam um projeto de engenharia que recaem em uma abordagem recursiva, na qual cada sistema pode ser analisado recursivamente como resultante da interligação de subsistemas, cada um destes também estruturado na forma de sub-subsistemas, e assim por diante, isto é, existe uma hierarquia a ser seguida no desenvolvimento de produtos, no qual há níveis de abstrações diferentes para a solução do projeto.

Engenharia de Sistemas visa o desenvolvimento e organização de sistemas complexos, pois os sistemas, cada vez mais, tornam-se complexos, os quais incluem hardware, software, informação e processos. E quanto maior o número de relações existentes, mais complexo o sistema será, uma vez que a complexidade do sistema depende da conexão entre cada elemento, em outras palavras, o que determina a complexidade de um sistema são as relações presentes no mesmo.

A Engenharia de Sistemas propõe solucionar problemas de integração dos componentes de um sistema complexo, esta abordagem tem como objetivo capturar as necessidades dos stakeholders e transformá-las em requisitos, os quais serão validados com os próprios stakeholders para garantir a satisfação dele em relação ao produto.

Na década de 40, o termo Engenharia de Sistemas foi criado, nos Laboratórios Bell, surgindo da necessidade de identificar e manipular as propriedades de um sistema como um todo, o que em projetos complexos pode ser diferente de apenas somar as diversas propriedades de cada componente individual. (HALL, 1962)

Quando não foi mais possível confiar no progresso dos projetos para aprimorar um sistema, devido às ferramentas disponíveis também não serem mais suficientes para atender à crescente demanda. Por causa do exposto, novos métodos, os quais abordavam diretamente a complexidade, começaram a ser desenvolvidos. (SAGE, 1992)

Para a aplicação do processo de Engenharia de Sistemas deve-se identificar o todo, por exemplo, a) conhecer e identificar os stakeholders; b) Analisar o contexto funcional em cada cenário do ciclo de vida do sistema.

A evolução da Engenharia de Sistemas, que ainda hoje continua, compreende o desenvolvimento e a identificação de novos métodos e técnicas de análise. Esses métodos buscam melhorar a compreensão do sistema como um todo, já que, muitos processos de Engenharia de Sistemas tendem a serem documentos intensivos e empregam uma mistura de técnicas que muitas vezes são imprecisas e inconsistentes.

De modo que, os documentos são gerados depois da finalização do processo de Engenharia de Sistemas, sendo a formalização de todo o trabalho e esforço empregado durante as análises e estudos do sistema.

6.3. Engenharia Simultânea

A Engenharia Simultânea reconhece o impacto, do ciclo de vida do produto em um projeto sistêmico, em virtude disso, aborda cada fase do ciclo de vida como uma etapa isolada. De modo que, é tradicionalmente aplicada no desenvolvimento de elementos e não de sistemas. (PRASAD, 1996)

Em outras palavras, é a antecipação dos requisitos relativos aos processos do ciclo de vida do produto e às organizações que implementam esses processos. Salvo a antecipação de cada requisito ocorre separadamente para fase inicial do projeto, ou seja, a abordagem sistêmica não é necessariamente considerada neste processo.

6.4. Método do Professor Dr. Geilson Loureiro

A figura a seguir apresenta o framework, conjunto de conceitos a serem seguidos em um projeto, de engenharia simultânea de sistemas, elaborado pelo Professor Dr. Geilson Loureiro, o qual possui três tipos de análises, a análise de requisitos, a análise funcional e a análise física. A sua dimensão de integração possui três itens a serem integrados, o produto, o processo e a organização.

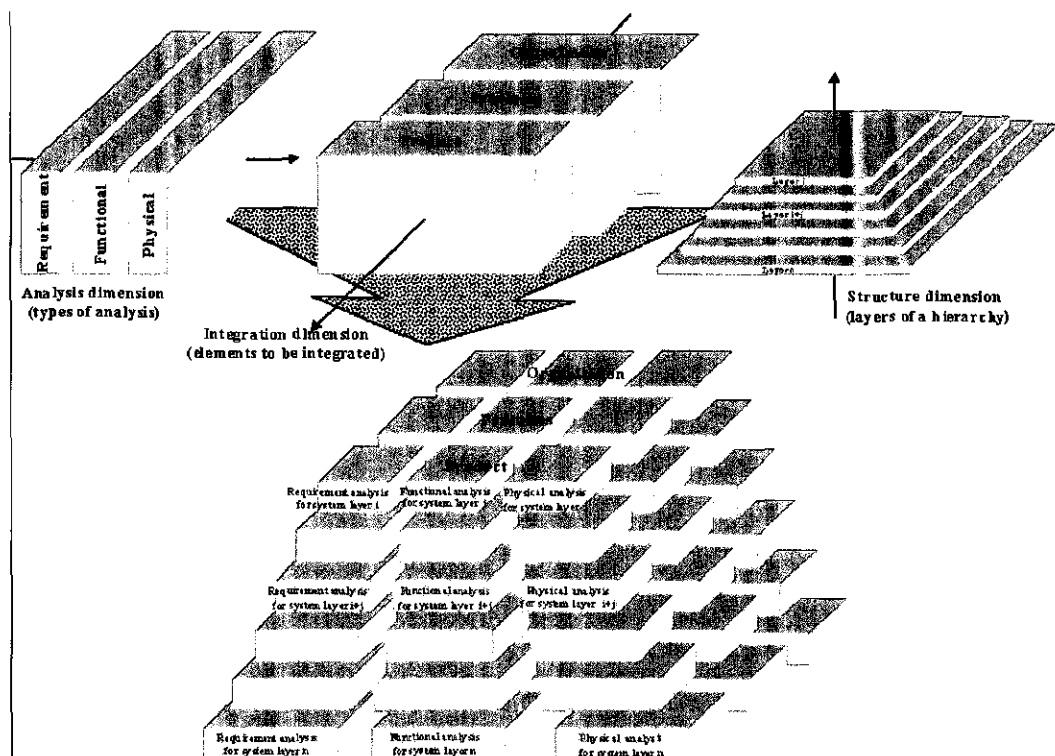


Figura 1 - Método de engenharia simultânea de sistemas.

Fonte: Professor Dr. Geilson Loureiro.

O framework ilustrado em seguida, contém um tipo a mais de análise, a análise de stakeholders. Esta inclusão surge da evolução do método do Professor Dr. Geilson Loureiro com a iniciativa de integrar o stakeholder durante todo o processo de engenharia de sistemas. Logo, o método evoluído possui quatro tipos de análises, a análise de stakeholders, a análise de requisitos, a análise funcional e a análise física.

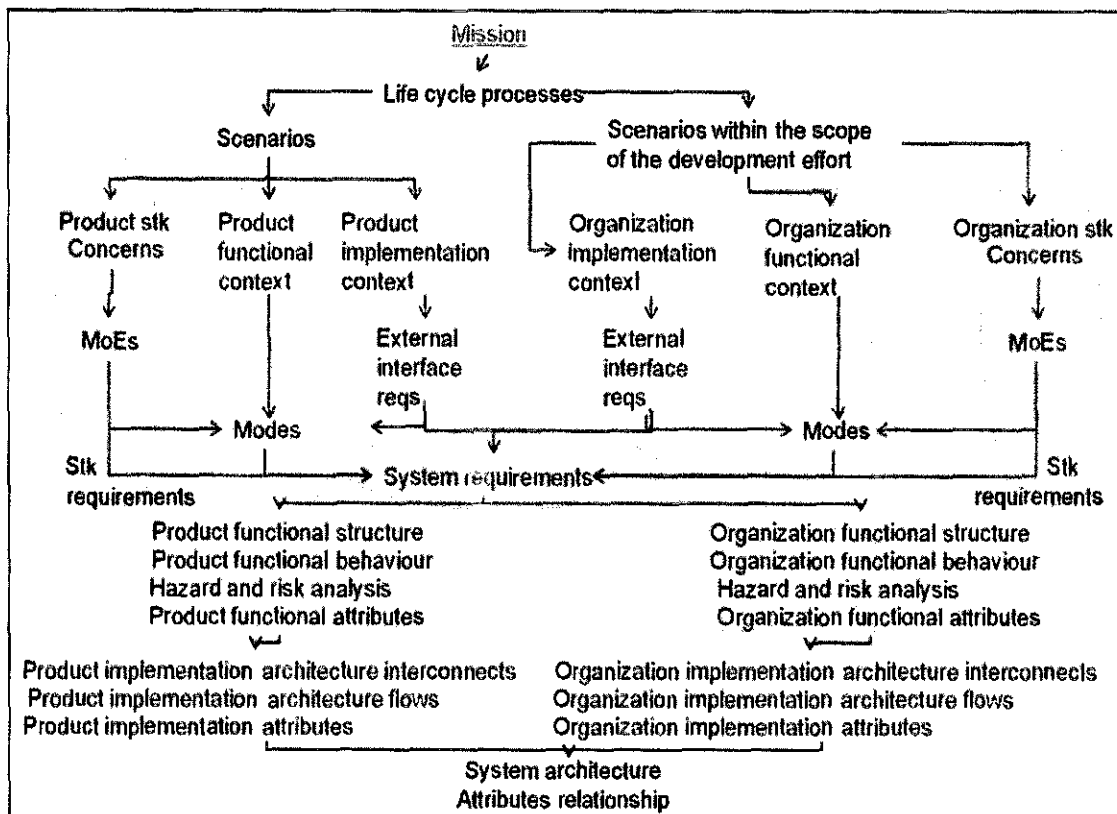


Figura 2 - Método evoluído de engenharia simultânea de sistemas.

Fonte: Professor Dr. Geilson Loureiro.

O método apresentado neste trabalho pode simultaneamente realizar análises tanto para produto como para organização desenvolvedora desse produto. As iterações existentes entre essas análises permitem que novos requisitos sejam capturados, auxiliando no entendimento da necessidade dos stakeholders.

As soluções para o projeto também são derivadas durante e a partir dessas interações. Esse loop de verificação visa confirmar que o sistema a ser implementado atende os requisitos identificados na análise de requisitos. E a validação do sistema ocorre diretamente com o stakeholder, ou seja, é feita a confirmação da satisfação do stakeholder perante o produto.

6.4.1. Principais etapas do método

A seguir serão apresentadas as etapas principais do método do Professor Dr. Geilson Loureiro, e para ilustrar melhor a hierarquia e relações entre elas serão usadas algumas telas do template construído neste trabalho.

A fim de organizar essas etapas foram utilizados os pacotes da linguagem SysML, os quais funcionam como se fossem pastas que guardam informações relacionadas.

Desta forma, é possível compreender e visualizar melhor o todo, possibilitando uma modelagem consistente. O software utilizado foi o IBM Rational Rhapsody.

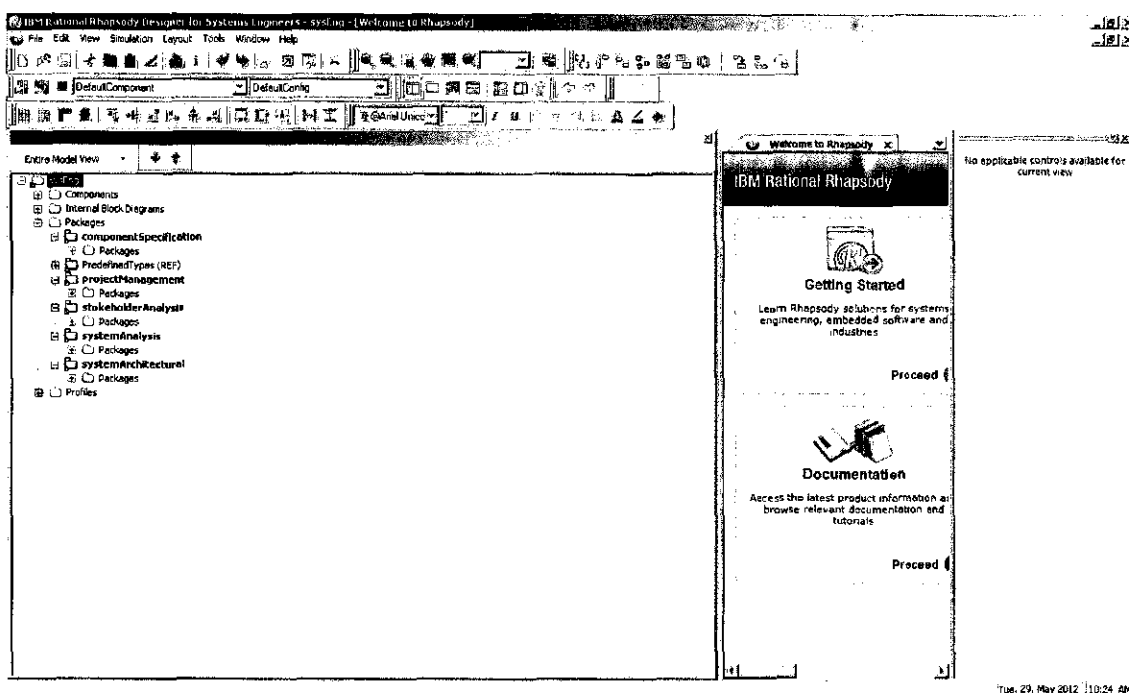


Figura 3 – Principais etapas do método.

6.4.1.1. Gestão de Projetos

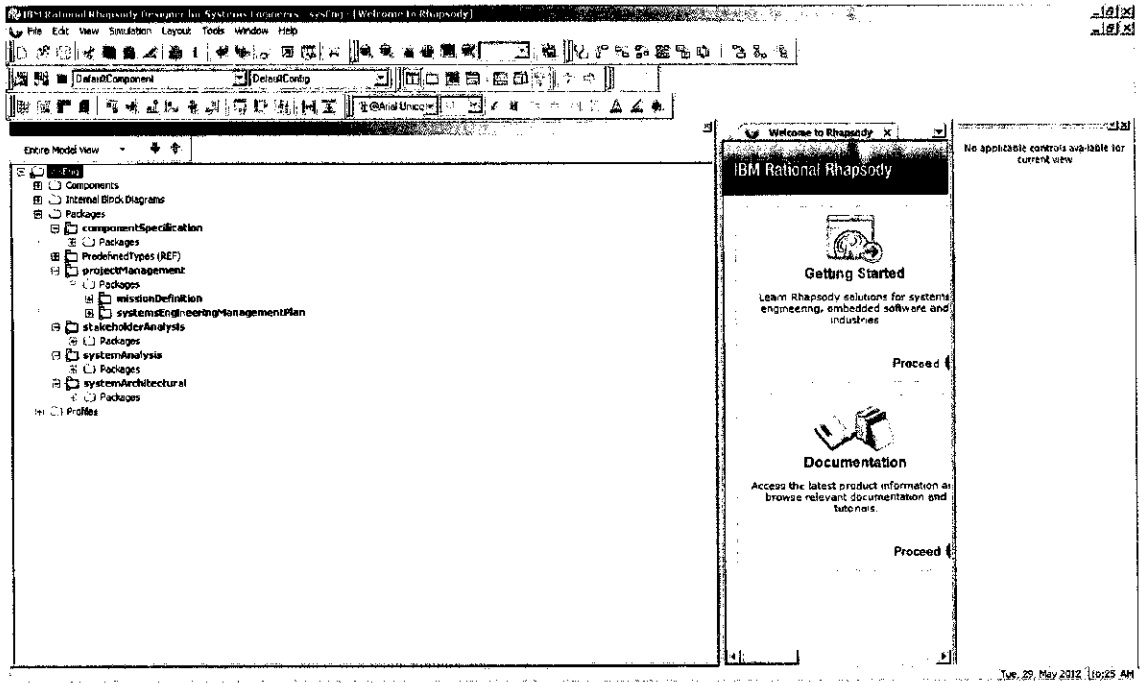


Figura 4 – Gestão de Projetos.

Durante a Gestão de Projetos é feita a definição da missão do sistema, a qual se trata da declaração da razão de existir do sistema, devendo ser documentada segundo a estrutura: O sistema deve { }. Entre as chaves é descrito o propósito do sistema.

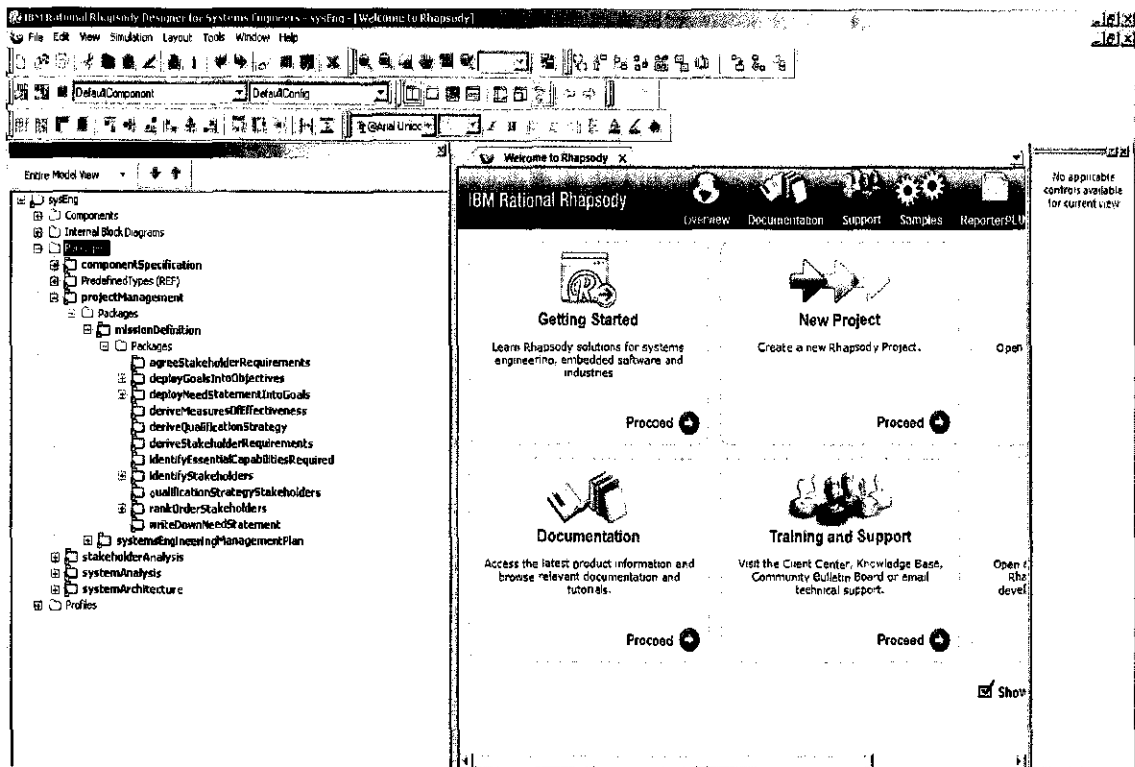


Figura 5 - Definição da missão

6.4.1.2. Análise de requisitos e Measure of Effectiveness (MoEs)

Os stakeholders expressam suas necessidades e a partir desses interesses são identificados os requisitos de stakeholders e as Measure of Effectiveness (MoEs), em português, medidas de efetividade, as quais, podem incluir, por exemplo, as facilidades de uso e o desempenho de manutenção do produto.

Stakeholders são aqueles que são afetados, afetem, acham que são afetados ou afetam o produto a ser desenvolvido por uma organização. Trata-se de elementos/pessoas essenciais ao planejamento do projeto do sistema.

Para compreender melhor um interesse do stakeholder, precisa realizar a quebra dessa necessidade, para dominá-la. Desta maneira, derivam-se as medidas dos interesses expressos por eles anteriormente, através de perguntas usando a palavra “como?”. São sucessivamente derivadas até que um conjunto de medidas seja obtido, por exemplo, custo, prazo, economia de combustível, produtividade. Esse processo trata-se das MoEs.

As MoEs representam uma forma de avaliar a satisfação dos stakeholders em relação ao produto a ser desenvolvido, ou até mesmo à organização, a qual realizará os processos dentro do escopo de desenvolvimento. A atribuição de valor e tolerância a essas medidas geram requisitos de stakeholders.

A análise de requisitos de stakeholders inclui a negociação com os stakeholders daqueles requisitos, que farão e que não farão parte do sistema, da alocação de requisitos para produto e para organização.

E a partir da identificação dos requisitos de stakeholders derivarão as capacidades (o que o sistema deve fazer) e aqueles que derivarão restrições (e.g. desempenho, limitações físicas ao que o sistema deve fazer).

E os requisitos de sistema são derivados a partir da análise dos requisitos de stakeholders, que por sua vez são derivados da necessidade dos stakeholders e das MoEs.

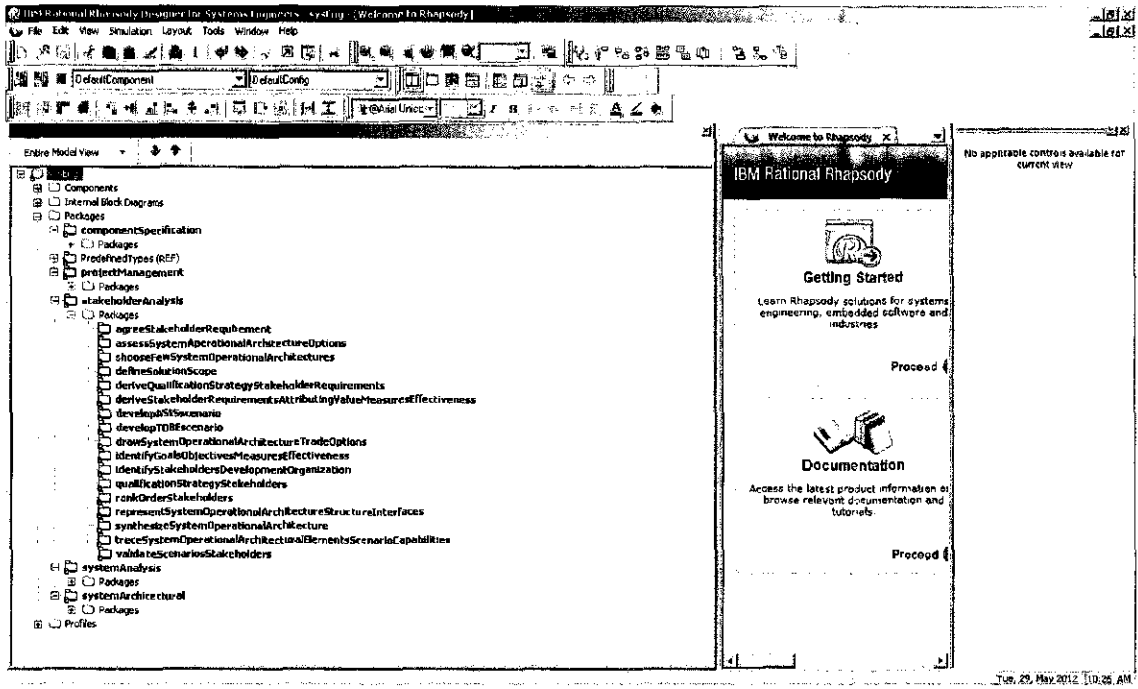


Figura 7 – Análise de stakeholders

6.4.1.3. Contexto Funcional

O contexto funcional define o escopo em que se encontra o sistema, engloba o que está no ambiente do sistema, que por sua vez contém os elementos que estão fora do sistema, mas que trocam material, energia e informação com o sistema de interesse.

Os elementos do ambiente podem ter diferentes estados, e o conjunto desses estados são chamados de circunstâncias. Os estados são sensoriais, por exemplo, som, imagem, posição, temperatura, isto é, as características físicas externas do sistema.

O sistema ainda deve ter diferentes modos, dependendo das circunstâncias. Os modos são um conjunto de funcionalidades que o produto realiza dependendo das circunstâncias.

O contexto funcional é identificado em cada cenário do ciclo de vida de produto.

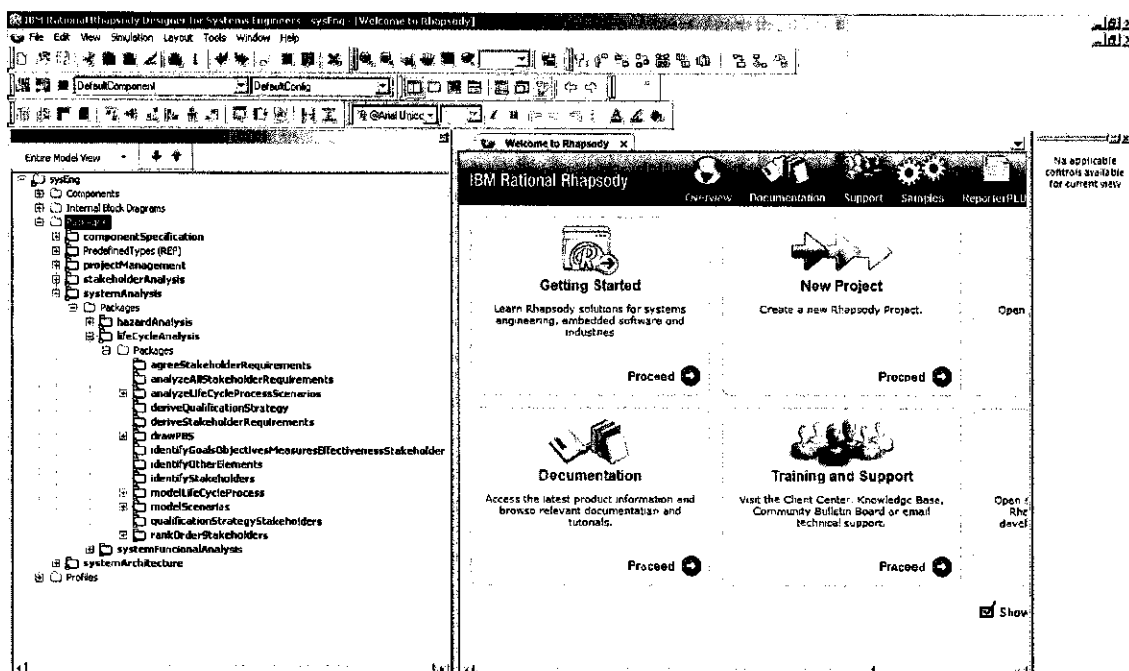


Figura 8 – Ciclo de vida do sistema.

6.4.1.4. Ciclo de vida

O ciclo de vida descreve os cenários de vida de um sistema, podendo ser modelado através de um diagrama de atividade da linguagem SysML, pois descreve um comportamento do sistema, mostrando os cenários, os quais podem ser obtidos a partir do desdobramento do diagrama de atividade quando desejar representar os processos do ciclo de vida do sistema.

Os cenários são as alternativas em cada processo do ciclo de vida, quer dizer, trata-se de uma seqüência específica de ações que ilustram o comportamento do sistema. Também pode ser entendido como, a decomposição de cada processo.

Processos do Ciclo de Vida	Cenários dos Processos do Ciclo de Vida
Desenvolvimento	<div style="display: flex; justify-content: space-around;"> <div style="border: 1px solid black; padding: 5px; width: 45%;">Desenvolvimento de um modelo completo de certificação</div> <div style="border: 1px solid black; padding: 5px; width: 45%;">Melhorias e correções no modelo de certificação</div> </div>
Implantação	<div style="display: flex; justify-content: space-around;"> <div style="border: 1px solid black; padding: 5px; width: 45%;">Implantação do modelo de certificação na organização certificadora</div> <div style="border: 1px solid black; padding: 5px; width: 45%;">Implantação do modelo de certificação nos stakeholders</div> </div>
Execução	<div style="display: flex; justify-content: space-around;"> <div style="border: 1px solid black; padding: 5px; width: 20%;">Abertura de processos de certificação</div> <div style="border: 1px solid black; padding: 5px; width: 20%;">Análise de processos de certificação</div> <div style="border: 1px solid black; padding: 5px; width: 20%;">Condução de processos de certificação</div> <div style="border: 1px solid black; padding: 5px; width: 20%;">Finalização de processos de certificação</div> </div>
Suporte	<div style="display: flex; justify-content: space-around;"> <div style="border: 1px solid black; padding: 5px; width: 30%;">Manutenção - Correção - Melhoria</div> <div style="border: 1px solid black; padding: 5px; width: 30%;">Treinamento o reciclagem</div> <div style="border: 1px solid black; padding: 5px; width: 30%;">Suprimento de informações</div> </div>
Desacarte	<div style="border: 1px solid black; padding: 5px; width: 100%;">Gerenciamento da configuração do processo de certificação</div>

Figura 9 – Processos do ciclo de vida e seus cenários

Fonte: Nota de aula do Professor Dr. Geilson Loureiro.

6.4.1.5. Análise funcional

A análise funcional é a definição das principais funções que um produto deve ter, isto é, a especificação das configurações funcionais com relação às configurações físicas correspondente, que depois serão implementadas.

A partir do contexto funcional do produto é feita a identificação das funções internas do produto em um determinado cenário, através de uma Lista de Eventos (YOURDON, 1990) considerando os elementos do ambiente e a troca de material, energia e informações com o sistema de interesse.

Cenários	Eventos	Respostas
Concepção		
Integração		
Transporte		
Manutenção		

Tabela 2 - Lista de Eventos.

Lista de eventos, os eventos tratam-se do que os elementos do ambiente realizam sobre o sistema de interesse. E as respostas do sistema são os resultados em relação aos eventos feitos pelos elementos do ambiente. Essas respostas serão as funções essenciais do produto.

A estrutura funcional é composta pelas funções do produto e pelos fluxos de material, energia e informação trocados entre o sistema e os elementos do ambiente.

O comportamento funcional do produto indica quando e sob que condições uma determinada função é iniciada, interrompida ou encerrada, ou seja, indica a seqüência no tempo da realização das funções, apresentando os aspectos causais e lógicos do sistema. É o comportamento funcional que define as transições entre estados e entre modos do sistema.

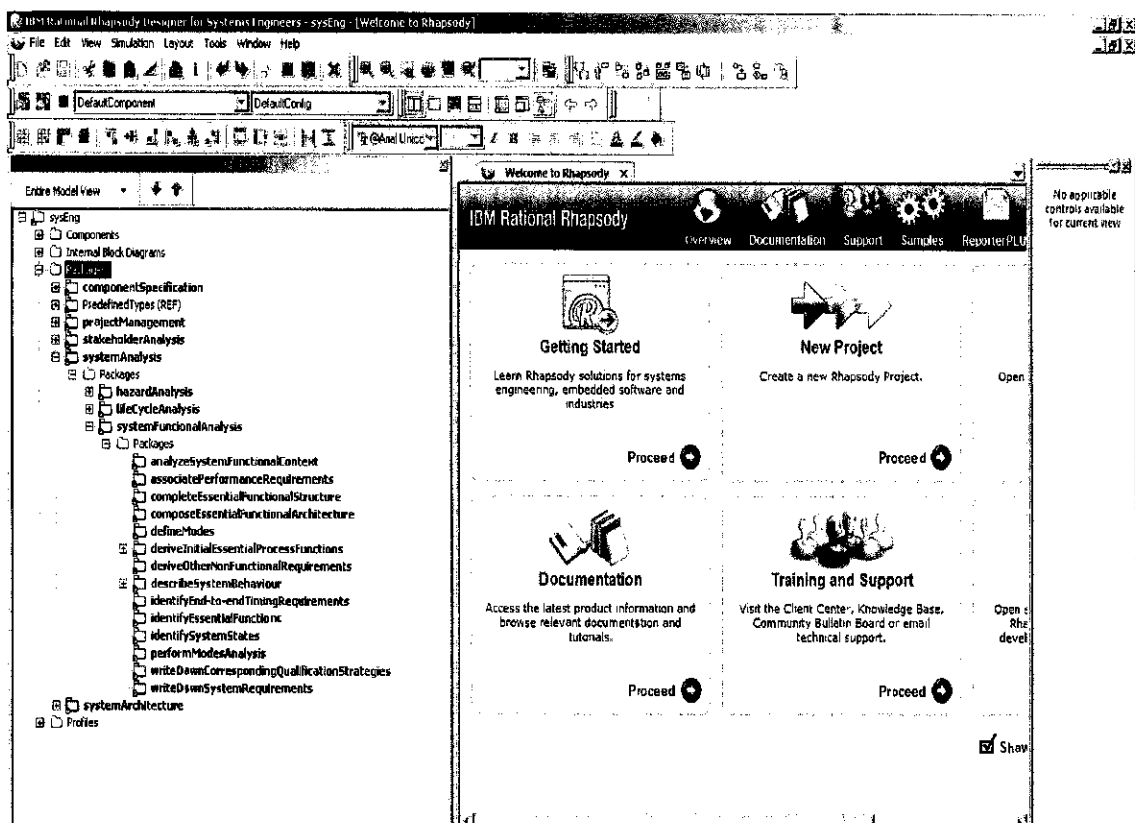


Figura 10 – Análise funcional do sistema.

6.4.1.6. Análise de perigo

A análise de perigo e risco é realizada para cada perigo potencial identificado a partir de a) circunstâncias, b) falhas nos fluxos entre o produto e os elementos do ambiente do sistema c) falhas nas funções internas do produto.

Perigo é qualquer dano imposto pelo produto ou organização ao próprio produto ou organização, ou a qualquer elemento no ambiente do produto ou da organização.

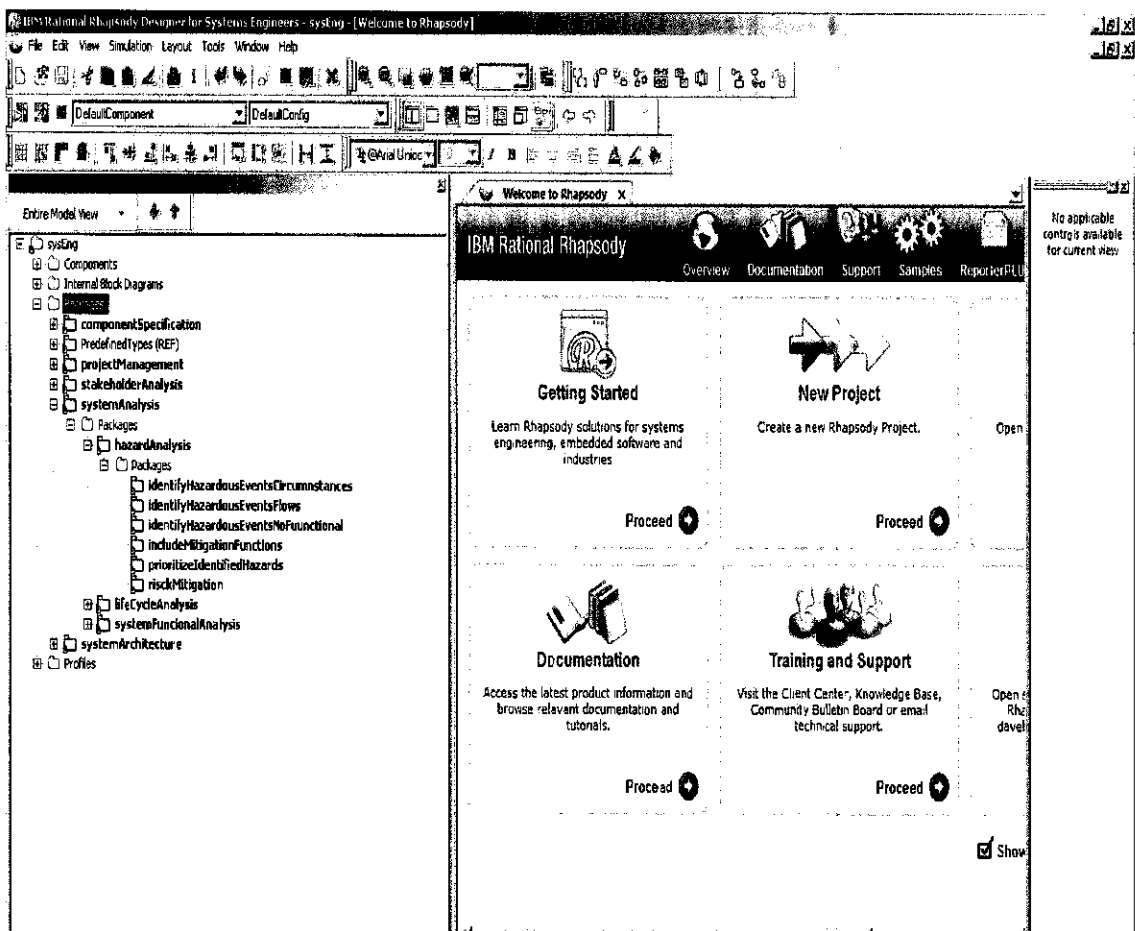


Figura 11 – Análise de perigo.

Uma tabela de falhas e de análise crítica FMECA (JURAN e GODFREY, 1998) é construída listando as circunstâncias, as falhas nos fluxos e as falhas nas funções internas, suas consequências, sua gravidade, suas causas, sua probabilidade e ocorrência, sua dificuldade de detecção e o risco computado a partir da composição das avaliações de gravidade, probabilidade e dificuldade de detecção.

Dependendo da avaliação do risco, funções de mitigação de risco são identificadas. Essas funções podem ser de a) detecção, b) prevenção, c) proteção, d) correção.

Descarte	Manutenção	Operação	Transporte	Testes	Integração	Fabricação	Concepção	Cenários
								Elemento
Normal								
								Modo
								Circunstancias
								Estado
								Nível
Falha								
								Modo
								Circunstancias
								Estado
								Nível
								Perigo
								Consequência
								Causa
								Deteção
								Gravidade
								Probabilidade
								Risco
								FMECA
								Verificação

Tabela 3 - Falhas e análise crítica FMECA

6.4.1.7. Contexto Físico

O contexto físico também é identificado em cada cenário do processo do ciclo de vida, apresentando o sistema, os elementos do ambiente, e as conexões físicas entre os componentes existentes.

A diferença entre o contexto físico e o contexto funcional, está nos elementos de ligação entre o sistema e os elementos do seu ambiente, resumindo, as interfaces. Pois, no funcional, são apresentados os fluxos de material, energia e informação. Já no físico, são apresentadas as conexões físicas entre o sistema e os elementos do seu ambiente. Portanto os fluxos do contexto funcional fluem pelas conexões do contexto físico.

As conexões físicas entre o sistema e os elementos do ambiente, definem os requisitos de interface externa de produto.

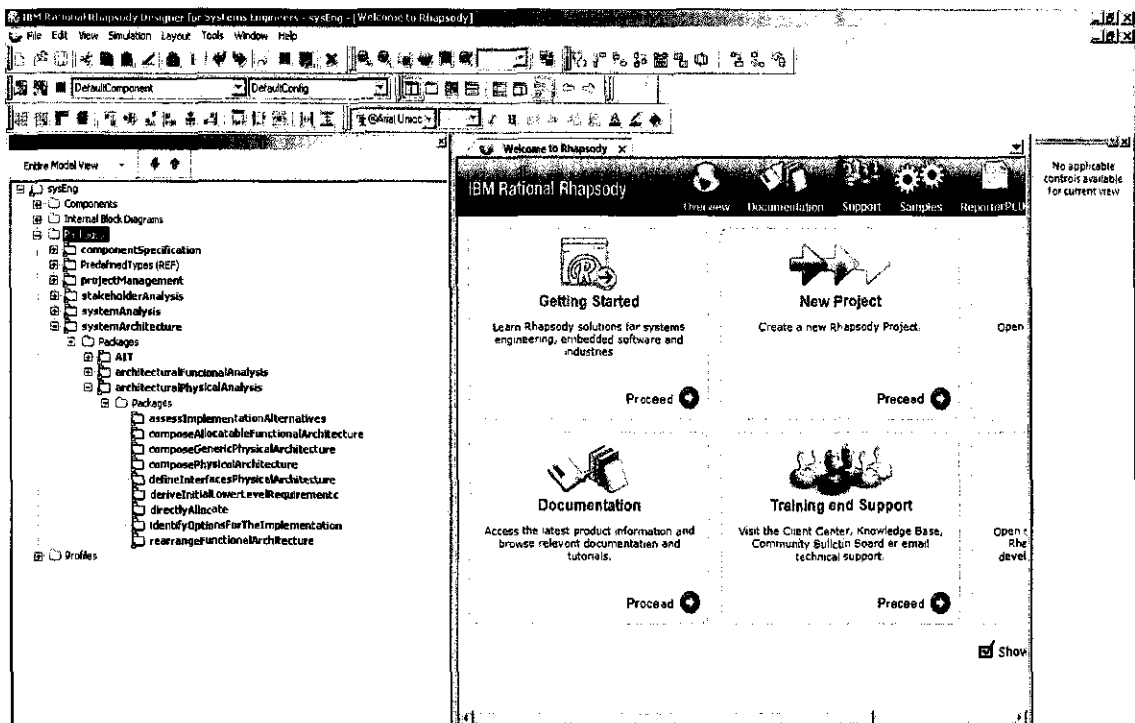


Figura 12 – Contexto físico.

6.4.1.8. Arquitetura de sistemas

A arquitetura de sistemas é o resultado do processo de Engenharia de Sistemas que apresenta os elementos do produto e da organização, sejam eles componentes, fluxos, interfaces e relacionamentos existentes.

A implementação provê as entradas necessárias para os processos de projeto detalhado, isto é a descrição de componentes, como, aquisição ou reuso de componentes, integração e testes do sistema composto por elementos do produto e da organização.

A arquitetura física é a implementação das funções descrita pelos elementos dos modelos de arquitetura funcional, como os componentes e os fluxos do sistema.

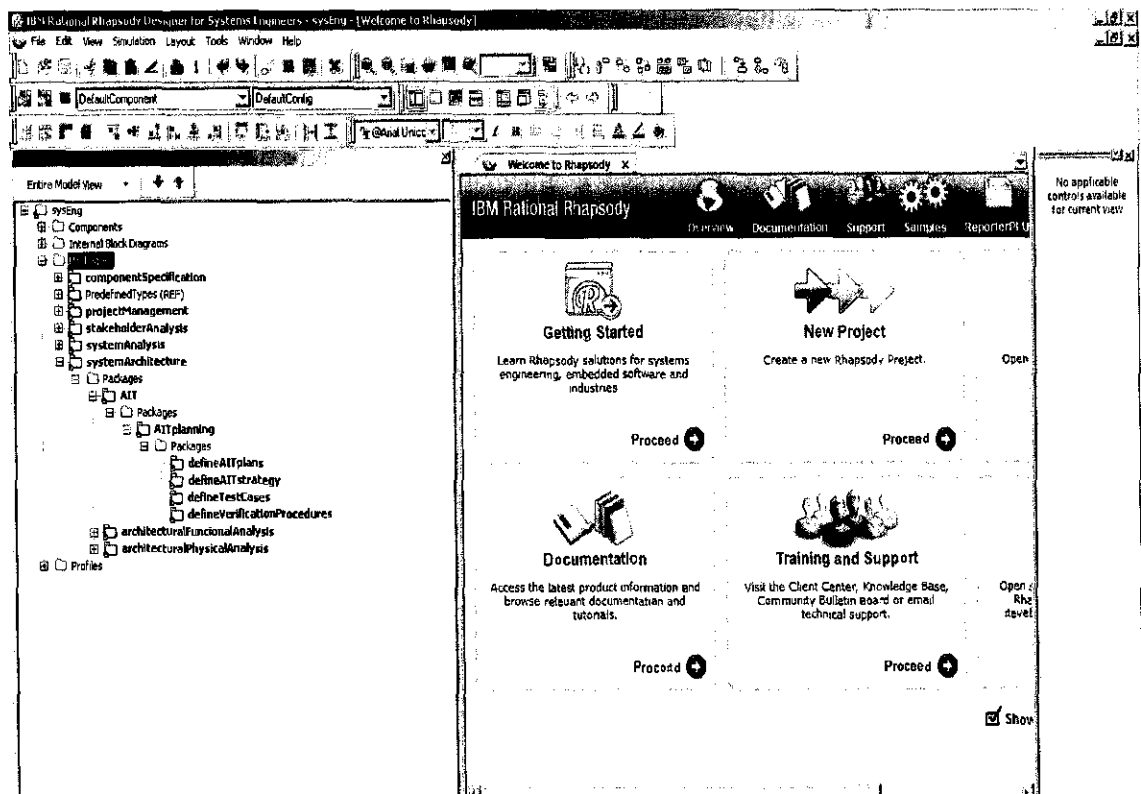


Figura 13 – AIT.

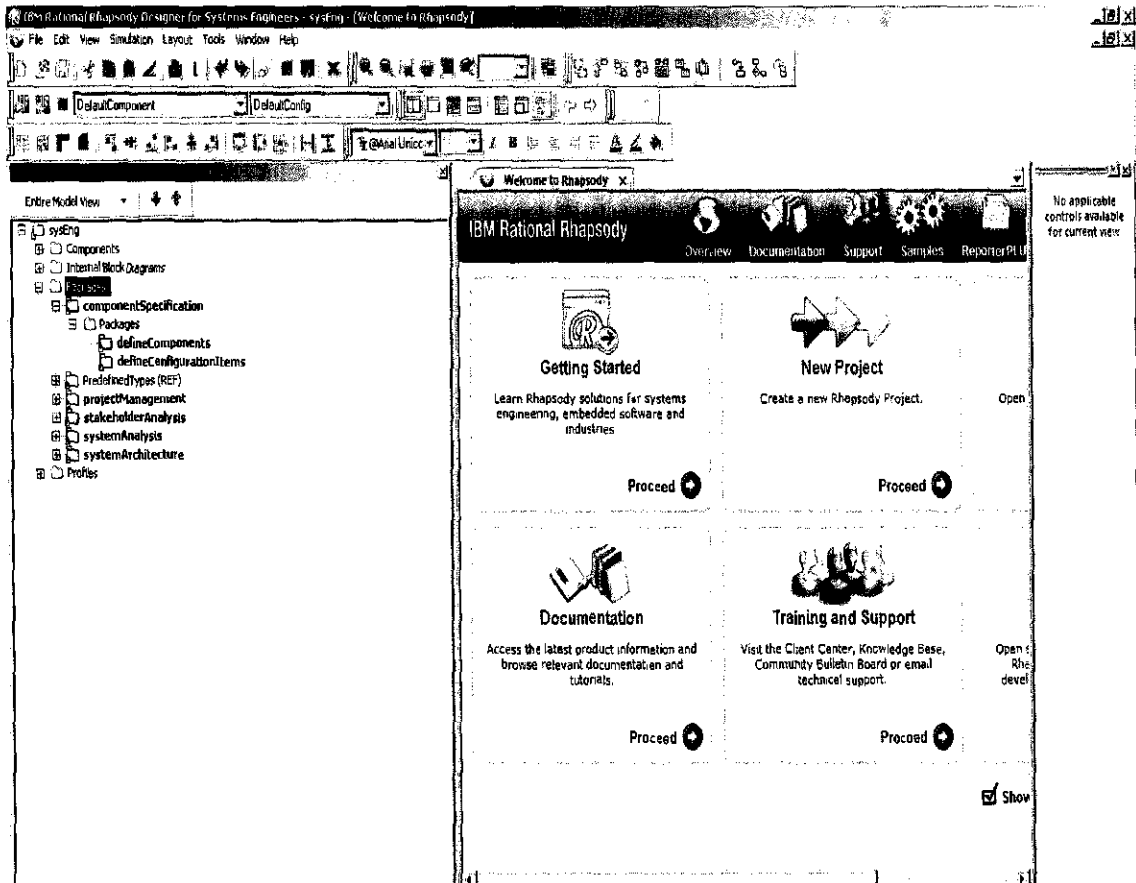


Figura 14 – Especificação de componentes.

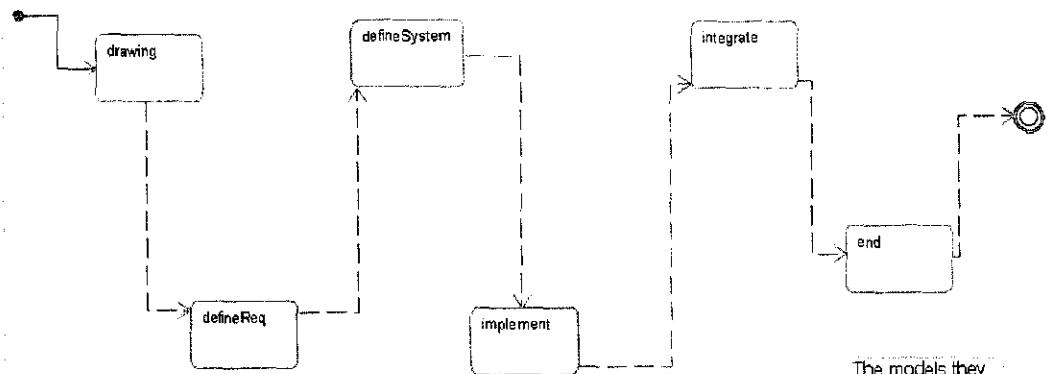
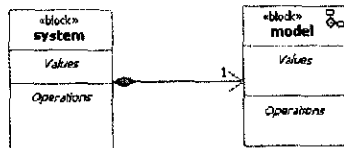
7. MODELAGEM GRÁFICA

As informações são veiculadas por representações, a qual possui semântica e sintaxe, sendo uma dessas representações a modelagem, a qual facilita o processamento e apresentação da informação de sistemas complexos.

A modelagem é uma técnica de Engenharia já aprovada e bem aceita que vai desde modelos matemáticos, computacionais, até os protótipos. Um modelo é a simplificação da realidade, a qual também pode ser entendida como uma abstração de um sistema, auxiliando na visualização do todo. Desta forma, os modelos não só podem como devem ser aproveitados a fim de gerenciar os riscos.

Com o auxílio da informática é mais fácil de corrigir possíveis erros na modelagem, sendo que a escolha da ferramenta faz diferença na qualidade da modelagem do sistema, por isso é fundamental fazer uma pesquisa de softwares que atendem a necessidade do modelador.

Para obter uma boa modelagem de um sistema complexo é preciso uma boa ferramenta computacional. Pois, apenas com um software é possível realizar, por exemplo: a) Controle de versões de um modelo. b) Análises estruturais e comportamentais do sistema em estudo. c) Rastreamento de requisitos tanto de stakeholders como os de sistemas. d) Captura de novos requisitos; pois a partir dos modelos é possível extrair requisitos do sistema modelado.



The models they depict have enormous value to the SE process

Figura 15- Modelagem de um sistema.

Fonte: Template deste trabalho

É através da modelagem, o fluxo de material, energia e informação é mais fácil de ser identificado. Já que, as formas gráficas são mais puras por serem mais fieis ao raciocínio original. (MEDINA, 2005)

Diversas vezes a modelagem gráfica gera desconforto porque é mais fácil e comum as pessoas expressarem-se por palavras, porém uma das grandes vantagens de utilizar os modelos gráficos é substituir um grande número de palavras por convenções de desenhos, isto é, símbolos. (MEDINA, 2005)

Os modelos permitem que o sistema seja melhor compreendido por todos envolvidos em um projeto. Pois a modelagem permite: a) Visualizar um sistema inteiramente. b) Especificar a estrutura e o comportamento de um sistema. c) Proporcionar um template para construção de um sistema (objetivo deste trabalho). d) Documentar as decisões tomadas em cada fase do projeto.

Além do mais é possível modelar sistemas desde os planos detalhados até os planos de mais alto nível de abstração, os quais este trabalho será baseado. E quanto mais complexo for o sistema, mais importante se torna a modelagem do mesmo.

Há vários tipos de modelagem gráfica desde os simples fluxogramas, passando pela Análise Estruturada, até a modelagem baseada em orientação a objeto, como, a Unified Modeling Language (UML), a qual contém diagramas de classes, e recentemente lançou a sua nova extensão chamada Systems Modeling Language (SysML), a qual foi criada para modelar sistemas complexos.

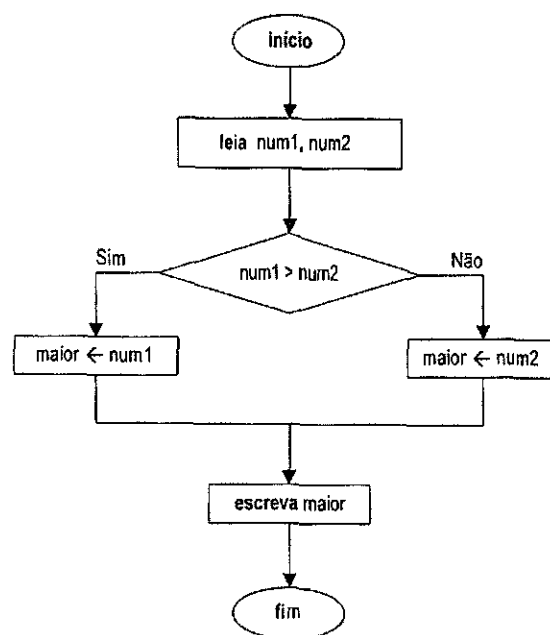


Figura16 - Fluxograma.

Fonte: Medina, 2005.

7.1. AS QUATRO BASES DA MODELAGEM

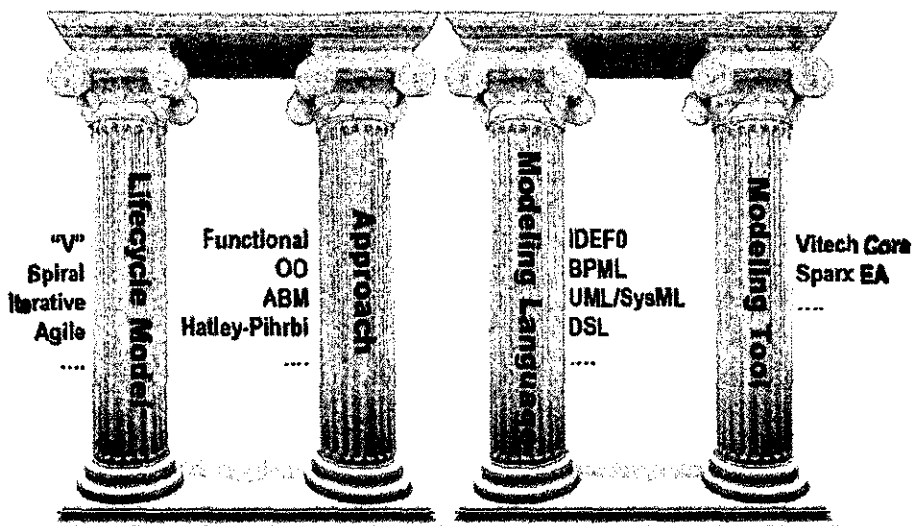


Figura 17 - As quatro bases da modelagem.

Fonte: INCOSE, 2011.

- 1) Ciclo de vida: O modelador deve estar ciente sobre qual será o modelo de ciclo de vida (V, Espiral, Cascata, entre outros) a ser utilizado no projeto de processo de engenharia de sistemas.
- 2) Approach: O approach da modelagem deve ser escolhido segundo o sistema que será modelado para que o resultado seja fiel a realidade do produto em estudo. A rigidez sintática e semântica faz parte das linguagens de modelagem, porque garante ao modelo que não haja mais de uma interpretação possível, ou seja, não gere ambiguidade.
- 3) Linguagem: Deve-se pesquisar qual é a linguagem mais apropriada para modelar o sistema a ser desenvolvido.
- 4) Software: Deve-se ter um bom domínio sobre a ferramenta computacional escolhida para modelar o produto conhecendo suas funções disponíveis, para garantir que se tenha ao final um trabalho com boa qualidade.

8. ANÁLISE ESTRUTURADA

No final da década de 70, a Análise Estruturada, modelagem bottom-up, permitiu a especificação de requisitos lógicos de um sistema em um modelo gráfico de alto nível de abstração, possibilitando que usuários compreendessem o que estava documentado nos diagramas, além de garantir que a arquitetura do sistema fosse mapeada.

O modelo gráfico introduzido pela Análise Estruturada representa: a) os dados utilizados por um sistema; b) os fluxos que transportam; c) os processos que os transformam.

Essa modelagem trata-se da construção de diagramas com fluxo de dados e informações sendo dividida em elementos funcionais, os quais descrevem o que deve ser construído.

A Análise Estruturada contém modelos que possibilita a demonstração de fluxo de informação, usando símbolos, juntamente com um dicionário de dados e narrativas dos processos como complemento aos modelos de fluxo de informação. Um modelo de fluxo pode ser criado para qualquer sistema baseado em computador, independentemente do tamanho e complexidade.

A técnica estruturada contém a modelagem comportamental, a qual representa o comportamento de um sistema, descrevendo seus estados e os eventos que fazem com que o sistema mude de estado.

Um diagrama muito conhecido e utilizado para expressar a troca de dados é o Diagrama de Fluxo de Dados (DFD), o qual descreve o fluxo de informação e transformações que são realizadas no sistema conforme os dados se movem da entrada para a saída. Um DFD pode representar todos os níveis de abstração.

Para auxiliar a modelagem em Análise Estruturada, como, rastrear e gerenciar mudanças no sistema, as ferramentas computacionais tornaram-se fundamentais para garantir a qualidade da modelagem. Pois estas constroem uma hierarquia interna de forma que cada bolha "pai" e seus "filhos" sejam automaticamente associados entre si.

Todos os comandos necessários podem ser acessados por menus e outras funções que também estão disponíveis, ajudando na visualização e entendimento do sistema. Dessa forma, o modelo criado, podendo ser copiado, alterado e impresso apenas com alguns comandos, economizando tempo e recursos financeiros, além de diminuir riscos.

Existem na Análise Estruturada extensões para modelar sistemas de tempo real, os quais são orientados pelo controle dos eventos com interação ao mundo real em um determinado período. Uma dessas extensões é a do Hatley e Pirbhai, a qual visa a representação e especificação dos aspectos orientados ao controle do software, criando um modelo de sistema de tempo real.

9. UNIFIED MODELING LANGUAGE (UML)

A Unified Modeling Language (UML) é uma linguagem de modelagem gráfica estabelecida no campo de desenvolvimento de software, é um padrão internacional especificado pelo OMG e também é aceito como padrão ISO/IEC 19501. (WEILKIENS, 2007)

Como a linguagem UML foi projetada para software, é ineficaz para aplicação em outros sistemas, como por exemplo, sistemas espaciais, mecânicos, médicos e financeiros. Outro problema encontrado é o fato da linguagem UML ter sido desenvolvida para análise e validação de softwares ao passo que a modelagem dos outros sistemas envolve outras disciplinas além dos softwares, ou seja, são interdisciplinares.

Várias iniciativas de padronizar os processos de engenharia de sistemas sugeriram, mas até o início dos anos 2000, nenhuma linguagem de modelagem havia sido criada, exigindo a utilização de diversas linguagens em um mesmo projeto, o que dificultava a comunicação entre os profissionais.

Na linguagem UML há falta de diagramas específicos para modelagem de requisitos e parâmetros de sistemas complexos, sem serem softwares.

Devido a essa necessidade de modelar outros sistemas, em 2001 a Object Management Group (OMG) juntamente com o International Council on Systems Engineering (INCOSE) decidiu criar um padrão da linguagem UML específica para engenharia de sistemas, a Systems Modeling Language (SysML).

A linguagem UML foi escolhida, pois já atendia alguns requisitos, já era amplamente utilizada e aceita pelos profissionais do mundo todo e possuía muitas ferramentas que davam suporte a ela, o que facilitaria fazer as adaptações para essa nova necessidade. Outro fator importante para a sua utilização foi a sua propriedade de extensão através do mecanismo de estereótipos, o qual permite a definição e adaptação para um domínio específico.

A atualização da linguagem UML para a linguagem SysML foi a criação de dois novos tipos de diagramas: diagrama paramétrico e diagrama de requisitos. Nesse processo ocorreu também omissão de alguns elementos da linguagem UML específicos para o desenvolvimento de software.

Para a modelagem de sistemas, a linguagem SysML possui algumas vantagens consideráveis em relação à linguagem UML. O principal é o fato de possibilitar a compreensão de forma clara e objetiva das propriedades, das restrições e dos requisitos de um sistema complexo.

10. SYSTEMS MODELING LANGUAGE (SYSML)

A Systems Modeling Language (SysML) é uma linguagem de modelagem gráfica de propósito geral, para a análise, especificação, projeto, verificação e validação de sistemas complexos, sendo que nesses sistemas podem estar incluídos hardware, software, dados, pessoas, procedimentos, facilidades e outros elementos de sistemas naturais. (FRIEDENTHAL et al, 2009)

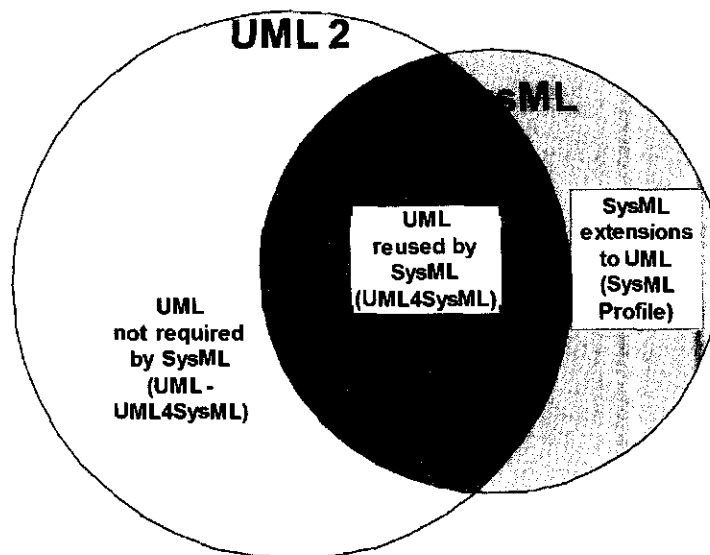


Figura 18 - Relação entre as linguagens UML e SysML.

Devido à sabida importância e necessidade de documentar tudo que se produzem os engenheiros de sistemas perceberam que a UML não estava sendo suficiente para atender seus interesses e, então, a partir de uma iniciativa do International Council on Systems Engineering (INCOSE) com o Object Management Group (OMG), responsável pela linguagem UML, criaram em 2001 um grupo de trabalho que definiu os requisitos de uma linguagem de modelagem para Engenharia de Sistemas.

Estes requisitos foram publicados em 2003 como uma chamada para propostas (UML for Systems Engineering Request for Proposal - UML for SE RFP). Foi então organizado o SysML Partners, um grupo de trabalho composto por representantes do setor industrial e produtores de ferramentas CASE.

Esse grupo iniciou a elaboração da linguagem SysML Open Source, uma linguagem que respondesse aos requisitos especificados na UML for SE RFP. A versão draft da linguagem SysML foi publicada em 2004, enquanto que a primeira versão, a linguagem SysML 1.0, foi proposta no final de 2005. A versão pública formal da linguagem SysML OMG™ v 1.2 foi publicada pela OMG em Junho de 2010.

Industry and government
American Systems
BAE Systems
The Boeing Company
Deere & Company
EADS Astrium GmbH
Eurostep Group AB
Lockheed Martin Corporation
Motorola, Inc.
National Institute of Standards and Technology (NIST)
Northrop Grumman Corporation
oose.de Dienstleistungen für innovative Informatik GmbH
Raytheon Company
Thales

Tool vendors
ARTISAN Software Tools
EmbeddedPlus Engineering
Gentleware AG
IBM
I-Logix, Inc.
Mentor Graphics
PivotPoint Technology Corporation
Sparx Systems
Telelogic AB
Vitech Corp

Academia
Georgia Institute of Technology

Liaison organizations
INCOSE
ISO 10303-233 (AP233) Working Group

Tabela 4 – SysML Partners.

A linguagem SysML derivou-se da linguagem UML, a qual também vem sendo aperfeiçoada e atualmente encontra-se na versão 2.4.1., que, por sua vez, é baseada no paradigma de orientação ao objeto, o qual aumenta a velocidade no desenvolvimento dos programas devido a reutilização dos objetos, sendo assim a linguagem SysML também reaproveita os diagramas, fator que ajuda a modelagem a ficar com concisa e menos suscetível a erros, possibilitando mais análises e simulação a testes físicos, gerando grande economia e mais segurança à organização.

UML diagrams	SysML diagrams
Class diagram	Borrowed and adapted with the concept of blocks within the Block Definition diagram
Object diagram	N/A
Package diagram	Borrowed
Component diagram	N/A
Composite Structure diagram	Borrowed and adapted with the concept of blocks within the Internal Block diagram
Deployment diagram	N/A
Use case diagram	Borrowed as-is
Activity diagram	Borrowed and extended
Interaction diagram	N/A
Communication diagram	N/A
Interaction Overview diagram	N/A
Sequence diagram	Borrowed and adapted
Timing diagram	N/A
State machine diagram	Borrowed and adapted
N/A	Requirement diagram
N/A	Parametric diagram

Tabela - Comparação entre as linguagens UML e SysML.

A linguagem SysML vem se tornando padrão para modelar sistemas complexos, encontra-se em fase de disseminação e aceitação pela comunidade científica, pois é uma linguagem gráfica nova. A qual surgiu com o propósito de eliminar as interpretações equivocadas ou ambíguas aos sistemas complexos, possuindo conteúdo necessário para satisfazer as necessidades da Engenharia de Sistemas.

A principal motivação do uso da linguagem SysML é conseguir reunir e integrar todos os requisitos, sejam eles, dos stakeholders, do sistema e dos subsistemas, sendo necessário uma modelagem sólida, que aceite atualizações e não comprometa o resto do projeto.

O fato de já existirem estudos buscando a sua integração com Modelica é outro incentivo para aprender a linguagem SysML, uma vez que Modelica já é amplamente utilizada na indústria, principalmente automotiva.

As principais funcionalidades e vantagens da linguagem SysML são: a) Gerenciamento de complexidade através da modelagem de pacotes. b) Comunicação entre os stakeholders e profissionais da organização que desenvolverá o produto. c) Redução de ambiguidade e erros nos modelos. d) Rastreabilidade de requisitos para validação futura. e) Possibilidade de simulações, uma vez que, modelos podem ser simulados, documentos não.

Um grande diferencial que a linguagem SysML traz é possuir cabeçalhos em todos os seus diagramas.

10.1. Diagramas da linguagem SysML

A linguagem SysML possui nove tipos de diagramas.

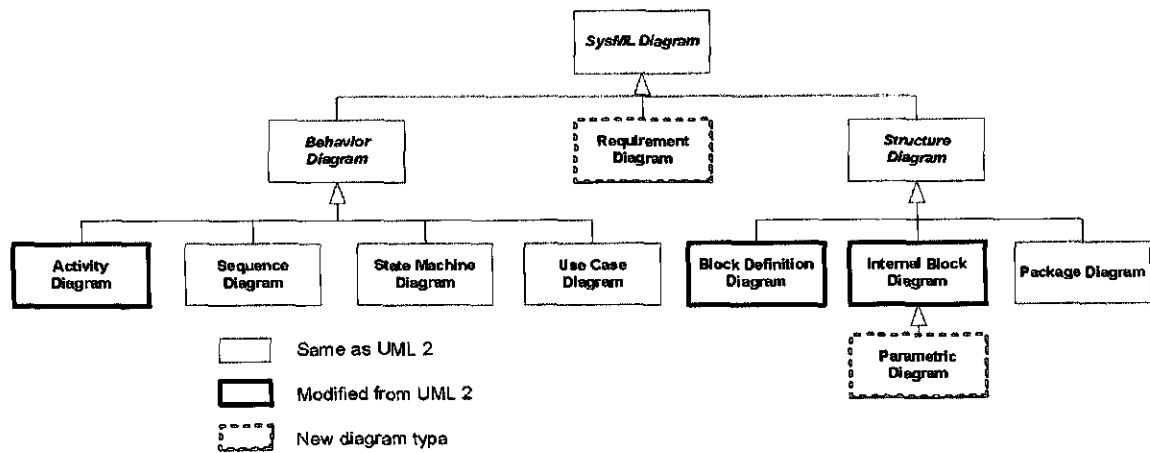


Figura 19 - Diagramas da linguagem SysML.

A seguir, serão apresentados todos os diagramas da linguagem SysML e suas respectivas características principais.

10.1.1. Diagrama de atividade

- a) Exibe etapas de um processo, com foco no fluxo de material, energia e informação entre uma etapa e outra do sistema.
- b) Existem dois modos de criar o diagrama de atividades:
 - ✓ Fluxo de trabalho: especifica, constrói e documenta processos.
 - ✓ Operação: empregado como fluxograma, pois contém ramificações, bifurcações e estados de união. Representa os fluxos conduzidos por processamentos. É essencialmente um gráfico de fluxo, mostrando o fluxo de controle de uma atividade para outra. Comumente isso envolve a modelagem das etapas sequenciais em um processo.
- c) Mostra o comportamento expresso pelos fluxos de controle e pelos fluxos de dados de uma atividade para a outra.
- d) Não são importantes somente para a modelagem de aspectos dinâmicos de um sistema ou um fluxograma, mas também para a construção de sistemas executáveis por meio de engenharia de produção reversa.
- e) Permite a criação de caminhos alternativos, para isso utilizam-se as expressões booleanas.
- f) Possui uma visão dinâmica do sistema modelado.

10.1.2. Diagrama de sequência

- a) Descreve de forma simples e lógica como as atividades se comportam ao longo do tempo em um sistema.
- b) Enfatiza a ordenação temporal das atividades de um sistema, através das mensagens que são trocadas entre os eventos do sistema. Entende-se por mensagens os serviços solicitados de uma atividade a outra, e as respostas desenvolvidas para as solicitações.
- c) Existe um caminho que deve ser seguido pela mensagem no diagrama, sendo que os comandos lógicos não deixam de cumprir um evento, ou seja, a mensagem passa por todas as atividades modeladas.
- d) Registra o comportamento de um único caso de uso, por vez, exibindo os componentes e as mensagens passadas entre esses componentes no diagrama de caso de uso.
- e) Representa interações entre elementos de um cenário, realizadas através de operações ou métodos (procedimentos ou funções).

10.1.3. Diagrama de transição de estados

- a) Demonstra a visão dinâmica de um sistema, há softwares de modelagem, que permitem animações e simulações no próprio modelo.
- b) É formado por estados, transições de eventos e de atividades.
- c) Exerce grande importância na modelagem comportamental das interfaces do sistema.
- d) Representa o estado ou a situação em que um evento pode encontrar-se no decorrer da execução do processo de um sistema complexo. Com isso, o evento transforma-se de um estado inicial para um estado final através de transição.

10.1.4. Diagrama de caso de uso

- a) Apresenta uma visão externa sobre como os elementos dos sistemas podem estar relacionados.
- b) Descreve as funcionalidades propostas para um novo sistema, o qual será projetado, podendo estender um caso de uso como seu próprio comportamento.
- c) Expõe como um sistema se comporta e como esse comportamento será implementado.
- d) Facilita organização e modelagem comportamental do sistema.
- e) Modela aspectos dinâmicos do sistema, envolvendo a modelagem do contexto do sistema e a modelagem de requisitos do comportamento desses elementos do ambiente.
- f) Representa a interação entre um ator (humano ou máquina) e um sistema.
- g) Segundo Ivar Jacobson, trata-se de um documento narrativo que descreve a sequência de eventos de um ator que usa um sistema para realizar um processo.
- h) Trata-se de um conjunto de casos de uso do sistema, e atores, sendo que seus relacionamentos apresentam uma visão estática do sistema.
- i) Trata-se de uma unidade de trabalho significativa, ou seja, uma tarefa a ser executada pelo ator no sistema, por exemplo, o login para um sistema e o registro em um sistema.

10.1.5. Diagrama de definição de blocos

- a) Descreve as relações entre cada subsistema e a troca do fluxo de material, energia e informação.
- b) Diferem dos fluxogramas por representarem pequenas partes de um grande sistema com foco no processo lógico.
- c) Modela a visão estática do projeto ou do processo do sistema, mostrando os conjuntos de blocos e seus relacionamentos no nível mais alto de abstração.
- d) Representa a modelagem gráfica estrutural de um processo ou modelo de um sistema complexo.

10.1.6. Diagrama de blocos internos

- a) Descreve as relações entre cada subsistema e a troca do fluxo de material, energia e informação.
- b) Modela a visão estática do projeto ou do processo do sistema, mostrando os conjuntos de blocos e seus relacionamentos, no nível mais detalhado do projeto, o nível mais baixo de abstração.
- c) Representa a modelagem gráfica estrutural de um processo ou modelo de um sistema complexo, possibilitando a modelagem dos componentes de um sistema.
- d) Trata-se da subdivisão do diagrama de definição de blocos.

10.1.7. Diagrama de pacotes

- a) Trata-se de pastas que se relacionam com outras pastas através de uma relação de dependência, ou seja, pacotes representam os subsistemas do sistema divididos em agrupamentos lógicos, os quais mostram as dependências entre eles. Entende-se por pastas, as pastas de um sistema operacional (documentos, imagens, músicas, vídeos, etc.) para organizar o sistema.

- b) Usado especialmente para ilustrar a arquitetura de um sistema, podendo ser usado em qualquer fase do processo de modelagem, pois visa organizar os modelos.

10.1.8. Diagrama de parâmetros

- a) Derivam-se a partir dos parâmetros as restrições do sistema.
- b) Descreve sucintamente as variáveis do processo, as quais, sendo cumpridas, habilitam a satisfação dos requisitos.
- c) Relaciona as variáveis e equações que modelam o comportamento do sistema, com o software de simulação matemática MATLAB.

10.1.9. Diagrama de requisitos

- a) Trata-se de um tipo específico do diagrama de definição de blocos.
- b) Usado para modelar tanto os requisitos de stakeholder como os requisitos do sistema, mostrando os seus relacionamentos.

11. FERRAMENTAS

As ferramentas de engenharia de sistemas são fundamentais para modelar um sistema complexo, pois realiza o controle de versões dos documentos gerados a cada fase do processo da engenharia de sistemas. Quanto mais complexo é um produto maior a necessidade de um software que auxilie o andamento do processo durante o projeto. Devido a essa necessidade surgiram várias ferramentas para auxiliar a aplicação dos conceitos de engenharia de sistemas.

Segue a descrição de algumas ferramentas que auxiliam o processo de engenharia de sistemas.

11. 1. Cradle

O Cradle é um software que ajuda a implementação de um ambiente de engenharia de sistemas, a qual permite a gestão de requisitos e sistemas.

Algumas de suas funções são: a) Enumeração e definição de requisitos. b) Análise de sistemas e design. c) Definição da arquitetura de sistemas. d) Modelagem de processos de negócios e engenharia. e) Definição e entrega da capacidade de serviço.

Com este aplicativo é possível realizar alterações na configuração, com a compra dos pacotes de extensão da ferramenta é possível ao modelador maximizar o controle, a eficiência, a segurança e a durabilidade dos projetos, garantindo o orçamento e o tempo previsto para o projeto.

Este programa possibilita que mais de uma pessoa trabalhe simultaneamente no mesmo projeto, a fim de criar, atualizar e gerenciar colaborativamente todos e quaisquer tipos de informação envolvida em produtos complexos.

11.2. Enterprise Architect (EA)

O software Enterprise Architect (<http://www.sparxsystems.com/products/ea/trial.html>) juntamente com o plug-in, extensão para a linguagem SysML (<http://www.sparxsystems.com/products/mdg/tech/sysml/index.html>) é capaz de modelar, projetar e construir um sistema usando a linguagem SysML. Nos links apresentados é possível obter o TRIAL dessa ferramenta e usá-la por trinta dias.

A partir dos modelos elaborados na ferramenta é possível gerar códigos automaticamente em Java, C#, C++, VB.NET, VB, Python e DLL. Desta forma, contribui para que a execução seja mais rápida, já que se derivam os códigos fontes diretamente dos modelos gráficos. Além disso, cria relatórios, gerencia e simula testes, entre outros recursos e disponíveis.

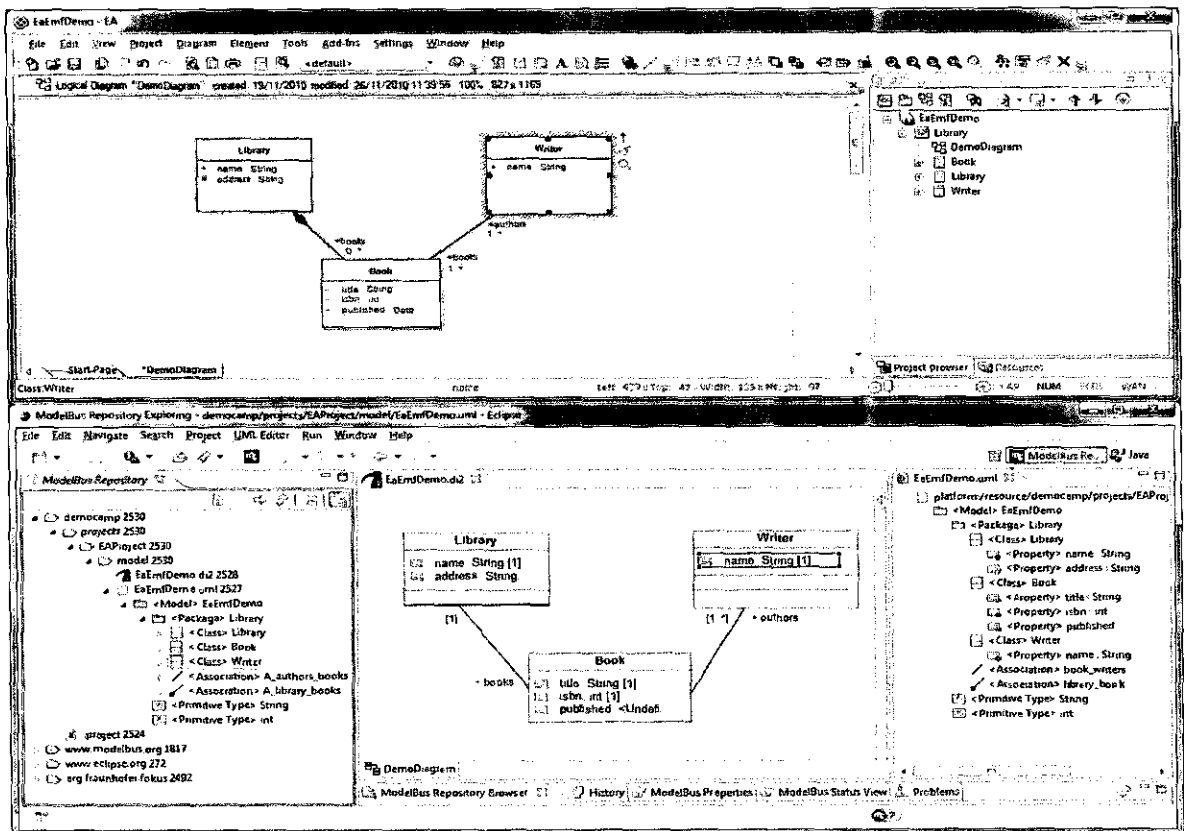


Figura 20 – Tela do software Enterprise Architect.

11.3. IBM Rational DOORS

O software DOORS permite que a comunicação entre os requisitos e os modelos seja eficiente, colaborando para a validação dos requisitos com os stakeholders. E assim possibilita a entrega de produtos e sistemas com maior qualidade e em menor tempo.

Esta ferramenta também é capaz de capturar, relacionar, analisar e atualizar os requisitos, ajudando as organizações responsáveis pelo desenvolvimento de produtos complexos para maximizar a satisfação dos stakeholders.

O DOORS importa e exporta XMI e XML e gera código (C, C++, JAVA) e documentos automaticamente.

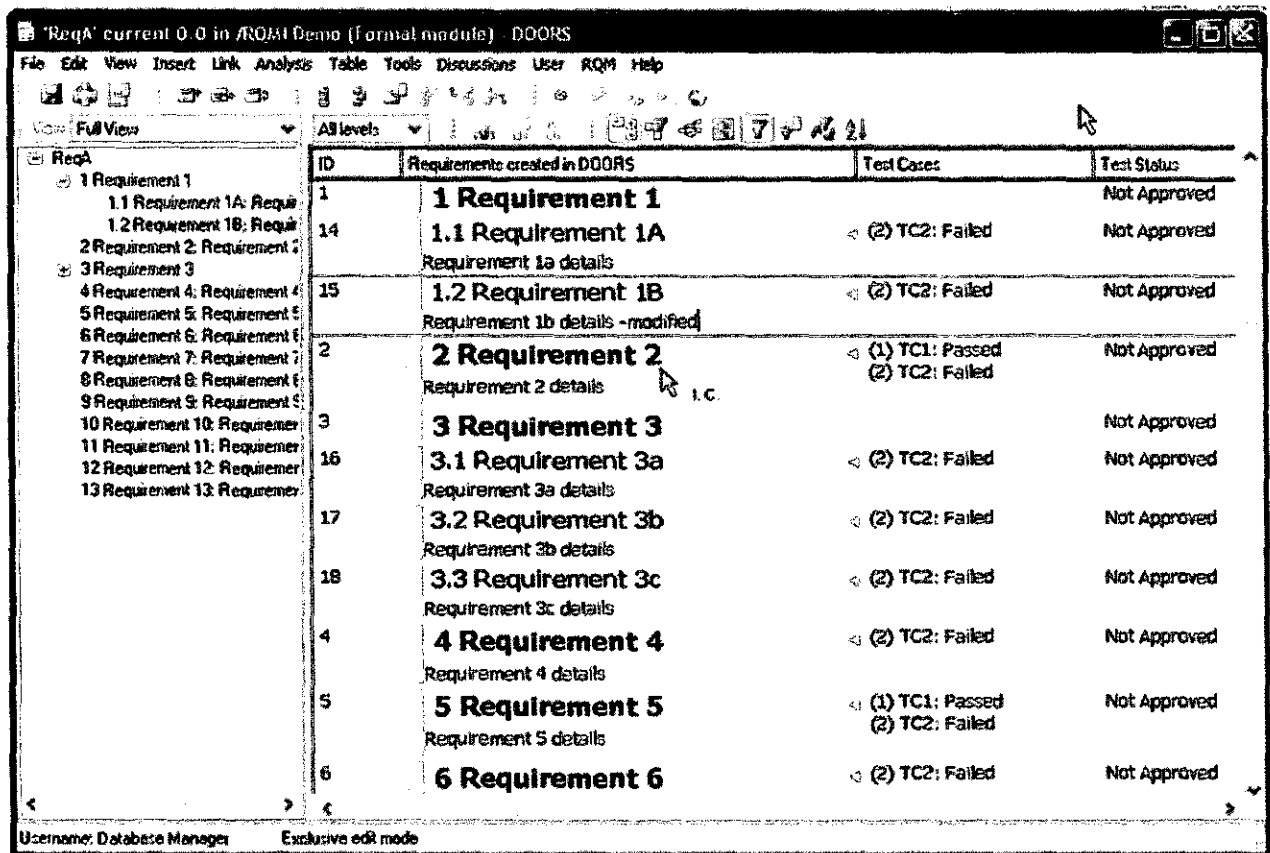


Figura 21 – Tela do software DOORS

11.4. IBM Rational Rhapsody

A ferramenta Rhapsody possui varias funcionalidades, as quais atendem as etapas do processo de Engenharia de Sistemas, pois suporta análise, design, desenvolvimento, testes e simulações. Para realizar simulações é integrada com o MatLAB, o qual extrai as informações do diagrama de parâmetros para realizar os testes.

Modela em diversas linguagens gráficas, não somente na linguagem SysML. Suporta XMI e XML como forma de partilhar dados e modelos bem como o standard AP233 (em desenvolvimento) E gera código (C, C++, JAVA) e documentos automaticamente.

O Rhapsody possibilita também a rastreabilidade e consistência entre os diagramas de requisitos e os requisitos armazenados no banco de dados do IBM Rational Doors, através da integração desses programas, os quais são da mesma empresa a IBM.

No site da IBM é possível, após fazer o cadastro, realizar o download da licença TRIAL por trinta dias.
(http://www.ibm.com/developerworks/downloads/r/rhapsodydeveloper/?S_CMP=rnav&ca=qapromo&s0rat-b0rat-10rat-d0rat-n021-o0F129911Y76724U66-g0usen)

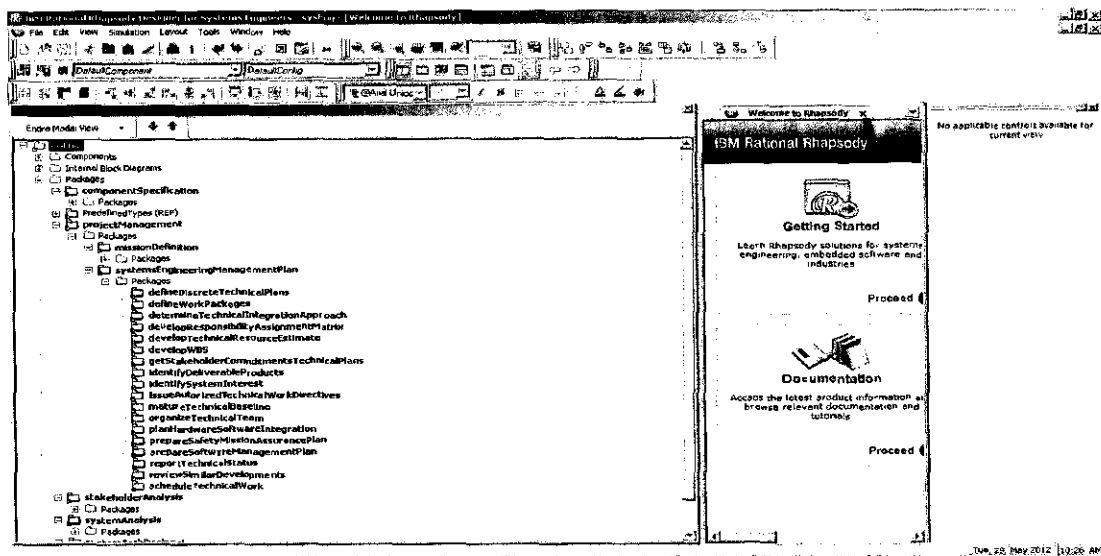


Figura 22 - Tela do software Rhapsody

11.5. MagicDraw - SysML Plug-in

O MagicDraw em conjunto com o SysML Plug-in permite a modelagem de requisitos, através do armazenamento desses em um banco de dados interno ao programa, em outras palavras, os elementos do modelo serão ligados aos requisitos armazenados, para que testes e simulações validem os requisitos. Esse diferencial auxilia o desenvolvimento do sistema complexo, detalhando o projeto, por meio de análises e validações automaticamente.

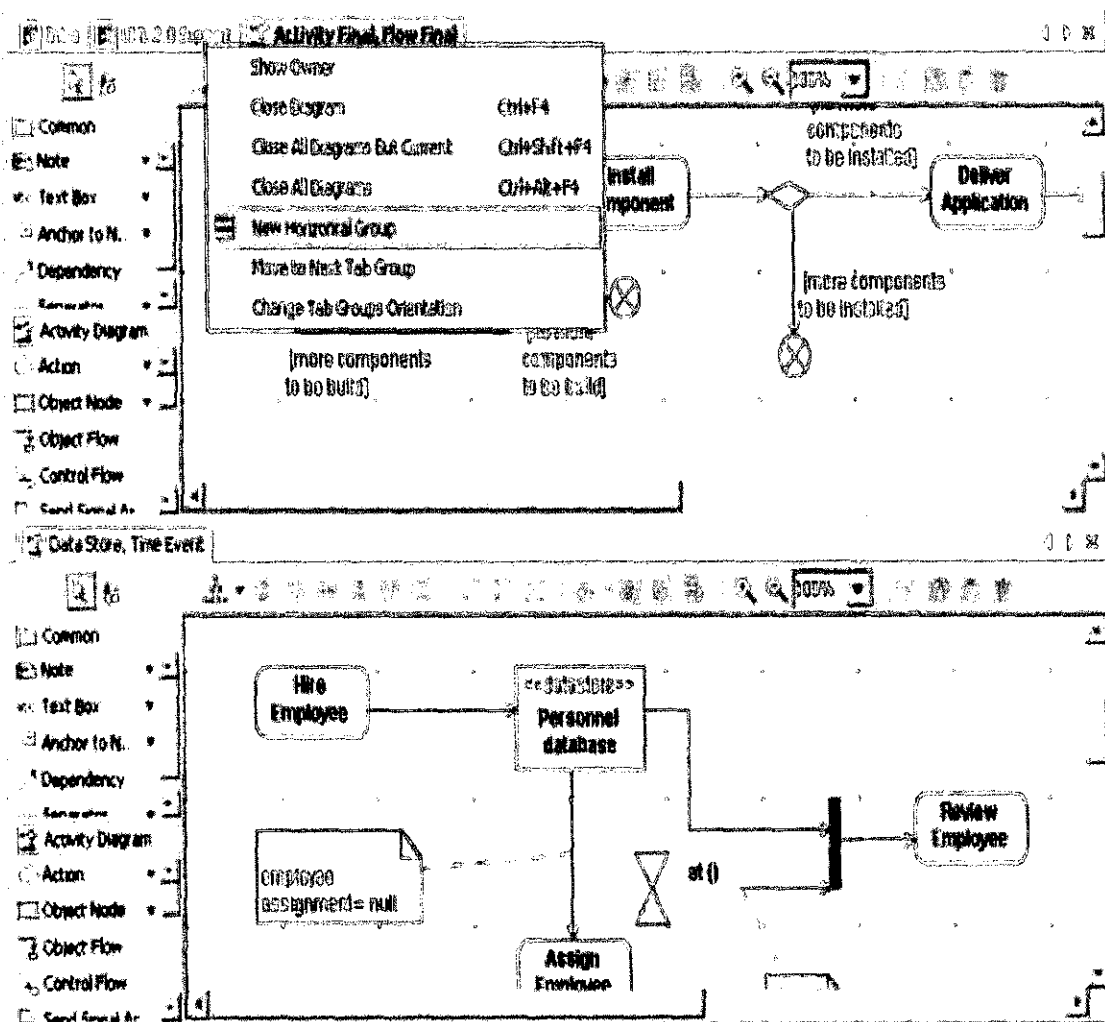


Figura 23 – Tela do software MagicDraw

11.6. Modelo System Architecture

O programa de modelagem gráfica Modelio System Architecture (<http://www.modeliosoft.com/en/download/system-architect-solution.html>) possui integração com as mais importantes linguagens de modelagem padrão atualmente, como, BPMN, UML, MDA, SysML, TOGAF.

A interface gráfica é pratica e fácil de ser manipulada pelo usuário.

No site é possível encontrar a licença TRIAL (<http://www.modeliosoft.com/en/download/free-downloads.html>) dessa ferramenta, que vale durante dez dias. Porém esta versão não oferece o diagrama de requisitos para ser modelado

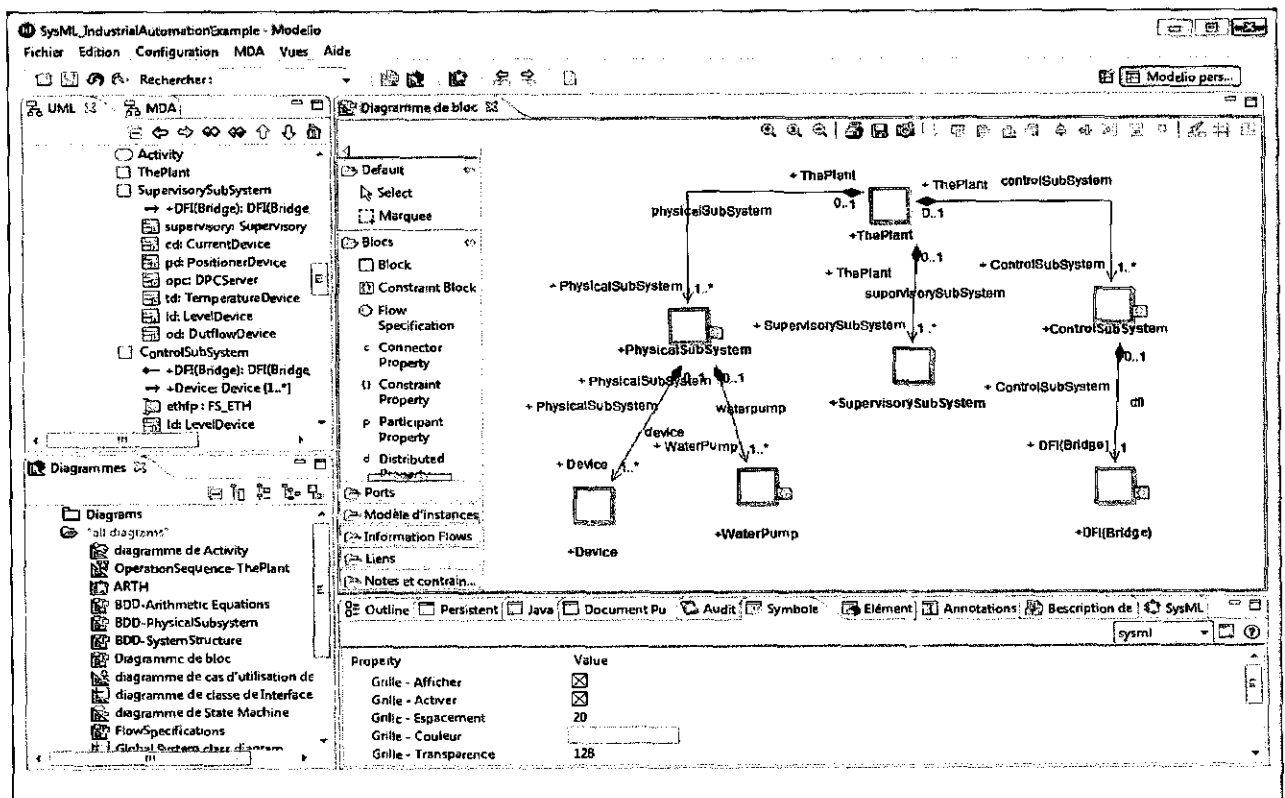


Figura 24 – Tela do software Modelio System Architecture

11.7. Papyrus UML for SysML

O Papyrus UML for SysML é uma ferramenta open-source (<http://www.papyrusuml.org/scripts/home/publigen/content/templates/show.asp?P=128&L=EN>), a qual apresenta limitações em relações as outras ferramentas computacionais apresentadas anteriormente neste trabalho

O plug-in proporciona ao usuário modelar na linguagem SysML, implementado a maioria dos estereótipos e propriedades relacionadas a essa linguagem gráfica, segundo a especificação SysML V1.1 (08-11-02)

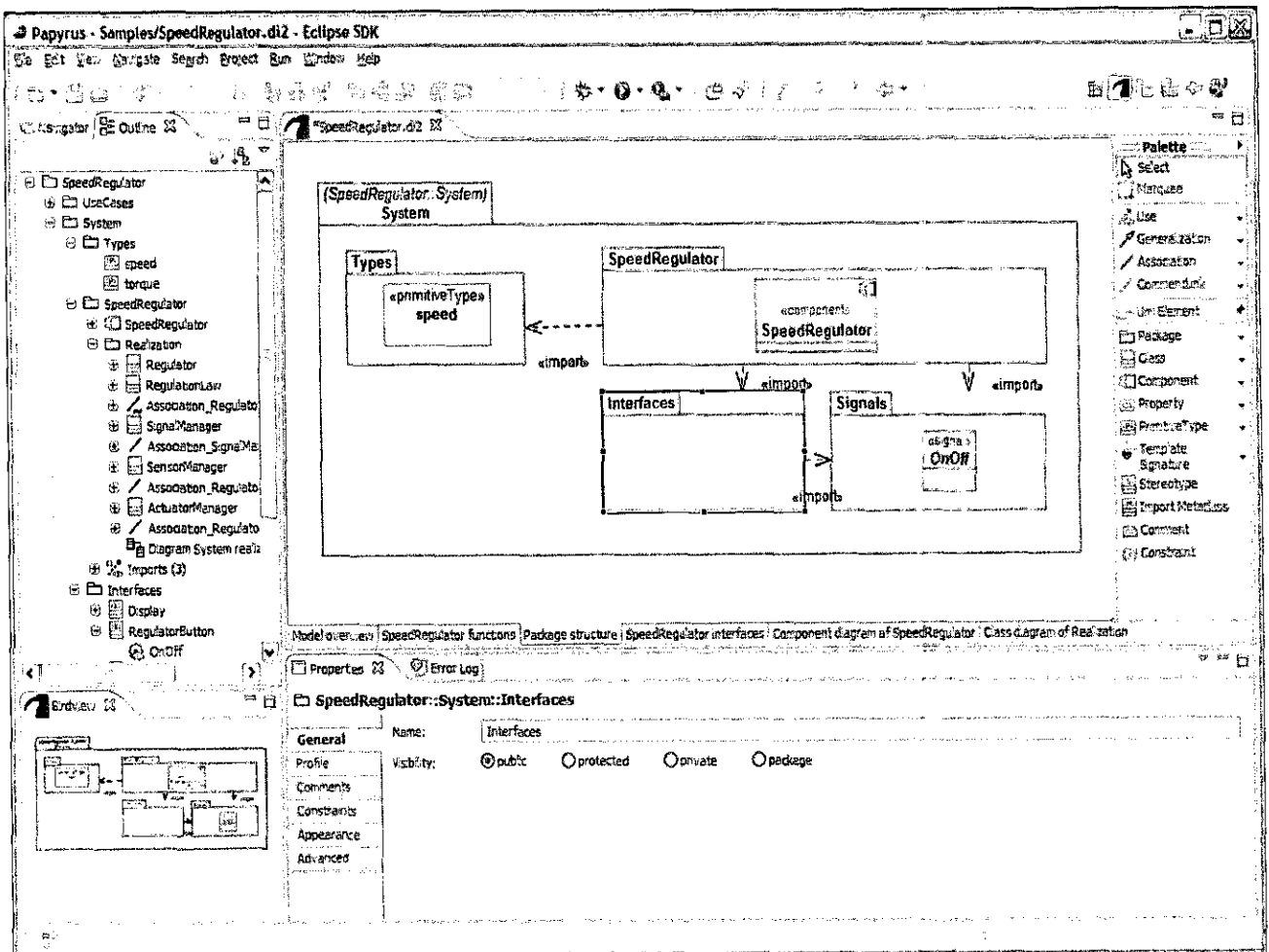


Figura 25 – Tela do software Papyrus.

11.8. TopCased- SysML

O aplicativo TopCased- SysML é open source (free/libre/open-source) A plataforma que suporta este aplicativo e o Eclipse

Primeiramente foi projetado para atender tanto a análise crítica de softwares com de hardwares. Mas ainda não possui todas as funções necessárias para desempenhar tal tarefa, atualmente encontra-se em desenvolvimento.

Igualmente a ferramenta Papyrus também apresenta restrições no desenvolvimento dos modelos.

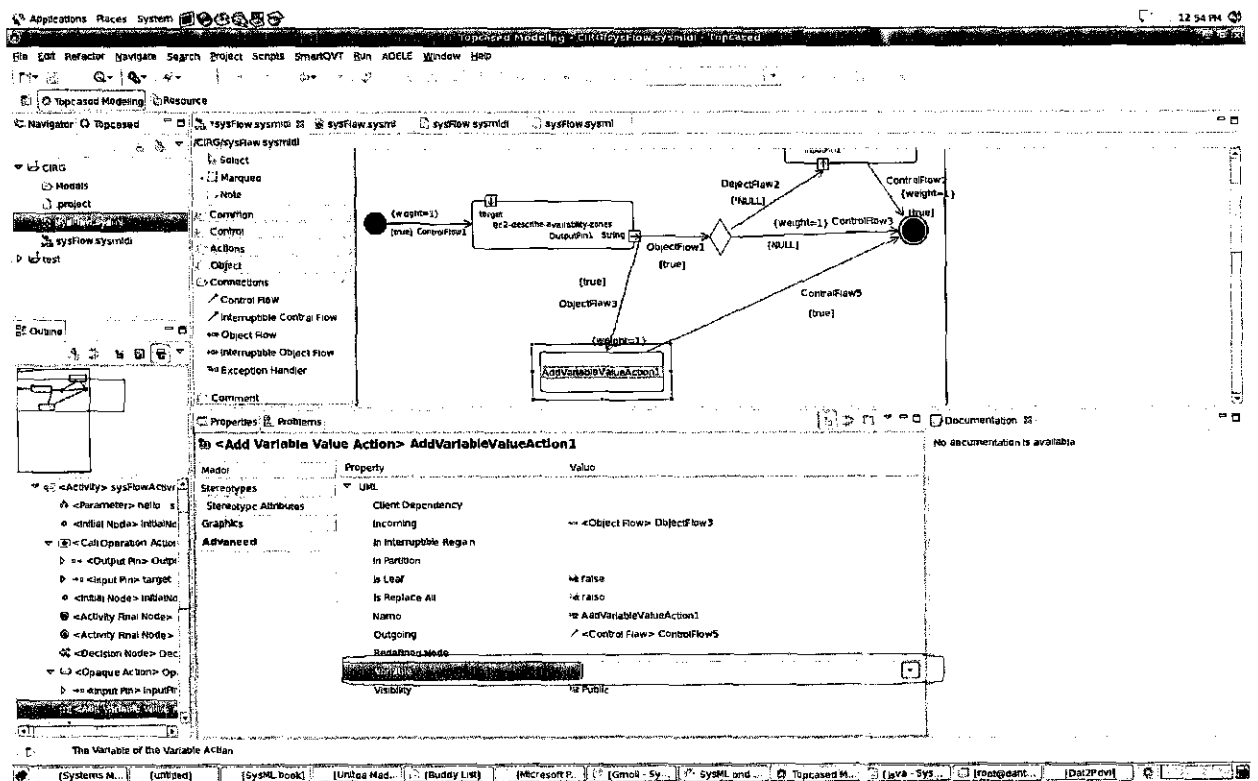


Figura 26 – Tela do software TopCased

12. RESULTADO

A fim de mapear as etapas do método do Professor Dr. Geilson Loureiro com a linguagem SysML, realizou-se a modelagem de um produto da área espacial, a qual não abrangeu a organização do produto.

O modelo desenvolvido é a primeira versão do template que se pretende criar, é o primeiro passo para uma série de adaptações e criações de novos estereótipos tanto de diagramas como de relacionamentos para modelar um produto e sua respectiva organização como é esperado, no método do professor Dr. Geilson Loureiro utilizando diagramas padrão da linguagem SysML.

A engenharia de sistemas, abordagem usada para modelar o template, ajuda a antecipar os problemas que muitas vezes depois de já implementado afetará o produto, sendo que setenta por cento do custo do ciclo de vida de um produto é gasto no início do ciclo.

Por isso é importante pensar em cada nível do sistema. É fundamental que este pensamento seja o mais abstrato possível, ou seja, que seja pensado de maneira funcional, por exemplo, muitas vezes o cliente pede um manômetro, mas o que ele precisa realmente é de um simples objeto que meça a pressão.

O pensamento sistêmico ajuda na análise funcional, uma vez que o todo sempre depende das relações entre as partes, sendo a parte algo indivisível.

Algumas telas do template modelado serão apresentadas a seguir. É importante ressaltar que o resultado apresentado neste trabalho abrangeu a modelagem do segmento da estação terrena do sistema CANSAT.

O modelo abaixo ilustra o sistema CANSAT e seus subsistemas, com trata- se da parte estrutural do sistema foi usado um diagrama de definição de blocos da linguagem SysML para representar as partes desse sistema espacial.

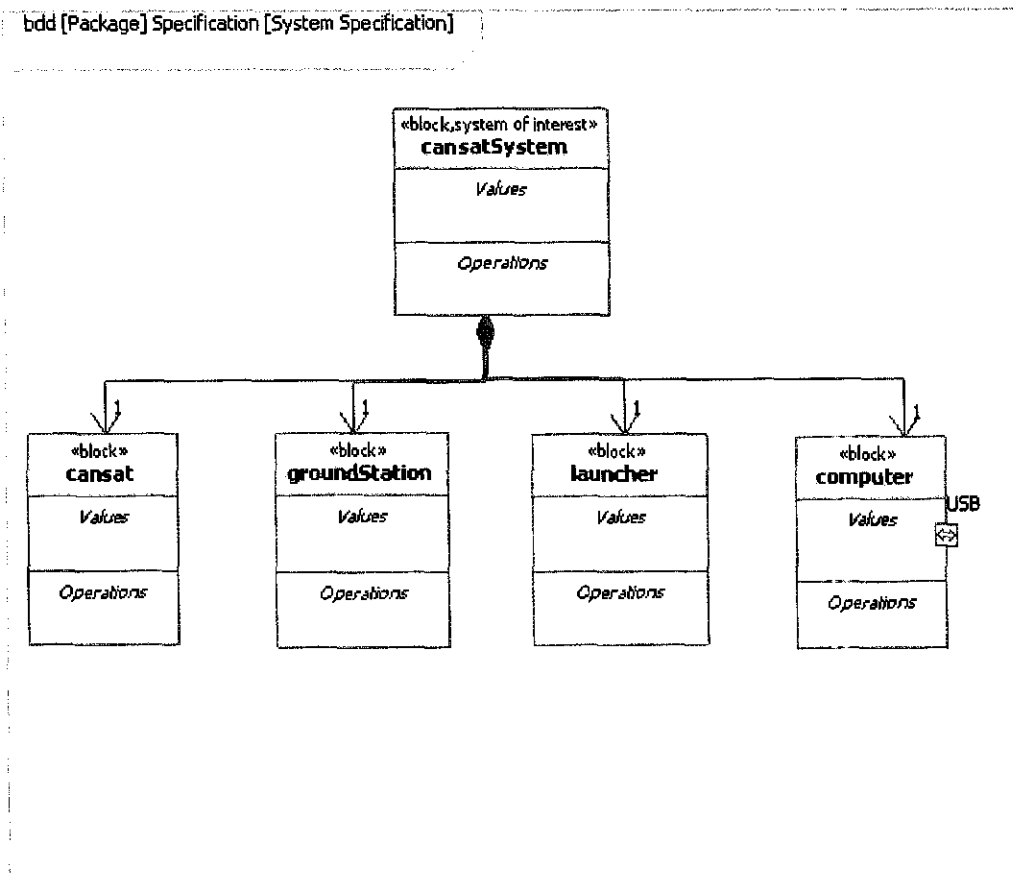


Figura 27 – Sistema CANSAT

No proximo diagrama foi dado um zoom no bloco da estação terrena, mostrando os subsistemas pertinentes a estação terrena. Também foi usado um diagrama de bloco, pois esse permite que seja exposto as partes e relacionamentos de subsistema.

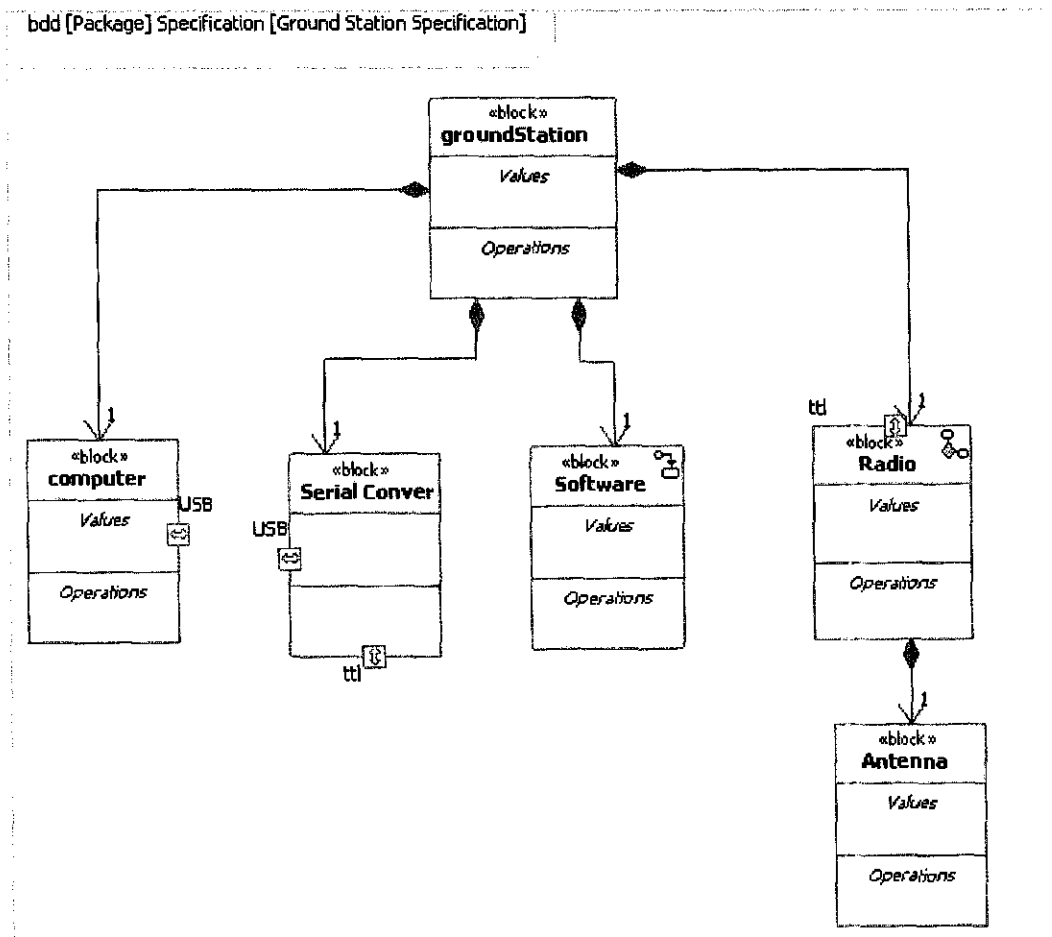


Figura 28 – Subsistemas da estação terrena

Na figura abaixo, a qual foi extraída do documento do projeto CANSAT observa-se os mesmos elementos modelos com diagrama de blocos. Porém o diagrama contém uma grande vantagem e relação a simples figura, geração de código automaticamente, através da ferramenta Rhapsody. A área técnica desta disciplina, através do uso da A linguagem SysML, vem obtendo ganhos significativos ao substituir, na documentação de sistemas complexos, páginas descritivas por modelos estruturados, cuja utilização consiste em um dos objetivos deste trabalho.

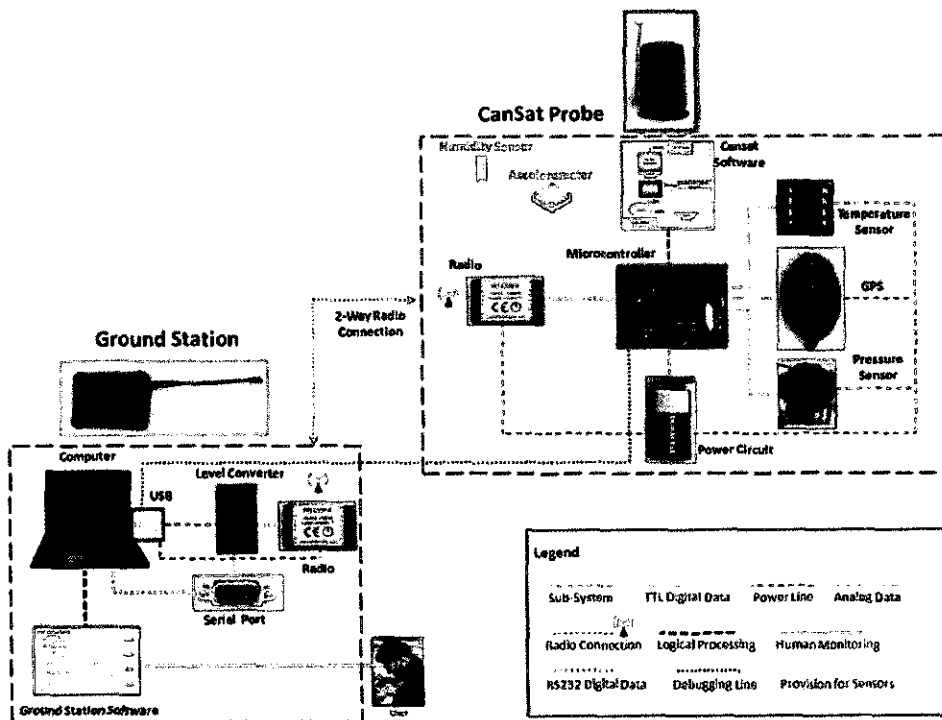


Figura 29 – Estação Terrena.

No diagrama a seguir há um nível de detalhamento maior do que os diagramas anteriores, demonstrando que é importante usar uma ferramenta computacional para modelar sistemas complexos, pois o desdobramento de um sistema pode ser muito grande.

O nível de detalhamento pode ser bem maior que do que o apresentado neste trabalho chegando a especificar os componentes, ou seja, mostrar todo o conjunto de interfaces do sistema.

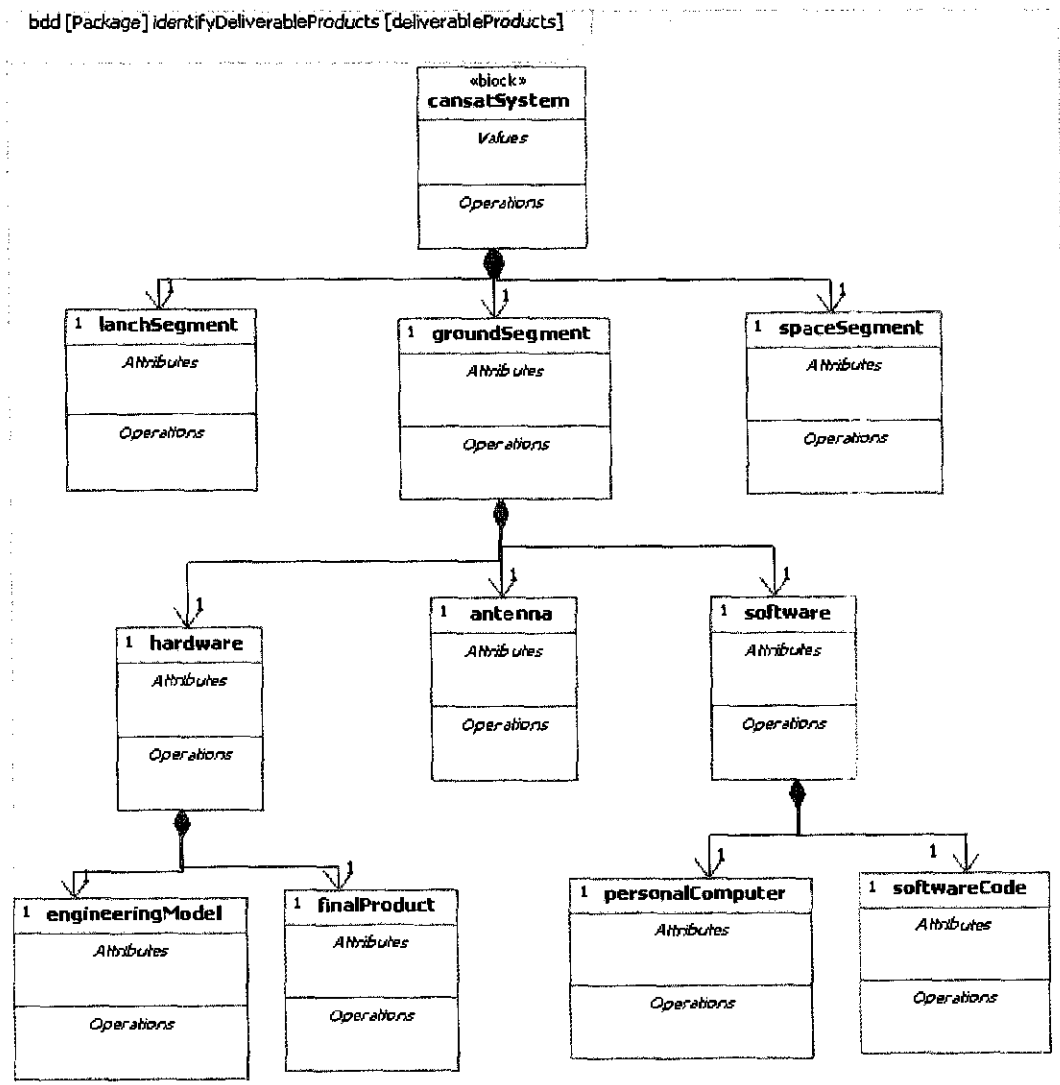


Figura 30- Detalhamento dos subsistemas da estação terrena

Com a modelagem apresentada é possível visualizar o todo de maneira organizada, facilitando a análise do sistema pois sintetizar é um processo lento e difícil, sendo que construir modelos é fazer uma síntese da realidade, em razão disso o problema deve ser bem estruturado no início do projeto para assegurar que a modelagem seja fiel à realidade do produto.

O grande diferencial do UML para o SysML é como as ações são modificadas durante o processo, ou seja, a saída pode ser diferente da entrada.

Pode-se criar segundo a norma da própria linguagem SysML novos diagramas, novos estereótipos de elementos e relações, que faz dessa linguagem não ser prescritiva.

Desta forma, para identificar os stakeholders, neste trabalho foi criado o diagrama de contexto, o qual também serve para representar tanto o contexto funcional com o contexto físico, este novo diagrama foi elaborado a partir do diagrama de blocos.

Foi criado esse diagrama dada a importância do stakeholder no processo de engenharia de sistemas. Por isso é importante valorizar a informação que o stakeholder oferece, as respostas que eles dão nas entrevistas, e por isso é importante perguntar várias vezes sobre o mesmo assunto sempre tentando pensar funcionalmente.

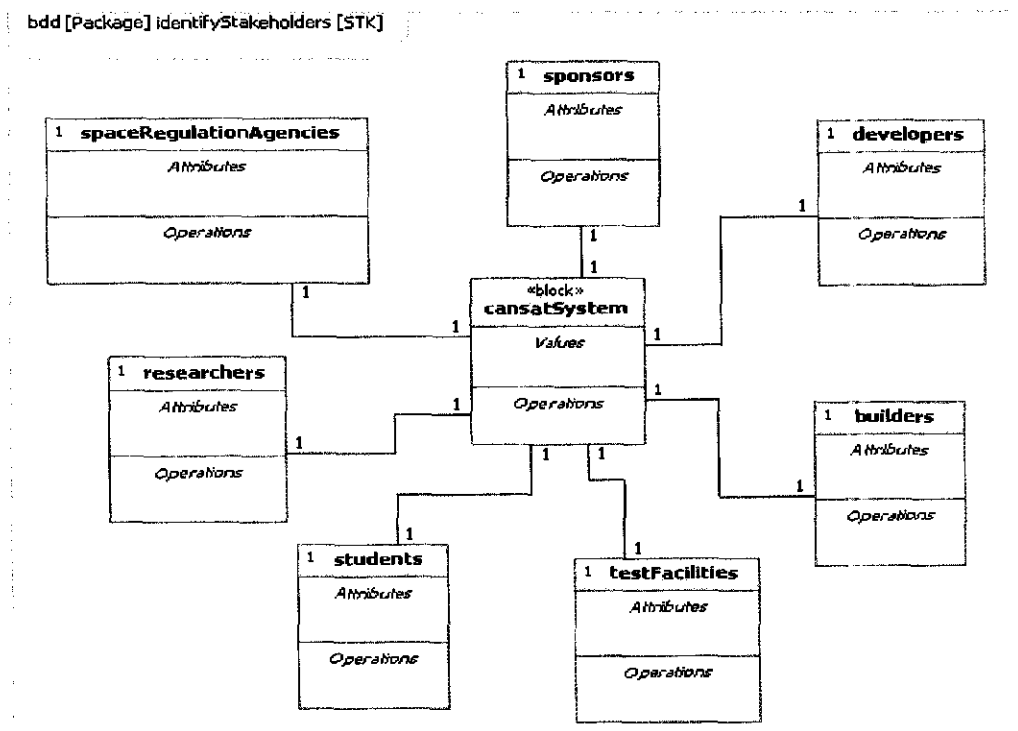


Figura 31- Identificação de stakeholders

DIAGRAMA DE CONTEXTO:

- a) Apresenta o conjunto de elementos relacionados para um determinado propósito, como especificar uma operação
- b) Define o ambiente em que o sistema vai pertencer.
- c) Demonstra as características do sistema.
- d) É composto por fluxos de dados que mostram as interfaces entre o sistema e as entidades externas.
- e) Permitem identificar os limites dos processos, as áreas envolvidas com o processo e os relacionamentos com outros processos e elementos externos à empresa (ex.: clientes, fornecedores).
- f) Representa o objeto do estudo, o projeto, e sua relação ao ambiente.
- g) Representa todo o sistema como um único processo.

A seguir será apresentado o ciclo de vida do subsistema Estação Terrena, a modelagem segue o pensamento funcional e hierárquico, como poderá ser comprovados nos diagramas de alguns cenários do ciclo de vida do sistema espacial

Foram modelados em um diagrama de atividade, pois expressa o comportamento do produto espacial, uma vêz que, o comportamento denota dinâmica do elemento, como por exemplo, uma operação no sistema.

A modelagem dos cenários são importantes, porque é através dessa modelagem que serão derivados os requisitos de sistema

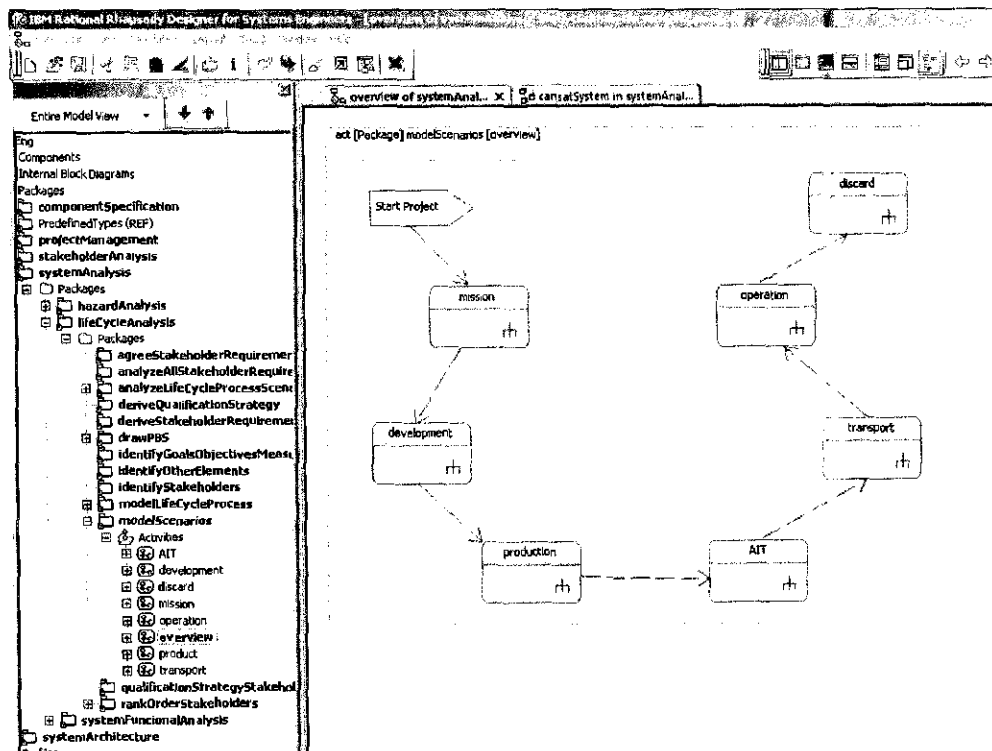


Figura 32- Ciclo de vida do sistema

Os fluxos de material energia e informação são os requisitos de interfaces , e com a modelagem é facil de capturar esses requisitos.

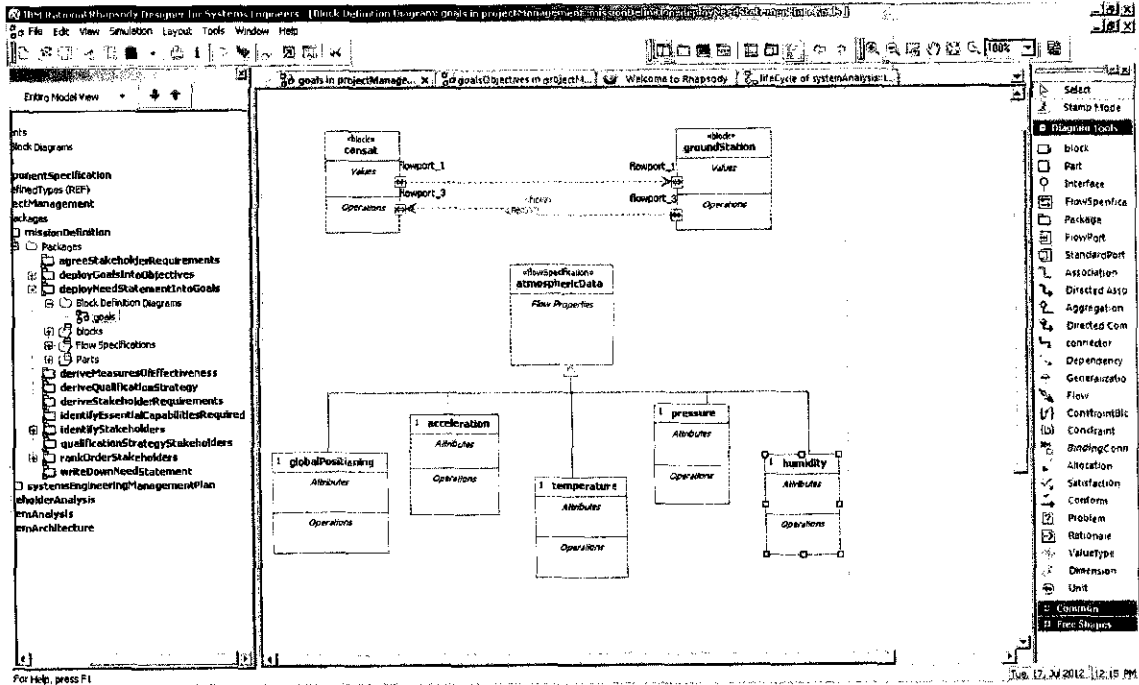


Figura 33- Fluxo de material, energia e informação.

Cenário de AIT:

- a) Descrição das características gerais do produto espacial a ser desenvolvido
- b) Identificação das interfaces necessárias para realização dos testes.
- c) Seleção dos parâmetros do sistema e subsistemas, com base na análise funcional e arquitetura de sistemas

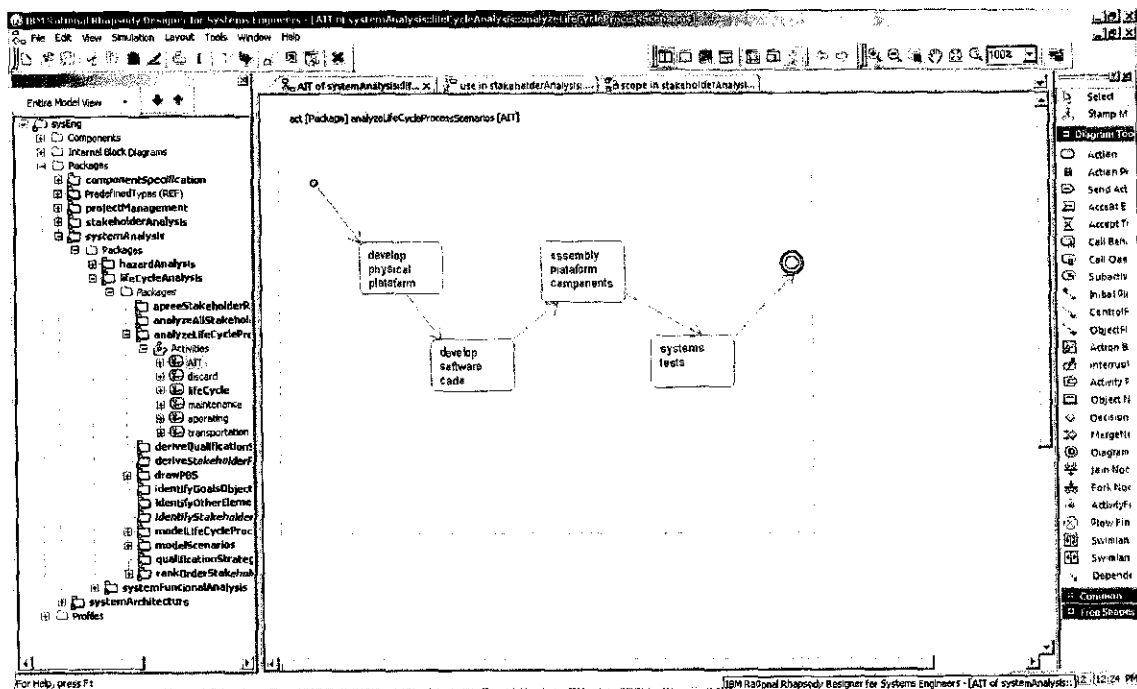


Figura 34 – Cenário AIT.

Cenário Manutenção:

- a) Garantia através de controles e monitoramento da qualidade da transmissão de sinal entre o satélite e a estação terrena do sistema CANSAT.
- b) Acompanhamento do processo de operação a fim atingir dos objetivos da operação.

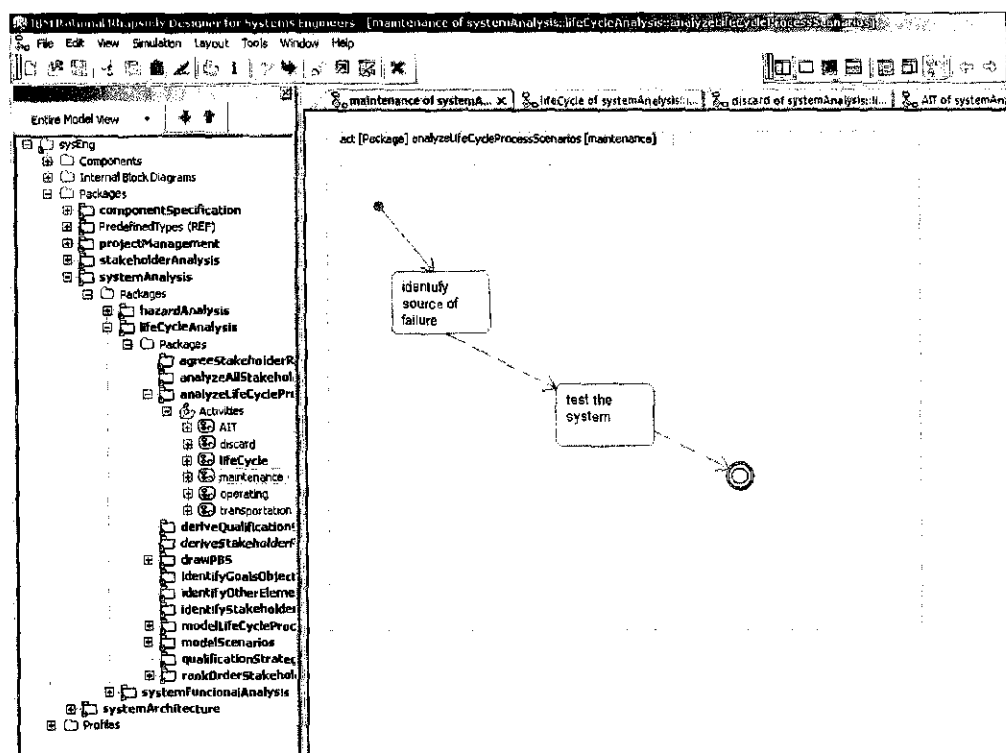


Figura 35 - Cenário Manutenção.

Cenário Descarte:

- a) A partir dos resultados obtidos após a operação do sistema, verificar qual será o destino adequado dos hardwares e softwares.

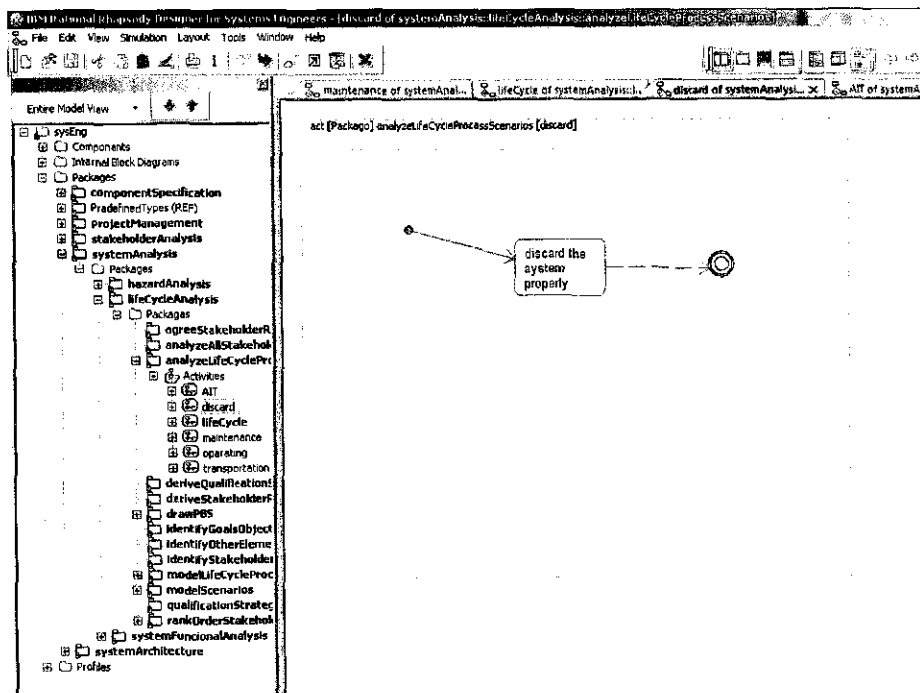


Figura 36 – Cenário Descarte

Cenário Produção:

- a) Com base nos objetivos, requisitos e especificações do sistema será realizada a construção do hardware e a implementação do software.

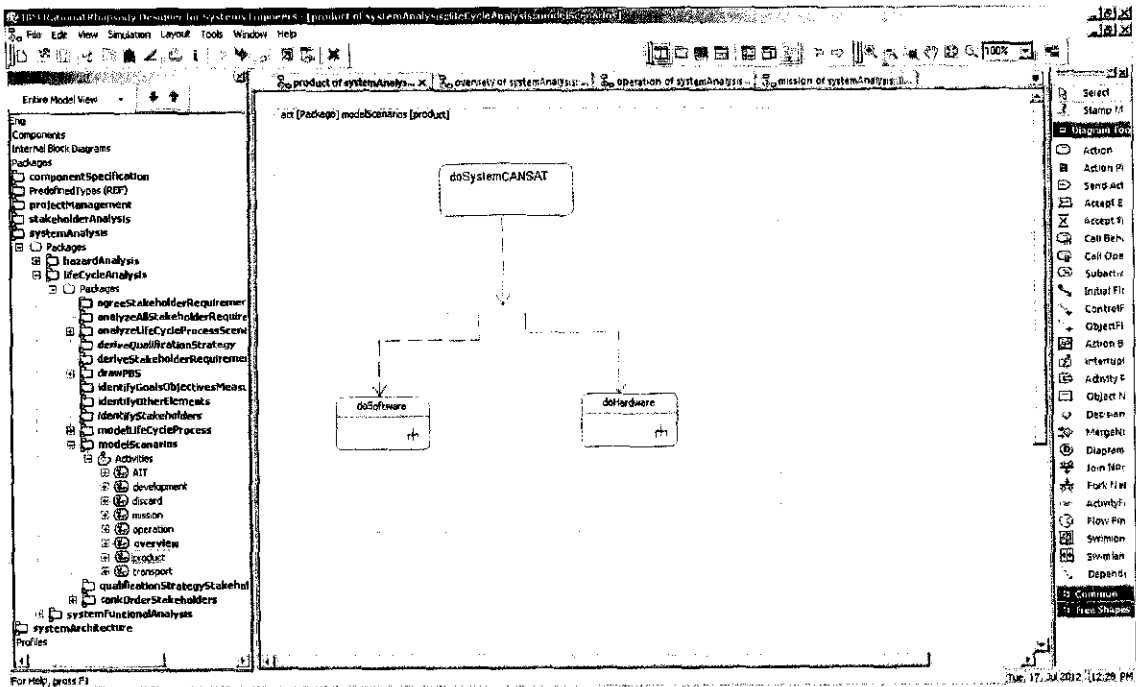


Figura 37 – Cenário Produção

12. TRABALHO FUTURO

Como trabalho futuro espera-se avançar na modelagem da primeira versão do template, deixando o mais próximo a realidade estudada permitindo a visualização do todo.

E para alcançar esse objetivo, o próximo trabalho visa responder Qual o melhor diagrama de SysML que pode ser usado em cada etapa do método do Professor Dr. Geilson Loureiro?

Uma vez que, a linguagem SysML apresenta aspectos a serem incrementados, por ser uma linguagem nova.

14. CONSIDERAÇÕES FINAIS

Este trabalho buscou realizar a modelagem utilizando a linguagem SysML, de um produto espacial seguindo o método de engenharia simultânea de sistemas do Professor Dr. Geilson Loureiro, como este foi modelado primeiramente em análise estruturada, foi preciso um estudo constante para realizar adaptações na modelagem e criações de novos diagramas e relacionamentos, já que a linguagem SysML permite isso.

A engenharia de sistemas tem se tornado cada vez mais requisitada na comunidade científica devido à evolução das tecnologias de implementação e ao aumento da complexidade dos próprios sistemas. Logo, novos métodos têm sido desenvolvidos na tentativa de diminuir custos, riscos e tempo gastos nas etapas de modelagem e desenvolvimento do projeto.

Os engenheiros de sistemas devem ser capazes de traduzir a necessidade dos stakeholder em requisitos de stakeholder e também em requisitos de sistemas.

Para ter uma boa modelagem é preciso uma boa ferramenta computacional. Apenas com recurso computacional é possível fazer controles de versões, realizar análises estruturais ou comportamentais do modelo, o software possibilita também o rastreamento de requisitos. Sendo fundamental a estruturação do modelo em pacotes antes da criação dos diagramas.

É importante ter a rastreabilidade dos requisitos durante o processo de Engenharia de Sistemas para que qualquer mudança no projeto seja comunicada aos envolvidos através de documentos que demonstrem qual a parte de sistema que será afetada direta, indiretamente ou não com as atualizações feitas nele.

A linguagem SysML trata-se de um grande auxiliar no desenvolvimento de produtos complexos, pois apresenta alguns benefícios muito valiosos para um projeto como: a) Alteração no projeto com menor custo. b) Facilidade de documentação do sistema. c) Reutilização de módulos desenvolvidos. d) Facilidade e rapidez na detecção de erros, que podem ser corrigidos nas primeiras etapas do projeto. e) Melhor entendimento do sistema por todos os envolvidos no projeto. f) Redução do tempo gasto entre as etapas de concepção, validação e teste do sistema.

A linguagem SysML só se torna completa com a criação de estereótipos, e possui grande potencial ainda a ser desenvolvido.

A linguagem SysML possui elementos para modelar o método de engenharia simultânea de sistemas do Professor Dr. Geilson Loureiro, pois trata-se de uma linguagem não prescritiva. Com o objetivo de mapear as fases do método usando a linguagem SysML, realizou-se a modelagem do sistema CANSAT. O modelo obtido serve como exemplo de implementação do método.

A seguir são apresentados exemplos de projetos relacionados à engenharia de sistemas, nos quais a linguagem SysML contribuiria de forma positiva, como já foi mencionado anteriormente neste trabalho:

Projeto SACI (Satélite Avançado Comunicações Interdisciplinares)

A motivação deste projeto foi à necessidade de viabilizar um sistema avançado de tecnologia educacional de alcance nacional para o Brasil.

A documentação deste projeto encontra-se na biblioteca central do INPE em São José dos Campos. São vários relatórios em três idiomas diferentes (português, inglês e espanhol), o que já indica uma falta de padrão na documentação em relação ao uso da linguagem A linguagem SysML. Além disso, os documentos estão desatualizados e desgastados pelo tempo.

As informações que eles contêm são importantíssimas, pois se trata do projeto mais revolucionário em educação usando a tecnologia, já idealizado no INPE pelo seu diretor da época Dr. Fernando Mendonça. É evidente que na época não existia a linguagem A linguagem SysML. Mas, se esse projeto tivesse sido realizado nos dias de hoje o seu ganho no projeto seria muito maior.

LSIS (Laboratório de Engenharia de Sistemas)

Este tem por objetivo integrar ferramentas de análise de stakeholders, engenharia de requisitos, modelagem conceitual de sistemas, modelagem física de sistemas, projeto CAD de produtos, ferramentas de verificação e validação de sistemas. Essas ferramentas darão suporte a um processo de engenharia simultânea de sistemas que antecipa, para as etapas iniciais do desenvolvimento do sistema, os requisitos dos processos do ciclo de vida do sistema. Esse processo, apoiado pelas ferramentas descritas, também tem o potencial de ser utilizadas para o fornecimento de serviços para clientes do LIT, tais como a engenharia do INPE, a AEB, empresas fornecedoras do programa espacial e de outros programas complexos, bem como pelos clientes de ensaios do LIT.

O template elaborado na linguagem SysML neste trabalho ajudará o desenvolvimento deste projeto, que também segue o método do Professor Dr. Geilson Loureiro.

ENGESIS (Engenharia, Gestão e Simulação de Sistemas Espaciais)

Este grupo foi criado em 2010, a partir da união dos esforços realizados pelos pesquisadores e Professor de grupos análogos já existentes nas instituições participantes, objetivando a realização de estudos em temas relacionados com o ciclo de vida completo (o projeto, a construção, a operação, o monitoramento, o controle, a análise e a melhoria contínua) do desenvolvimento de modelos, produtos e processos da área espacial ou sistemas semelhantes de alta tecnologia.

As principais áreas do conhecimento envolvidas nos estudos realizados pelo grupo ENGENESIS são: a Engenharia da Produção, a Engenharia e Gestão de Desenvolvimento de Software, Gestão de Projetos, Gestão por Processos e Modelagem e Simulação de Sistemas. A característica principal que o distingue é a utilização de uma abordagem interdisciplinar, denominada Ciência e Tecnologia de Processos, na qual os conceitos, métodos e ferramentas dessas áreas são integrados e aplicados para a solução de problemas da área espacial.

Com a linguagem SysML é possível modelar e documentar todos os processos propostos por este grupo. O que garante uma melhor compreensão do todo, visão sistêmica, por todos os integrantes do grupo.

Como conclusão tem se que a linguagem de SysML ainda precisa de adaptações. No entanto, ela já possui as recursos suficientes para uma implementação do método, pois a linguagem SysML permite ao Profissional que sejam criados novos tipos de diagramas, já que se trata de uma linguagem não prescritiva, ao contrário da Análise Estruturada que é prescritiva.

Não pode dizer que o trabalho acaba por aqui, pois a própria linguagem SysML ainda tem muito a melhorar e contribuir para a engenharia simultânea de sistemas.

15. REFERÊNCIAS BIBLIOGRÁFICAS

3SL. Cradle user manual, 2010. Disponível em: < <http://www.threesl.com/>> Acesso em: ago. 2011.

BALMELLI. **An overview of the systems modeling language for products and systems development.** IBM Technical Report. 2006.

FRIEDENTHAL, S.; MOORE, A.; STEINER, R. **A practical guide to SysML: the systems modeling language.** The MK/OMG Press. Elsevier. Amsterdam, 2009.

FULINDI, J. **Auxílio computacional a um processo de engenharia simultânea de sistemas espaciais.** 2011. 273p. (sid.inpe.br/mtc-m19/2011/04.04.12.08-TDI).
Dissertação (Mestrado em Engenharia e Gerenciamento de Sistemas Espaciais) - Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2011. Disponível em: <<http://urlib.net/8JMKD3MGP7W/39EQGDP>>. Acesso em: ago. 2011.

HALL, Arthur D. **A Methodology for systems engineering.** Van Nostrand. New York, 1962.

HOLT, J.; PERRY, S. **SysML for systems engineering.** IET. Stevenage. 2008.

IBM. DOORS user manual, 2010.
Disponível em: < <http://www-01.ibm.com/software/awdtools/doors/>>
Acesso em: ago. 2011.

IBM. Rhapsody 7.0 user manual, 2010.
Disponível em: < <http://www-01.ibm.com/software/awdtools/rhapsody/>>
Acesso em: ago. 2011.

INCOSE San Francisco Bay Area Chapter. **Systems engineering handbook: a how to guide for all systems engineers.** San Francisco, 1998.

JURAN, J.M.; GODFREY, A.B. **Jura's quality handbook**. MCGRAW- HILL. United States of America, 1998.

LOUREIRO, G. Engenharia de sistemas. 2011. São José dos Campos. ITA. Notas de aula do curso de graduação de Engenharia Aeroespacial. ITA, 2011.

LOUREIRO, G. Engenharia de sistemas. 2012. São José dos Campos. ITA. Notas de aula do curso de pos-graduação de Modelagem de Sistemas. ITA, 2012.

LOUREIRO, G. Engenharia de sistemas. 2012. São José dos Campos. ITA. Notas de aula do curso de graduação de Engenharia Aeroespacial. ITA, 2012.

LOUREIRO, G. **A systems engineering and concurrent engineering framework for the integrated development of complex products**. 1991. (PhD Thesis) - Loughborough University, Loughborough, UK. 1999.

LOUREIRO, G. 2010. Lessons learned in 12 years of space systems concurrent engineering. In: INTERNATIONAL ASTRONAUTICAL CONGRESS, 61. 2010. **Proceedings** Prague, CZ. 2010.

MEDINA, M.; FERTIG, C. **Algoritmos e programação: teoria e prática**. Novatec. São Paulo, 2005.

OMG SYMML, 2012. Disponível em: < <http://www.omgsysml.org/>> Acesso em: jul. 2012.

PRASAD, B. **Concurrent engineering fundamentals: integrated product and process organization**. PTR Prentice Hall. New Jersey, 1996

SAGE, Andrew P. **Systems Engineering**. John Wiley & Sons, Inc. New York, 1992.

WEILKIENS, Tim. **Systems engineering with SysML/UML: modeling, analysis, design**. The MK/OMG Press. Elsevier. Amsterdam, 2009.