

INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO

MESTRADO EM ENGENHARIA INFORMÁTICA

isep



ANÁLISE DA CONCEPÇÃO DE SISTEMAS DE SISTEMAS COM A NOTAÇÃO SysML

Tiago André Queiroz Soares da Costa

Maio de 2009





***Análise da Concepção de Sistemas de Sistemas
com a notação SysML***

Departamento de Engenharia Informática

Tiago André Queiroz Soares da Costa

***Dissertação para obtenção do Grau de Mestre em
Engenharia Informática***

Área de especialização em Arquitectura de Sistemas e Redes

Orientador: Doutor Alberto A. C. Sampaio

Co-orientador: Doutor Gustavo R. C. Alves

Júri:

Presidente: Doutora Maria de Fátima Coutinho Rodrigues, Professora
Coordenadora, Instituto Superior de Engenharia do Porto/Departamento de
Engenharia Informática

Vogais: Doutor Pedro Miguel Gonzalez de Abreu Ribeiro, Professor Auxiliar,
Universidade do Minho/Departamento de Sistemas de Informação
Doutor Alberto António Chalupa Sampaio, Professor Adjunto, Instituto Superior
de Engenharia do Porto/Departamento de Engenharia Informática

Porto, Maio de 2009

Aos meus pais

Agradecimentos

A realização desta Tese não demonstra apenas o trabalho de uma pessoa, mas sim a união de determinadas circunstâncias que tornaram possível a criação de um trabalho criterioso e sustentável, com base no rigor incutido por todos os elementos do Instituto Superior de Engenharia do Porto.

No entanto, gostaria de deixar o meu primeiro agradecimento aos meus pais. A sua incansável dedicação, para me darem as “ferramentas” que possam conceder-me um futuro melhor, juntamente com o seu carinho e amor, fez de mim aquilo que sou hoje. Os valores e a disciplina que me foi ensinada desde pequeno ajudaram-me neste longo percurso, por vezes tortuoso e difícil, mas sempre recompensador. Além disso, a vossa confiança e a persistência fizeram com que nunca baixasse os braços e encarasse os desafios sem recear o resultado. Desta forma, para eles vai o meu maior agradecimento, pois sem eles, hoje não seria quem sou. Pai e Mãe, um bem-haja a vocês.

À minha namorada, tenho de lhe agradecer por ter compreendido que as várias horas e dias em que não pudemos estar juntos, porque era preciso “trabalhar na Tese”, sempre recompensam. A força para continuar, sempre com a mesma dedicação, reflecte-se nesta Tese. Por isso, um grande obrigado a ti, Susana.

Ao meu orientador, Doutor Alberto Sampaio, o meu grande obrigado pela disponibilidade que mostrou para me receber e apoiar ao longo desta Tese. Ao LABORIS em geral e ao Doutor Gustavo Alves em particular, o meu agradecimento por me ter disponibilizado os meios necessários para o desenvolvimento desta Tese e me ter indicado o caso prático a abordar. Finalmente, à Eng.^a Teresa Pimenta, o meu agradecimento por me ter recebido e esclarecido todas as dúvidas que tive ao longo da modelação.

O meu último agradecimento vai para os meus amigos, presentes ao longo destes anos no Instituto Superior de Engenharia do Porto. Em particular, um abraço especial ao João, ao Vasco, ao Pedro, ao Bruno e ao Duarte, numa fase em que terminamos esta longa caminhada.

A todos, um grande obrigado!

Resumo

Nos últimos anos, a existência de sistema de sistemas é considerada como uma banalidade, algo que aceitamos como um dado adquirido. Um exemplo deste tipo de sistemas é o automóvel, composto por vários sistemas secundários, que entre si permitem criar um sistema único, capaz de nos mover de um local A para um local B.

No entanto, a projecção de tais sistemas envolve um elevado número de equipas, cada uma delas com os seus próprios recursos, metodologias e conhecimentos, levando a que a inter-comunicação seja seriamente afectada.

A notação SysML, derivada do UML, possibilita a comunicação destas equipas entre si, utilizando para isso uma linguagem comum a todos. Através das várias características da notação (diagramas, estereótipos, entre outras), é possível transmitir as ideias de forma clara e inequívoca.

O objectivo desta Tese passa por analisar todos estes aspectos teóricos e práticos, para que seja possível apresentar conclusões sobre a notação e a sua utilidade na modelação de sistema de sistemas.

Palavras-chave: Sistema de Sistemas, Modelação de Sistemas, UML, SysML

Summary

In recent years, the existence of system of systems is considered usual, as something that we can rely on. One example of this type of systems is the car, made from various secondary systems, which as a whole, allow us to create an unique system, capable of moving us from point A to point B.

However, the making of such systems involves a great number of teams, each one with their own resources, methodologies and knowledges, leading to a rather difficult inter-communication.

The SysML notation, derived from UML, allows the communication between the different teams, using a common language. With the help of several characteristics of the language (such as diagrams, stereotypes, among others), the clear and undoubtful transmission of ideas and knowledges is made possible.

The objective of this Thesis is to analyze all theoretical and practical aspects of the notation, so that is possible to conclude about its utility in the modeling of system of systems.

Keywords: System of Systems, Systems Modeling, UML, SysML

Índice

Agradecimentos	V
Resumo	VII
Summary	VII
Índice	IX
Índice de Figuras	XV
Índice de Tabelas	XVII
Glossário	XIX
1. Introdução	1
1.1. Origem da notação	1
1.2. Objectivos Propostos.....	1
1.3. Organização da Tese	2
1.4. Estrutura do documento.....	2
2. Sistemas.....	3
2.1. Tipos de Sistemas	3
2.2. Sistema de sistemas	4
2.3. Engenharia de Sistemas.....	4
2.4. Metodologias na Engenharia de Sistemas	5
2.5. Engenheiro de Sistemas.....	6
2.6. Modelação de Sistemas.....	7
2.7. Conclusão	8
3. Notação SysML	9
3.1. Importância do SysML.....	9
3.2. SysML na indústria	10
3.3. Estrutura do SysML	10
3.4. Estereótipos e Perfis	11

3.5.	Alterações no SysML.....	12
3.6.	Diagrama de Requisitos.....	15
3.6.1.	Requisitos.....	15
3.6.2.	Contentor de Espaços de Nome (Namespace Container)	16
3.6.3.	Relação Satisfação (“Satisfy”) do Requisito	16
3.6.4.	Relação Derivação (“Derive”) do Requisito	17
3.6.5.	Relação Cópia (“Copy”) do Requisito.....	17
3.6.6.	Relação Refinamento (“Refine”) do Requisito.....	17
3.6.7.	Relação Rastreabilidade (“Trace”)	18
3.6.8.	Relação Verificação (“Verify”) do Requisito.....	18
3.6.9.	Caso de Teste	19
3.6.10.	Notação Tabela	20
3.7.	Diagrama de Definição de Blocos (Block Diagram)	20
3.7.1.	Blocos	20
3.7.2.	Tipos de Dados (Data Types)	22
3.7.3.	Definição da Distribuição	22
3.7.4.	Unidade e Dimensão	22
3.7.5.	Tipo Valor (Value Type)	23
3.7.6.	Porto de Fluxo (Flow Port).....	23
3.7.7.	Fluxo de Itens (Item Flow)	24
3.7.8.	Bloco de Associação (Association Block)	24
3.8.	Diagrama Paramétrico (Parametric Diagram)	24
3.8.1.	Bloco de Restrição (Constraint Block).....	25
3.9.	Diagrama de Actividade (Activity Diagram)	26
3.9.1.	Árvore de Funções (Function Tree)	26
3.9.2.	Operador de Controlo.....	27

3.9.3.	Taxa (Rate)	27
3.9.4.	Nó de Objecto Especial (Special Object Node)	27
3.9.5.	Probabilidade	28
3.10.	Alocação	28
3.10.1.	Relação Alocação (Allocation Relationship)	29
3.10.2.	Alocação de Partição de Actividade (Allocate Activity Partition)	30
3.10.3.	Notação Tabela	30
3.11.	Elementos de Modelação Genéricos	30
3.11.1.	Rationale	30
3.11.2.	Janela do Diagrama (Diagram Frame)	31
3.11.3.	Visão do Modelo (Model View and Viewpoint)	31
3.11.4.	Problema	31
3.12.	UML Testing Profile	31
3.13.	Discussão	32
3.13.1.	Pontos fortes e fracos	32
3.13.2.	Análise técnica	33
3.14.	Conclusão	36
4.	Ferramentas e Notações	39
4.1.	Ferramentas disponíveis	39
4.2.	Outras notações	40
4.2.1.	MARTE	40
4.2.2.	AADL	40
4.2.3.	SystemC	41
4.2.4.	JMCAD	42
4.2.5.	Modelica	43
4.2.6.	Bond Graph	44

4.3.	Conclusão	45
5.	Caso Prático	47
5.1.	Descrição do Caso Prático	47
5.2.	Método de Investigação.....	48
5.3.	Metodologia de Concepção do Sistema.....	51
5.4.	Escolha da Ferramenta.....	52
5.5.	Fases de Investigação	53
5.6.	Análise do Contexto.....	53
5.7.	Modelo de Domínio.....	55
5.7.1.	Avaliação.....	58
5.8.	Análise de Requisitos.....	58
5.8.1.	Avaliação.....	63
5.9.	Casos de Uso	64
5.9.1.	Avaliação.....	67
5.10.	Interacções Identificadas	67
5.10.1.	Avaliação.....	70
5.11.	Actividades e Parâmetros.....	71
5.11.1.	Avaliação.....	81
5.12.	Blocos internos.....	82
5.12.1.	Avaliação.....	85
5.13.	Estados e Pacotes	86
5.13.1.	Avaliação.....	91
5.14.	Discussão.....	91
5.14.1.	Limitações do SysML	91
5.14.1.1.	Validações estabelecidas	91
5.14.1.2.	Principais dificuldades.....	92

5.14.1.3.	Áreas de aplicação	93
5.14.2.	Aplicações da Notação e Metodologias	94
5.14.2.1.	Sobre a notação SysML.....	94
5.14.2.2.	Utilização do SYSMOD.....	95
5.14.2.3.	Interesse da Investigação em Acção.....	95
5.14.2.4.	Simulação.....	96
5.14.3.	Consistência do modelo.....	97
5.14.3.1.	Especificação da notação.....	97
5.14.3.2.	Relações na ferramenta.....	98
5.14.3.3.	Adaptabilidade dos modelos	99
5.15.	Conclusão	100
6.	Conclusão	101
6.1.	Conclusões do Estudo	101
6.2.	Trabalho Futuro.....	102
	Referências	105
	Anexo A	107
1.	Introdução	107
2.	Diagrama de Sequência (Sequence Diagram).....	107
3.	Diagrama de Casos de Uso (Use Case Diagram).....	107
3.1.	Caso de Uso	108
3.2.	Actor	108
3.3.	Relação Incluir (Include Relationship)	108
3.4.	Relação de Extensão (Extend Relationship)	109
3.5.	Generalização (Generalization)	109
4.	Diagrama de Máquina de Estado (State Machine Diagram).....	109

Índice de Figuras

Figura 1 – Árvore de Diagramas do SysML _____	14
Figura 2 – Fotografia do Sistema de Experimentação de Transferência de Calor _	47
Figura 3 – Diagrama de contexto do sistema do Departamento de Química _____	55
Figura 4 – Comportamentos possíveis no sistema em causa _____	56
Figura 5 – Detalhe do sistema de sistemas analisado _____	57
Figura 6 – Diagrama de requisitos do tanque principal _____	59
Figura 7 – Diagrama de requisitos do tanque secundário _____	61
Figura 8 – Diagrama de requisitos do sistema de aquisição de dados _____	62
Figura 9 – Diagrama de casos de uso relativo ao actor Aluno _____	64
Figura 10 – Diagrama de casos de uso relativo ao actor Professor _____	66
Figura 11 – Diagrama de sequência da acção Ligação do Sistema _____	68
Figura 12 – Diagrama de sequência da acção Ligação do Sistema _____	68
Figura 13 – Diagrama de sequência da acção Encerramento do Sistema _____	69
Figura 14 – Diagrama de sequência da acção Encerramento do Sistema _____	69
Figura 15 – Diagrama de sequência da acção Alteração de Temperatura _____	69
Figura 16 – Diagrama de sequência da acção Alteração de Temperatura _____	70
Figura 17 – Diagrama de actividade do processo Ligação do Sistema _____	72
Figura 18 – Diagrama de actividade do processo Ligação Tanque 1 _____	72
Figura 19 – Diagrama de actividade do processo Ligação Tanque 2 _____	73
Figura 20 – Diagrama de actividade do processo Ligação Aquisição de Dados ____	74
Figura 21 – Diagrama de actividade do processo Encerramento do Sistema _____	75
Figura 22 – Diagrama de actividade do processo Verificar Tanque 1 _____	75
Figura 23 – Diagrama de actividade do processo Verificar Tanque 2 _____	76
Figura 24 – Diagrama de actividade do processo Verificar Aquisição de Dados ____	77
Figura 25 – Diagrama de actividade do processo Alteração de Temperatura _____	77

Figura 26 – Diagrama de actividade do processo Existência de Líquido _____	78
Figura 27 – Diagrama de actividade do processo Alteração de Refrigeração ____	79
Figura 28 – Diagrama paramétrico do Controlo de Temperatura _____	80
Figura 29 – Diagrama paramétrico do Transporte de fluido _____	80
Figura 30 – Diagrama de blocos internos do Sistema de Transferência de Calor _	82
Figura 31 – Diagrama de blocos internos do Tanque 1_____	83
Figura 32 – Diagrama de blocos internos do Tanque 2_____	84
Figura 33 – Diagrama de blocos internos do Sistema de Aquisição de Dados ____	85
Figura 34 – Diagrama de máquinas de estado do Sistema de Transferência ____	86
Figura 35 – Diagrama de máquinas de estado do Tanque 1 _____	87
Figura 36 – Diagrama de máquinas de estado do Tanque 2 _____	88
Figura 37 – Diagrama de máquinas de estado do Sistema de Aquisição _____	89
Figura 38 – Diagrama de pacotes do Sistema de Transferência de Calor _____	90

Índice de Tabelas

Tabela 1 – Tipos de Sistemas Existentes	3
Tabela 2 – Estrutura do SysML	13
Tabela 3 – Estrutura do SysML	13
Tabela 4 – Diagramas presentes no SysML	13
Tabela 5 – Diagramas ocultados no SysML	14
Tabela 6 – Ferramentas com suporte para SysML	39
Tabela 7 – Tabela de requisitos relativa ao tanque principal	60
Tabela 8 – Tabela de requisitos relativa ao tanque secundário	61
Tabela 9 – Tabela de requisitos relativa ao sistema de aquisição de dados	63
Tabela 10 – Especificação do caso de uso [Aluno] Ligação do Sistema	64
Tabela 11 – Especificação do caso de uso [Aluno] Encerramento do Sistema	65
Tabela 12 – Especificação do caso de uso [Aluno] Alteração da temperatura	65
Tabela 13 – Especificação do caso de uso [Professor] Ligação do Sistema	66
Tabela 14 – Especificação do caso de uso [Professor] Encerramento do Sistema	66
Tabela 15 – Especificação do caso de uso [Professor] Alteração da temperatura	67

Glossário

D

Diagrama: Representação gráfica de uma parte específica de um sistema, definindo os respectivos aspectos característicos.

I

INCOSE: International Council of Systems Engineering, uma das principais organizações no campo da engenharia de sistemas.

M

Modelação de Sistemas: Consiste na criação de diagramas interligados entre si, os quais irão permitir a descrição de um sistema.

Modelo: Abstracção da realidade, consiste num conjunto de diagramas que entre si, definem a estrutura de um sistema, com as suas operações e atributos próprios.

O

OMG: Object Management Group, criador da linguagem SysML e UML.

S

SIMILAR: Metodologia utilizada na Engenharia de Sistemas.

Sistema de Sistemas: Um sistema composto por vários blocos, os quais podem ser sistemas independentes.

SysML: Systems Modeling Language, linguagem gráfica utilizada para modelação de sistema de sistemas.

SYSMOD: Metodologia de aplicação da linguagem SysML na elaboração de sistemas de sistemas.

U

UML: Unified Modeling Language, linguagem gráfica utilizada para modelação e descrição de software.

1. Introdução

Tal como é patente nos dias de hoje, a engenharia de sistemas é cada vez mais complexa, o que juntamente com a integração de sistemas dentro de outros sistemas, por sua vez também complexos, resulta num sistema final difícil de compreender, desenvolver e prever. Foi a partir deste pressuposto que surgiu o SysML, uma extensão de um subconjunto da Unified Modeling Language (UML), desenvolvido pela Object Management Group (OMG). A criação desta extensão teve como objectivo principal o desenvolvimento de uma linguagem única, que pudesse ser usada por todas as equipas de concepção, qualquer que fosse o sistema que estivesse em desenvolvimento e/ou manutenção.

1.1. Origem da notação

Para que tal objectivo fosse possível, analisou-se o que poderia ser efectivamente usado noutras áreas além do desenvolvimento de software, qual a relevância da componente de engenharia de software que realmente integrava o UML e como é que se poderia adaptá-lo para ser utilizado na concepção de sistemas de engenharia de outras áreas de aplicação. Concluiu-se então uma linguagem que se mostrou capaz e útil para os diversos ambientes em que poderia ser inserida, sem se mostrar demasiado específica para ser restritiva na sua utilização.

Tendo como principal potencial, facilitar a criação de sistemas de sistemas, permite expor o funcionamento dos vários componentes de forma gráfica e intuitiva. No entanto, o SysML também apresenta potenciais problemas, sendo que um dos principais não se encontra directamente relacionado com a própria linguagem, mas sim com as metodologias de modelação utilizadas. Uma vez que a especificação da linguagem não apresenta uma metodologia de aplicação própria, torna-se necessário o recurso a metodologias externas, em reduzido número e com acesso dificultado.

1.2. Objectivos Propostos

A partir desta análise inicial do SysML, foi desenvolvido todo o trabalho relativo a esta Tese, em que o principal objectivo é:

- Aumentar o conhecimento sobre a notação SysML

Enquadrado neste objectivo principal, podemos destacar outros objectivos secundários que estão directamente ligados à concretização do objectivo principal:

- Estudo e análise da notação SysML
- Recolha de dados práticos para validação e melhoramento da análise anterior
- Avaliar as potencialidades da notação

A determinação dos pontos fortes e fracos foi feita através de uma análise crítica da literatura existente. Quanto à validação, esta foi realizada através de um caso prático de utilização do SysML, onde se modelou um sistema de sistemas utilizando esta linguagem. Os resultados do caso prático ajudaram a melhorar a análise inicialmente realizada.

1.3. Organização da Tese

No que concerne à organização da Tese e de toda a sua elaboração, esta foi dividida nas seguintes fases:

- Análise teórica da área de sistemas e da notação SysML
- Análise prática da aplicabilidade da notação, com recurso a um caso prático
- Apresentação dos resultados e validações efectuadas

Através desta divisão da Tese por diferentes fases e períodos, tornou-se possível especificar quais os parâmetros teóricos referente à notação que necessitavam de avaliação prática, a partir da qual foi elaborado um conjunto de recomendações válidas e credíveis. Além disso, o estudo teórico e consequente integração da notação num caso prático permitiu analisar a sua adaptação a um novo meio, real e representativo dos cenários em que poderia ser utilizada.

1.4. Estrutura do documento

Este documento encontra-se dividido em 6 secções. A Secção 1 introduz a temática da Tese, apresentando também a sua organização e desenvolvimento, enquanto a Secção 2 descreve a área de Sistemas, bem como as suas derivações. Na Secção 3, é apresentada a notação SysML em detalhe enquanto na Secção 4, são referidos os pontos fortes e fracos da notação, bem como ferramentas que permitem a sua utilização. Finalmente, na Secção 5 é relatado o caso prático abordado nesta Tese, com a Secção 6 a apresentar as conclusões finais obtidas. No Anexo, são apresentados os diagramas que não sofreram qualquer alteração relativamente ao UML.

2. Sistemas

Iniciando a abordagem da temática pelo ponto mais básico, devemos definir um sistema:

Um sistema trata-se de um conjunto de vários componentes, de determinadas disciplinas que unidos devem funcionar em conjunto e realizar uma determinada tarefa para a qual foram desenvolvidos [21].

Com isto, um sistema pode significar qualquer objecto que possa ser criado, tal como um frigorífico, um carro, uma casa, levando à interacção de variadas disciplinas de engenharia, tais como engenharia química, informática, mecânica, entre outras, devendo estas funcionar em sintonia.

2.1. Tipos de Sistemas

Dentro dos sistemas, podemos distinguir vários tipos, devidamente enquadrados nos diversos ramos da engenharia. Na tabela apresentam-se alguns exemplos desses tipos de sistemas:

Tipos de Sistemas	Área de Engenharia	Exemplos
Comunicação	Engenharia de telecomunicações	Sistemas de comunicações móveis e fixas, analógicas e digitais
Eléctricos	Engenharia electrotécnica	Sistemas de automação e controlo de tarefas, sistemas modulares
Mecânicos	Engenharia mecânica	Sistemas de suspensão e direcção de automóveis, sistemas industriais
Informação	Engenharia informática	Sistemas de gestão, sistemas estratégicos

Tabela 1. Tipos de sistemas existentes

Apesar da designação de sistemas, os conceitos aplicados em cada um destes tipos de sistemas difere de ramo para ramo, havendo poucas ou nenhuma semelhanças nas metodologias, ferramentas e conhecimentos aplicados no desenvolvimento dos mesmos.

Desta forma podemos inferir que um sistema poderá apresentar problemas específicos, dependendo da área em que se encontra integrado.

2.2. Sistema de sistemas

Podendo um sistema ser composto por vários subsistemas, podemos definir um sistema de sistemas da seguinte forma:

Um sistema de sistemas é a designação para um conjunto de sistemas dedicados à tarefa que, utilizando os seus recursos e as suas capacidades de forma conjunta, permite um novo sistema, mais complexo, com melhor desempenho e maiores capacidades, do que seria possível com a simples soma dos seus constituintes [18].

Desta forma, o desenvolvimento de um sistema completo deverá ter em conta a integração dos vários subsistemas, para que seja possível fornecer uma solução funcional, que cumpra todos os requisitos inicialmente propostos e que não introduza novas falhas em virtude de uma integração deficiente.

2.3. Engenharia de Sistemas

Devido à complexidade crescente, presente no desenvolvimento de sistemas, existe a necessidade de dividir as tarefas, de acordo com as disciplinas abordadas. No entanto, a comunicação entre as mesmas deve ser garantida para que a correcta integração dos vários componentes interdisciplinares seja bem sucedida.

Partindo desta necessidade, a criação e o desenvolvimento de uma disciplina dedicada à engenharia de sistemas levou a que se prestasse atenção não apenas à componente técnica em si, mas também à sua correcta interligação, optimizando todos os processos existentes na gestão de um projecto.

A engenharia de sistemas congrega todas as disciplinas da engenharia, sobressaindo delas, atravessando todas as fases do desenvolvimento de um sistema completo, abordando os problemas e as soluções para problemas existentes durante o projecto. Os processos utilizados para a elaboração do projecto deverão ser abordados segundo uma metodologia correcta e que garanta fiabilidade e correcção do projecto.

Segundo o International Council on Systems Engineering (INCOSE), a engenharia de sistemas pode ser definida desta forma:

A engenharia de sistemas é uma abordagem interdisciplinar, através de meios que permitem a realização bem sucedida de sistemas. Foca-se na definição das necessidades do consumidor e funcionalidades requisitadas numa fase precoce do ciclo de desenvolvimento, documentando os requisitos e procedendo à sintetização do

desenho e validação do sistema, enquanto se consideram vários pontos como custo, treino, suporte, desempenho, entre outros.

A engenharia de sistemas integra todas as disciplinas e grupos de especialidade numa só equipa, formando um processo de desenvolvimento estruturado que evolui desde o conceito, para a produção e operação. A engenharia de sistemas também considera as necessidades técnicas e de negócio dos consumidores, com o objectivo de providenciar um produto com qualidade, que cumpre as necessidades do utilizador [6].

Na prática, a engenharia de sistemas consiste sobretudo na definição e documentação dos requisitos de um sistema, a concepção do mesmo e sua verificação, relativamente aos requisitos inicialmente propostos [1]. A definição de um sistema pode levar à criação de blocos (ou componentes), podendo cada bloco corresponder a software, hardware, pessoa, ou qualquer outro tipo de sistema, devendo todos os blocos existentes ser correctamente detalhados. No entanto, o nível de detalhe de cada um dos blocos (e do próprio modelo criado para o sistema) pode depender da complexidade que se pretenda, podendo ser muito simples, ou altamente detalhados até ao mais ínfimo pormenor.

Juntamente com o desenvolvimento imediato do projecto, a engenharia de sistemas deve também levar à criação de um modelo temporal que irá prever a utilização do sistema, desde a sua fase de introdução no meio ambiente até à sua remoção e desactivação. Todos os detalhes referentes a este modelo deverão ser abordados numa fase inicial do projecto, de modo a que possam ser efectuadas correcções, as quais serão bastante difíceis (senão impossível) em fases mais avançadas do projecto.

2.4. Metodologias na Engenharia de Sistemas

Como foi referido anteriormente, a elaboração de um projecto e a descoberta de soluções para os problemas enfrentados no mesmo deve seguir uma metodologia rigorosa e que garanta confiança no resultado obtido. Existem várias metodologias que poderão ser aplicadas, tal como as metodologias SYSMOD [15], SIMILAR [21], sendo que esta última consiste nestas fases:

Abordar o problema - Descreve as tarefas envolvidas, devendo estas estar bem formuladas, pois os erros nesta fase são bastante custosos.

Investigar alternativas – Analisa as várias soluções encontradas, devendo ter em consideração aspectos como tamanho, custo, tempo, etc.

Modelar o sistema – Detalhe do modelo escolhido, sendo um desenvolvimento inicial do mesmo.

Integrar – Integra o sistema no ambiente de destino, definindo as interacções com os restantes elementos do mesmo ambiente.

Inicializar o sistema – Inicia as actividades do sistema, implementando o desenho especificado inicialmente.

Avaliar o desempenho – Teste do sistema e avaliação dos resultados de acordo com os requisitos propostos na elaboração do projecto.

Reavaliação – Nova avaliação geral do sistema, de acordo com os resultados obtidos na fase anterior e que serão utilizados para correcções futuras e análise de soluções para problemas pontuais.

Além destas fases, deve-se ter em conta a avaliação dos riscos a tomar, bem como a sua própria gestão, sendo que esta não é considerada na metodologia SIMILAR. A correcta avaliação de eventuais riscos, permite identificar potenciais áreas problemáticas e definir medidas para as enfrentar, sendo que a sua identificação requer uma observação atenta de todos os desenvolvimentos do projecto.

2.5. Engenheiro de Sistemas

A função do engenheiro de sistemas é bastante importante para o correcto desenvolvimento de uma solução eficiente. Podemos definir o engenheiro de sistemas desta forma:

O engenheiro de sistemas é o responsável pela união e interacção das diferentes partes do projecto, o que, tendo em conta que todas essas partes podem ser substancialmente distintas, leva a que a complexidade desta tarefa seja elevada e de grande responsabilidade [15].

O engenheiro de sistemas (ao contrário dos engenheiros das outras disciplinas) possui uma visão geral do sistema, independentemente de cada uma das disciplinas envolvidas. Além disso, o mesmo possui poder decisivo sobre todo o projecto, podendo abortar ou não as resoluções apresentadas.

A existência de um engenheiro de sistema é preponderante para a correcta gestão de um projecto, visto que este será responsável pela manutenção de uma

visão coerente e correcta do projecto, evitando contratempos que possam afectar o resultado final e a sua avaliação relativamente à proposta inicial.

Outro factor a ter em conta deve ser o domínio de uma determinada disciplina sobre todas as outras, sendo que este facto deve ser evitado, porque tal poderá levar a potenciais conflitos e falhas de comunicação. Além disto, as soluções propostas por cada disciplina devem funcionar em conjunto, não causando problemas adicionais ao projecto, sendo que a decisão final passará obrigatoriamente pelo engenheiro de sistemas.

Em última análise, o papel do engenheiro de sistemas deve ser integrado na estrutura do projecto, de forma a criar regras claras e tarefas bem definidas e transparentes para todos. Desta forma, será possível criar um elemento de controlo, que será o responsável pelo sucesso final de todo um projecto.

2.6. Modelação de Sistemas

Com a introdução das tecnologias de informação, nos últimos 50 anos, a área de modelação e simulação de sistemas foi-se desenvolvendo e penetrando nas diversas disciplinas de engenharia, bem como noutras áreas da ciência. Podemos definir um modelo de um sistema, desta forma:

Um modelo pode ser considerado uma abstracção da realidade, desenhada para responder a questões específicas sobre o mundo real, uma representação de um processo ou estrutura do mundo real ou uma ferramenta física, conceptual ou matemática para ajudar na escolha de uma decisão. [13]

Assim sendo, a modelação de sistemas consiste na criação de modelos de determinados sistemas, de tal forma que possam ser considerados como representações credíveis dos respectivos sistemas físicos.

Qualquer que seja o sistema em causa, a modelação de sistemas pode ajudar na obtenção de um conhecimento mais aprofundado do funcionamento interno do mesmo, bem como prever o seu comportamento, com base em dados mais fiáveis. Além disso, a simulação é frequentemente utilizada como substituto real dos testes físicos.

Desta forma, poderemos obter como principais vantagens [5], redução dos custos associados aos testes físicos (quer financeiros, quer temporais), possibilidade de experimentação e recolha de dados em situações adversas (repetição infinita dos testes, testes destrutivos, entre outros), além do controlo e monitorização de todos os parâmetros associados à simulação.

No entanto, também existem diversas desvantagens associadas à utilização da modelação e simulação de sistemas [5], tais como o facto de cada teste virtual necessitar de uma modelação completa, válida e verificada do sistema, bem como as próprias simulações estarem limitadas pela capacidade de processamento dos computadores utilizados. Adicionalmente, um modelo é sempre uma simplificação da realidade.

A validação e verificação do modelo também são aspectos cruciais da modelação de sistemas, podendo-se definir cada uma delas desta forma:

A verificação do modelo analisa se o modelo implementado reflecte o modelo conceptual, dentro dos limites de precisão estabelecidos. No que concerne a validação do modelo, esta permite avaliar se o modelo implementado é adequado para à tarefa em causa, no seu domínio de aplicação [5].

A validação e a verificação apresentam-se como aspectos que permitem validar as especificações de um sistema e verificar o desenho criado para o mesmo.

2.7. Conclusão

Através da análise da problemática de sistemas e sua modelação, é possível observar que existem aspectos que poderiam ser facilitados pela utilização de uma notação que permitisse agregar todas as áreas dos projectos, sob uma linguagem única e compreensível por todas as partes.

Tendo em conta que um sistema de sistemas é constituído por diversos subsistemas, cada um destes desenvolvidos por metodologias próprias e de acordo com regras específicas, a interligação dos mesmos é um aspecto determinante para o sucesso dos projectos. Além disso, a existência de uma visão coerente na realização do projecto é essencial para que este consigo ultrapassar as dificuldades enfrentadas durante o percurso, sendo que essa tarefa é garantida pelo engenheiro de sistemas, em conjunto com uma metodologia de criação de sistemas rigorosa e merecedora de confiança.

Como complemento, a utilização de uma notação única, com interpretação igual para todos os intervenientes, seria interessante para facilitar a tarefa do engenheiro de sistemas, que assim poderia recorrer a uma ferramenta que lhe possibilitasse apresentar a sua visão, a todos os elementos da equipa, sem ambiguidades e falhas de interpretação.

3. Notação SysML

Apesar de existirem diversas linguagens de modelação de sistemas, as mesmas continuavam a utilizar metodologias tradicionais, bastante específicas, pouco adaptáveis a outras disciplinas de engenharia que pudessem estar envolvidas num projecto.

Tornou-se por isso necessário a criação de uma linguagem de modelação normalizada, que pudesse ser abordada por todas as disciplinas, utilizando uma ferramenta única, sem no entanto possuir especificidades de cada uma delas. Essa linguagem trata-se do SysML, a qual será abordada nesta Secção em detalhe, com base nos dados disponibilizados pela referência [21].

3.1. Importância do SysML

A criação do SysML partiu da iniciativa do OMG em tornar o UML, uma linguagem utilizada extensivamente a nível global, na linguagem normalizada para a engenharia de sistemas. Dai que uma pergunta poderá surgir: porque não utilizar o UML, ao invés de desenvolver uma nova linguagem? A resposta a essa pergunta passa pelo facto de o UML não possuir uma extensão normalizada, necessária para satisfazer determinadas necessidades, tais como capacidade para modelar diferentes tipos de requisitos, suporte para funções contínuas e estruturas de decomposição [21]. Além disto, o UML apresentava demasiadas ligações à engenharia de software (conceitos como classes e orientação a objectos, pouco perceptíveis a quem não seja desta área), os quais impossibilitavam a utilização do UML nouro tipo de engenharia. Estes factos levaram assim a uma necessidade de alterar o UML, para que se pudesse adequar a um novo tipo de modelos.

Recorrendo ao mecanismo de extensões do UML, foi criado um vocabulário adicional, específico para modelação de sistemas, além de terem sido omissas e alteradas algumas funcionalidades do UML. O resultado foi o SysML 1.0, o qual foi aceite como norma pelo OMG em Abril de 2006 e que é actualmente suportado por diversos fabricantes como a ARTiSAN Software Tools, EmbeddedPlus Engineering, Telelogic, NoMagic e Sparx Systems [15].

Actualmente, o SysML encontra-se na sua versão 1.1, tendo esta sido lançada em Maio de 2008. O objectivo principal desta versão passa sobretudo por pequenas correcções de determinados aspectos do SysML, não afectando o funcionamento e o conhecimento já obtido a partir da versão anterior.

Visto que o SysML é a linguagem normalizada para a engenharia de sistemas, a sua análise será de vital importância, quer para comparação com eventuais notações que existem actualmente e eventualmente existirão, bem como para avaliação da sua utilidade e reais potencialidades no meio empresarial. Será partindo desta necessidade que se desenvolverá todo o trabalho realizado nesta Tese, analisando se o SysML se adequa à concepção de sistemas de sistemas, bem como sistemas embebidos, relativos a outras áreas de engenharia, como a engenharia química e mecânica.

3.2. SysML na indústria

O SysML, apesar do seu período de vida relativamente curto, obteve um apoio e suporte bastante alargado por parte da indústria. Tal contribuição partiu da iniciativa deste grupo de empresas, juntamente com a OMG, de criar um novo perfil da linguagem UML para a engenharia de sistemas.

Numa fase inicial, assistiu ao envolvimento por parte de empresas como a Gentleware AG, Motorola Inc., Telelogic AB, PivotPoint Technology Corporation e Northrop Grumman Corporation, as quais ajudaram no desenvolvimento da especificação inicial do SysML. Passada esta fase e após o lançamento comercial do SysML, mais empresas aderiram a esta notação, integrando-a nos seus próprios procedimentos. Das várias empresas que adoptaram o SysML, de destacar a Boeing, American Systems Corporation, I-Logix Inc., IBM, Israel Aircraft Industries, Lockheed Martin, NASA, entre outras.

O crescimento do SysML na indústria tem sido consistente, com o número de empresas a adoptarem-no como referência a aumentar de dia para dia.

3.3. Estrutura do SysML

O SysML é actualmente a linguagem a utilizar quando a temática é engenharia de sistemas. É uma linguagem orientada a objectos e a sua caracterização pode ser feita através de 4 pontos: diagrama, modelo, elementos estruturais e elementos comportamentais.

A caracterização do SysML através destes pontos permite o conhecimento da estrutura básica da linguagem e dos seus conceitos principais, sobre os quais se rege toda a sua concepção, levando assim a um entendimento mais fácil da sua organização e utilização.

De forma a esclarecer qual a importância de cada um destes pontos no SysML, podemos defini-los desta forma [21]:

- *Um diagrama representa um aspecto específico de um determinado sistema, sendo a sua representação da totalidade do sistema incompleta.*
- *Um modelo representa a descrição completa desse mesmo sistema, contendo todos os diagramas que descrevem os atributos, operações e associações existentes.*
- *Os elementos estruturais permitem descrever a estrutura do sistema, através de elementos como blocos e componentes.*
- *Os elementos comportamentais permitem descrever o funcionamento do sistema, bem como as suas interações, utilizando para isso actividades, máquinas de estado e interações.*

Existem, no entanto, alguns elementos que não poderão ser enquadrados dentro de uma única categoria, visto que além de serem estruturais, são também comportamentais.

3.4. Estereótipos e Perfis

As raízes do SysML originam no UML, a partir do qual foi expandido e adaptado, alterando substancialmente a estrutura e conceitos utilizados. Entre as várias alternativas para extensão da linguagem, deve-se destacar os estereótipos. Estes podem ser definidos desta forma:

O estereótipo é um mecanismo de extensibilidade que permite aumentar a flexibilidade dos blocos, através de uma subclassificação [11].

Os estereótipos permitem criar novos elementos de modelação, derivados a partir de elementos existentes mas possuindo propriedades específicas para um determinado problema ou utilização. A utilização de nós estereotipados permite resolver estas lacunas e representar os elementos em falta no modelo. Um exemplo simples da importância dos estereótipos trata-se da criação de símbolos necessários para a representação de uma determinada situação (routers e hubs numa rede informática, por exemplo).

Um estereótipo é representado pela identificação determinada para o elemento, acompanhado de << e >>, identificando-o como um estereótipo. Deve estar localizado acima do nome do elemento estereotipado. Pode também ser representado por um ícone específico, podendo este substituir a representação gráfica original do UML.

Além dos estereótipos, outro mecanismo de extensão do SysML são os perfis. Podemos definir os perfis desta forma:

Os perfis consistem num mecanismo genérico de extensão da linguagem, sendo definidos por elementos como estereótipos e restrições aplicadas a determinados elementos de modelação [15].

Desta forma, a reunião de todos as definições destes elementos permite adaptar a linguagem a uma determinada plataforma ou domínio, sendo esse o objectivo da utilização dos perfis em linguagens como UML. Pode-se também referir que o SysML é um perfil do UML, definido pela OMG e destinado à utilização na engenharia de sistemas.

3.5. Alterações no SysML

Relativamente ao UML, as alterações marcantes no SysML restringem-se à introdução de novos estereótipos e diagramas. Enquanto os estereótipos não necessitam de uma ferramenta de modelação específica, os diagramas requerem uma ferramenta específica para o SysML que tenha suporte ao nível do diagrama. Juntamente com as extensões adicionadas, também alguns elementos do UML foram retirados, por não serem relevantes para a linguagem. Podemos assim resumir as principais mudanças do SysML:

Classes - As classes do UML, utilizadas para descrição de estrutura e comportamentos dos objectos com as mesmas características e semânticas, denominam-se de blocos no SysML. Além disso, a denominação do diagrama de classes é alterada para diagrama de definição de blocos.

Diagramas de estrutura composta – este tipo de diagramas, presente no UML, são denominados de diagrama de blocos internos no SysML.

Fluxos internos – Os fluxos entre elementos, no diagrama de blocos internos, podem ser modeladas e representados.

Funções contínuas – Este tipo de funções são suportadas por acções e nós de objectos, nos diagramas de actividade. Além disso, os diagramas Enhanced Functional Flow Block são suportados.

Novos diagramas – Foram adicionados novos tipos de diagramas, tais como o diagrama de requisitos e o diagrama paramétrico.

ISO AP233 – Integração da norma ISO AP233, um protocolo de transferência de dados para sistemas, baseada na norma ISO 10303. Esta norma suporta o conjunto

de fases que integram o ciclo de desenvolvimento de um sistema, desde os requisitos até à validação. Para tal, recorre ao XMI e ao XML Schema para representação dos dados transferidos.

Voltando à caracterização do SysML e aos seus 4 pontos, podemos enquadrar os seus elementos na Tabela 2:

	Elementos estruturais	Elementos comportamentais
Diagrama	Diagrama de definição de blocos Diagrama de blocos internos Diagrama paramétrico Diagrama de pacotes	Diagrama de actividades Diagrama de casos de uso Diagrama de máquina de estados Diagrama de sequência
Modelo	Modelo de estrutura	Modelo de comportamento

Tabela 2. Estrutura do SysML

Quanto aos restantes elementos que não podem ser enquadrados unicamente como elementos estruturais ou comportamentais, a sua caracterização é realizada desta forma:

Elementos estruturais/comportamentais
Diagrama de requisitos, estereótipo, visão do modelo, AP-233, meta dados XMI e formato de transferência

Tabela 3. Estrutura do SysML

Relativamente aos diagramas presentes no SysML, a sua identificação (nome original e nome actual), bem como as suas mudanças relativas ao UML, podem ser caracterizadas desta forma:

Diagramas	Novo	Alterado	Estendido	Nome original
Diagrama de definição de blocos	Não	Sim	Sim	Diagrama de classes
Diagrama de actividades	Não	Não	Sim	-
Diagrama de casos de uso	Não	Não	Não	-
Diagrama paramétrico	Sim	Não	Não	-
Diagrama de pacotes	Não	Não	Não	-
Diagrama de sequência	Não	Não	Não	-

Diagrama de blocos internos	Não	Sim	Sim	Diagrama de estrutura composta
Diagrama de requisitos	Sim	Não	Não	-
Diagrama de máquina de estados	Não	Não	Não	-

Tabela 4. Diagramas presentes do SysML

De referir que os diagramas encontram-se divididos entre 3 tipos distintos: diagramas de comportamento, de estrutura e de requisitos.

Dentro do primeiro tipo, encontramos os diversos sub-diagramas, tais como os diagramas de actividades, de sequência, de máquinas de estado e casos de uso. Relativamente aos diagramas de estrutura, encontramos os sub-diagramas de definição de blocos, de pacotes e de blocos internos. Neste dois tipos principais de diagramas (Comportamento e Estrutura), os mesmos não podem ser utilizados directamente, sendo obrigatória a utilização de um sub-diagrama. Finalmente, no que concerne aos diagramas de requisitos, não existem sub-diagramas.

Além destes diagramas que permanecem (ou foram introduzidos) no SysML, existem outros diagramas que foram ocultados por não serem relevantes para esta linguagem. Os diagramas que tiveram esse destino estão indicados na Tabela 5.

Diagramas ocultados
Diagrama de componentes, diagrama de objectos, diagrama de instalação, diagrama de temporização, diagrama de comunicação, diagrama de interacção

Tabela 5. Diagramas ocultados no SysML

Resumindo num gráfico simples, a estrutura do SysML pode ser descrito através desta árvore de diagramas:

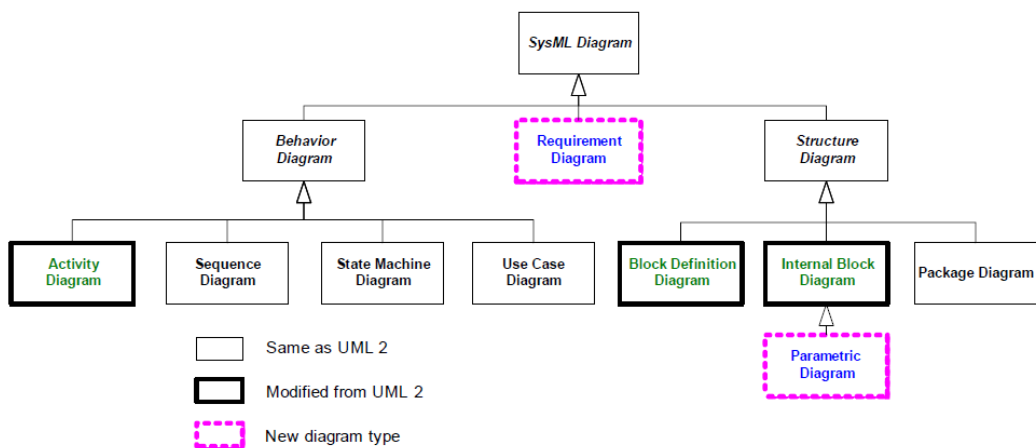


Figura 1. Árvore de Diagramas do SysML [15]

Nas próximas secções irão ser detalhados alguns dos aspectos do SysML, tais como os diagramas introduzidos e alterações efectuados sobre o UML.

3.6. Diagrama de Requisitos

O SysML define elementos para modelar e descrever requisitos, tais como tempos de resposta e tamanho, aspectos não funcionais que não poderiam ser representados no UML. Enquanto no UML, os requisitos funcionais poderiam ser descritos com recurso aos casos de uso, os requisitos não funcionais não poderiam ser inseridos e definidos.

O diagrama de requisitos do SysML permite colmatar essa lacuna, dando a possibilidade de descrever todos os requisitos existentes, bem como as suas relações com os outros elementos presentes no modelo. De realçar que este diagrama está presente e é suportado por todas as ferramentas de modelação do SysML.

3.6.1. Requisitos

Um requisito descreve uma ou mais propriedades ou comportamentos que o sistema tem de cumprir obrigatoriamente. Refere-se ao estereótipo requisito, presente no elemento classe do UML.

Os requisitos descrevem o contracto entre uma entidade e o criador do desenho do sistema, especificando os fluxos e as condições que têm de ser cumpridas pelo sistema. Um requisito é um estereótipo, tendo a semântica de uma classe à qual foi estendida e adicionada novas semânticas e propriedades.

As propriedades elementares de um requisito são o seu ID (identificador único, string sem estrutura definida) e o texto de descrição (string, podendo conter texto ou ligação para fontes externas), sendo estes atributos do próprio estereótipo que permitem a modelação de qualquer requisito.

Como limitações dos requisitos, não são permitidas operações e a adição de novos atributos sobre estes. No entanto, outras propriedades poderão ser introduzidas através de uma extensão independente do estereótipo, tais como categorias de requisitos, entre outros.

Os requisitos são abstractos, não permitindo a criação de instâncias a partir destes, nem a criação de generalizações e respectiva herança de operações e atributos. Relativamente à notação gráfica utilizada, os requisitos utilizam a notação por defeito do UML.

3.6.2. Contentor de Espaços de Nome (Namespace Container)

Um Contentor de Espaços de Nome permite descrever se um requisito está contido dentro de outro requisito. Os requisitos podem estar localizados em diferentes níveis hierárquicos, podendo um requisito geral conter vários requisitos mais específicos e detalhados, sendo que o requisito geral só é cumprido quando todos os requisitos inferiores são cumpridos.

Desta forma, esta relação descreve a relação entre um requisito principal e um requisito inferior, que se encontra contido dentro do requisito geral. O requisito principal não poderá conter requisitos com o mesmo nome.

Relativamente aos requisitos inferiores, estes não poderão fazer parte de um ou mais requisitos, não sendo por isso permitido a reutilização directa de requisitos. No entanto, poderá haver reutilização destes, através da relação Cópia de Requisito.

O Contentor de Espaços de Nome tem como correspondência directa as classes encapsuladas, presentes no UML. No que concerne a sua notação gráfica, é utilizado um símbolo (circulo com uma cruz) no requisito principal, de onde derivam todos os requisitos inferiores.

3.6.3. Relação Satisfação (“Satisfy”) do Requisito

A relação Satisfação do Requisito refere-se ao cumprimento de um requisito por parte de um elemento do modelo e representa o estereótipo Satisfação, presente na relação Realização do UML.

Cada elemento do modelo tem o objectivo (directo ou indirecto) de cumprir um determinado requisito. Esta relação permite por isso, ver quais os efeitos que o desenho terá no requisito, além de permitir identificar quais os elementos que serão modificados com as alterações efectuadas aos requisitos.

O ponto de partida da relação deve ser um elemento do modelo, preferencialmente um bloco ou pacote, de modo a não sobrecarregar o modelo (o que seria prejudicial, complicando a compreensão do mesmo).

O SysML utiliza uma notação gráfica diferente do UML, utilizando uma seta com o estereótipo Satisfy. Não é sempre necessário desenhar todas as relações presentes no diagrama, quando se utiliza esta notação gráfica.

A relação Satisfação do Requisito não define se o requisito é satisfeito inteiramente ou parcialmente, podendo-se adicionar um comentário ou estereótipo para complementar essa informação. Se um requisito é cumprido por vários elementos, pode-se usar uma única relação para representar todas estas sub-

relações. De realçar que uma relação Satisfação do Requisito pode ter várias fontes e vários destinos.

3.6.4. Relação Derivação (“Derive”) do Requisito

Uma relação de derivação do requisito descreve um requisito que foi derivado de outro requisito. Representa o estereótipo `deriveReq` do elemento abstracção do UML.

Está disponível apenas entre requisitos, permitindo descrever derivações de requisitos noutros requisitos. Apesar de ser possível a derivação no UML, é introduzida uma relação separada desta no SysML, de modo a facilitar a distinção das várias derivações existentes.

Relativamente à notação gráfica utilizada, recorre a uma seta com o estereótipo `deriveReq`, lendo-se no sentido da mesma. Além disto, a descrição da derivação pode ser complementado através da utilização de comentários.

3.6.5. Relação Cópia (“Copy”) do Requisito

A Relação Cópia do Requisito descreve a cópia de um determinado requisito a partir de outro requisito, representando a especialização do estereótipo `Traçagem` do UML.

Foi introduzido no SysML devido à existência de problemas formais na cópia directa de um requisito a partir de outro, especialmente se estiver contido dentro de um requisito principal. Permite a reutilização de um requisito em variados contextos, criando uma cópia que é consistente com o original, embora o nome e o ID sejam diferentes.

A descrição do requisito é sempre igual ao original, encontrando-se protegida contra cópia. Relativamente à notação gráfica utilizada, utiliza uma seta que tem como origem a cópia e destino o requisito original, contendo o estereótipo `Copy`. Relativamente ao comentário, podem ser utilizados como alternativa mas apenas numa única direcção, recorrendo à palavra-chave `Master`, à qual é anexa o elemento do modelo, junto da cópia efectuada.

3.6.6. Relação Refinamento (“Refine”) do Requisito

Esta relação especifica se um elemento do modelo descreve as propriedades de um requisito com maior detalhe. Representa o estereótipo `Refinamento` da relação `Abstracção` do UML.

Um requisito necessita de mais detalhes além do ID e do texto acessório, insuficientes para uma descrição detalhada. Para isso, é utilizada esta relação dentro do modelo, servindo para refinação do requisito pretendido, através da introdução de mais detalhes.

Esta relação não permite a refinamento através de fontes externas, sendo que para isso será utilizado o texto de descrição do próprio requisito. O SysML aproveita esta relação do UML sem qualquer alteração, adicionando-lhe uma notação alternativa que permite definir uma relação dentro de um comentário.

3.6.7. Relação Rastreabilidade (“Trace”)

Uma Relação Rastreabilidade descreve uma relação entre um requisito e um determinado elemento do modelo, utilizada apenas para efeitos de rastreabilidade. Representa o estereótipo Rastreabilidade, da relação Abstracção do UML.

É uma relação bastante geral e vaga, com semânticas fracas. Indica que existe uma relação entre determinados elementos do modelo, sem as detalhar. O único objectivo desta relação é possibilitar a rastreabilidade entre os diversos elementos do modelo.

O SysML aproveita esta relação directamente do UML, sem qualquer alteração, adicionando-lhe apenas uma notação alternativa, utilizando o comentário para detalhar a relação. Utiliza os estereótipos TracedTo e TracedFrom, os quais podem ser definidos da seguinte forma:

TracedTo – Representa um emissor da Relação Rastreabilidade e deriva de todos os requisitos que se apresentam como receptores desta relação.

TracedFrom – Representa um receptor da Relação Rastreabilidade e deriva de todos os requisitos que se apresentam como emissor desta relação

Esta notação é bastante útil quando se pretende determinar se os elementos intervenientes devem ou não aparecer no diagrama, devendo por isso ser utilizada nestas circunstâncias.

3.6.8. Relação Verificação (“Verify”) do Requisito

A Relação Verificação refere-se à ligação de um caso de teste com os requisitos referentes a esse mesmo caso e representa o estereótipo Verificação, sendo uma especialização do estereótipo Traçagem do UML.

O modelo de teste define vários casos de teste para verificar se a implementação dos requisitos foi ou não bem efectuada. A relação Verificação define qual teste vai avaliar qual requisito. Relativamente à notação gráfica, esta relação utiliza uma seta com o estereótipo Verify, apontando do caso de teste para o requisito. No que concerne os comentários, podem ser utilizados para mostrar que um elemento é participante activo nesta relação, sem recorrer à utilização da notação gráfica.

Esta relação não indica o grau de cumprimento do caso de teste, relativamente ao requisito, podendo-se utilizar os comentários ou estereótipos para complementar essa mesma informação.

3.6.9. Caso de Teste

Descreve um fluxo experimental, desenvolvido para testar um sistema e verificar se este satisfaz ou não um determinado requisito. Representa o estereótipo Caso de Teste, presentes nos elementos Operação e Comportamento do UML.

Para que seja possível analisar um diagrama de requisitos, temos de recorrer a casos de teste. No entanto, o SysML encontra-se bastante limitado neste aspecto, sendo recomendado a utilização do UML Testing Profile. Este perfil consiste num conjunto de pacotes, que podem ser utilizados para modelar diversos casos de teste, no SysML e UML, de acordo com as necessidades apresentadas (ver Secção 3.12).

De referir que os Casos de Teste e a Relação Verificação estão intimamente ligados, visto que os testes são efectuados sobre a implementação do sistema, podendo ser introduzidos diversos parâmetros para que seja possível testar o comportamento do sistema em diversas circunstâncias. Enquanto isso, a Relação Verificação permite analisar se os requisitos estabelecidos são cumpridos pela especificação apresentada para o sistema.

Relativamente ao Caso de Teste apresentado pelo SysML, este é suficiente para modelos simples, assegurando que determinados cenários são aceites pela implementação do sistema e servindo de base para linguagens de teste mais especializadas.

Os elementos Comportamento e Operação podem ser utilizados para descrição detalhada e executável de um caso de teste. Relativamente à notação gráfica, é utilizado um rectângulo com o nome do caso de teste, juntamente com o estereótipo TestCase.

3.6.10. Notação Tabela

Refere-se à notação disponibilizada pelo SysML, a qual é uma extensão do diagrama que não pode ser definida por estereótipos. Também tem que ser explicitamente suportada pela ferramenta de modelação.

As colunas de uma tabela de requisitos representam as propriedades, tais como o ID e o texto de descrição do requisito, bem como as relações desse requisito com outros requisitos. Esta notação permite visualizar milhares de requisitos, sem qualquer complicação visual.

3.7. Diagrama de Definição de Blocos (Block Diagram)

Os blocos apresentados no SysML são os elementos centrais de toda a linguagem. Os blocos derivam das classes e objectos, conceitos essenciais em linguagens orientadas a objectos e que se encontravam associadas ao UML. No entanto, visto que tinham uma conotação demasiado ligada à engenharia de software, os conceitos foram reestruturados e adaptados à engenharia de sistemas, adoptando a forma de blocos.

Os diagramas de definição de blocos descrevem a estrutura do sistema, sendo que cada bloco pode possuir um atributo (com informação sobre o próprio elemento) ou então referenciar outros blocos (através de associações). Os blocos podem descrever também operações que possam ser executadas. Em resumo, os blocos utilizados no diagrama de definição de blocos permitem descrever o sistema como um conjunto, onde cada parte tem um determinado papel a desempenhar, num contexto específico.

Os blocos podem ser descritos como classes (já presentes no UML), mas às qual foram alteradas alguns pormenores. Entre eles, de referir algumas nomenclaturas utilizadas (os diagrama de classes passaram a denominar-se de diagrama de definição de blocos, enquanto que o termo diagrama de estrutura composta passa a ser denominado de diagrama de blocos internos), que assim permitem uma melhor identificação da sua utilização.

3.7.1. Blocos

Refere-se a uma determinada parte da estrutura geral de um sistema e representa o estereótipo Bloco do elemento Classe, do UML.

Um bloco pode referir-se a uma unidade lógica ou física (podendo ser ambas). Quanto à sua representação textual, a notação utilizada difere da representação

normalizada de classes estereotipadas. O estereótipo Block pode ser ocultado e o rectângulo que representa o bloco possui um cabeçalho principal e várias divisões, com os respectivos cabeçalhos secundários.

Desta forma, é possível especificar os vários aspectos do bloco, sem qualquer incómodo ou confusão visual. Pode-se indicar cada uma das secções desta forma:

Cabeçalho **Values** – apresenta atributos com dados utilizados pelo SysML

Cabeçalho **Operations** – refere-se às operações executáveis pelo bloco

Cabeçalho **Constraints** – apresenta as condicionantes referentes ao bloco

Cabeçalho **Parts** – indica as propriedades compostas, existentes no bloco

Cabeçalho **References** – refere-se às propriedades não compostas, existentes no bloco

Cabeçalho **Properties** – utilizado quando não se pretende fazer distinções no bloco

Cada secção do bloco pode ser apresentada ou ocultada e pode-se adicionar secções personalizadas, de acordo com as intenções do utilizador. Todas as propriedades do bloco são públicas e visualizáveis por todos.

Relativamente à representação gráfica, o SysML permite representar cada uma das secções através de elementos visuais. Desta forma, é utilizado uma estrutura denominada de compartement, a qual contem um diagrama de blocos internos descrevendo como é estruturado o bloco. Cada uma das partes do bloco é representada através de um rectângulo, enquanto as referências são representadas através de linhas.

Além disto, o SysML permite a extensão dos blocos de acordo com o contexto onde se encontra inserido, de modo a evitar a criação de um novo bloco, derivado de um bloco existente e com meras variantes. No que concerne os valores utilizados no bloco, o SysML permite a definição dos valores iniciais numa divisão denominada de defaultValues.

Outra característica permitida pelo SysML refere-se à possibilidade de recuperar blocos internos, de segundo plano e elevada profundidade, para o primeiro plano, facto esse que não era possível no UML. Esta propriedade permite interligar blocos internos de diversas profundidades entre si, o que viola o princípio de encapsulamento existente nas linguagens orientadas a objectos.

No entanto, um bloco pode ser modelado como uma caixa negra, escondendo o seu funcionamento interno e fornecendo apenas acesso a determinados aspectos,

visíveis do exterior. Um bloco deste tipo será denominado de Encapsulated e não utilizará interfaces para os acessos externos (disponibilizadas pelo SysML), visto que estas estão associadas à engenharia de software apenas, não tendo representação física.

3.7.2. Tipos de Dados (Data Types)

O SysML, além dos tipos de dados definidos pelo UML, acrescenta mais alguns tipos novos. Assim, os tipos de dados presentes nesta linguagem são: Boolean, Integer, String, Unlimited Natural, Real e Complex.

3.7.3. Definição da Distribuição

Descreve como são distribuídos os valores de uma determinada gama, especificada anteriormente. Representa o estereótipo Propriedade Distribuída, do elemento Propriedade do UML.

As definições de distribuição têm de ser especificadas individualmente, através de estereótipos especializados. O SysML fornece como exemplos de distribuição, a distribuição intervalar e a distribuição normal.

Quanto à notação gráfica utilizada, corresponde à especificada para os estereótipos comuns, sem qualquer alteração.

3.7.4. Unidade e Dimensão

O elemento Unidade representa a estrutura de uma unidade física, representando o estereótipo Unidade do elemento Especificação de Instância do UML. Relativamente à Dimensão, descreve a quantidade de uma Unidade, representando o estereótipo Dimensão do elemento Especificação de Instância do UML.

Uma Unidade representa uma unidade física específica, sendo que a Unidade não é um tipo. As unidades são utilizadas na engenharia de sistemas para especificação de valores, os quais serão utilizados na execução e teste dos respectivos sistemas. O SysML incorpora na sua modelação de sistemas as definições para as unidades básicas do modelo SI, podendo mais unidades serem adicionadas à biblioteca da linguagem.

3.7.5. Tipo Valor (Value Type)

Um tipo de dados que define elementos que não têm identidade e não podem ser referenciados por um bloco, mas que têm no entanto uma dimensão ou unidade atribuída. Representa o estereótipo Tipo Valor, do elemento Tipo Dados do UML.

Este tipo trata-se de um Tipo Dados com duas propriedades adicionais (a unidade e dimensão), podendo-se assim definir uma unidade como um Tipo.

Quanto à representação gráfica, é idêntica ao Tipo Dados do UML, sendo que o estereótipo utilizado é valueType. Este estereótipo também é utilizado para identificar o cabeçalho onde a unidade e a dimensão são definidos.

3.7.6. Porto de Fluxo (Flow Port)

Descreve um ponto de interacção usado por um bloco para interagir com o seu ambiente externo, tais como outros objectos. Representa o estereótipo Porto de Fluxo do elemento Porta do UML.

Relativamente ao fluxo, existe também o elemento Especificação de Fluxo, que é utilizado para especificar dados de entrada e saída de um Porto de Fluxo. Representa o estereótipo Especificação de Fluxo, do elemento Interface do UML.

As interacções de um bloco com o seu ambiente externo podem ser realizadas de várias formas, pelo que existe a necessidade de representar todas essas possibilidades. No UML, tal era possível através de um Porto de Fluxo ou através da descrição de serviços, tal como acontece no SysML.

No entanto, o Porto de Fluxo inclui restrições aos objectos que podem ser transferidos para os blocos, sendo que a Especificação do Fluxo descreve quais os dados que podem ser transferidos.

No que concerne a representação gráfica da Especificação de Fluxo, a descrição dos dados tem um cabeçalho com o título Flow Properties, enquanto os dados de entrada tem o estereótipo In e os dados de saída, o estereótipo Out. Os dados transferidos em ambas as direcções possuem o estereótipo inout.

Relativamente ao Porto de Fluxo, esta possui pequenas setas de indicação no interface de entrada e saída. De referir que o Porto de Fluxo do bloco destino (que comunica com o bloco origem) deve utilizar a mesma Especificação de Fluxo, ao passo que a interface de entrada e saída de dados é invertida.

Além deste tipo de Porto de Fluxo, existe também um Porto de Fluxo Atómico, o qual permite a transferência de um tipo de dados (igual ao tipo do Porto de Fluxo), numa única direcção, não sendo necessária qualquer Especificação de Fluxo. Na

representação gráfica, a direcção da transferência é definida pela seta, presente na interface do Porto de Fluxo.

3.7.7. Fluxo de Itens (Item Flow)

Representa um fluxo de informação especial, que descreve num conector presente num diagrama de blocos internos quais os objectos que irão ser transferidos. Refere-se ao estereótipo Fluxo de Itens do elemento Fluxo de Informação do UML.

Visto que o Fluxo de Informação presente no UML apenas descrevia tipos abstractos, no SysML foi estendido para que pudesse utilizar o conector presente no diagrama de blocos internos, para descrição o fluxo dos objectos especificados. Esta descrição é feita através de uma propriedade definida no contexto do bloco.

Ao contrário do Porto de Fluxo (que descreve o que pode ser transferido), o Fluxo de Itens descreve quais os objectos que são realmente transferidos entre blocos.

3.7.8. Bloco de Associação (Association Block)

Descreve as propriedades estruturais de uma associação, sendo a representação do elemento Classe de Associação, presente no UML.

Além deste elemento, a Propriedade de Participação descreve o fim de uma determinada associação, presente numa estrutura interna do Bloco de Associação. Representa o estereótipo Propriedade de Participação do elemento Propriedade do UML.

Sendo o Bloco de Associação uma extensão da Classe de Associação, presente no UML, a novidade reside no facto de se poder descrever a estrutura deste tipo de blocos num diagrama de blocos internos. No que concerne a Propriedade de Participação, representa uma propriedade dos Blocos de Associação que contém a referência para uma propriedade especificada pelo participante.

Quanto à representação gráfica, enquanto o Bloco de Associação utiliza a notação normalizada para os blocos, a Propriedade de Participação é apresentada com setas, no diagrama de blocos internos, juntamente com o estereótipo end entre chavetas.

3.8. Diagrama Paramétrico (Parametric Diagram)

O SysML permite integrar nos seus diagramas diversas relações paramétricas entre as propriedades de um bloco (parâmetros). Isto permite, entre outros aspectos,

modelar relações físicas entre elementos de uma equação matemática (tal como a lei de Newton, por exemplo), possibilitando uma multiplicidade de hipóteses que poderão ser utilizadas pelos criadores de um sistema. No entanto, não representam um comportamento, apresentando-se somente como a relação entre as propriedades quantificadas de determinados itens.

Uma das principais vantagens reside na possibilidade de integrar o desempenho e a fiabilidade na modelação dos sistemas, bem como criar modelos específicos para esses aspectos. Desta forma, é possível analisar sistemas alternativos, tendo em conta aspectos como custos e tempo, entre outros. Além desta vantagem, a possibilidade de integração de equações de outras áreas ligadas à engenharia (Física, Química, Mecânica, entre outras), algo que era complicado de realizar no UML (senão impossível) [21], permite alargar o leque de utilização da notação a novas áreas, com as eventuais contrapartidas daí surgidas. Visto que é possível a especificação física das diversas reacções dos sistemas, torna-se assim possível utilizar a simulação virtual nestes modelos e eventualmente aplicar a notação para fins comerciais.

As relações entre os blocos podem ser descritas e definidas através dos diagramas paramétricos (uma variante do diagrama de blocos internos e utilizados para este propósito), sendo que todas as restrições definidas irão criar uma rede que irá afectar todos os blocos existentes no modelo.

3.8.1. Bloco de Restrição (Constraint Block)

Descreve as restrições aplicadas à estrutura do sistema e os parâmetros necessários para tal. Representa uma especialização do estereótipo Bloco do SysML.

Um Bloco de Restrição é definido sem ter em conta qualquer contexto, podendo ser reutilizado, levando à criação de uma biblioteca de restrições, facilmente aplicável a qualquer sistema modelado.

As restrições criadas são iguais às presentes no UML, podendo ser formuladas em qualquer linguagem. Relativamente ao Bloco de Restrição, este descreve os parâmetros das próprias restrições, sendo definidos como atributos.

Relativamente à notação gráfica, é similar à utilizada nos blocos, com a alteração do estereótipo block para constraint. Existem duas secções para cada Bloco de Restrição: constraints (utilizada para a representação das restrições) e parameters (utilizada para representar os parâmetros das restrições).

3.9. Diagrama de Actividade (Activity Diagram)

O Diagrama de Actividade presente no SysML apresenta o que será executado, qual a sua sequência, quais os dados que serão requisitados e apresentados. Para que tal seja possível, especifica quais os dados de entrada e saída necessários durante um determinado fluxo, podendo este ser paralelo, sincronizado ou dividido.

Relativamente ao UML, o Diagrama de Actividade acrescenta novas propriedades, tais como, controlo do fluxo actualizado (apresenta informações adicionais que permitem parar determinadas acções e controla o fluxo através de operadores de controlo), suporte para modelação de sistemas contínuos (fluxo de objectos discreto ou contínuo), probabilidade dos fluxos e regras de modelação para actividades (através do formato do diagrama de definição de blocos).

O Diagrama de Actividade do SysML é compatível com o diagrama EFFBD [15]., utilizado na engenharia de sistemas, através do estereótipo effbd, que maximiza a compatibilidade através da aplicação de restrições.

Relativamente à notação gráfica, o Diagrama de Actividade presente no SysML permite que um Fluxo de Controlo seja apresentado com uma seta, permitindo uma melhor distinção entre os vários elementos presentes.

3.9.1. Árvore de Funções (Function Tree)

As actividades representam blocos especiais, podendo ser executáveis. Desta forma, é possível generalizar as actividades e permitir a sua participação em associações, levando à criação de uma árvore de funções.

É possível invocar acções de outras acções, dentro de uma actividade, sendo que graficamente, as acções são representadas com uma nova divisória de saída. Se a chamada for contínua, significa que a execução de uma determinada acção chamada é terminada, mal essa mesma chamada é terminada.

Os nós de objectos podem ser representados numa actividade, representando desta forma uma associação de uma actividade com o elemento que descreve o tipo do objecto.

A utilização das árvores de funções deve ser equacionada, visto que a sua integração no modelo é complexa, devido aos vários passos necessários para a sua modelação correcta.

3.9.2. Operador de Controlo

Especifica um comportamento que pode levar à activação ou desactivação das acções, através dos respectivos valores de controlo. Representa o estereótipo Operador de Controlo presente no elemento Comportamento e Operação do UML.

Quanto ao valor de controlo, é um Tipo Enumeração com valores, que descreve o controlo efectuado pelo Operador de Controlo.

Tendo em conta que as actividades de um sistema contínuo envolvem inúmeras acções, por sua vez contínuas, o Operador de Controlo irá permite o controlo externo dessas mesmas acções. Para tal, irá introduzir valores de controlo (que poderão variar entre enable e disable, podendo ser acrescentados mais valores), que irão iniciar ou terminar a execução de uma determinada acção, no respectivo sistema.

Quanto à representação gráfica, é utilizado o estereótipo controlOperator para identificar o Operador, além de poderem ser representadas as ligações do mesmo aos restantes elementos.

3.9.3. Taxa (Rate)

Descreve a frequência pela qual os elementos entram e saem de um determinado parâmetro, localizado no limite de uma actividade. Representa o estereótipo Taxa, dos elementos Parâmetro e Limite de Actividade.

As transferências entre acções, nós de objectos e parâmetros de actividades são efectuadas através de um determinado limite, sendo que a cadência pela qual são transferidos é importante para a especificação do próprio fluxo. Para tal, é utilizada a Taxa, a qual permite especificar essa mesma cadência, através de uma frequência pré-estabelecida.

A Taxa pode ser dividida em dois tipos: Taxa Contínua e Taxa Discreta, sendo que a primeira é representada graficamente através do estereótipo continuous e a segunda através do estereótipo discrete. Ainda na notação gráfica, a Taxa é apresentada dentro de chavetas, junto ao limite de uma actividade.

3.9.4. Nó de Objecto Especial (Special Object Node)

Um Nó de Objecto Especial pode ter determinadas propriedades específicas, não presentes num nó de objecto convencional.

Uma das propriedades é representada graficamente através do estereótipo nobuffer e descreve nós de objectos que podem rejeitar determinados dados, se eles não puderem ser aceites pelas respectivas acções e limites apresentados.

A segunda propriedade consiste na descrição de nós de objectos que podem reescrever dados que possuam, por dados recebidos posteriormente. A sua representação gráfica é feita através do estereótipo `overwrite`.

Finalmente, a última propriedade descreve os parâmetros que não necessitam de valores para executar uma determinada acção, sendo a sua representação gráfica especificada através do estereótipo `optional`.

Relativamente a um nó de objecto normal, cujo comportamento é igual ao de um pilha (recebe elementos e faz o seu tratamento, através do princípio `First In First Out`), estas propriedades alteram substancialmente o comportamento de um nó de objecto convencional e permitem alterar a ordem de execução.

Por exemplo, a propriedade `nobuffer` permite ignorar o comportamento do tipo pilha, evitando um sobre carregamento de dados de entrada com fluxo contínuo. Relativamente à propriedade `overwrite`, esta permite que na eventualidade do nó estar cheio, se elimine os dados armazenados mais antigos, para dar lugar aos dados de entrada mais recentes.

Desta forma, é possível alterar substancialmente o funcionamento de um nó, a ser utilizado em determinadas circunstâncias bastante específicas.

3.9.5. Probabilidade

Descreve a probabilidade de utilização de um limite de saída de um nó, por determinados dados. Representa o estereótipo `Probabilidade`, do elemento `Limite de Actividade`, presente no UML.

Visto que os limites de saída de um nó podem ter condições de escolha, a opção é tomada caso uma determinada condição seja verdadeira.

O cálculo da probabilidade dessa escolha permite ter uma melhor percepção do funcionamento do nó, sendo que o valor da probabilidade de uma opção varia entre 0 e 1. O resultado da soma das probabilidades de todas as opções de escolha de um nó tem de apresentar um valor 1.

Relativamente à representação gráfica, a probabilidade é apresentada junto ao limite de uma actividade, especificando o seu valor mínimo e máximo.

3.10. Alocação

O mecanismo de alocação permite interligar elementos de diferentes áreas de modelação, sendo que esses aspectos não são perceptíveis ao engenheiro de sistemas, tal é a sua profundidade e detalhe.

Um exemplo que permite verificar a utilidade da alocação consiste na especificação de um fluxo de objecto, presente num diagrama de blocos internos, como um fluxo de objecto de um diagrama de actividades. Tal especificação apenas é indicada no modelo completo, através de detalhes bastante complexos e pouco visíveis, pelo que a introdução da alocação permite modelar as ligações, numa fase precoce do projecto e com relativa facilidade.

O SysML propõe 3 tipos de alocações gerais (alocação de comportamento, alocação de estrutura e alocação de fluxo de objectos), os quais podem ser alterados pelos modeladores, de acordo com o seu projecto. As diferenças entre cada um dos tipos podem ser referidas desta forma:

Alocação de Comportamento – permite para alocar um comportamento (actividade ou acção) a um bloco que execute tal comportamento.

Alocação de Estrutura – permite separar as estruturas lógicas das estruturas físicas, facilitando o desenvolvimento do sistema em partes faseadas. Esta alocação permite a indicação de elementos de cada uma das áreas.

Alocação de Fluxo de Objecto – permite ligar um fluxo de um elemento, presente num diagrama de estrutura, com um fluxo de objecto, presente num diagrama de actividade.

3.10.1. Relação Alocação (Allocation Relationship)

Esta relação disponibilizada pelo SysML trata-se de uma relação abstracta entre elementos de diferentes tipos ou níveis. Permite alocar um elemento final a um ou mais elementos iniciais. Representa o estereótipo Alocar do elemento Abstracção do UML.

Visto que elementos como o comportamento ou estrutura são analisados separadamente na fase de modelação, a sua interligação só é realizada numa fase posterior. Para que seja possível efectuar a mesma numa fase inicial da modelação, pode-se recorrer à utilização da Relação Alocação, a qual permite modelar este aspecto, sem necessitar de especificar detalhadamente o aspecto final da alocação.

A notação gráfica da Relação Alocação utiliza uma seta com o estereótipo Allocate, indicando qual a alocação efectuada. Além desta notação gráfica, é possível introduzir um comentário, quer no elemento inicial, como no elemento final.

Esta notação pode ser utilizada preferencialmente, visto que existe interligação de vários tipos de diagramas, pelo que os seus elementos nunca estarão presentes num único diagrama. Além destas duas notações de representação das alocações, é

possível utilizar uma terceira, na qual se reparte as alocações por blocos, utilizando os estereótipos `allocatedFrom` e `allocatedTo` para indicar a origem e destino de cada uma das alocações.

3.10.2. Alocação de Partição de Actividade (Allocate Activity Partition)

Trata-se de uma Partição de Actividades específica, que permite alocar cada acção dentro de uma partição a uma estrutura que represente a mesma partição. Representa o estereótipo `Alocar Partição de Actividades` do elemento `Partição de Actividades` do UML.

Esta alocação faz parte do tipo `Alocação de Comportamento` e permite agrupar as acções pelas suas características, formando um bloco para o qual estas acções são alocadas. A acção passará a ser o elemento final de uma `Relação de Alocação`, enquanto a estrutura será o elemento inicial da mesma.

3.10.3. Notação Tabela

Além do que foi referido para os diagramas de requisitos, as tabelas presentes no SysML permitem a representação das alocações, indicando detalhes como o elemento inicial e elemento final, nome, entre outros aspectos.

Existe também uma matriz adicional que permite a descrição das várias relações presentes no SysML. No entanto, na especificação actual, apenas permite a representação de alocações e dos seus pormenores.

3.11. Elementos de Modelação Genéricos

Existem vários elementos que não podem ser enquadrados num diagrama específico, visto que podem ser utilizados em diversos diagramas. De seguida serão apresentados estes mesmos elementos.

3.11.1. Rationale

Descreve os princípios e a motivação por detrás de uma determinada escolha de modelação. Representa o estereótipo `Rationale` do elemento `Comentário`, presente no UML.

Visto que é difícil a um observador entender o modelo, especialmente se este for complexo, deve-se comentar o mesmo, utilizando para isso o `Rationale`. Este elemento será utilizado para descrever as escolhas efectuados no modelo, bem como ajudar na

sua compreensão. Para a representação gráfica, é utilizado o estereótipo rationale na sua apresentação.

3.11.2. Janela do Diagrama (Diagram Frame)

A Janela do Diagrama permite identificar o tipo e nome do elemento do modelo que se encontra modelado, bem como o diagrama e o tipo de do modelo a que pertence. Pode recorrer a abreviaturas para identificação (não obrigatório), derivadas do UML e devidamente adaptadas às novas nomenclaturas do SysML.

3.11.3. Visão do Modelo (Model View and Viewpoint)

Descreve a representação do sistema na sua totalidade, vista de um Ponto de Vista definido. Representa o estereótipo Visão do elemento Pacote, presente no UML.

Relativamente ao Ponto de Vista, este especifica a estrutura da Visão do Modelo utilizada, com base em objectivos definidos e requisitos apresentados. Representa o estereótipo Ponto de Vista do elemento Classe, existente no UML.

Finalmente, a Relação Conformidade faz a ligação entre um Ponto de Vista com a Visão do Modelo e verifica quais os requisitos que se encontram satisfeitos. Representa o estereótipo Conformidade, o elemento UML denominado de Dependência.

3.11.4. Problema

Documenta um potencial erro ou falha no modelo ou sistema. Representa o estereótipo Problema, do elemento Comentário, presente no UML.

Este elemento permite identificar potenciais problemas e documentar essa informação no modelo. Desta forma, é possível resolver essas falhas numa fase posterior da modelação do sistema, em que os recursos para a sua resolução estarão disponíveis.

3.12. UML Testing Profile

O UML Testing Profile, definido pela OMG em Abril de 2002, é um perfil para o UML que permite a captura de toda a informação necessária para avaliar a correcção das implementações de sistemas criadas [2]. O seu objectivo é fornecer o suporte para testes eficientes e automatizados, sobre os modelos UML e SysML criados. Os objectivos, estabelecidos pela OMG e que deveriam ser cumpridos, eram os seguintes:

- Possibilitar a especificação de testes para aspectos estruturais (estáticos) e comportamentais (dinâmicos) dos modelos computacionais criados com o UML
- Ser baseado no meta-modelo do UML
- Ser capaz de operar com tecnologias de teste existentes

Este perfil é baseado no UML 2.0, estando dividido em 3 pacotes:

Teste de comportamentos – analisa as observações e actividades realizadas durante o teste

Teste de arquitectura – contém os elementos e as suas relações, envolvidas no teste

Teste de dados – verifica as estruturas e significado dos valores processados durante o teste

De referir que um modelo de teste, desenhado utilizando este perfil, necessita que se especifique os objectivos para o teste, bem como os correspondentes casos de uso. Além do UML Testing Profile, existem outras implementações concorrentes que se encontram de acordo com o documento apresentado pela OMG, tais como o AML, apresentado pela AGEDIS [2].

3.13. Discussão

A partir desta análise teórica, pode-se definir alguns pontos fortes e fracos, bem como estabelecer alguns elementos, apresentados pela literatura, cuja validação será discutida no caso prático.

3.13.1. Pontos fortes e fracos

De acordo com [21], [4] e [8], existem diversas pontos fortes e fracos da utilização do SysML, podendo-se resumir da seguinte forma:

Pontos fortes

- Utilização de semânticas específicas da engenharia de sistemas, bem como de dois novos diagramas.
- Maior facilidade de aprendizagem do SysML, relativamente ao UML.
- Linguagem mais compacta e simplificada do que o UML.
- Suporte para vários tipos de tabelas de alocação

- Aplicação de equações matemáticas, representativas do mundo real, nos modelos criados (utilizando diagramas paramétricos)

Pontos fracos

- Pouca experiência com a utilização do SysML na modelação de simulações
- Metodologia de utilização da linguagem inexistente
- Inexistência de exemplos avançados da linguagem
- Herda problemas já verificados no UML, tal como a definição vaga e imprecisa da semântica, além da elevada extensão da linguagem
- Necessita de ser estendida para providenciar suporte para especificidades de certas engenharias, não incluídas inicialmente

3.13.2. Análise técnica

A partir de diversos artigos técnicos referentes ao SysML, irão ser detalhadas com maior precisão algumas das principais falhas desta notação. Começando pelo artigo [4], os três autores (Edward Huang, Randeep Ramamurthy e Leon F. McGinnis) destacam um problema inerente a esta linguagem: a utilização de mecanismos de controlo e a sua representação nos modelos analíticos. Esta é uma área que se encontra em investigação, prometendo mais evoluções num futuro próximo.

Outro aspecto que apresenta alguns problemas refere-se ao desenvolvimento de meta modelos para determinados domínios em particular, os quais iriam fornecer uma plataforma entre as aplicações de simulação (baseadas na literatura existente) e a notação em si.

No entanto, aspectos como a integração de modelos de simulação de eventos discretos, referentes à manufacturação e logística, apresentam grande potencial, podendo facilitar a transição das linguagens dedicadas a simulação para uma linguagem independente e que permita a criação de um modelo de um sistema de sistemas, facilmente perceptível por todos.

Concluindo a análise deste artigo, os seus autores referem também a problemática que envolve a verificação do sistema modelado e da sua concordância com o sistema real, processo esse que até à criação do SysML, era tipicamente ad-hoc. Com a nova notação a introduzir o conceito de modelos executáveis (devido à sua derivação do UML), torna-se possível automatizar a geração de software de simulação, tal como era possível no UML. No entanto, a falta de documentação e

exemplos de tais utilizações são pormenores que condicionam este tipo de utilização da notação.

Referindo o artigo [8], os autores Mihai Fonoage e Ionut Cardei preferem destacar as desvantagens da linguagem SysML. Entre os pontos menos positivos da notação, de destacar as semânticas complexas, bem como a necessidade de expandir a linguagem para fornecer suporte para o desenho de diversos sistemas (sistemas sem fios, por exemplo).

Desta forma, a linguagem, sendo compacta e simplificada, deixa o desenvolvimento de diversos aspectos mais específicos a cargo dos seus utilizadores. Este mesmo aspecto poderá levar a um processo inicial de adaptação da linguagem à temática em causa e maior demora no arranque dos projectos.

No que concerne o artigo [10], os autores Michel dos Santos Soares e Jos Vrancken apresentam os diagramas de requisitos como uma ponte entre a engenharia de requisitos (que recorre frequentemente a elementos textuais para apresentação dos requisitos) e os casos de uso, implementados inicialmente no UML.

Uma vez que grande parte dos projectos falhados, têm os seus principais problemas na fase de elaboração dos requisitos, esta é uma área que deve ser abordada com algum rigor. Além disso, a existência de demasiado detalhe inicial também restringe a liberdade criativa dos projectistas, levando a soluções mais convencionais e menos criativas.

O diagrama de requisitos apresenta-se por isso como uma ferramenta normalizada, contribuindo para um melhor entendimento das necessidades do sistema. Além disso, o recurso a tabelas de requisitos permite detalhar as relações do sistema, bem como contribui para uma melhor decomposição interna.

Analisando o artigo [22], da autoria de Yariv Grobshtein, Valeriya Perelman, Eliyahu Safra, Dov Dori, podemos constatar que apesar de ser uma linguagem mais simples de ser utilizada, quando comparada com uma linguagem de modelação concorrente, a OPM (Object Process Methodology), poderá ser eventualmente considerada algo complexa. Este aspecto parte sobretudo dos inúmeros diagramas existentes na linguagem, que dificultam a compreensão geral do sistema, especialmente para alguém que não tenha experiência anterior com UML.

Neste comparativo em particular, a linguagem OPM apresentou um modelo mais limpo e compreensível, sendo também mais fácil de navegar. O SysML, pelo contrário, apresentava um modelo demasiado detalhado, o que em conjunto com a multiplicidade de diagramas existentes nesta linguagem, levou a um dificultar da

aprendizagem numa fase inicial. No entanto, o detalhe existente nos modelos criados é também um aspecto positivo, visto que ajuda a detalhar todos os aspectos do sistema em causa.

Resumidamente, apesar de ser mais complexo e exigir conhecimentos de diversos conceitos subjacentes à linguagem em si, o SysML apresenta capacidades que não estão ao alcance de outras linguagens mais simples, mas também bastante limitadas. A curva de aprendizagem apresentada pelo SysML é relativamente elevada, devido sobretudo à sua multiplicidade de diagramas, que permitem fornecer uma visão final do sistema coerente e credível.

No que concerne o artigo [9], redigido por Mariusz Grecki, Zheqiao Geng, Gohar Ayvazyan, Stefan Simrock, Hamburg Bahtiar Aminov, a implementação de um sistema de sistema como o sistema de controlo LLRF, utilizando um arquitectura ATCA, permitiu concluir que a combinação da linguagem SysML com metodologias de engenharia de sistemas, permite atingir resultados bastante interessantes.

Visto que o projecto tem um carácter europeu, o envolvimento de diversas universidades, laboratórios de investigação e indústrias teve um grande destaque, pelo que a utilização de uma linguagem única e perceptível a todos era necessária. A adopção da notação SysML e das suas ferramentas de modelação facilitou a comunicação entre os vários elementos do projecto, além de ter permitido estimar com fiabilidade os custos de hardware e software de todo o projecto. As equações e regras utilizadas para tais cálculos encontram-se definidas na maioria das ferramentas, podendo no entanto ser alteradas, caso assim seja necessário.

Um problema também referido por este artigo trata-se da integração de diagramas provenientes do UML 2.0 e que sofreram poucas ou nenhuma alterações. Visto que a especificação do SysML é relativamente pequena, este aspecto torna a aprendizagem por parte de quem não tem qualquer experiência com UML bastante difícil. Uma consideração referida é o facto de a leitura dos modelos criados ser bastante mais simples do que a sua criação, demonstrando que embora o conhecimento prévio de UML seja necessário para o desenvolvimento de sistemas de sistemas, tal não se verifica para a leitura e compreensão dos modelos criados.

Em resumo, o artigo apresenta diversas conclusões que permitem indicar o SysML como uma linguagem acessível e capaz de ser utilizada pelos diversos ramos da engenharia, sem que os seus conhecimentos diferenciados coloquem entraves à comunicação. A linguagem apresenta assim um potencial bastante interessante,

podendo a sua utilização levar à diminuição do tempo previsto para o período entre a criação do modelo do sistema e a sua implementação no terreno.

Outros aspectos que poderão criar alguma apreensão por parte dos utilizadores serão as eventuais actualizações à linguagem. Num espaço de 6 meses e até ao lançamento da versão 1.0, em Abril de 2006, foram lançadas 4 actualizações intermédias, que alteravam consideravelmente a utilização da linguagem. No entanto, tal ritmo baixou consideravelmente, tendo-se apenas registado o lançamento da versão 1.1, em Novembro de 2008, pelo que é credível que a linguagem não será actualizada tão frequentemente. Desta forma, será possível ter uma especificação estável, sobre a qual se poderá desenvolver os sistemas de sistemas.

Podemos assim concluir que, apesar das diversas lacunas ainda existentes no SysML, a sua utilização na engenharia de sistemas deve ser tida em consideração. No cômputo geral, a linguagem oferece boas capacidades de modelação aos engenheiros, podendo ser adaptada às suas áreas específicas e respectivas necessidades. Embora seja relativamente recente, a sua aprendizagem é relativamente rápida e existe documentação dedicada à linguagem (pouca mas com qualidade), pelo que a sua expansão futura deverá ser interessante.

3.14. Conclusão

A utilização do SysML permite colmatar várias lacunas presentes no UML e que impossibilitavam a utilização do mesmo noutras áreas de engenharia, além da engenharia de software. A presença de novos elementos, tais como estereótipos, notações gráficas, entre outros aspectos, permite que o SysML seja acessível a quem já teve alguma (mesmo que pouca) experiência de utilização do UML. No entanto, visto que é uma linguagem distinta das restantes utilizadas noutras áreas (inclusive na engenharia de software), poderá levar a um período de adaptação inicial à mesma.

Um aspecto interessante do SysML reside na sua capacidade de abstrair a complexidade existente em diversos sistemas, apresentando apenas os detalhes principais, necessários para a compreensão dos mesmos. Além disso, visto que é independente da área onde é aplicado, permite uma melhor comunicação entre os vários elementos existentes numa equipa de desenvolvimento.

Outro aspecto a realçar no SysML prende-se com a sua redução de erros e ambiguidades, através do fornecimento de uma notação precisa e livre de possíveis erros de interpretação. A presença de elementos como requisitos (Secção 3.6) e

alocações (Secção 3.10) permite também verificar a aplicabilidade do modelo criado e, caso seja necessário, permite adaptá-lo às exigências apresentadas.

Um aspecto que poderá parecer algo simbólico, mas que se apresenta como um passo importante para a adopção da notação, trata-se da criação e utilização de diagramas paramétricos. Estes diagramas apresentam-se como uma novidade bastante importante, visto que permitem a integração de aspectos físicos (derivados da Física, Química, entre outros) na modelação de sistemas de sistemas, possibilitando também que a sua simulação também seja possível. Embora possam ser aspectos que tenham a sua quota-parte de imprevisibilidade, recorrendo às equações matemáticas existentes, é possível introduzir esses parâmetros e considerá-los na elaboração de um determinado modelo.

Finalmente, a elaboração de modelos utilizando o SysML permite definir várias especificações possíveis para um determinado sistema, documentá-las e avaliar o seu desempenho e a sua aplicação à tarefa pretendida. No entanto, o SysML não ditará a sua implementação física no terreno, podendo esta ser alterada em função do meio ambiente onde o sistema for inserido e as suas condicionantes.

Desta forma, pode-se considerar o SysML como uma ferramenta interessante para ser utilizada na criação e desenvolvimento de sistemas para as diversas áreas de engenharia.

4. Ferramentas e Notações

Embora tenham sido referidos teoricamente, os elementos estruturantes da notação SysML, a sua aplicação prática é efectuada com recurso a aplicações informáticas, criadas especificamente para a modelação de sistema de sistemas. Na sua maioria, derivam de ferramentas utilizadas à modelação com a notação UML, existindo no entanto algumas que foram criadas especificamente para o SysML. No todo, estas permitem modelar um sistema completo, recorrendo a todas as funcionalidades disponibilizadas pela notação.

4.1. Ferramentas disponíveis

Apesar de ser uma notação relativamente recente, o SysML apresenta suporte de diversas ferramentas de modelação de renome. As ferramentas que apresentam esse mesmo suporte encontram-se indicadas na Tabela 6.

Ferramentas	Condições
Magic Draw	Necessita da instalação de software para suporte do SysML
Enterprise Architect	Suporta o SysML após a instalação do MDG Technology for SysML, desenvolvido pela Sparx Systems
Rational Software Modeler/Architect	Necessita da instalação do SysML Toolkit, desenvolvido pela EmbeddedPlus
Rhapsody	Não necessita de instalações adicionais
TAU G2	Não necessita de instalações adicionais
Artisan Studio	Suporta a notação SysML sem qualquer instalação adicional
TOPCASED-MF	Suporta a notação SysML sem instalações adicionais
Papyrus 4 UML	Necessita da instalação do Papyrus for SysML

Tabela 6. Ferramentas com suporte para o SysML

Como se pode verificar, necessitando ou não de instalação adicional, todos as principais ferramentas de modelação disponíveis no mercado apresentam suporte para a notação SysML. Embora algumas apresentem um suporte mais integrado com a aplicação original e conseqüentemente, mais fácil de utilizar, todas as ferramentas possuem as capacidades necessárias para a criação de sistemas complexos utilizando SysML.

4.2. Outras notações

Embora o SysML seja a principal notação abordada nesta Tese, existem outras notações alternativas, que permitem modelar alguns dos elementos e sistemas descritos pelo SysML, bem como introduzir a modelação de sistemas em novas áreas de aplicação.

4.2.1. MARTE

A notação MARTE (Modeling and Analysis of Real-Time and Embedded Systems) trata-se de um perfil do UML [14], estendendo-o para o desenvolvimento de sistemas embebidos e de tempo real.

Incorporando os conceitos fundamentais para a modelação e análise dos problemas desta área, a notação MARTE providencia suporte para as fases de especificação, desenho, verificação e validação. Não definindo novas técnicas de análise, pretende suportar as técnicas existentes, bem como as futuras. Além disto, fornece os recursos necessários para complementar os modelos com informação adicional, necessária para a execução de análises específicas.

A notação foca-se na análise do desempenho e da escalabilidade, definindo também uma base para análises quantitativas. Como principais benefícios, fornece um meio comum de modelar elementos de hardware e software dos sistemas de tempo real, permite a interoperabilidade entre ferramentas de desenvolvimento e analisa as características de hardware e software dos sistemas modelados, bem como os seus requisitos.

Esta notação divide-se em duas áreas (definidos em dois pacotes): desenho do sistema (incorporando as características dos sistemas embebidos e de tempo real) e análise (suportando a análise das propriedades dos sistemas). Tanto uma área como outra têm em comum a preocupação com aspectos como tempo e recursos concorrentes. Estes aspectos estão definidos na notação, através do pacote Fundações. Terminando, a notação também contém um pacote Anexos, o qual contém bibliotecas adicionais, bem como perfis complementares.

4.2.2. AADL

Tendo partido da iniciativa de Bruce Lewis (pertencente ao ARMDEC, Aviation and Missile Research, Development, and Engineering Center) e Peter Feiler, o AADL consiste numa norma internacional da indústria, a qual se foca na modelação da arquitectura em tempo de execução dos sistemas de software integrado.

Esta norma inclui diversas tecnologias fundadas pela DARPA (Defense Advanced Research Projects Agency) e abrange áreas como plataformas de integração de software embebido, ambiente físico em que o sistema embebido interagirá, temporização, entre outros aspectos.

A utilização desta notação é dividida por 4 fases: Engenharia de Requisitos, Desenho do Sistema, Desenho da Arquitectura de Software, Desenho dos Componentes de Software. Enquanto na primeira fase, é utilizada uma abordagem de alto nível (com aspectos como o orçamento e segurança a terem o principal destaque), no desenho do sistema é identificada a plataforma de computação, o interface com o ambiente físico e os principais componentes do software embebido, elementos a partir dos quais são tomadas as decisões relativas à arquitectura do sistema.

No desenho da arquitectura de software, a arquitectura de comunicações e tarefas do sistema embebido é refinada, além da tomada de decisões relativa a uma plataforma de computação em particular. Finalmente, no desenho dos componentes de software são desenhados e codificados os respectivos elementos, sendo estes divididos num conjunto de especificações funcionais e não-funcionais.

Em cada uma destas fases, diversas validações podem ser efectuadas, de forma a se determinar se a modelação das diversas características do sistema estão a ser efectuadas de forma pretendida.

A norma AADL encontra-se em aplicação nas seguintes iniciativas industriais: ASSERT (Automated System and Software Engineering for Real-Time Applications, fundada pela Comissão Europeia), TOPCASED (Toolkit in Open Source for Critical Applications & Systems Development, liderada pela Airbus) e SPICES (Support for Predictable Integration of Mission Critical Embedded Systems, focada na análise de problemas metodológicos inerentes ao uso do AADL).

4.2.3. SystemC

O SystemC é uma linguagem de descrição de sistemas [20], que fornece capacidade para modelação de comportamentos e transacções. Consiste num conjunto de rotinas e macros implementadas em C++, disponibilizadas por várias bibliotecas, permitindo assim simular processos concorrentes.

O SystemC permite também que os objectos descritos possam comunicar num ambiente de tempo real simulado, utilizando para isso os diversos tipos de dados disponibilizados pelo C++ e pelo próprio SystemC. A linguagem é estruturada em torno de uma hierarquia bem definida e delineada.

Destacando os módulos, estes são os principais blocos de construção de modelos SystemC. Geralmente, um modelo consistirá num conjunto de módulos que comunicarão entre si, utilizando para isso portas, interfaces e canais.

Sendo os canais, o principal elemento de comunicação do SystemC [20], estes podem variar desde simples fios directos até complexos mecanismos de comunicação, tais como barramentos. Além de poderem ser definidos pelos utilizadores (os quais estabelecem os interfaces, canais e portas utilizados), existem também 5 tipos de canais genéricos, pré-definidos pela linguagem: sinais de hardware, buffer, FIFO (First In First Out), exclusão mútua e semáforos.

Detalhando os aspectos de comunicação existentes no SystemC, devemos referir que uma interface consiste num conjunto de métodos implementados por um canal, sendo que esse mesmo canal pode implementar diversas interfaces. Relativamente às portas, estas servem de ligação entre os módulos e os canais, especificando também a interface requisitado para tal e fornecendo o acesso aos métodos disponibilizados pela mesma aos módulos.

No que concerne os canais, os mesmos podem ser primitivos ou hierárquicos. No primeiro caso, são entidades atómicas, que não contém qualquer estrutura interna visível e podem utilizar um esquema do tipo requisito/actualização, baseado em sinais de hardware. Quanto ao segundo caso, são módulos que podem implementar interfaces e utilizar portas, além de conter processos, módulos e canais. Ambos os tipos implementam interfaces.

Em resumo, o SystemC consiste numa linguagem de descrição e simulação, podendo o código compilado utilizar a componente de simulação para fornecer um ficheiro executável, cujo comportamento é exactamente igual ao modelo desenhado. No entanto, o desempenho desta componente de simulação é inferior à verificada nos simuladores projectados para VHDL e Verilog. A linguagem apresenta bastantes similaridades ao nível da semântica com o VHDL e o Verilog, oferecendo no entanto uma maior liberdade de desenvolvimento, através da utilização de templates e da programação orientada a objectos.

4.2.4.JMCAD

O JMCAD trata-se de uma linguagem que permite a criação de diagrama de blocos [7], tal como o SysML. Utilizando para isso um programa para modelação e simulação de sistemas dinâmicos complexos, possibilita a criação de diagramas de

blocos, em conjunto com a sua simulação, de modo visual e bastante simplificado, facilitando também a posterior alteração e manutenção.

Além disto, a sua velocidade de execução rápida é adequada para a utilização de modelo no treino de operadores, afinação de controladores sem acesso à rede e testes sucessivos de hardware.

O motor matemático de simulação providencia soluções rápidas e precisas para o desenho de sistemas lineares, não-lineares, híbridos, contínuos e discretos, pelo que os utilizadores poderão desenvolver protótipos de sistemas e processos que permitam demonstrar o comportamento dos mesmos, antes de avançarem para a construção dos protótipos.

A partir de uma biblioteca de blocos, o utilizador pode seleccionar um bloco pré-definido ou escrever o seu próprio bloco, sendo que cada um destes blocos desempenha uma função. Além destas capacidades, os utilizadores podem programar os seus blocos em Java e adicioná-los à biblioteca do JMCAD.

O JMCAD fornece suporte para várias ferramentas adicionais de áreas tão diversas como análise de frequências, criação de código Java, modelação de sistemas de comunicação, desenho de sistemas embebidos, redes neuronais e entrada e saída de dados digitais e analógicos.

4.2.5. Modelica

Sendo uma linguagem destinada à modelação de sistemas físicos complexos, o Modelica permite a descrição de variados sistemas, tais como sistemas eléctricos, hidráulicos e térmicos. No entanto, ao contrário do SysML, o Modelica é uma linguagem textual, na qual se descreve todas as partes do modelo, bem como a sua estrutura interna [12].

Um modelo Modelica contém componentes (tal como um cilindro, uma resistência ou outro elemento), sendo que esses componentes podem ser ligados entre si por conectores (também denominados de portas), os quais descrevem as suas possibilidades de interacção. A criação de ligações entre componentes leva também à criação de sub-modelos, pelo que um modelo em Modelica por conter vários modelos secundários, respeitantes a cada um dos seus componentes internos. Todos os elementos presentes no Modelica são criados utilizando equações discretas, diferenciais e algébricas.

Tratando-se de uma linguagem gratuita, começou a ser desenvolvida pela associação Modelica em 1996, tendo a mesma empresa desenvolvido uma biblioteca

de componentes e funções genéricas para o Modelica, denominada de Modelica Standard Library. O Modelica, embora semelhante ao C++ e Java, não se trata de uma linguagem de programação pura, visto que os componentes criados não são compilados mas sim traduzidos para objectos, os quais são executados pelo modelo de simulação (o qual não é definido pela própria linguagem, sendo um elemento externo).

O Modelica também pode ser expandido, utilizando para isso alguns produtos comerciais e gratuitos que lhe conferem novas capacidades. Entre esses produtos, podemos destacar o Dymola, um interface criado pela empresa Dynasim AB para modelação e simulação de sistemas ligados à indústria automóvel, aeroespacial e robótica, o MathModelica System Designer, um ambiente gráfico para modelação e simulação de sistemas dinâmicos multi-domínios e o Scicos, uma derivação gratuita do Modelica destinada essencialmente para a simulação.

4.2.6. Bond Graph

A notação Bond Graph trata-se de uma descrição gráfica para um sistema físico dinâmico, utilizando para isso a estrutura energética do mesmo [3]. Desta forma, é possível criar modelos matemáticos precisos para os referidos sistemas, sendo que a notação faz uso dos fluxos de energia entre os diversos componentes do sistema para esses cálculos.

Criada em 1959 por H.M. Paynter, cuja ideia inicial consistiu em definir os sistemas pelas suas trocas energéticas, utilizando para isso os elementos físicos em ligação com as estruturas de junção (onde são definidas as restrições existentes). A partir desta ideia inicial surgiu a notação Bond Graph, a qual pode ser orientada relativamente à potência ou à informação. Posteriormente, esta notação foi desenvolvida por outros investigadores, os quais expandiram a notação para outros domínios tais como a termodinâmica e hidráulica.

Com o Bond Graph, é possível representar um sistema físico através de símbolos e linhas, identificando assim os caminhos associados aos fluxos energéticos do mesmo. Elementos como resistência, capacitância e herança são interligados entre si utilizando junções e ligações, resultando numa rede complexa.

Graças à análise dos fluxos de energia entre os componentes, é possível obter as trocas de energia entre os vários elementos do sistema, o que possibilita a modelação de qualquer tipo de sistema físico (mecânico, eléctrico, hidráulico, entre

outros). Segundo o Bond Graph, a transmissão de potência entre componentes resulta da combinação de um esforço e fluxo, variando estes de sistema para sistema.

Na notação Bond Graph, é também possível introduzir relações de causalidade entre os diversos factores de potência, indicando as direcções de esforço e fluxo.

4.3. Conclusão

Como se pode constatar, o SysML não é a primeira notação gráfica existente, bem como também não é a única notação destinada à modelação de sistema de sistemas. No entanto, é a primeira que permite oferecer uma abordagem única a todos os intervenientes do projecto, sem interferir com os conhecimentos técnicos implícitos a cada um deles.

Enquanto outras notações privilegiam a componente matemática e física, fornecendo possibilidades de simulação mais detalhadas e avançadas (tais como o Modelica e o JMCAD), outras apresentam novas formas de pensamento sobre a própria elaboração do sistema (caso do Bond Graph e SystemC). No entanto, nenhuma delas permite se apresenta tão fácil de utilizar (algumas chegam a apresentar-se no formato textual), nem tão abrangente como o SysML (devido sobretudo às suas próprias restrições impostas pelas ferramentas e pelos âmbitos de utilização das mesmas).

Embora o SysML não possa ser considerado como a notação “ideal” para a modelação de sistema de sistemas, a sua utilização permite abranger diversos aspectos que outras linguagens não atingem. No entanto, a integração do SysML com essas mesmas linguagens poderá permitir obter resultados mais eficazes noutras áreas, tais como a simulação.

5. Caso Prático

O caso prático sobre o qual incide esta fase do desenvolvimento da Tese trata-se da modelação de um sistema de sistemas real, utilizado nos Laboratórios do Departamento de Engenharia Química e desenvolvido pela Engenheira Teresa Pimenta.

5.1. Descrição do Caso Prático

Com base no seu trabalho de Doutoramento, foi criado um sistema de experimentação, no qual se pode observar o fenómeno de transferência de calor entre dois fluidos, quando colocados em determinadas circunstâncias. O sistema é composto por dois tanques, sendo que um é utilizado para o armazenamento e controlo da temperatura de um dos fluidos, enquanto o outro contém o segundo fluido, que quando em contacto indirecto com o primeiro, faz com que a temperatura deste aumente ou diminua, dependendo da combinação de fluidos.



Figura 2. Fotografia do Sistema de Experimentação de Transferência de Calor

Para efectuar a circulação do fluido entre os dois tanques, é utilizado um sistema de bombas, juntamente com vários sensores de medição que quando ligados a um sistema de aquisição de dados, permitem recolher os registos dos mesmos e guardá-los para futuras análises. Relativamente aos sensores ligados ao sistema de aquisição

de dados, de referir que existem 9 sensores de medição de temperatura espalhados ao longo do circuito, além dos dois sensores de pressão (pressão inicial e diferencial de pressão) e um medidor de caudal.

O objectivo deste caso prático passa pelos seguintes pontos:

- Modelar um sistema de sistemas real, utilizando a notação SysML
- Avaliar a sua adaptação ao contexto apresentado pelo sistema
- Analisar as possibilidades de modelação disponibilizadas pela notação
- Avaliação da adaptabilidade do sistema aos modelos criados
- Apresentar algumas limitações da notação SysML

Desta forma, será possível, através deste conjunto de dados, obter uma avaliação sobre o real potencial da notação SysML. Todos estes pontos abordados têm como finalidade, esclarecer os principais objectivos desta investigação:

- Validar os pontos fortes e fracos da notação, estabelecidos na análise teórica
- Analisar a aplicação prática da notação em condições reais

Através destes dados, será possível obter algumas conclusões sobre a utilidade real do SysML, bem como validar os registos obtidos após a análise teórica da notação.

5.2. Método de Investigação

Para se garantir o rigor durante todo o desenvolvimento destes caso prático, recorreu-se à Investigação em Acção (Action Reseach), a qual consiste num conjunto de processos de reflexão que permitem levar à determinação de novas soluções para um mesmo problema. A Investigação em Acção pode ser definida da seguinte forma [19]:

Focada na investigação durante a acção, a Investigação em Acção consiste numa abordagem científica para estudar a resolução de diversos problemas sociais ou organizacionais juntamente com aqueles que os experienciam, recorrendo para isso a um ciclo de 4 passos: Diagnóstico, Planeamento, Acção e Avaliação.

Sendo um processo democrático, envolve os membros do sistema no processo de decisão, com estes a representarem uma parte importante do resultado final da Investigação em Acção. Além disto, visto que este método é executado concorrentemente com a acção, permite que a mesma seja mais efectiva enquanto o conhecimento científico sobre o sujeito em causa é alargado.

A Investigação em Acção não tem como principal objectivo a obtenção de soluções para problemas imediatos, mas sim a obtenção de conhecimento sobre os problemas e as suas consequências (directas e indirectas), contribuindo assim para o conhecimento científico e teórico.

Com a definição inicial da Investigação em Acção a ser dividida em 4 fases, podemos defini-las da seguinte forma:

- **Diagnóstico** – Trata-se da nomeação dos problemas, a partir da qual serão efectuados os planeamentos necessários.
- **Planeamento** – A partir da análise do contexto e objectivo do projecto, problema principal e respectivo diagnóstico, permite detalhar o conjunto de passos que terão que ser abordados.
- **Acção** – Consiste na tomada de acção, baseada no planeamento efectuado anteriormente.
- **Avaliação** – Com base nos resultados obtidos, é efectuada a análise de parâmetros tais como correcção do diagnóstico, correcção da acção tomada, entre outros aspectos.

A partir dos dados obtidos na Avaliação, é criado um novo ciclo com os novos dados, repetindo-se todas as fases novamente. Além desta divisão, a Investigação em Acção define também um ciclo de aprendizagem experimental (Experimental Learning), o qual consiste num conjunto de passos que são constantemente seguidos, durante todas as etapas da Investigação em Acção:

- **Experimentar** – Consiste num conjunto de passos, planeados ou não, que permitem a obtenção de informação a partir de uma situação.
- **Reflectir** – A partir dos dados experimentais obtidos, reflecte-se sobre o que se efectuou e o que eventualmente estará errado.
- **Interpretar** – Permite obter as respostas para os problemas surgidos na experimentação, a partir de teorias criadas a partir da reflexão.
- **Agir** – Tendo as respostas, pode-se definir um novo comportamento ou método de utilização, para futuras repetições de uma determinada situação.

Desta forma, em cada fase de diagnóstico, planeamento, acção e avaliação (processos inerentes ao acto de investigação), é efectuada uma auto-crítica disciplinada e que permite obter novos dados, a partir de uma determinada experiência.

Na modelação do sistema apresentado, foram seguidas todas estas etapas, de forma a obter-se a maior quantidade de informação possível a partir do caso prático. Para que seja possível ter-se uma melhor percepção da utilização deste método de investigação na modelação realizado no caso prático, pode-se referir alguns exemplos desse facto. Começando pela fase inicial da componente prática desta Tese, após as diversas reuniões com a responsável pelo sistema, foi necessário realizar diversas alterações aos diagramas de requisitos inicialmente criados.

A partir das diversas informações sobre o sistema em si e sobre qual o objectivo da responsável para o sistema (o qual pode ser integrada dentro da fase de Diagnóstico, definida pela Investigação em Acção), foram analisadas os erros que se havia realizado (por exemplo, identificação errónea de ligações entre os elementos do sistema, tais como o Sistema de Aquisição de Dados e os Tanques, as quais não existiam na realidade), bem como possíveis abordagens à modelação do sistema (definida na etapa Planeamento). A partir desta reflexão, foram realizadas as alterações necessárias e obteve-se um novo resultado, na forma do modelo representativo do sistema em causa.

Este passo permitiu a obtenção de novos conhecimentos sobre o sistema em si, bem como possíveis erros que se poderiam efectuar no futuro, caso tal não tivesse sido experimentado (a representação errónea do sistema teria levado a novas alterações do mesmo, caso não tivessem sido identificados os erros iniciais). Integrando-se nas etapas Acção e Avaliação, definiu-se um novo comportamento de utilização para este caso, o qual foi repetidamente utilizado até à finalização da modelação.

Além da sua aplicação na modelação de requisitos (onde se apresentou como uma ferramenta importante para a discussão da modelação dos requisitos do sistema), o ciclo de aprendizagem experimental permitiu obter novos dados em cada diagrama modelado. Visto que o sistema se apresentava como um todo, os diagramas modelados deveriam fazer parte de um conjunto global, pelo que a modelação de cada um deles levou a diversas alterações intermédias, para que fosse possível obter o resultado pretendido.

Como exemplo, pode-se referir a modelação dos diagramas de actividade para cada uma das operações apresentadas nos casos de uso, que levaram a diversas interpretações sobre o funcionamento do sistema. Desta forma, as actividades representadas sofreram alterações ao longo do tempo, em virtude do maior

conhecimento obtido acerca do sistema e da posterior reflexão sobre a sua modelação.

5.3. Metodologia de Concepção do Sistema

Para que o trabalho realizado fosse rigoroso, era necessária definir antes de qualquer modelação, qual o tipo de metodologia a utilizar para o processo de concepção do sistema, bem como as etapas necessárias para o cumprimento desse mesmo trabalho. Para tal, recorreu-se à metodologia SYSMOD, apresentada por Tim Weilkiens no seu livro [21] e a qual estabelece todos os pontos necessários para que seja possível modelar um sistema de acordo com as necessidades dos clientes, sendo também uma representação fiel do seu congénere real.

Esta metodologia foi escolhida por se encontrar definida especificamente para utilização com o SysML, abordando cada um dos diagramas disponibilizados pela notação e apresentando no final uma visão geral da constituição do sistema. Detalhando a referida metodologia, podemos dividi-la pelas seguintes fases:

- Descrição do contexto do projecto
- Determinação dos requisitos
- Modelação do contexto do sistema
- Modelação dos casos de uso e processos do sistema
- Modelação das estruturas do sistema e estados
- Finalização dos conhecimentos do domínio

Desta forma, através destas etapas, é possível determinar e modelar cada um dos aspectos do sistema, desde o seu contexto inicial até à sua forma final. Começando pela primeira etapa da metodologia, consiste sobretudo na análise do contexto onde será inserido o sistema (neste caso, onde o mesmo se encontra inserido) e na sua representação.

Posteriormente, são determinados e modelados os respectivos requisitos do sistema, partindo destes a criação dos casos de uso e de sequência, representando os diversos processos do sistema. Por fim, serão modeladas as actividades e parâmetros, bem como os blocos internos, para que seja possível estruturar todo o sistema de forma coesa. Para agregar todos estes elementos e culminar os conhecimentos deste modelo, recorre-se à utilização de pacotes, que permitem criar determinadas visões, em função dos respectivos utilizadores.

Visto que a metodologia SYSMOD é bastante abrangente, cobrindo todas as etapas desde a análise até ao desenho do sistema, existem vários passos cuja utilidade neste caso prático não se mostrou necessária (por exemplo, a criação de visões adicionais para outros utilizadores não foi considerada, visto que tanto o professor, como os alunos terão as mesmas funções neste sistema, embora com restrições no segundo caso). Desta forma, esses passos não foram tidos em consideração durante a modelação do sistema.

Além desta metodologia, Tim Weilkiens também criou o respectivo perfil para o SysML, apresentando novos diagramas e estereótipos que estendem o âmbito da sua metodologia. Este perfil encontra-se disponível unicamente para a ferramenta Magic Draw e pode ser descarregado a partir de www.sysmod.de.

Finalmente, de referir que a metodologia SYSMOD deriva da metodologia V-Model XT, a qual foi apresentada em 2004 pelo Estado Alemão. Esta metodologia aborda o planeamento e implementação de projectos de desenvolvimento de sistemas e fornece um conjunto de ferramentas tais como papeis definidos, produtos e actividades que podem ser adaptados a qualquer tipo de projectos. Ao contrário do SysML, que apresenta as ferramentas mas não inclui qualquer tipo de instruções de utilização, a metodologia V-Model XT indica quais os passos a seguir, sem no entanto restringir as ferramentas a utilizar.

A metodologia SYSMOD, tal como a V-Model XT, apresenta essas indicações, específicas para o desenvolvimento de sistemas de sistemas utilizando o SysML, sem no entanto restringir o conjunto de ferramentas a utilizar para tal.

5.4. Escolha da Ferramenta

Após a escolha da metodologia, foi necessário escolher a ferramenta de modelação que seria utilizada para o cumprimento desta tarefa. Tal escolha recaiu sobre o Artisan Studio, o qual apresenta diversas vantagens, tais como:

- Interface gráfico intuitivo e fácil de utilizar
- Licença gratuita e disponível para download (na versão Uno, utilizada para este projecto)
- Facilidade de importação e exportação de modelos (importação a partir do Rational Rose e exportação para outras aplicações)
- Possibilidade de criação de modelos em SysML, UML e híbridos
- Possibilidade de gerar documentação a partir dos modelos criados

Analisadas as vantagens da sua utilização, relativamente às suas concorrentes (comerciais, visto que não existe outra ferramenta com licença gratuita e que apresente as mesmas funcionalidades), não houve qualquer dúvida sobre a sua escolha e aplicação neste caso prático.

5.5. Fases de Investigação

Estando definidas a metodologia de concepção, o método de investigação e a ferramenta a serem utilizadas, estão definidos os pontos principais e basilares sobre os quais será efectuada a modelação do sistema. Foi definido um conjunto de fases:

- Estudo da metodologia, método e ferramenta a utilizar
- Modelação do sistema em causa
- Análise e escrita dos resultados obtidos

De referir que dentro do período de modelação do sistema apresentado neste caso prático, foi necessário dividir este processo em 3 fases distintas:

- Criação dos diagramas de contexto, definição de blocos e requisitos
- Criação dos diagramas de caso de uso, sequência e actividade
- Criação dos diagramas paramétricos, blocos internos e pacotes.

5.6. Análise do Contexto

Partindo da introdução anterior, verificamos que o sistema a modelar estava inserido no Laboratório de Química do Instituto Superior de Engenharia do Porto e que tem como principal função permitir a compreensão e experimentação dos fenómenos de transferência de calor entre fluidos, investigação essa levada a cabo pela responsável pelo sistema.

O interesse da modelação do sistema passava não só pela referida modelação, mas também por averiguar eventuais possibilidades de expansão do sistema para utilização remota, caso tal fosse possível. Desta forma, o modelo do sistema poderia ser utilizado para que, em fases futuras, tal fosse passível de ser analisado e averiguado, bem como as possibilidades reais de utilização da simulação neste sistema.

O sistema é constituído por 3 elementos principais: dois tanques e um sistema de aquisição de dados. Enquanto o tanque principal (referido nos diagramas como Tanque 1) é constituído por um fluido (variável de acordo com a experiência), que é

transportado através de serpentinas condutoras, colocadas no interior do referido tanque. O transporte do fluido desde o tanque secundário (Tanque 2) até ao tanque principal (Tanque 1) é assegurado por duas bombas (centrífuga e volumétrica) montadas em paralelo, mas onde apenas uma funciona de cada vez.

Dentro do tanque principal encontra-se também o óleo, outro líquido essencial à experimentação. O óleo é agitado através de um agitador mecânico, enquanto a sua temperatura é ajustada com recurso a uma resistência eléctrica, ligado a um controlador PID. Graças à presença destes dois líquidos (que nunca se encontram em contacto directo), é possível realizar as experimentações de transferência de calor entre fluidos.

Para efectuar o registo de dados, são utilizados dois PT100 e dois Termopares Tipo T, sensores que registam diferenças de temperaturas em gamas variadas, um medidor de caudal e dois medidores de pressão (pressão inicial à entrada do tanque e diferencial de pressão à saída). Todos estes sensores enviam os seus dados para o sistema de aquisição de dados, podendo também ser programados através do software deste.

O sistema de aquisição de dados é composto por uma placa de recepção de dados, onde são ligados todos os sensores do sistema, bem como por dois softwares independentes: AOIP LW1, destinado à recepção e gravação dos dados enviados pelos sensores e o VideoToTrace, software que permite a programação da placa e dos sensores.

Finalmente, no que concerne o tanque secundário (Tanque 2), este é utilizado para armazenamento do fluido a utilizar, tendo acoplado um sistema de refrigeração convencional, composto por uma bomba de calor (para arrefecimento) e uma resistência eléctrica (para aquecimento). Além disto, tem também sensores, tais como de temperatura (Termopar Tipo K) e indicador de nível, sendo que nenhum deles se encontra ligado ao sistema de aquisição de dados.

Estando concluída a apresentação do referido sistema a modelar, falta-nos definir as pessoas que poderão trabalhar com o referido sistema. Neste caso, as pessoas que poderão fazer alguma interacção com o sistema tratam-se da responsável pelo sistema (definida nos diagramas como Professora) e eventuais alunos que utilizarão o sistema para algum trabalho de investigação (referidos nos diagramas como Aluno).

5.7. Modelo de Domínio

Estando definidos os pontos principais do sistema, partiu-se para a criação do modelo de domínio, sendo este modelo constituído pelos diagramas apresentados pela notação SysML (contexto, definição de blocos, requisitos, blocos internos, paramétricos, sequência, casos de uso, paramétrico, actividade e pacotes). Iniciando a modelação pelo primeiro diagrama (contexto), utilizou-se o mesmo para exemplificar facilmente todos os componentes envolvidos no referido sistema. Trata-se assim do primeiro passo referido na metodologia SYSMOD, essencial para completar a abordagem inicial à análise do sistema [21]. Utilizando a ferramenta Artisan Studio Uno, tal foi realizado com relativa facilidade, visto que para tal tarefa utiliza-se os diagramas definição de blocos, como é referido na metodologia.

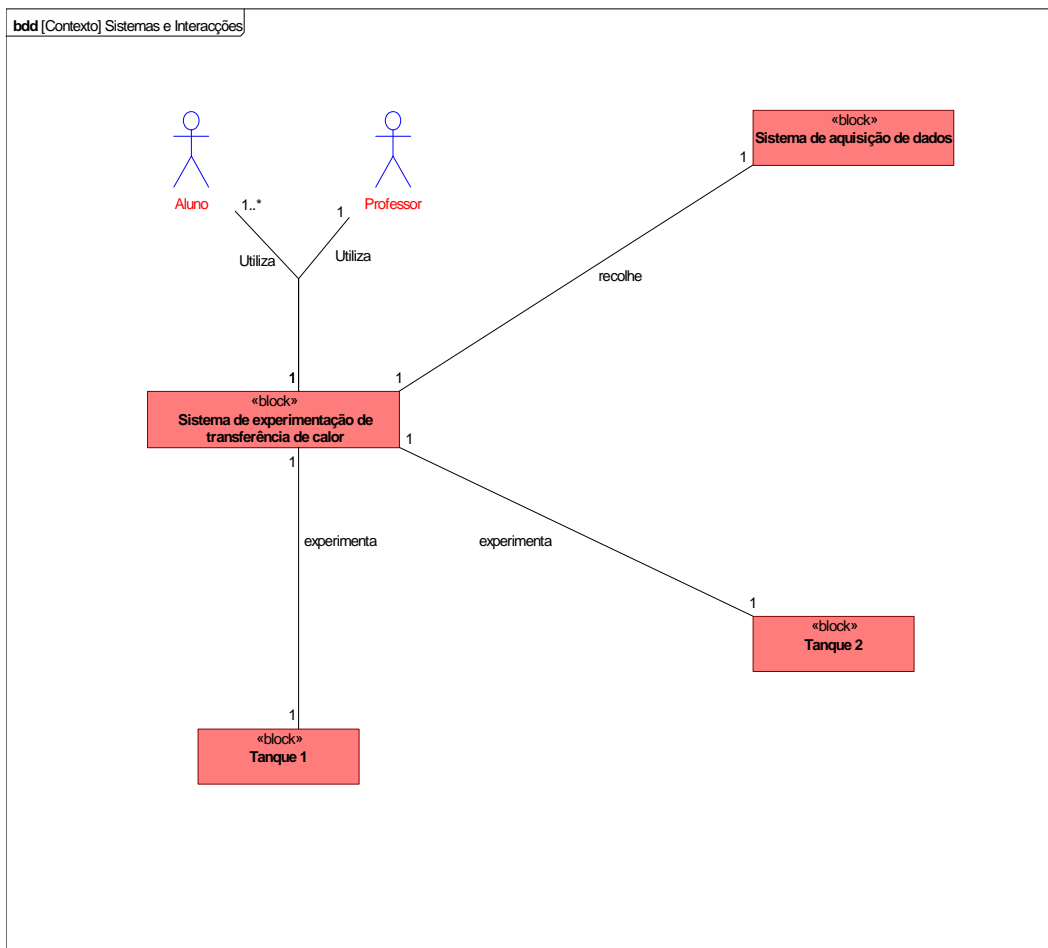


Figura 3. Diagrama de contexto do sistema do Departamento de Química

Como se pode verificar neste diagrama, os tanques e o sistema de aquisição de dados foram acoplados a um sistema central, referido como Sistema de Experimentação de Transferência de Calor. Esta opção foi tomada de forma a facilitar a compreensão do sistema, integrando os seus componentes como sistemas

integrados num sistema principal (sistema de sistemas). Desta forma, pode-se considerar o sistema central sem que o seu local de utilização seja um factor perturbador. Visto que este tanto pode ser local (estado actual) como remoto (hipótese futura), a adaptação do modelo do sistema a um possível cenário de experimentação remota é algo que facilmente poderá ser realizado, sem grandes custos temporais.

De forma a detalhar os comportamentos de cada um dos componentes do sistema, procedeu-se então à modelação dos comportamentos de cada um dos actores (Professor e Aluno), recorrendo novamente ao diagrama de definição de blocos, tal como é referido na metodologia.

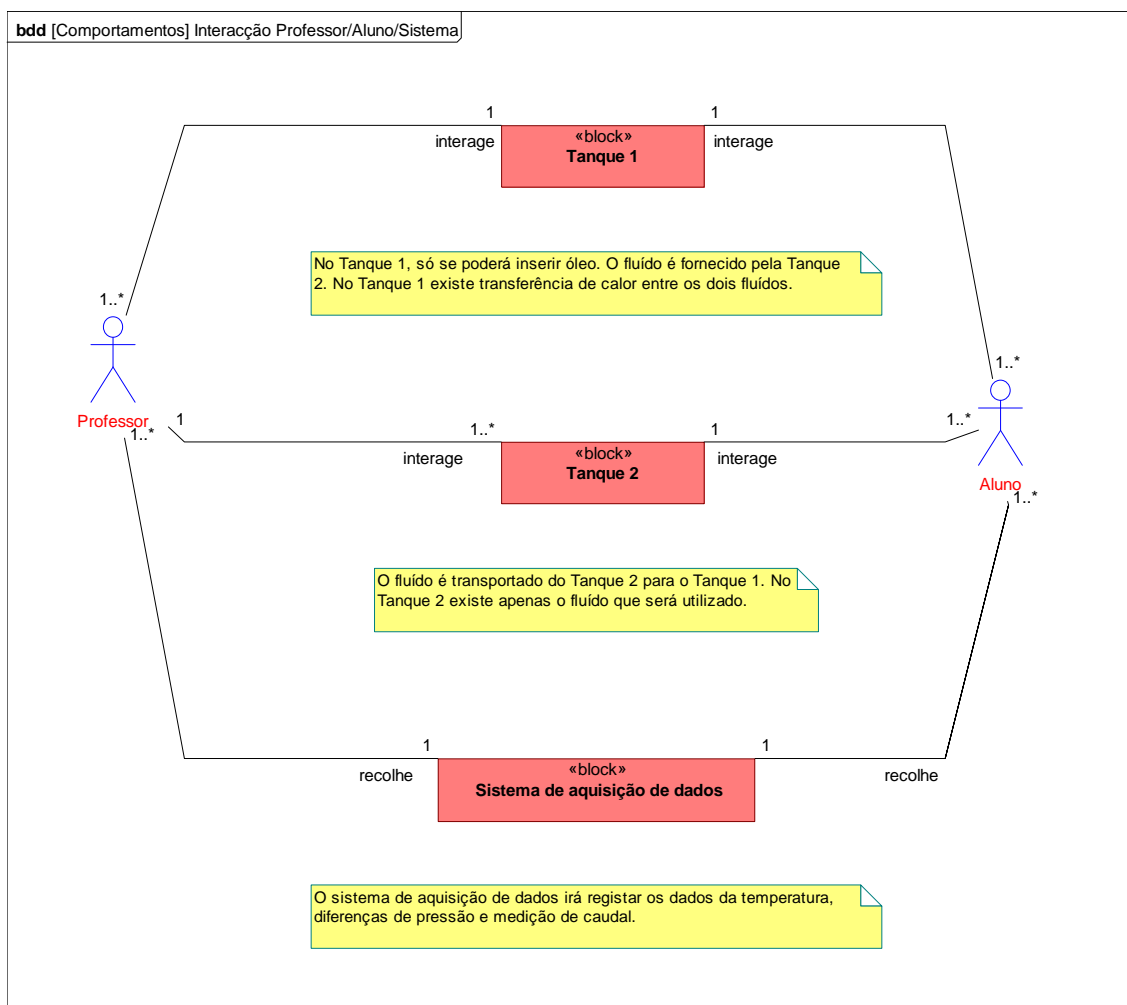


Figura 4. Comportamentos possíveis no sistema em causa

No que concerne o tanque principal (Tanque 1), a única interacção possível consiste na colocação do óleo no mesmo, visto que não existe qualquer outro tipo de interacção possível no mesmo. Quanto ao tanque secundário (Tanque 2), é possível alterar a temperatura do fluido a transportar para o tanque principal, através da

utilização de um controlador de temperatura (ligado ao Termopar Tipo K instalado no tanque), que irá activar a bomba de calor ou a resistência eléctrica para alterar a temperatura. Finalmente, o sistema de aquisição de dados pode ser utilizado para alterar a programação dos registos dos sensores, de modo a que se analisem outros parâmetros importantes.

Para que a análise de contexto ficasse concluída, procedeu-se à criação de um outro diagrama de definição de blocos, no qual são especificadas as variáveis passíveis de registo e análise. Algumas destas variáveis poderão não ser registadas actualmente, mas caso o sistema seja utilizado para experimentação remota, terão invariavelmente de ser controladas.

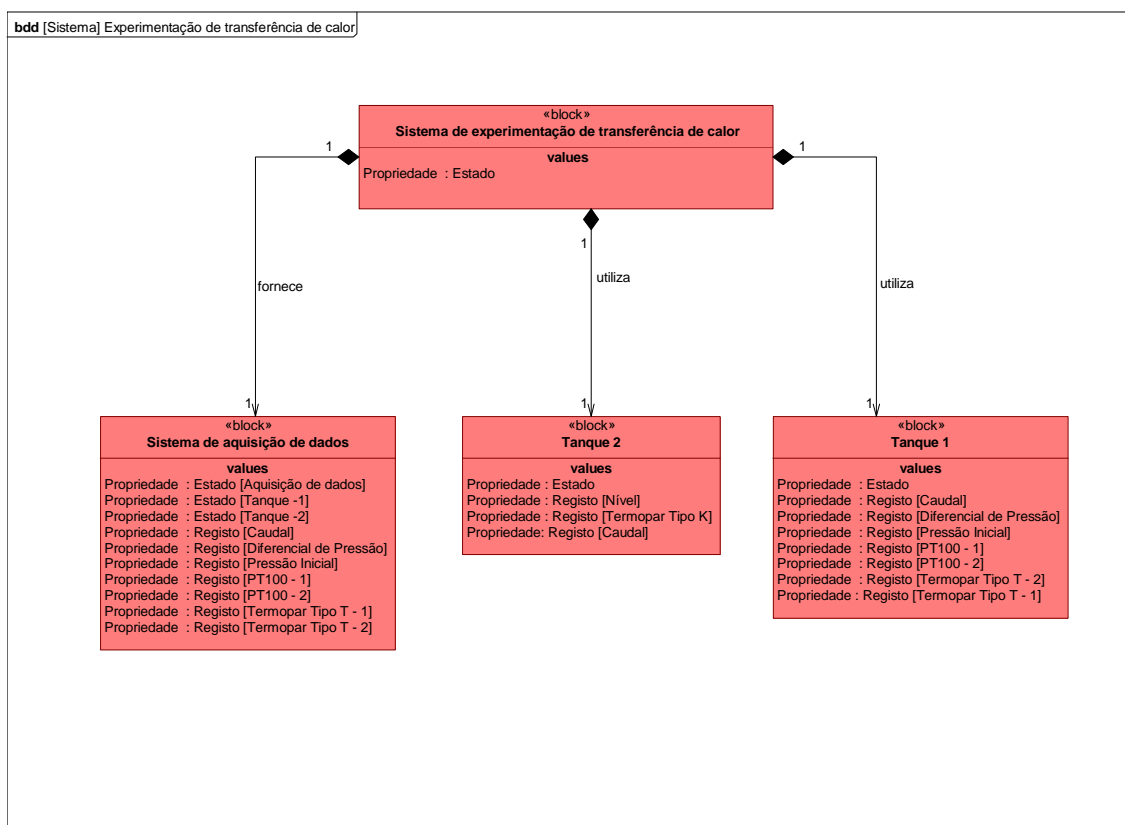


Figura 5. Detalhe do sistema de sistemas analisado

Para cada um dos componentes foi inserida uma variável de estado, a qual permite verificar se o componente se encontra ligado ou desligado. Além disso, foram inseridas variáveis de registos de cada um dos sensores utilizados, para que os mesmos possam ser devidamente controlados e registados. Com a inserção destas variáveis, torna-se possível o controlo de um componente a partir de outro (sistema de aquisição de dados interagir autonomamente com os tanques), tal como realizar operações de vigilância dos sensores (ligação dos mesmos e correcção de sensores

com funcionamento incorrecto, por exemplo). Desta forma, o sistema ganhará autonomia de funcionamento, podendo acrescentar-se funcionalidades de verificação e segurança, quer do sistema, quer dos dados recolhidos, caso tal fosse interessante e vantajoso.

5.7.1. Avaliação

Com estes três diagramas foi possível obter uma visão geral do sistema e do que será o seu modelo, utilizando a notação SysML. De referir que o processo de modelação dos diagramas apresentados não revelou qualquer entrave, tendo os únicos problemas surgido devido ao período inicial de adaptação à ferramenta.

Os diagramas de definição de blocos presentes no SysML são semelhantes aos diagramas de classes do UML, com a sua utilização a ser relativamente fácil, mesmo para quem não possua qualquer tipo de conhecimento deste tipo de linguagens.

Estando modelado os aspectos básicos relativos ao sistema em causa, a fase seguinte (de acordo com a metodologia utilizada), consistia na modelação dos requisitos do sistema, de acordo com as necessidades apresentadas.

5.8. Análise de Requisitos

O passo seguinte apresentado pela metodologia SYSMOD consiste na análise dos diversos requisitos do sistema, para que cada um dos seus sistemas secundários possa funcionar correctamente e atingir os objectivos desejados [21]. Partindo da divisão do sistema principal (Sistema de Experimentação de Transferência de Calor) em diversos sistemas secundários (Tanque 1, Tanque 2 e Sistema de Aquisição de Dados), esta tarefa tornou-se menos morosa e mais simples de se efectuar. Para a modelação dos requisitos, utilizou-se o diagrama de requisitos, disponibilizado pelo SysML para modelar os mesmos.

A análise dos requisitos foi realizada através de diversas reuniões com a responsável. Desta forma, foi possível obter todos os dados necessários para que o resultado final fosse o pretendido e de acordo com o sistema real. Além disso, através da apresentação de esboços intermédios dos diagramas, foi possível identificar potenciais erros e corrigi-los atempadamente. Após as devidas rectificações, foi possível apresentar os diagramas no seu formato final à responsável, a qual os validou.

Com isto, foi possível obter os diagramas de requisitos representativos do sistema em causa, a partir dos quais se abordaram as fases seguintes, bem como o conjunto de requisitos aos quais o sistema modelado deveria ser fiel:

- Tanque 1 deve ser referido como o local de experimentação para a transferência de calor entre os fluidos em questão, podendo estes ser alterados ao longo do tempo.
- Tanque 2 deve apresentar-se como local de armazenamento do fluido transportado, podendo este ser diferente para cada experimentação.
- Enquanto o fluido armazenado no Tanque 2 é transportado para o Tanque 1 por intermédio de serpentinas e bombas, o fluido secundário inserido unicamente no Tanque 1 nunca deve estar em contacto com o fluido principal e deve estar localizado permanentemente neste tanque.
- O sistema de armazenamento de dados deve recolher informação sobre as diversas temperaturas ao longo do sistema, bem como do caudal e pressão do fluido transportado.
- Qualquer alteração dos parâmetros do sistema levará à alteração da experimentação e por consequência do resultado final da mesma.

No que concerne os diagramas de requisitos criados, tendo como base os requisitos enumerados, bem como as informações obtidas a partir da responsável, foi possível desenhar um conjunto de diagramas representativos do sistema em causa.

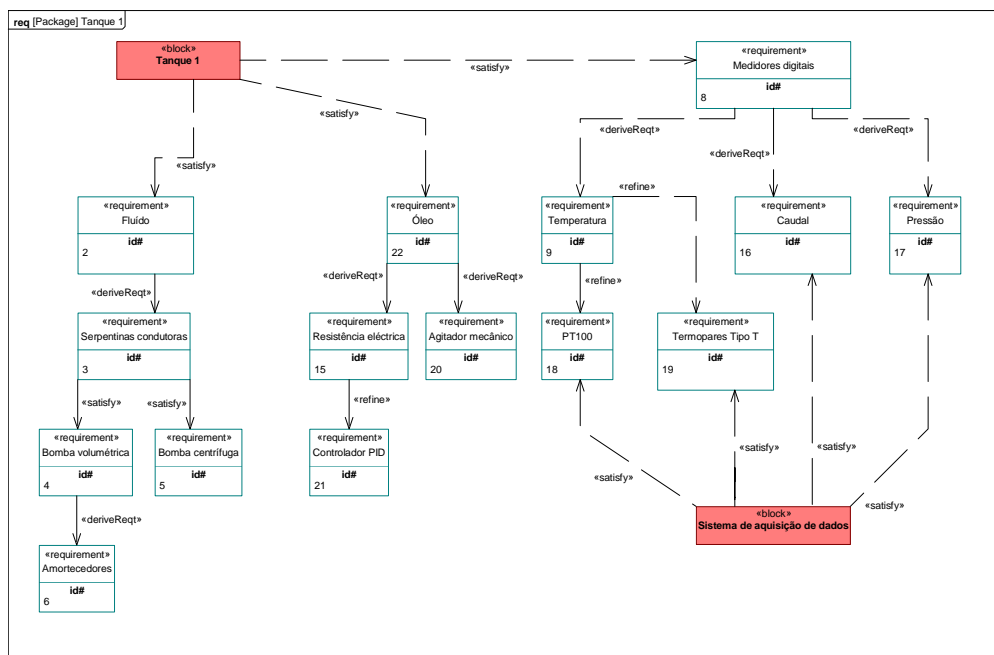


Figura 6. Diagrama de requisitos do tanque principal

Identificador	Nome	Relação	Nome
2	Fluido	satisfy	Tanque 1
3	Serpentinas condutoras	deriveReq	Fluido
4	Bomba volumétrica	satisfy	Serpentinas condutoras
5	Bomba centrífuga	satisfy	Serpentinas condutoras
6	Amortecedores	deriveReq	Bomba volumétrica
8	Medidores digitais	satisfy	Tanque 1
9	Temperatura	deriveReq	Medidores digitais
15	Resistência eléctrica	deriveReq	Óleo
16	Caudal	deriveReq	Medidores digitais
17	Pressão	deriveReq	Medidores digitais
18	PT100	refine	Temperatura
19	Termopares Tipo T	refine	Temperatura
20	Agitador Mecânico	deriveReq	Óleo
21	Controlador PID	refine	Resistência eléctrica
22	Óleo	satisfy	Tanque 1

Tabela 7. Tabela de requisitos relativo ao tanque principal

Começando pelo tanque principal (Tanque 1), podemos reparar que existem determinados elementos que devem estar presentes para que os objectivos da experiência sejam cumpridos. Entre estes, de destacar o fluido (um dos líquidos a utilizar na experiência), o óleo (outro líquido em utilização) e os medidores digitais.

No que concerne o fluido, para o seu transporte é necessária a utilização de serpentinas condutoras, que serão o elemento em contacto com o óleo no interior do tanque. Para que o transporte seja efectuado, é necessário recorrer à utilização de uma bomba volumétrica ou de uma bomba centrífuga, ambas montadas em paralelo. Quanto ao óleo, para que esteja a uma dada temperatura e num estado ideal, é necessária a utilização de uma resistência eléctrica (ligada a um controlador de temperatura, onde se define a temperatura) e um agitador mecânico.

Finalmente, relativamente aos medidores digitais, são utilizados 4 tipos de sensores: PT100 e Termopares Tipo T (para determinação da temperatura em variadas gamas e em pontos distintos do circuito, por exemplo no interior das serpentinas e no interior do tanque), caudal (para determinação do caudal do fluido em qualquer instante da experiência) e pressão (pressão à entrada do tanque e diferença de pressão à saída). Desta forma, é possível avaliar estes 3 parâmetros, dependendo das condições impostas à experiência.

No que concerne o tanque secundário (Tanque 2), embora as necessidades deste sejam relativamente semelhantes às do tanque principal, existem diversos elementos que divergem deste, como se poderá verificar no diagrama e tabela de requisitos seguinte.

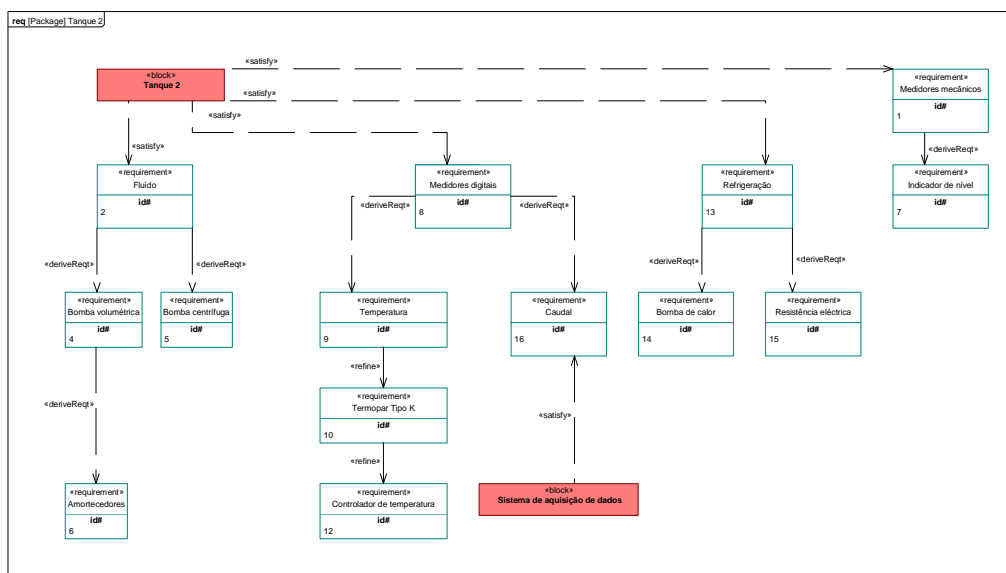


Figura 7. Diagrama de requisitos do tanque secundário

Identificador	Nome	Relação	Nome
1	Medidores mecânicos	satisfy	Tanque 1
2	Fluido	satisfy	Tanque 1
3	Serpentinas condutoras	deriveReq	Fluido
4	Bomba volumétrica	satisfy	Serpentinas condutoras
5	Bomba centrífuga	satisfy	Serpentinas condutoras
6	Amortecedores	deriveReq	Bomba volumétrica
7	Indicador de nível	deriveReq	Medidores mecânicos
8	Medidores digitais	satisfy	Tanque 1
9	Temperatura	deriveReq	Medidores digitais
10	Termopar Tipo K	refine	Temperatura
12	Controlador de temperatura	refine	Termopar Tipo K
13	Refrigeração	satisfy	Tanque 1
14	Bomba de calor	deriveReq	Refrigeração
15	Resistência eléctrica	deriveReq	Refrigeração
16	Caudal	deriveReq	Medidores digitais

Tabela 8. Tabela de requisitos relativo ao tanque secundário

Relativamente ao fluido, podemos constatar que já não é necessária a utilização de serpentinas condutoras no transporte do mesmo. Tal verifica-se pois o fluido não se encontra em contacto com outro líquido no interior deste tanque, visto ser o único líquido presente no mesmo. Desta forma, é apenas necessária a utilização das bombas volumétrica e centrífuga, para realizar o transporte do fluido. No que concerne os medidores digitais, dos sensores presentes no tanque principal, apenas o medidor de caudal se encontra presente, visto que é utilizado à saída do tanque secundário (e respectiva entrada no tanque principal). Além deste, existe também um medidor de temperatura (termopar Tipo K), que através do seu controlador de temperatura (ligado ao sistema de refrigeração instalado neste tanque), permite alterar e adequar a temperatura do fluido em qualquer altura.

Este sistema de refrigeração também se encontra inserido neste diagrama de requisitos, sendo constituído por dois elementos: uma bomba de calor (para arrefecimento) e uma resistência eléctrica (para aquecimento). Desta forma e dependendo da temperatura regulada, será possível alterar a temperatura do fluido de acordo com as necessidades das experiências a realizar. Finalmente, existe também um medidor mecânico adicional, destinado à medição de nível, que é utilizado apenas quando o fluido presente no tanque é água.

Para conclusão da modelação dos requisitos do sistema, analisou-se também os requisitos do sistema de aquisição de dados, representado no diagrama e tabela de requisitos seguinte.

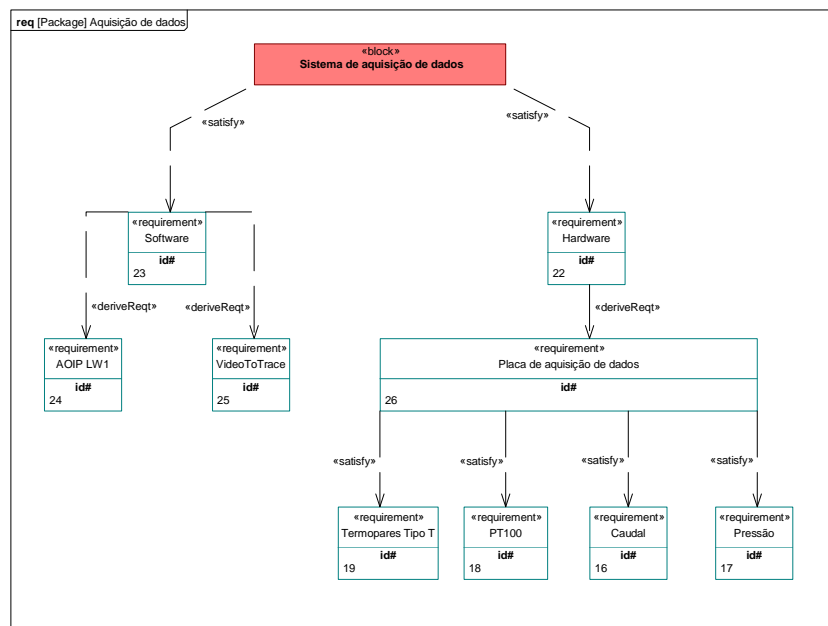


Figura 8. Diagrama de requisitos do sistema de aquisição de dados

Identificador	Nome	Relação	Nome
16	Caudal	satisfy	Placa de aquisição
17	Pressão	satisfy	Placa de aquisição
18	PT100	satisfy	Placa de aquisição
19	Termopares Tipo T	satisfy	Placa de aquisição
22	Hardware	satisfy	Sistema de aquisição
23	Software	satisfy	Sistema de aquisição
24	AOIP LW1	deriveReq	Software
25	VideoToTrace	deriveReq	Software
26	Placa de aquisição	deriveReq	Hardware

Tabela 9. Tabela de requisitos relativo ao sistema de aquisição de dados

No que concerne o sistema de aquisição de dados, existem dois elementos essenciais para que as suas tarefas possam ser realizadas: software e hardware.

No que concerne o software, é necessária a utilização de duas aplicações distintas, cada uma delas utilizada para a realização de um determinado serviço. Enquanto a aplicação AOIP LW1 é utilizada para armazenar os registos dos sensores utilizados ao longo do circuito, a aplicação VideoToTrace permite programar a placa de aquisição de dados de acordo com as necessidades das experiências, alterando os parâmetros e gamas de valores a registar.

Quanto à componente de hardware, tem um elemento principal (Placa de Aquisição de Dados Externa), à qual se ligam todos os sensores utilizados e monitorizados pelo sistema de aquisição de dados. Entre eles, de destacar os Termopares Tipo T e os PT100 (destinados à medição da temperatura), além dos medidores de caudal e pressão (pressão inicial e diferença de pressão à saída). O Termopar Tipo K e o indicador de nível não se encontram ligados ao sistema de aquisição de dados, pelo que não são considerados neste sistema.

5.8.1. Avaliação

Através destes diagramas de requisitos, é possível apresentar os diversos elementos que constituem o sistema completo e as suas necessidades e ligações, possibilitando a sua eventual modificação no futuro, sem que tal implique a alteração física do sistema. De referir que a lista de requisitos inicialmente apresentada foi devidamente seguida, quer na elaboração dos diagramas de requisitos, bem como nos restantes diagramas do modelo em causa.

5.9. Casos de Uso

A metodologia SYSMOD especifica que, após a realização dos diagramas de requisitos, se proceda à criação dos casos de uso do sistema em causa, que serão necessários para definir as possibilidades de interacção directa com o sistema [21].

Tendo em conta que se trata de um sistema experimental, a sua adaptação física poderá ser realizada a qualquer momento. No entanto, visto que tal é imprevisível e impossível de modelar, não foi considerada como interacção e como tal, não foi incluída nos diagramas de casos de uso.

Desta forma, foram só consideradas interacções credíveis com o sistema e que alterassem os seus resultados experimentais, sem envolver qualquer alteração do sistema projectado. Além disso, teve-se em consideração uma eventual adaptação do sistema para experimentação remota, pelo que alguns procedimentos tiveram de ser incluídos, muito embora possam ser ignorados no sistema actual. Esses procedimentos foram detalhados nos diagramas de sequência presentes na secção 5.10.

Tendo como actores os alunos (representados pelo actor Aluno) e a responsável (representada pelo actor Professor), identificou-se três tipos de acções possíveis: Ligação do Sistema, Encerramento do Sistema e Alteração de Temperatura.

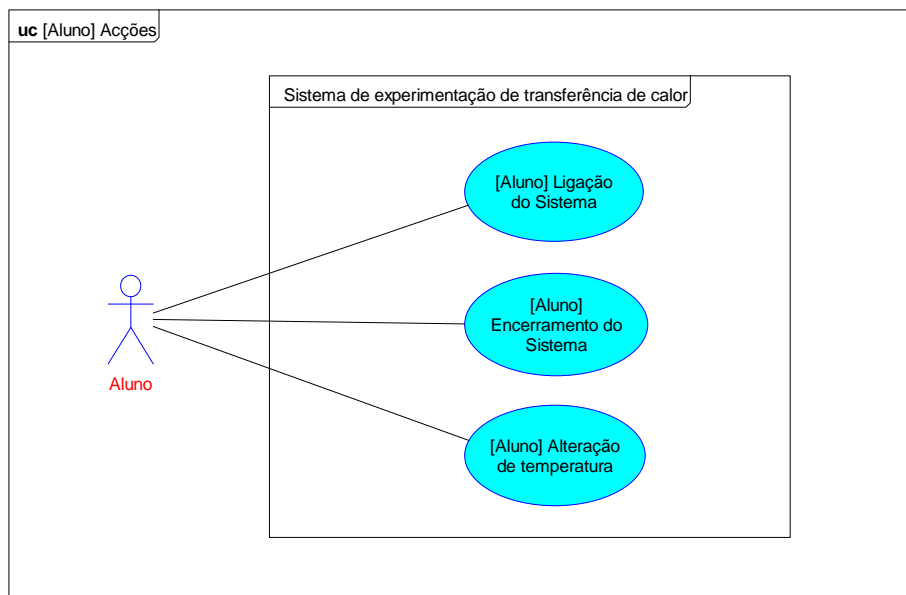


Figura 9. Diagrama de casos de uso relativo ao actor Aluno

Detalhes	Descrição
ID	1
Nome	[Aluno] Ligação do Sistema

Descrição	A ligação do sistema consiste em várias etapas: ligação do tanque principal e secundário, bem como do sistema de aquisição de dados. Após estes procedimentos, é iniciada a experimentação prevista.
Actores	Aluno
Pré-condições	Sistema de experimentação pronto a funcionar
Pós-condições	O actor deve poder iniciar a sua experiência e recolha de dados

Tabela 10. Especificação do caso de uso [Aluno] Ligação do Sistema

Detalhes	Descrição
ID	2
Nome	[Aluno] Encerramento do Sistema
Descrição	O encerramento do sistema do sistema consiste em várias etapas: verificação do tanque principal e secundário, bem como do sistema de aquisição de dados. Após estes procedimentos, o sistema estará encerrado
Actores	Aluno
Pré-condições	Sistema de experimentação em funcionamento
Pós-condições	O sistema deve encontrar-se devidamente encerrado

Tabela 11. Especificação do caso de uso [Aluno] Encerramento do Sistema

Detalhes	Descrição
ID	3
Nome	[Aluno] Alteração de temperatura
Descrição	A alteração da temperatura consiste na verificação da existência de líquido no tanque secundário. Caso este exista, será efectuada a alteração da temperatura, de acordo com o limite máximo imposto anteriormente.
Actores	Aluno
Pré-condições	Sistema de experimentação em funcionamento Não pode estar em curso qualquer experiência
Pós-condições	A temperatura no fluido deve estar devidamente alterada

Tabela 12. Especificação do caso de uso [Aluno] Alteração de temperatura

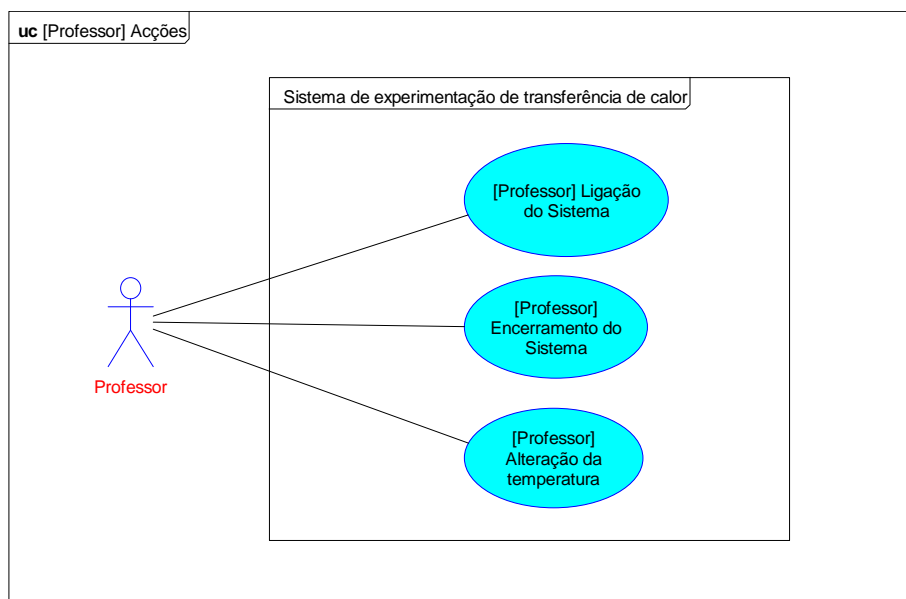


Figura 10. Diagrama de casos de uso relativo ao actor Professor

Detalhes	Descrição
ID	4
Nome	[Professor] Ligação do Sistema
Descrição	A ligação do sistema consiste em várias etapas: ligação do tanque principal e secundário, bem como do sistema de aquisição de dados. Após estes procedimentos, é iniciada a experimentação prevista.
Actores	Professor
Pré-condições	Sistema de experimentação pronto a funcionar
Pós-condições	O actor deve poder iniciar a sua experiência e recolha de dados

Tabela 13. Especificação do caso de uso [Professor] Ligação do Sistema

Detalhes	Descrição
ID	5
Nome	[Professor] Encerramento do Sistema
Descrição	O encerramento do sistema do sistema consiste em várias etapas: verificação do tanque principal e secundário, bem como do sistema de aquisição de dados. Após estes procedimentos, o sistema estará encerrado
Actores	Professor
Pré-condições	Sistema de experimentação em funcionamento
Pós-condições	O sistema deve encontrar-se devidamente encerrado

Tabela 14. Especificação do caso de uso [Professor] Encerramento do Sistema

Detalhes	Descrição
ID	6
Nome	[Professor] Alteração de temperatura
Descrição	A alteração da temperatura consiste na verificação da existência de líquido no tanque secundário. Caso este exista, será efectuada a alteração da temperatura. Visto que o Professor terá maior conhecimento do sistema, será este actor que irá determinar qualquer a temperatura máxima que poderá ser imposta para o fluido
Actores	Professor
Pré-condições	Sistema de experimentação em funcionamento Não pode estar em curso qualquer experiência
Pós-condições	A temperatura no fluido deve estar devidamente alterada

Tabela 15. Especificação do caso de uso [Professor] Alteração de temperatura

Como se pode verificar, enquanto os dois primeiros são facilmente perceptíveis, o último refere-se à alteração da temperatura no tanque secundário, que assim irá influenciar a temperatura do fluido e os respectivos resultados finais. Incluídas num sistema geral (Sistema de Experimentação de Transferência de Calor), adequam-se a uma utilização do sistema em experimentação remota, visto que estas acções poderiam ser disponibilizadas por um interface apresentado aos utilizadores sobre a forma local ou remota (através da Internet, por exemplo). De referir que a divisão dos actores em Aluno e Professor, permite o estabelecimento de restrições de utilização a cada um destes, com o Professor a ter os maiores privilégios e o Aluno a ter as maiores condicionantes de experimentação e controlo.

5.9.1. Avaliação

Enquanto nestes diagramas apenas são especificadas as acções possíveis pelos actores envolvidos, cada uma dessas acções é detalhada com maior profundidade nos diagramas de sequência, apresentados na secção seguinte.

A utilização dos diagramas de casos de uso facilitou a identificação das potenciais interacções entre os actores do sistema e o respectivo sistema, tendo sido uma ferramenta muito útil para a correcta modelação do sistema em causa.

5.10. Interacções Identificadas

Partindo dos casos de uso identificados, irá proceder-se ao detalhe das acções especificadas, através da utilização dos diagramas de sequência. Estes diagramas irão descrever a par e passo todas as interacções entre os diversos sistemas

secundários para a realização de uma determinada acção, bem como as respostas que serão esperadas durante o decorrer destas interacções [21].

Começando pela acção Ligação do Sistema, poderá especificar-se dois diagramas de sequência, destinados a cada um dos actores. No entanto, as interacções para os dois actores são iguais, pelo que os procedimentos identificados também o são.

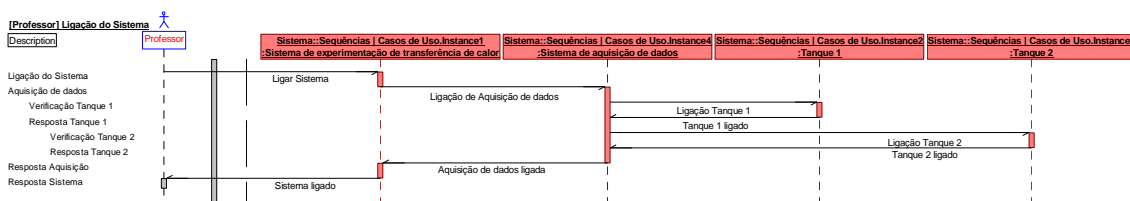


Figura 11. Diagrama de sequência da acção Ligação do Sistema, relativa ao actor Professor

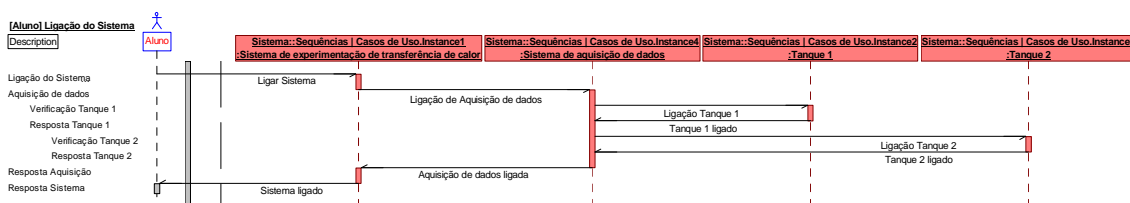


Figura 12. Diagrama de sequência da acção Ligação do Sistema, relativa ao actor Aluno

Como se pode verificar, a acção Ligação do Sistema irá desencadear uma sequência de acções intermédias, que terão como objectivo final a realização da tarefa indicada, seja através de interacção local ou remota.

Desta forma, a ligação do sistema de experimentação de transferência de calor levará à ligação dos seus sistemas secundários: sistema de aquisição de dados, tanque principal e tanque secundário.

Enquanto a ligação do sistema de aquisição de dados trata sobretudo de colocar em funcionamento o software e hardware necessários ao registo dos dados experimentais, a ligação do tanque principal e secundário passa pela ligação dos seus sistemas de transporte e dos seus sensores, ligados ao sistema de aquisição de dados. Desta forma, é possível controlar o funcionamento de cada um destes sistemas, podendo-se identificar em qualquer momento alguma anomalia no funcionamento do mesmo. Todos os sistemas enviam uma resposta ao sistema que interagiu com estes, com o sistema de aquisição de dados a devolver a resposta final ao sistema de experimentação de transferência de calor. Com o mesmo objectivo, a acção Encerramento do Sistema é modelada de modo semelhante.

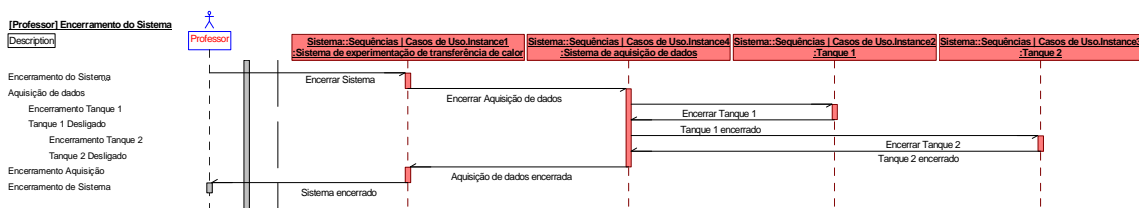


Figura 13. Diagrama de seqüência da acção Encerramento do Sistema, relativa ao actor Professor

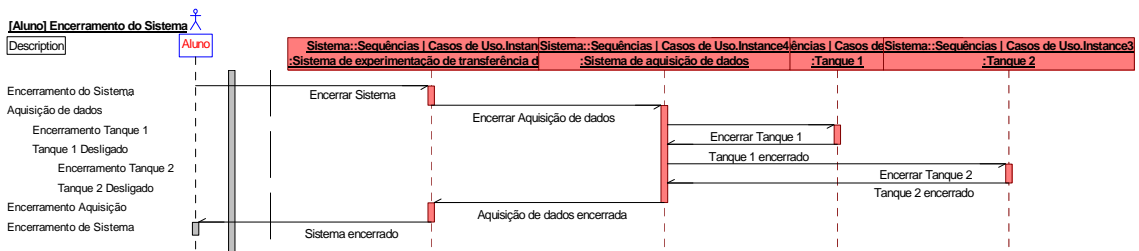


Figura 14. Diagrama de seqüência da acção Encerramento do Sistema, relativa ao actor Aluno

Neste caso, em vez se proceder à ligação de todos os sistemas secundários, procede-se ao encerramento dos mesmos. Tal como acontece na Ligação do Sistema, o sistema de experimentação de transferência de calor envia uma mensagem ao sistema de aquisição de dados, o qual procede à ligação dos sensores e transporte do tanque principal e secundário, recebendo uma mensagem de confirmação no final de cada operação.

No final, o sistema de aquisição de dados envia uma mensagem de confirmação ao sistema de experimentação de transferência de calor, sendo este o interface entre o utilizador e o sistema. Tal como se pode verificar, os procedimentos são iguais para os dois actores, visto que a interacção destes com o sistema é realizada do mesmo modo.

Finalmente, para concluir a criação dos diagramas de seqüência relativo às acções que os actores poderão realizar, era necessário especificar os diagramas relativos à acção Alteração de Temperatura, tanto para o actor Aluno, como para o actor Professor. Os mesmos são apresentados abaixo.

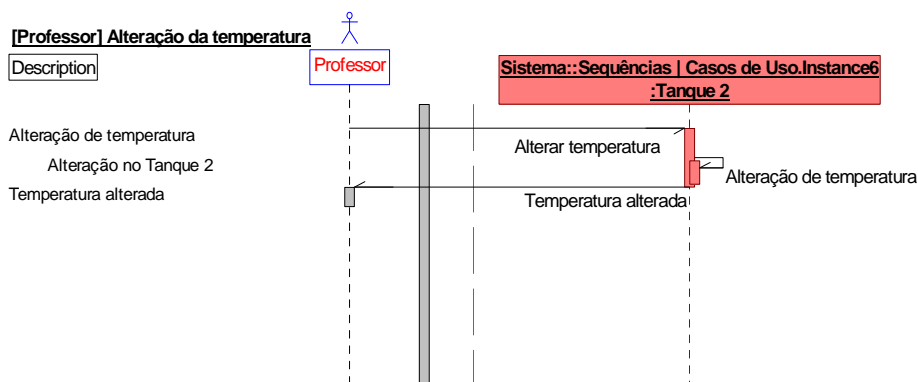


Figura 15. Diagrama de seqüência da acção Alteração de Temperatura, relativa ao actor Professor

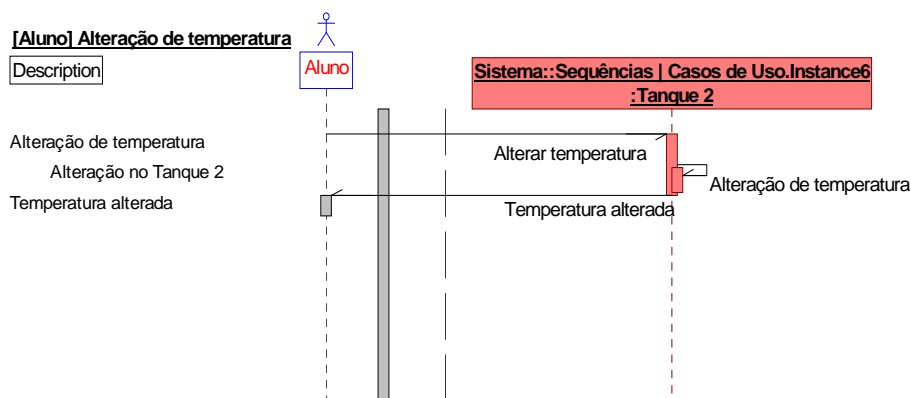


Figura 16. Diagrama de sequência da acção Alteração de Temperatura, relativa ao actor Aluno

Os diagramas criados representam os procedimentos envolvidos na acção de alteração da temperatura do tanque secundário (Tanque 2), para que se possa adequar a mesma às experiências a realizar. Tal como se pode verificar, tanto o Aluno como o Professor podem alterar a referida temperatura, sendo que a mesma é alterada directamente no tanque. Como se poderá ver nos diagramas paramétricos apresentados na secção 5.11, a temperatura estabelecida será verificada e caso a mesma esteja acima de um determinado valor pré-estabelecido, será colocada num valor seguro.

Tal procedimento é actualmente ignorado no sistema real (visto que o sistema é controlado pela responsável, que assim não permite qualquer alteração mais exagerada), pelo que não existe qualquer controlo de segurança da temperatura definida no controlador de temperatura utilizado no sistema de refrigeração do tanque secundário. No entanto, tal deverá ser considerado numa eventual utilização através de experimentação remota, visto que um potencial erro dos utilizadores poderá colocar em causa a integridade física de todo o sistema.

5.10.1. Avaliação

Com estes diagramas de sequência, ficaram definidas as principais acções possíveis pelos utilizadores do sistema, bem como a sua interacção com o sistema. A sua utilização também permite que para eventuais alterações dos casos de uso, se possa ajustar os respectivos diagramas de sequência para que representem adequadamente as acções identificadas. No entanto, visto que representam uma parte importante do modelo, a sua modelação deve ser feito com todo o cuidado necessário (sobretudo com a ajuda da Investigação em Acção, a qual irá permitir uma maior reflexão sobre as capacidades e interacções do sistema), para o resultado final seja fiel ao sistema real existente.

5.11. Actividades e Parâmetros

O próximo passo a seguir na metodologia SYSMOD consiste na criação dos diagramas paramétricos e de actividade. Desta forma, é possível estabelecer padrões de comportamento de utilização do sistema para tarefas repetitivas e que são resolvidas sempre da mesma forma [21], podendo-se assim melhorar a compreensão do sistema, bem como criar uma plataforma para possíveis alterações do mesmo para outro tipo de actividades.

Começando pelos diagramas de actividade, existem diversos tipos de padrões de comportamento passíveis de utilização [21], entre os quais o padrão Sucesso (que verifica após cada acção se esta foi bem sucedida ou não), Terminação (que permite sempre o encerramento do processo após cada acção) e o padrão Desvio (que permite o regresso a uma acção anterior, após conclusão de outra acção).

Relativamente a este sistema de sistemas e sua modelação, foi utilizado o padrão Sucesso, visto que o sistema irá verificar o resultado das suas acções e apresentar a resposta adequada a estas e só terminará caso os utilizadores assim o desejem.

Utilizando como referência os anteriores casos de uso e respectivos diagramas de sequência, são representadas nos diagramas de actividades os respectivos processos identificados (ligação e encerramento do sistema de experimentação de calor, bem como a alteração de temperatura do tanque secundário), detalhando-se o seu funcionamento interno e as suas respostas previstas.

Começando pelo processo Ligação do Sistema de Experimentação de Transferência de Calor, foi criado o respectivo diagrama.

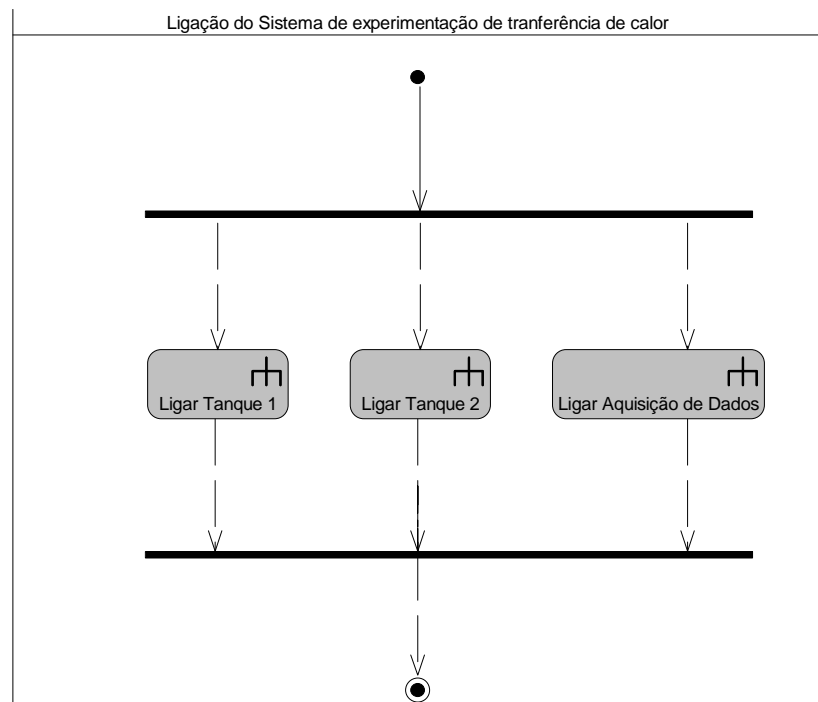


Figura 17. Diagrama de actividade do processo Ligação do Sistema de Transferência de Calor

Como se pode verificar, este processo deriva em diversos processos em paralelo (Ligar Tanque 1, Ligar Tanque 2 e Ligar Aquisição de Dados), destinados a ligar os sistemas secundários presentes no interior do sistema principal. Desta forma, é necessário também especificar os respectivos diagramas de actividade para cada um destes processos.

Relativamente ao processo Ligar Tanque 1, o diagrama de actividade referente a este processo é o seguinte.

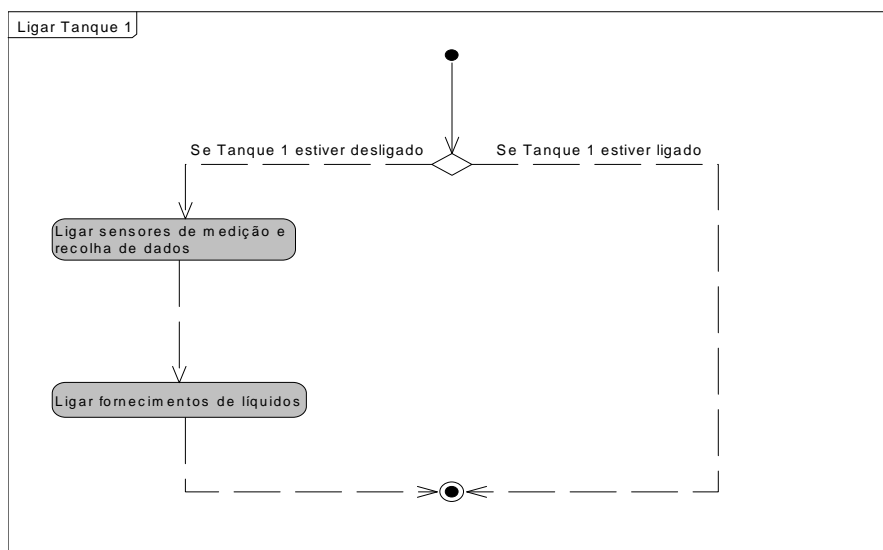


Figura 18. Diagrama de actividade do processo Ligar Tanque 1

No que concerne este processo, pode-se atentar ao facto de se verificar se o tanque se encontra ligado ou desligado, antes de efectuar qualquer acção. Caso este esteja ligado, não será efectuada qualquer acção, visto que os sensores e respectivo fornecimento de líquidos estarão já em funcionamento, pelo que não será necessária qualquer intervenção.

Caso o tanque se encontre desligado, são executadas as acções de ligação dos sensores de medição e recolha de dados, bem como o fornecimento dos líquidos, iniciando assim a actividade do tanque principal.

Para o tanque secundário (Tanque 2), é também apresentado um diagrama de actividade semelhante, tal como se pode verificar.

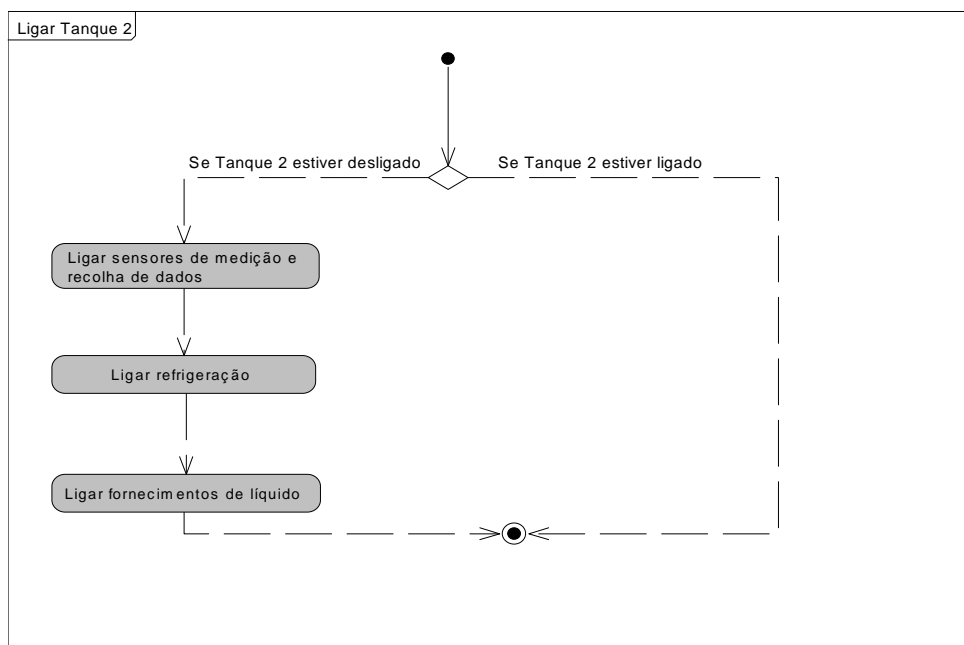


Figura 19. Diagrama de actividade do processo Ligar Tanque 2

Tal como no tanque principal, é verificado se o tanque já se encontra ligado ou não, para evitar qualquer procedimento desnecessário. Caso o mesmo não esteja ligado, são iniciadas as respectivas acções de ligação dos sensores de medição e de recolha de dados e fornecimento de líquidos, bem como a ligação da refrigeração (presente apenas neste tanque).

Finalmente, fica a falta o processo Ligar Aquisição de Dados, que se pode ver no seguinte diagrama.

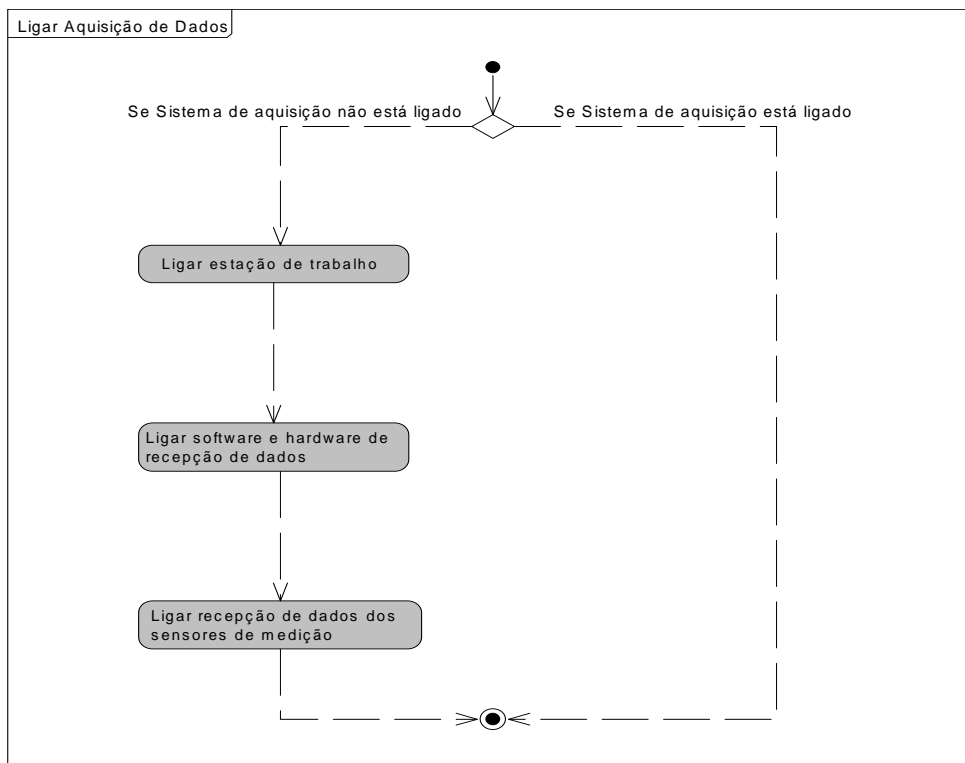


Figura 20. Diagrama de actividade do processo Ligar Aquisição de Dados

No caso deste processo, também é verificado se o sistema de aquisição de dados já se encontra ligado, evitando qualquer acção caso assim seja. No entanto, caso não esteja ligado, são efectuadas as acções de ligação da estação de trabalho (onde se encontra localizado todo o software a utilizar), ligação do software e hardware de recepção de dados (para iniciar o registos dos dados experimentais) e inicio da recepção dos dados provenientes dos sensores de medição.

Desta forma, através destes quatro diagramas é possível detalhar todos os processos envolvidos na ligação do sistema de experimentação de transferência de calor. No entanto, é também necessário criar os diagramas relativas ao processo oposto, encerrar o respectivo sistema. Tais diagramas são apresentados de seguida.

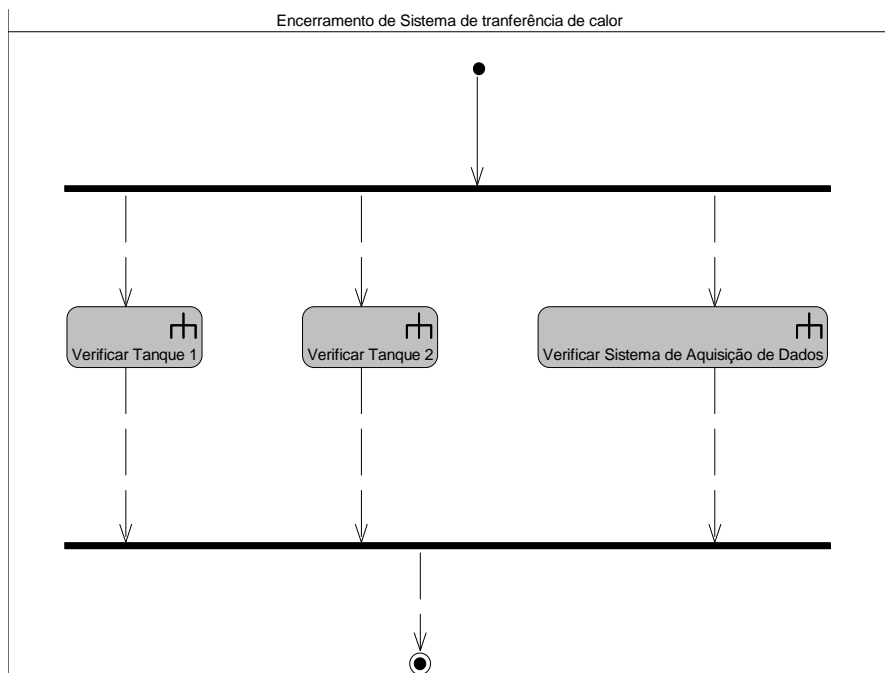


Figura 21. Diagrama de actividade do processo Encerramento do Sistema de Transferência de Calor

Tal como acontece no processo Ligação do Sistema de Transferência de Calor, são também verificados cada um dos sistemas secundários existentes, de modo a que o seu encerramento seja efectuado da forma correcta. E tal como no primeiro processo, é necessário detalhar cada um dos procedimentos secundários através dos respectivos diagramas de actividade.

No que concerne ao processo Verificar Tanque 1, é detalhado no seguinte diagrama.

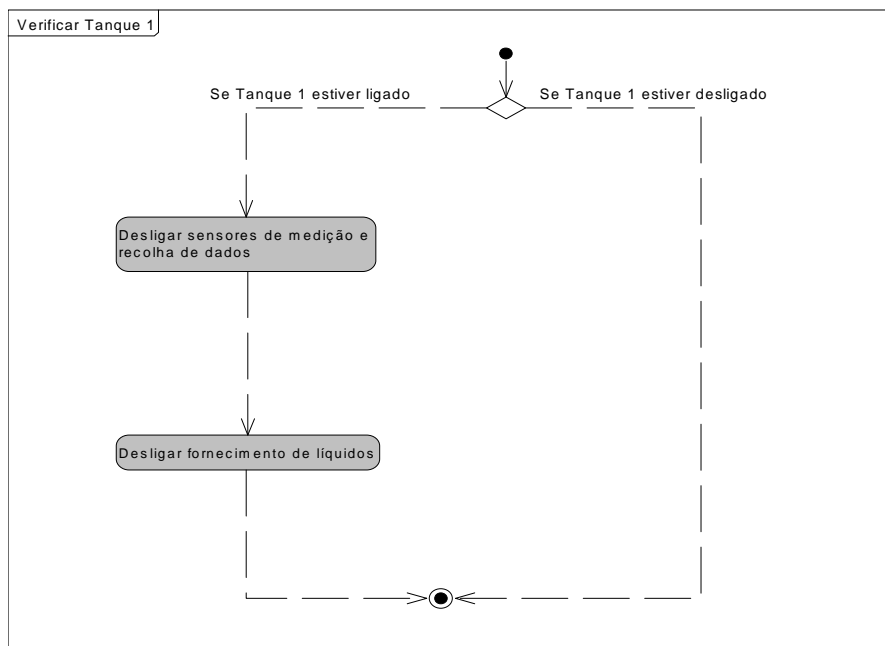


Figura 22. Diagrama de actividade do processo Verificar Tanque 1

Embora semelhante aos processos de ligação dos respectivos tanques, este processo diferencia-se pelo facto de efectuar as acções contrárias, ou seja, desligarem-se os sensores de recolha de dados, bem como o fornecimento de líquidos. Tais acções só serão efectuadas se o tanque se encontrar ligado, pois caso assim não seja, os seus componentes internos não se encontrarão em funcionamento e não necessitarão de qualquer intervenção.

Para o tanque secundário (Tanque 2), é também apresentado o respectivo diagrama de actividade, semelhante ao do tanque principal.

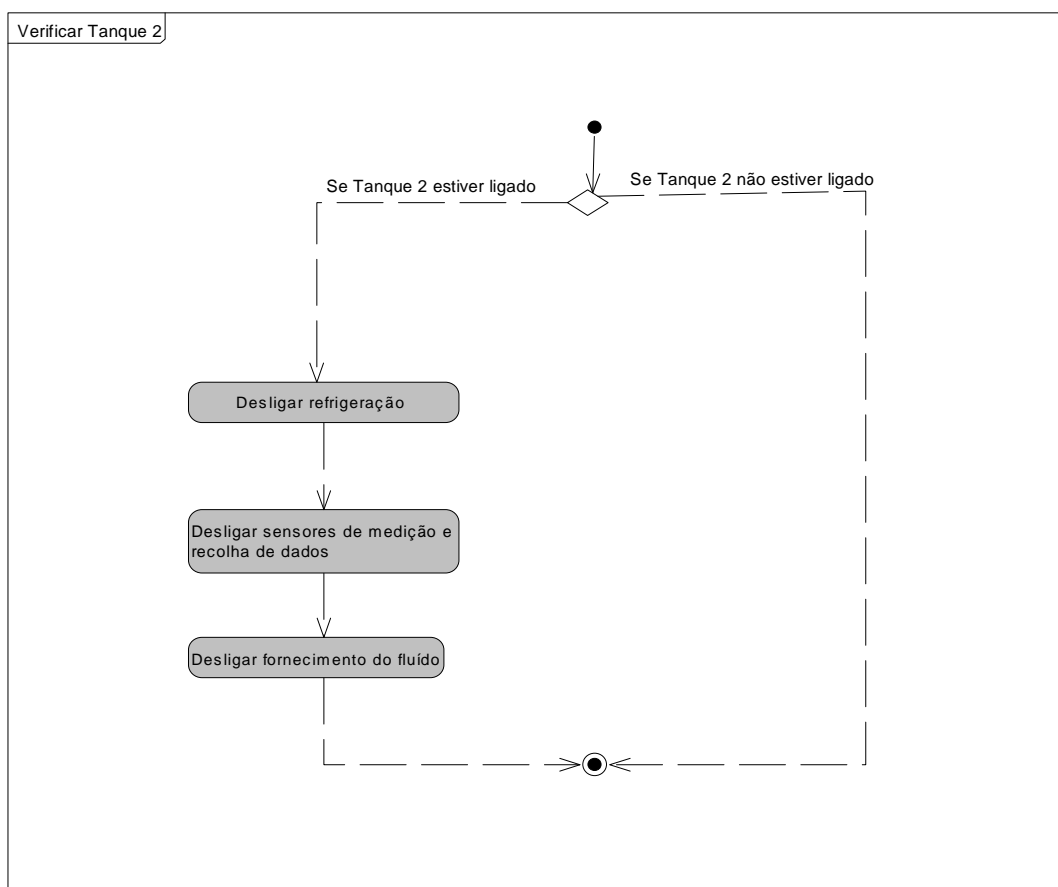


Figura 23. Diagrama de actividade do processo Verificar Tanque 2

Tal como é possível constatar, as acções realizadas no tanque secundário são em tudo semelhantes às realizadas no tanque principal. No entanto, existe uma acção adicional, que consiste no encerramento do sistema de refrigeração, presente apenas neste tanque e que por isso necessita de especial atenção.

Finalizando o detalhe do processo Encerramento do Sistema de Transferência de Calor, resta apenas especificar o diagrama de actividade relativo à verificação e encerramento do sistema de aquisição de dados.

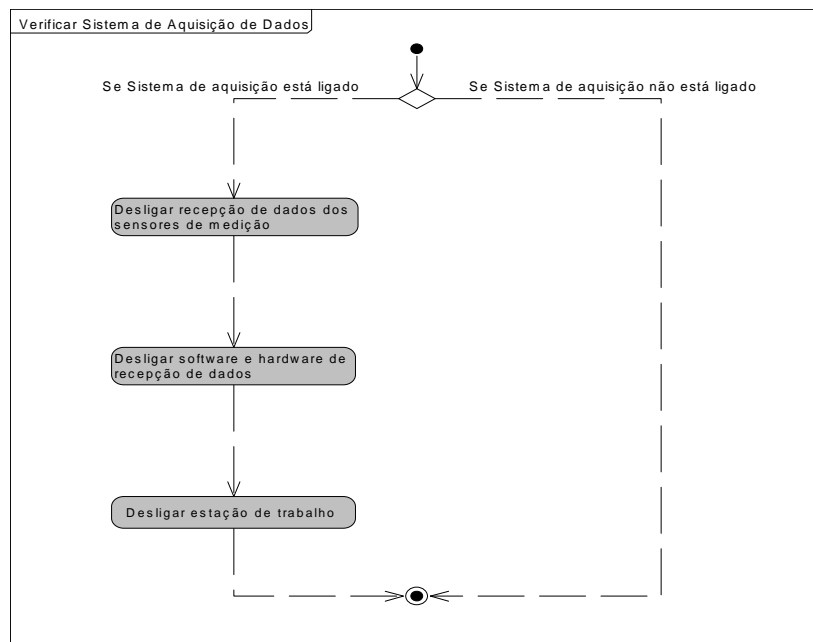


Figura 24. Diagrama de actividade do processo Verificar Sistema de Aquisição de Dados

Relativamente a este processo, é semelhante ao processo de ligação do mesmo sistema, só que se diferencia por se realizar em ordem inversa. Desta feita, começa-se por desligar a recepção dos dados (para evitar qualquer corrupção dos mesmos), procedendo-se ao encerramento do software e hardware utilizado nessas actividades. Por fim, encerra-se a estação de trabalho utilizada.

No caso da estação já se encontrar desligada, não é executada qualquer acção, visto que não é necessária qualquer intervenção.

Além dos processos Ligação e Encerramento do Sistema de Transferência de Calor, existe um processo adicional que necessita de especificação: Alteração da Temperatura no Tanque 2. Tal é apresentado no seguinte diagrama.

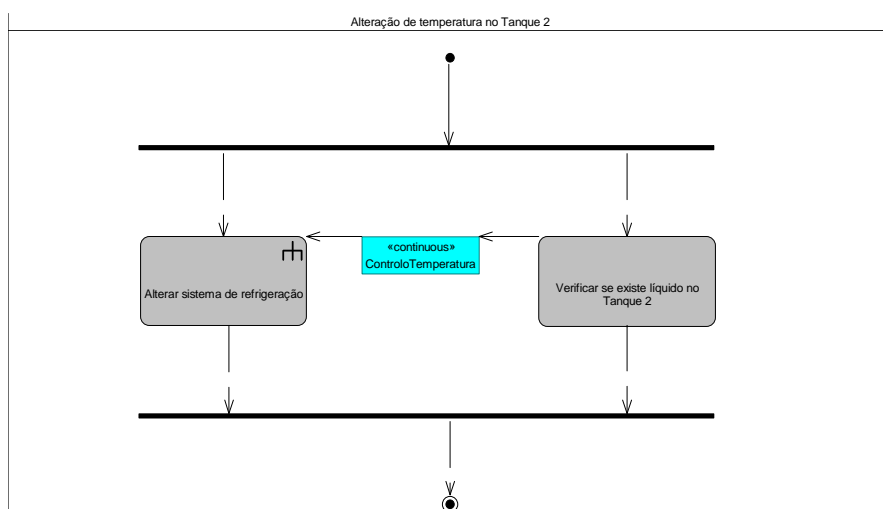


Figura 25. Diagrama de actividade do processo Alteração de temperatura no Tanque 2

Neste processo, pode-se identificar dois processos secundários: verificação da existência de líquido no tanque secundário e alteração do sistema de refrigeração. Tal deriva das necessidades existentes neste tanque, em particular do sistema de refrigeração instalado.

Visto que o mesmo se encontra ligado a um controlador de temperatura, qualquer alteração da mesma, em condições não recomendáveis, poderá levar a potenciais danos neste tanque. Torna-se por isso necessário verificar se existe algum líquido no tanque secundário e caso tal exista, verificar a sua temperatura e alterar a mesma de acordo com o pretendido.

Relativamente ao processo de verificação do líquido, tal é apresentado neste diagrama.

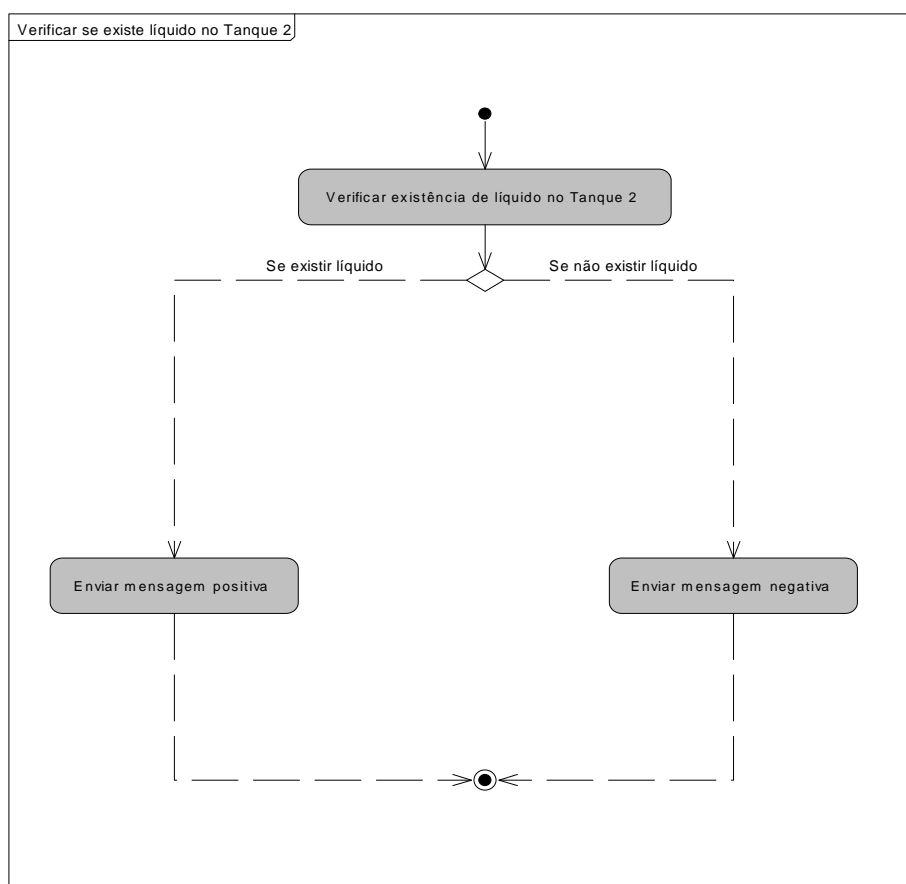


Figura 26. Diagrama de actividade do processo Verificação da existência de líquido no Tanque 2

Tal como se pode constatar, este processo consiste sobretudo numa validação da existência do líquido, enviando-se uma resposta adequada caso este exista ou não. Tal resposta será utilizada no processo de Alteração do Sistema de Refrigeração, para concretizar a respectiva acção ou não.

Relativamente a este último processo, o diagrama de actividade que especifica o seu funcionamento é o seguinte.

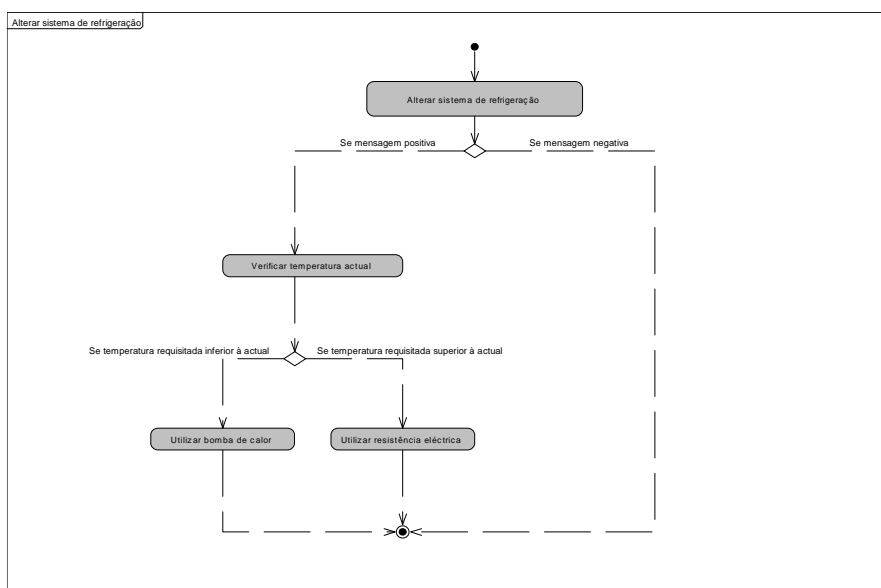


Figura 27. Diagrama de actividade do processo Alteração do Sistema de Refrigeração

Utilizando a resposta enviada pelo processo anterior, são desencadeadas ou não determinadas acções. Caso a resposta enviada seja negativa, a temperatura não é alterada. No entanto, caso a resposta seja positiva, são executadas diversas acções, tais como Verificar a Temperatura Actual.

Em função da temperatura actual do líquido, é utilizado a bomba de calor (para arrefecimento) ou a resistência eléctrica (para aquecimento), que irão assim influenciar a temperatura do líquido presente no tanque. A temperatura do tanque é fornecida pelo controlo da temperatura, que funciona continuamente.

Com estes diagramas de actividade é possível destacar os detalhes mais importantes dos procedimentos internos de cada processo, bem como as possíveis respostas. Além desta ferramenta, a metodologia SYSMOD sugere a utilização dos diagramas paramétricos para especificação de determinados variáveis de restrição, utilizadas nos procedimentos referidos [21]. Estes diagramas apresentam-se como dos mais importantes em toda a modelação do sistema, visto que representam as relações físicas entre os diversos elementos do modelo.

No que concerne o controlo da temperatura do líquido existente no tanque secundário, especificou-se um diagrama paramétrico no qual se especifica o controlo da referida temperatura, impondo uma restrição de controlo para que tal não exceda valores pré-estabelecidos.

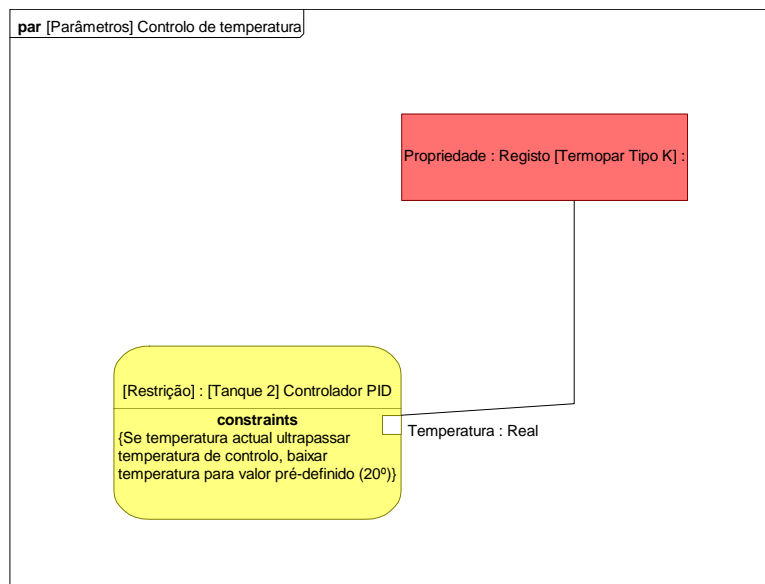


Figura 28. Diagrama paramétrico do Controlo de Temperatura

Obtendo os registos da temperatura directamente a partir do termopar Tipo K, o Controlador PID tem pré-definido um valor máximo de temperatura que poderá ser atingido (temperatura de controlo), a partir do qual irá descer até um valor razoável, que também se encontra pré-definido. O valor da temperatura é especificado neste diagrama como um valor real, podendo assim obter-se valores com casas decimais e não arredondamentos.

Além deste diagrama paramétrico, foi necessário especificar outro diagrama relativo ao transporte do fluido e controlo do caudal do mesmo. Tal é apresentado no seguinte diagrama.

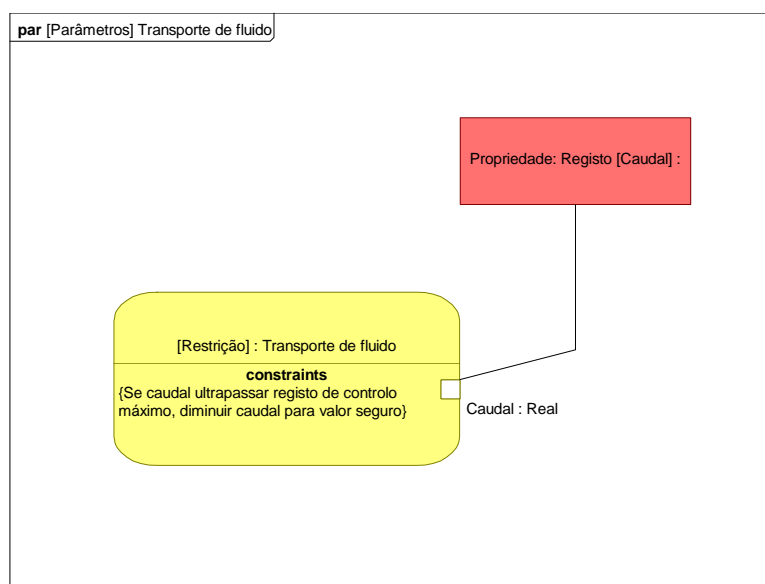


Figura 29. Diagrama paramétrico do Transporte de fluido

Tal como se verifica para o controlo da temperatura, para o caudal também é estabelecido um valor de controlo, utilizado para verificar se este se encontra dentro dos limites físicos do equipamento ou não.

Tal valor é obtido a partir do sensor de medição do caudal, que regista os valores do mesmo e actua de acordo com as necessidades experimentais e dos utilizadores. Se o valor do caudal ultrapassar em algum momento o valor de controlo pré-definido, será alterado para um valor seguro e que não coloque em causa a integridade física do equipamento.

Estes diagramas paramétricos permitem assim estabelecer os propriedades relativas as estes dois elementos, especificando os parâmetros que devem ser controlados e devidamente acautelados.

5.11.1. Avaliação

Estando definidos os diagramas de actividade e paramétricos relativos aos vários processos existentes no sistema em causa, pode-se verificar que estes especificam informação importante acerca do funcionamento interno do sistema de sistemas, bem como os seus comportamentos e interacções possíveis entre si. Além disso, permitem estruturar toda a composição do sistema, sendo ferramentas bastante interessantes na elaboração de um modelo de um sistema de sistemas, utilizando a notação SysML.

No que concerne os diagramas paramétricos, estes permitem acima de tudo, estabelecer os propriedades relativas as estes dois elementos, especificando os parâmetros que devem ser controlados e devidamente acautelados. Apesar da sua importância poder parecer algo relativa, a utilização dos diagramas paramétricos permite introduzir componentes como a Física e a Química na modelação de sistemas, podendo essas componentes ser devidamente integradas nos sistemas de sistemas.

Como principais dificuldades, de destacar a dificuldade em determinar quais os elementos que deveriam ser modelados através dos diagramas paramétricos, visto que poucas propriedades apresentavam a necessidade de estar em controlo constante. No entanto, após uma análise mais cuidada, foi possível determinar quais os alvos que necessitavam de rigor e de vigilância, resultando nos diagramas apresentados.

Outro aspecto que apresentou alguns problemas iniciais foi a falta de interligação inicial entre os vários diagramas, em virtude de uma falta de experiência de modelação com SysML, o que pode levar eventualmente à criação de diagramas

distintos e sem qualquer interligação. Tal facto prejudicará a elaboração do modelo e poderá levar, mais tarde ou mais cedo, a uma eventual reformulação dos diagramas criados, para que possam funcionar como um conjunto e fornecer uma visão global e geral da composição interna do sistema.

5.12. Blocos internos

Embora os diagramas de actividade e paramétricos forneçam uma visão do funcionamento interno do sistema, não detalham o funcionamento interno dos sistemas internos apresentados. Tal é apenas possível com a utilização dos diagramas de blocos internos, ferramenta apresentada na notação SysML para especificação dos detalhes internos dos sistemas integrados no sistema principal [21]. É precisamente partindo desta necessidade que a metodologia SYSMOD indica a sua utilização para a próxima fase, de forma a detalhar ainda mais o funcionamento interno do sistema.

Desta forma e começando pelo início, é necessário definir os diagramas para cada um dos sistemas existentes: Sistema de Experimentação de Transferência de Calor, Sistema de Aquisição de Dados, Tanque 1 e Tanque 2. Relativamente ao Sistema de Experimentação de Transferência de Calor, o diagrama de blocos internos é o seguinte.

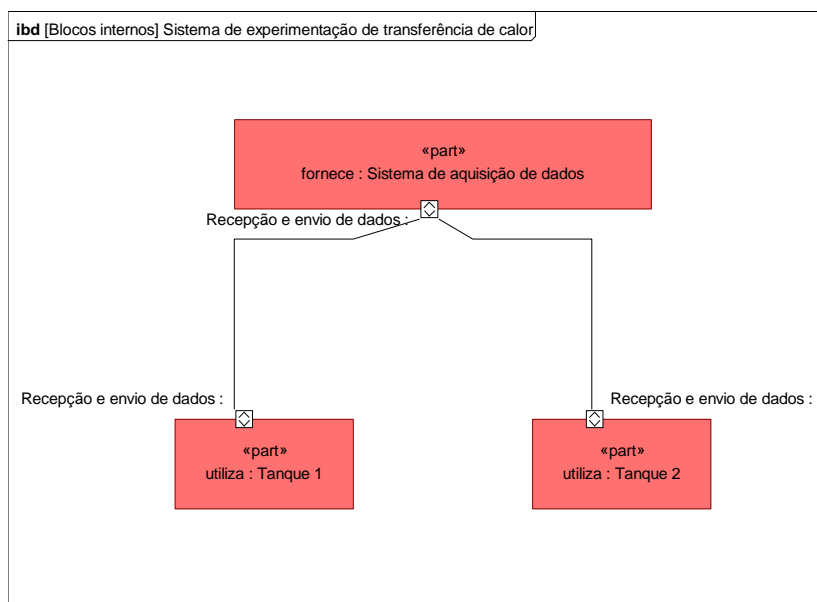


Figura 30. Diagrama de blocos internos do Sistema de Transferência de Calor

Como se pode verificar, este sistema inclui os respectivos sistemas secundários, servindo de interface para os utilizadores (já considerando a experimentação remota).

Desta feita, o sistema de aquisição de dados interage com o tanque principal e tanque secundário através da recepção de dados e envio de programação dos respectivos sensores, utilizando para isso determinadas portas de fluxo bidireccionais.

No que concerne o tanque principal (Tanque 1), foi elaborado o seguinte diagrama de blocos internos.

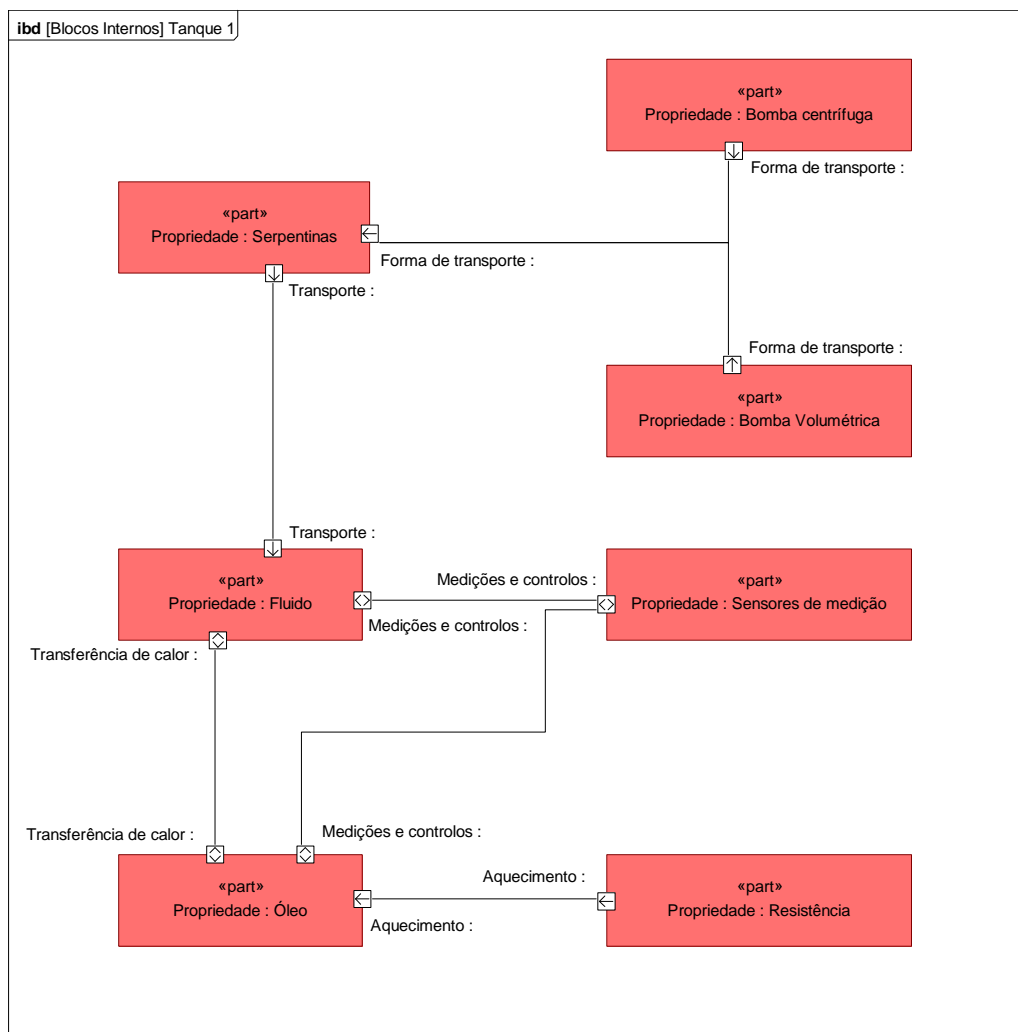


Figura 31. Diagrama de blocos internos do Tanque 1

Neste caso, este diagrama permite a especificação das interacções dos diversos elementos internos deste sistema. Desta feita, de destacar que o transporte, efectuado com recurso às diversas bombas, através de uma porta de fluxo que representa o transporte efectivo, o fluido que se encontra ligado às serpentinas (visto que é por estas que ele será transportado) e que também envia registos de medição através de portas bidireccionais específicas (envio e recepção de dados), além do óleo, que irá interagir com o fluido de forma indirecta (visto que nunca se encontram em contacto directo) e cuja temperatura será mantida e controlada por uma resistência eléctrica. De

facto, a única interacção entre o fluido e o óleo será a transferência de calor, que representa o principal alvo de atenção de toda a experiência.

Além do tanque principal, também foi elaborado um diagrama de blocos internos para o tanque secundário, diferindo bastante do primeiro.

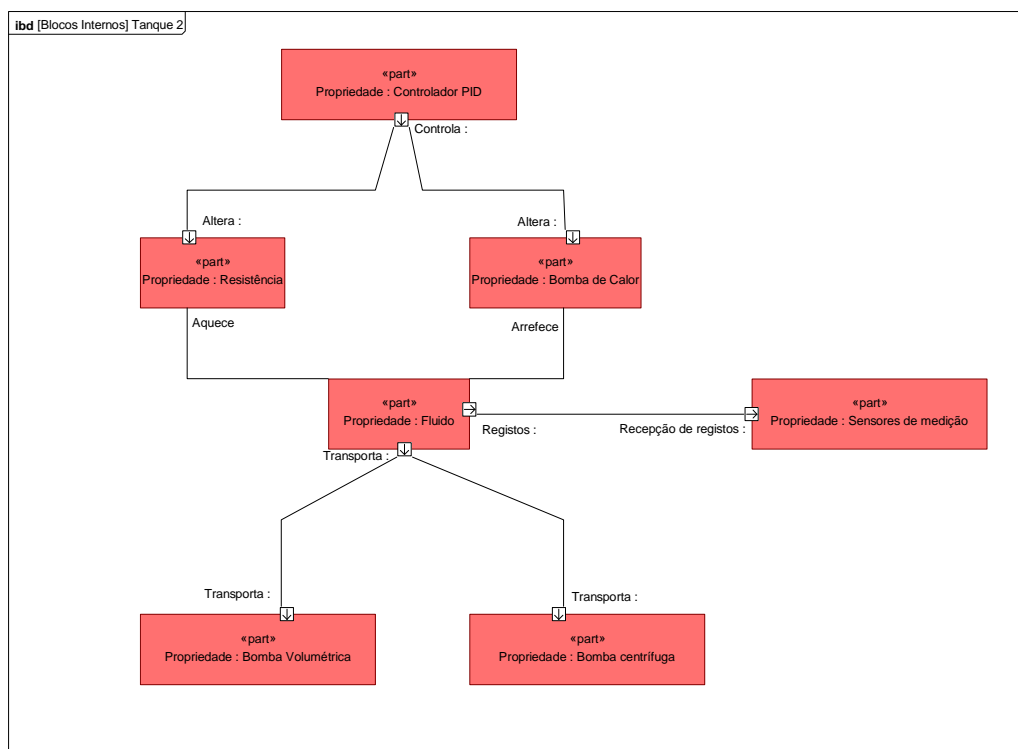


Figura 32. Diagrama de blocos internos do Tanque 2

No que concerne o tanque secundário, o principal destaque vai para o controlo da temperatura do fluido, utilizando para isso o controlador PID, juntamente com a bomba de calor e a resistência eléctrica, que assim irá influenciar a temperatura final. Além disso, o fluido encontra-se ligado a sensores de medição que irão permitir o registo das diversas flutuações do mesmo, ao longo das experiências.

Relativamente ao transporte do fluido, o funcionamento é semelhante ao apresentado pelo tanque principal, recorrendo às bombas existentes (volumétrica e centrífuga) para a o transporte, mas dispensando a utilização das serpentinas condutoras. As interacções entre os diversos elementos são realizadas através de portas fluxos, sendo que a única interacção bidireccional apresentada existe entre os sensores de medição e o fluido. Para finalizar a elaboração dos diagramas de blocos internos, resta apenas a elaboração do diagrama relativa ao sistema de aquisição de dados, apresentado de seguida.

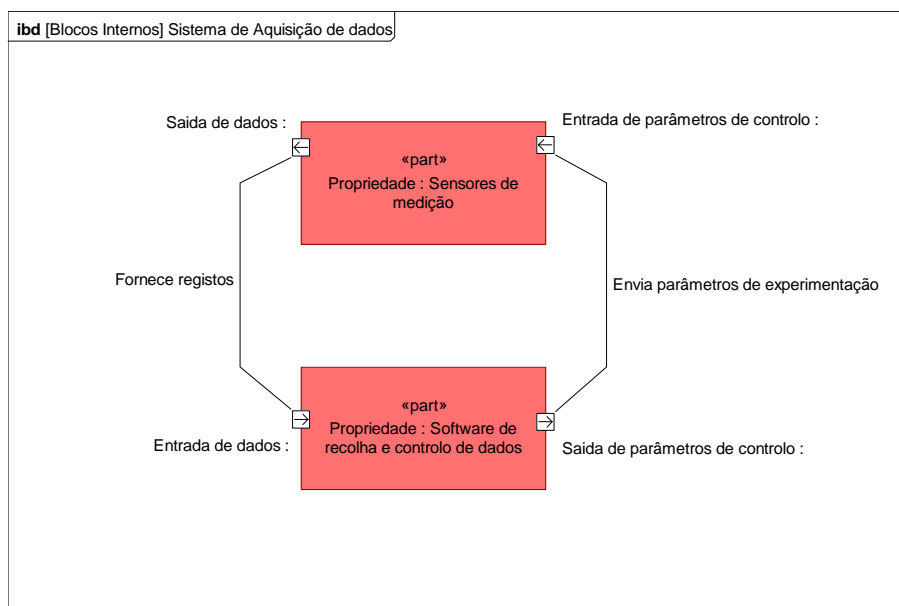


Figura 33. Diagrama de blocos internos do Sistema de Aquisição de Dados

O Sistema de Aquisição de Dados apresenta interacções bastante simples, visto que necessita apenas de receber e enviar dados do software instalado para o hardware montado e vice-versa. Desta feita, enquanto o hardware terá como principal funcionalidade, o fornecimento dos registos experimentais obtidos e que serão guardados pelo software utilizado, este último poderá também programar os sensores, enviando os respectivos parâmetros. Com isto, poderá alterar as gamas de valores a registar, bem como os próprios dados que deverão ser obtidos, abrangendo um elevado leque de possibilidades de recolha de dados.

5.12.1. Avaliação

Com os diagramas de definição de blocos internos, é possível estruturar o funcionamento interno dos sistemas integrados, identificando quais serão as necessidades de cada um deles, bem com os detalhes de interacção entre os mesmos que deverão ser considerados. Apresentam-se assim como uma ferramenta valiosa e complementar aos diagramas de actividade e paramétricos, devendo ser utilizados na elaboração do modelo de um sistema.

Como principal falha, de destacar a relativa ambiguidade na utilização das portas de fluxo, visto que embora representem os pontos de interacção entre os sistemas e elementos em causa, poderão ter um significado potencialmente diferente do inicialmente previsto e estipulado.

5.13. Estados e Pacotes

Embora a maioria dos aspectos do sistema já esteja modelado, ainda existem alguns aspectos que necessitam de atenção, tal como é referido na metodologia SYSMOD. Entre esses aspectos, devemos referir os estados possíveis em cada um dos sistemas existentes, bem como a divisão dos vários aspectos do modelo por pacotes, de forma a facilitar a estruturação de todo o sistema.

Começando pelos estados dos sistemas, estes são modelados através da utilização de diagramas de máquinas de estado, os quais permitem definir quais os vários estados possíveis durante o funcionamento do sistema, em função das acções executadas [21].

Embora os diagramas de sequência, actividade e casos de uso permitam definir os comportamentos do sistema de sistemas, é a partir destes que serão obtidos os diagramas de máquinas de estado, que permitirão detalhar ainda mais o funcionamento interno do sistema.

Começando com o sistema de experimentação de transferência de calor, que servirá de interface entre os utilizadores e o funcionamento interno do sistema, o respectivo diagrama de máquinas de estado é o seguinte.

Sistema de experimentação de transferência de calor

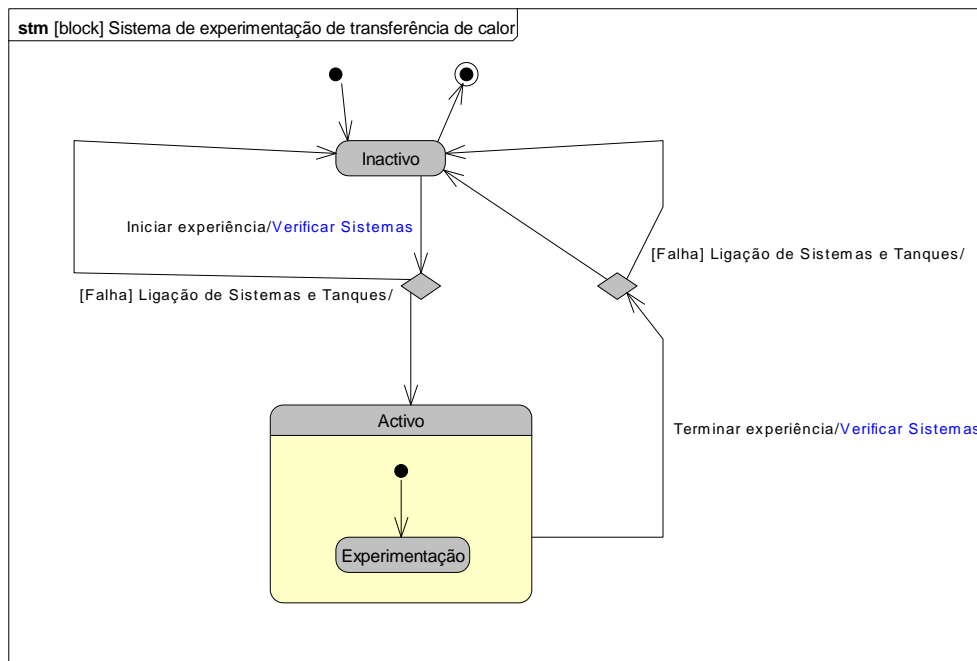


Figura 34. Diagrama de máquinas de estado do Sistema de Transferência de Calor

Neste diagrama, pode-se constatar que o sistema encontra-se inactivo, caso nenhuma acção seja executado. No entanto, caso seja iniciada a experiência, o

o sistema inicia o processo de verificação dos sistemas (modelado através do diagrama de sequência Ligação do Sistema de Experimentação de Transferência de Calor) e caso esta verificação falhe, o sistema continuará inactivo, visto que haverá algum problema num dos sistemas secundários.

Caso a verificação seja bem sucedida, o sistema começa o processo de experimentação, sendo apenas interrompido caso seja necessário terminar a experiência. Novamente, é efectuado o processo de verificação dos sistemas (também modelado pelo diagrama de sequência Encerramento do Sistema de Experimentação de Transferência de Calor), embora neste caso o sistema termine a sua execução em qualquer um dos casos. No entanto, caso exista alguma falha na verificação, será registado o motivo da falha para posterior avaliação.

No que concerne o tanque principal (Tanque 1), também foi criado o respectivo diagrama de máquinas de estado, apresentado de seguida.

Tanque 1

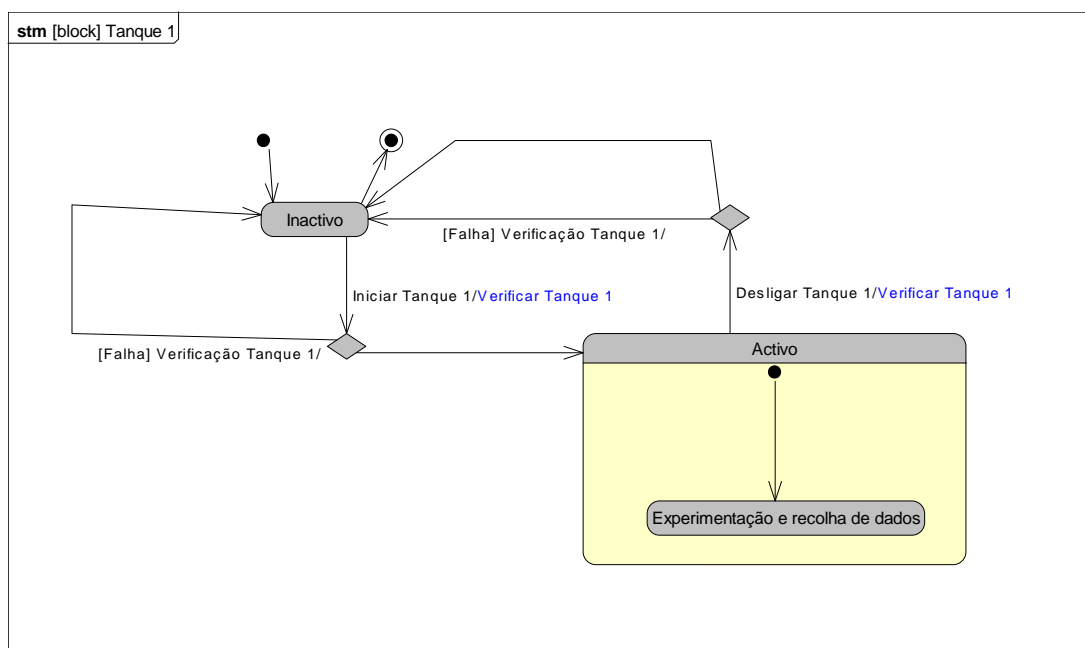


Figura 35. Diagrama de máquinas de estado do Tanque 1

Tal como aconteceu para o primeiro sistema, também o tanque principal se encontra inactivo até que seja requisitada a sua activação. Caso tal seja requisitada, é necessário que o tanque seja verificado, de modo a averiguar se existe algum problema com este. Se tal se verificar, o tanque voltará ao seu estado inactivo, retornando uma mensagem negativa (erro), evitando assim que haja algum dano devido ao funcionamento incorrecto.

No caso de não haver qualquer problema, o tanque ficará activo e começará a experimentação e recolha dos dados experimentais, até que seja ordenada a sua paragem. Tal como no sistema anterior, é verificado se existe algum problema no tanque, durante o seu encerramento, enviando uma mensagem de alerta caso tal aconteça. Em todo o caso, o tanque será sempre desligado.

Tal como aconteceu para o tanque principal, também é necessária a especificação de um diagrama de máquinas de estado para o tanque secundário, utilizado para efectuar o armazenamento dos fluidos.

Tanque 2

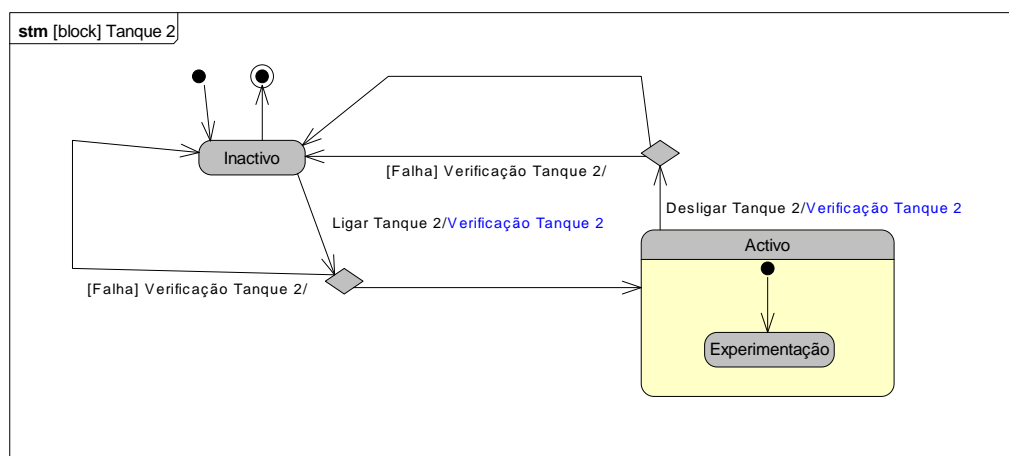


Figura 36. Diagrama de máquinas de estado do Tanque 2

Os estados apresentados no tanque secundário são em tudo semelhantes aos apresentados pelo tanque principal, sendo a única diferenciação o facto de quando este fica activo, não efectuar qualquer recolha de dados para o sistema de aquisição de dados (a medição do caudal é realizada entre o tanque secundário e o tanque principal). Além disso, na verificação do tanque, existe um aspecto adicional que também tem de ser considerado e que se trata do sistema de refrigeração, que também necessita de atenção.

Finalmente, o sistema de aquisição de dados também necessita de um diagrama de máquinas de estados, representativo dos seus vários estados de funcionamento.

Sistema de aquisição de dados

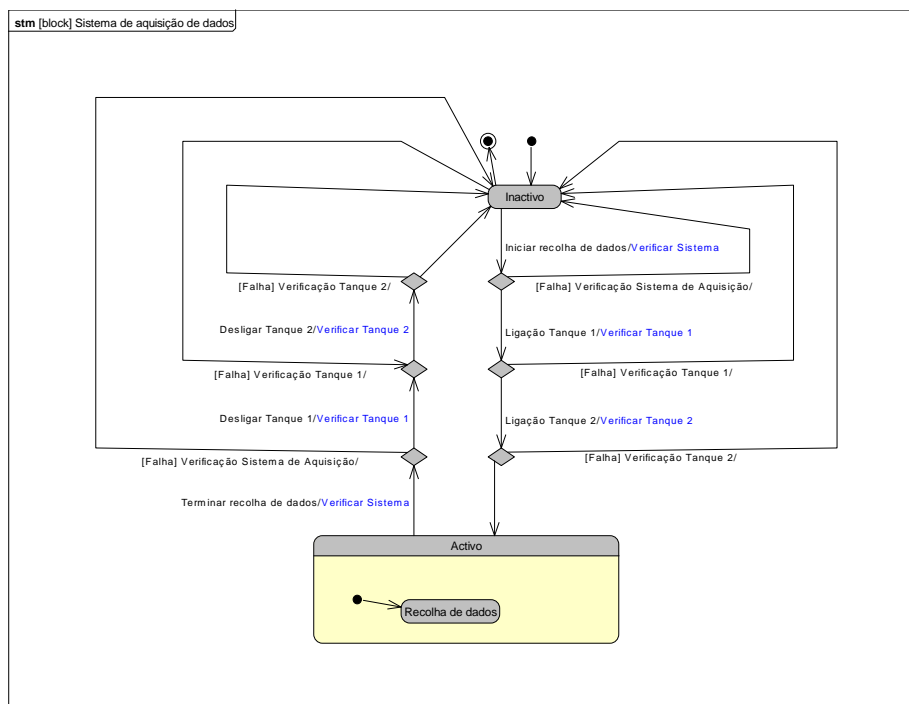


Figura 37. Diagrama de máquinas de estado do Sistema de Aquisição de Dados

Relativamente a este sistema e tal como os anteriores, este encontra-se inactivo até que seja requisitada a sua activação. Para que tal seja efectuada, o sistema de aquisição de dados procede à activação dos respectivos sensores de medição em cada um dos tanques, bem como à activação do hardware e software do próprio sistema.

Em cada um destes sistemas, caso seja encontrado algum erro, o sistema voltará a ficar inactivo, enviando uma mensagem de erro. Caso tal não aconteça, irá proceder-se à recolha de dados, iniciando-se assim as experimentações, até que seja ordenado o seu encerramento.

Em tal caso, tal como nos sistemas anteriores, serão efectuadas as verificações para o sistema de aquisição de dados, tanque principal e tanque secundário. Caso seja encontrado alguma anomalia, será enviado uma mensagem de erro para o sistema de experimentação de transferência de calor, indicando qual a anomalia encontrada.

Estando apresentados os diagramas de máquinas de estado de cada um dos elementos do sistema de sistemas, o último passo em falta segundo a metodologia SYSMOD consiste na criação do diagrama de pacotes referente a este sistema, o que irá permitir detalhar a estrutura do sistema através de uma visão geral do mesmo [21]. Tal diagrama será apresentado de seguida.

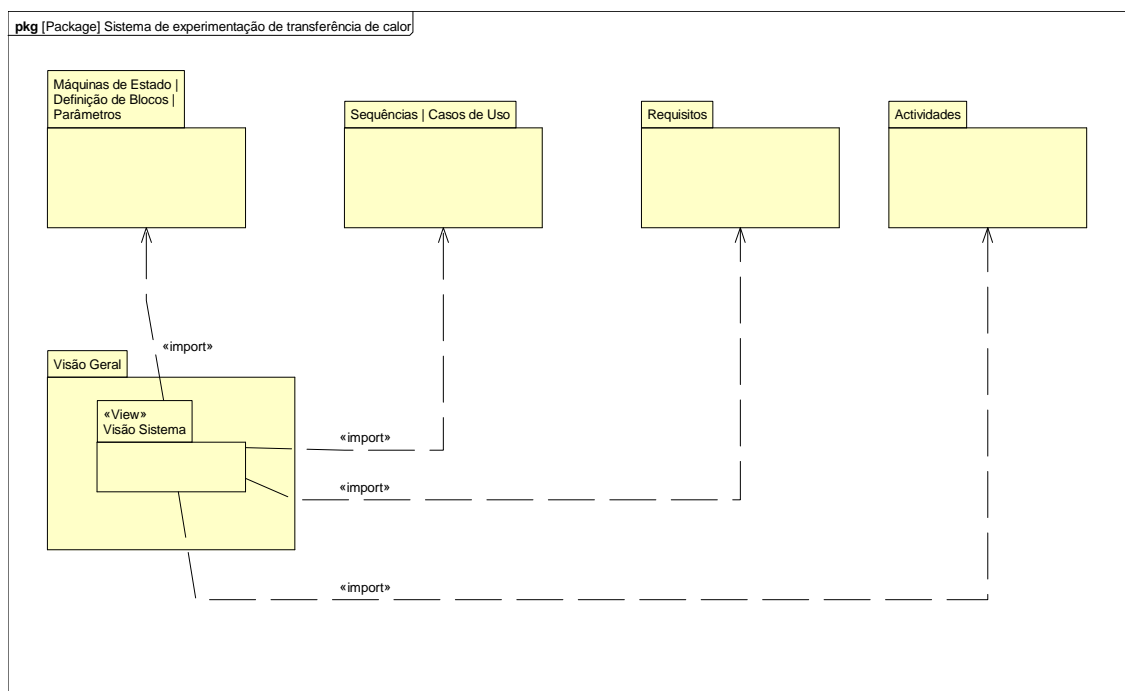


Figura 38. Diagrama de pacotes do Sistema de Transferência de Calor

Através deste diagrama, é possível detalhar quais os diversos pacotes secundários em que se inseriu os respectivos diagramas. Esta divisão foi realizada tendo em conta as especificidades de cada tipo de diagrama (necessidades de utilizar um tipo de diagrama para modelar outro tipo de diagrama, devendo ambos estar no mesmo local, por exemplo), bem como as necessidades da própria ferramenta de modelação utilizada (neste caso, o Artisan Studio Uno).

Assim sendo, foram criados 4 pacotes secundários, com o primeiro a englobar os aspectos relativos aos diagramas de máquinas de estado, definição de blocos e paramétricos. No segundo pacote foram englobados os diagramas de sequência e casos de uso, enquanto no terceiro pacote foram incluídos os diagramas de requisitos. Finalmente, no quarto pacote, foram incorporados os diagramas de actividades.

Todos estes pacotes serão incluídos num pacote geral, denominado de Visão Geral, que assim criará um modelo do sistema, partindo dos aspectos gerais e mais básicos (que facilitam a abordagem inicial ao sistema) para os pormenores mais específicos do mesmo (mais interessantes, visto que detalham em profundidade o funcionamento do sistema). Com a criação deste diagrama de pacotes, finaliza-se a modelação do sistema, obtendo-se assim uma representação virtual do sistema em causa que incorpora aspectos e pormenores que possibilitam a sua adaptação à experimentação remota, caso assim seja necessário.

5.13.1. Avaliação

Dada a possibilidade de criar várias visões de um mesmo modelo para os diferentes utilizadores do sistema, os diagramas de pacotes são uma ferramenta importante para reunir todos os detalhes respectivos a cada um dos utilizadores e apresentar uma visão completa e simplificada das possibilidades do sistema. A criação dos diagramas de estado, juntamente com o diagrama de pacotes, permite um maior aperfeiçoamento e detalhe do modelo do sistema.

Embora não sejam essenciais para o desenho do sistema, a sua utilização, leva à criação de uma estrutura mais limpa, que ajudará a esclarecer quaisquer dúvidas que surjam relativamente à execução do sistema (que deverá ser confirmada com o recurso à Investigação em Acção).

5.14. Discussão

Após a modelação do sistema referido neste caso prático, foi possível analisar vários aspectos ligados à notação SysML e sua aplicação, a partir dos quais foi possível retirar algumas informações sobre as suas possibilidades e limitações actuais.

5.14.1. Limitações do SysML

No que concerne as limitações do SysML, a validação dos pontos fortes e fracos apresentados anteriormente, a análise das principais dificuldades sentidas ao longo da modelação e a identificação das áreas de aplicação da notação, podemos estabelecer diversos dados comparativos, que ajudam a definir quais as capacidades da notação.

5.14.1.1. Validações estabelecidas

A elaboração deste caso prático permitiu validar alguns dos pontos fortes e fracos apresentados na análise teórica da notação SysML. Desta feita, começando pelos mecanismos de controlo, pode-se referir que apesar de podermos modelar os diversos fluxos e os seus controlos, existentes num determinado bloco, tais mecanismos são apenas modelados com recurso a elementos simples e vagos (tal como se encontra presente nos diagramas de blocos internos criados).

Tais expressões não poderão ser representadas directamente nos meta-modelos analíticos, razão pelo qual este aspecto ainda se encontra em investigação [4] e deve ser tido em conta quando se pretende ligar um modelo de domínio (tal como o

apresentado neste caso prático) a simulações que envolvam meta-modelos, utilizados para definir os aspectos necessários para a simulação de um sistema em particular.

No capítulo da existência de meta-modelos para domínios específicos, tal não se verificou neste caso prático. Apesar de existirem alguns modelos pré-definidos (servindo apenas de exemplo e obrigando à sua completa reestruturação), foi necessário recorrer ao modelo padrão, definido pela especificação e implementado pela ferramenta (a qual cria um modelo UML, incluindo posteriormente a notação SysML, bem como as suas alterações e restrições) e que disponibiliza elementos padrão (entre eles, diagramas convencionados na especificação, blocos simples representando as medidas do SI, portas de fluxo vulgares, entre outros). A partir deste mesmo modelo, foi delineado e criado todo o desenho do sistema, bem como os respectivos diagramas existentes.

Desta forma, a utilização da notação em áreas relativamente extensas e complexas (por exemplo, sistemas que impliquem conhecimentos de áreas tão diversas como Física, Química, Mecânica, entre outras e que para isso necessitem de elementos específicos, próprios e inexistentes na especificação geral da notação) será bastante afectada, visto que não existem elementos de representação próprios para modelar determinados aspectos ou comportamentos internos [8]. Com isto, torna-se necessário recorrer aos elementos de extensão (estereótipos e perfis) para que se consiga adicionar novos elementos, o que nem sempre se apresenta como uma tarefa intuitiva e fácil de conseguir, especialmente com utilizadores com pouca experiência na notação.

Finalmente, embora exista um perfil adequado à metodologia SYSMOD (apresentando novos elementos de modelação), os modelos criados continuam a ser relativamente vagos, visto que não são específicos para uma determinada área ou problema. Limitando ainda mais a sua aplicação, este perfil encontra-se disponível apenas para uma ferramenta de modelação (Magic Draw), pelo que a sua aplicabilidade é diminuta.

5.14.1.2. Principais dificuldades

Partindo-se de uma plataforma já existente, todos os seus aspectos reais teriam que ser modelados virtualmente, implicando um conhecimento relativamente profundo do seu funcionamento, bem como de todos os elementos envolvidos nos seus vários procedimentos.

Avaliando o desenvolvimento deste caso prático, a principal dificuldade encontrada passou por este facto. A modelação de um sistema já existente apresentou maiores problemas do que as inicialmente esperadas, levando a diversas reuniões com a responsável, além de posteriores trocas de emails, para clarificação do sistema.

Caso se tivesse partido de zero, tais dificuldades dificilmente seriam encontradas, embora novas adversidades provavelmente surgissem ao longo da modelação. A identificação do conjunto de requisitos que deveriam ser seguidos ao longo da modelação também permitiu que este processo se desenrolasse com alguma celeridade e preocupação para que estes fossem devidamente cumpridos, tal como se veio a verificar ao longo da modelação.

De referir que os diagramas utilizados para apresentar os desenvolvimentos do caso prático à responsável possibilitaram uma melhor comunicação com a mesma. A representação gráfica dos elementos presentes no sistema a modelar levou a uma rápida compreensão dos problemas encontrados durante a modelação, resultando na rápida resolução dos mesmos, além de diversas trocas de ideias sobre potenciais alterações a efectuar ao sistema.

O conhecimento do sistema a modelar, partindo do zero ou de um sistema já existente, deve ser tido em conta, para que seja possível chegar a uma solução credível e utilizável em ambiente real. Por esta ordem de ideias, qualquer alteração efectuada ao modelo irá influenciar por consequência o sistema real, caso o sistema afectado tenha sido modelado e avaliado com recurso à notação SysML (cenário expectável, com a crescente utilização da notação na criação de sistemas reais e capacidade da mesma para apresentar todo um sistema).

Desta forma, todo o processo de remodelação de um sistema deve ser acompanhado de um processo de estudo prévio, no qual se poderá obter conhecimento sobre os sistemas em causa, bem como quais as suas possibilidades de adaptação.

5.14.1.3. Áreas de aplicação

Outro aspecto que se encontrava em destaque durante a realização deste caso prático, seria a aplicação de áreas tão distintas como a Mecânica e a Química num mesmo modelo. Visto que o caso prático envolvia um sistema que continha tais elementos, a sua modelação seria um aspecto bastante interessante a considerar. Tal como se apresentou (embora os elementos modelados sejam relativamente simples, visto que os conhecimentos das duas áreas são obviamente limitados), é possível

demonstrar que a modelação de áreas tão distintas é credível, sem que tal implique que as partes contrárias fiquem alheias. Com recurso aos diagramas apresentados (diagrama de blocos internos e actividade sobretudo), é possível apresentar os fluxos de funcionamento dos diversos componentes, sem que tal se torne incompreensível para quem não possua quaisquer conhecimentos das áreas.

Um exemplo prático deste aspecto foi verificado durante a fase de análise de requisitos, a quando da necessidade de validar os mesmos junto da responsável pelo sistema. Para tal, recorreu-se aos diagramas já criados para apresentação do sistema, bem como para validação da sua utilidade em utilizadores sem conhecimento da notação. Embora tenha necessitado de algumas explicações iniciais, a responsável foi capaz de interpretar perfeitamente os diagramas apresentados e o seu funcionamento, sem necessitar de qualquer estudo da notação. Esta conclusão, além das anteriores, permite confirmar um dos principais objectivos (senão o principal) desta notação: tornar possível a interligação das diversas engenharias, envolvidas num projecto multi-disciplinar, sem falhas de comunicação ou interpretação entre elas.

5.14.2. Aplicações da Notação e Metodologias

A utilização da notação referida neste caso prático, bem como das metodologias adequadas à mesma, permitiu uma análise mais detalhada da aplicabilidade do SysML, SYSMOD e Investigação em Acção num caso real. Esses dados são assim uma fonte de informação bastante importante, visto que apresentam todos os aspectos envolvidos na sua aplicação prática no terreno, sendo por isso uma base comparativa para futuras utilizações da notação.

5.14.2.1. Sobre a notação SysML

Na modelação referida neste caso prático, pôde-se constatar que, apesar de a linguagem ser relativamente fácil de utilizar, a sua correcta utilização nem sempre é verdade. Tal como se referiu, basta que o entendimento do sistema seja incorrecto, para que a modelação efectuada também seja ela incorrecta. No entanto, tal não advém da dificuldade de aplicação da notação num caso real, mas sim da falta de compreensão da sua utilização. Para que tal fosse possível, deveria ser apresentada juntamente com a especificação da notação, uma metodologia de aplicação da mesma, colmatando desta forma eventuais falhas de interpretação da linguagem.

5.14.2.2. Utilização do SYSMOD

Uma vez que a especificação da notação não define qualquer tipo de metodologia de utilização, recorreu-se à metodologia SYSMOD, a qual apesar de recorrer à notação SysML, poderia ser facilmente utilizada noutra notação de modelação de sistemas de sistemas, visto que é totalmente independente. Sendo uma solução aceitável, não é a mais eficaz, visto que implica que os eventuais utilizadores da notação tenham que recorrer a literatura que possa não estar directamente ligada com a notação, dificultando a sua aprendizagem.

A metodologia SYSMOD, apresentada em [21], abrange um leque variado de pontos relacionados com a modelação de um sistema de sistemas. Através da apresentação de um exemplo prático na literatura referida, é possível observar a utilização da notação SysML num caso real. Mesmo sendo um caso bastante extenso e interessante, a existência de mais exemplos de utilização da metodologia seria bastante vantajosa (quer através desta literatura ou de outras), para o esclarecimento de eventuais dúvidas que surgissem durante a utilização da mesma.

De referir que alguns dos pontos mais específicos da metodologia SYSMOD foram ignorados ou adaptados para a utilização neste caso prático, devido à sua falta de adequação e enquadramento relativamente ao caso de exemplo apresentado na metodologia (modelação de um sistema de sistemas para uma rent-a-car). Sendo o exemplo de aplicação da metodologia apresentado por [21] bastante detalhado e complexo, não é expectável que todos os pontos apresentados por este sejam seguidos nos projectos de modelação, pelo que a aplicação e resultado verificados neste caso prático deverá ser representativa da utilização da metodologia nos restantes projectos.

5.14.2.3. Interesse da Investigação em Acção

Com a metodologia SYSMOD a apresentar-se como a metodologia de concepção do sistema a utilizar, a Investigação em Acção foi o método de investigação escolhido para se obter um maior rigor na investigação efectuada ao longo do caso prático.

Utilizada nas diversas fases do caso prático, a Investigação em Acção levou a uma maior auto disciplina no planeamento e execução, bem como a uma avaliação de todos os resultados obtidos. Tal levou a que nas diversas situações em que foi necessário alterar o modelo criado, se pudesse obter um maior conhecimento do

sistema, bem como de novas soluções que pudessem ser utilizadas para problemas que surgissem no futuro.

A sua utilização representou um aspecto positivo na modelação, contribuindo para um melhoramento constante de todos os aspectos do modelo e da sua representação final. No entanto, visto que se trata de um ciclo contínuo de aprendizagem, esse mesmo ciclo deve ter um fim, para que a conclusão do projecto seja possível.

Desta forma, a aplicação da Investigação em Acção é aconselhável a todos os projectos de investigação, visto que a sua aplicação levará a um maior detalhe e autocrítica nos mesmos, culminando num melhor produto final, com o devido aperfeiçoamento.

5.14.2.4. Simulação

Relativamente à componente de simulação, através da literatura existente sobre a notação SysML, chegou-se à conclusão de que embora a notação forneça as ferramentas necessárias para se poder utilizar os seus modelos em ambientes de simulação, não existe qualquer experiência sobre a execução destes modelos nos referidos ambientes [21].

Embora a ferramenta (apenas na sua licença comercial) permita algumas simulações básicas (validação do comportamento do sistema, recorrendo à integração entre o Artisan Studio e o Altia Design), a sua aplicação prática é bastante complexa e permite obter informação relativamente reduzida sobre o funcionamento do sistema, recorrendo para isso aos diagramas de texto (para descrição do sistema), contexto (para definição de entradas e saídas), casos de uso (para apresentação de funcionalidades) e sequência (para especificar a interacção entre os elementos).

No entanto, para se obter resultados mais precisos, é necessário recorrer a outras ferramentas que disponibilizem simulações de modelos (quer através de simples simulações, quer através de ligações com outras ferramentas), tais como o Telelogic Rhapsody e o Artisan Studio (na sua licença Simulation). A ferramenta utilizada na modelação deste caso prática (Artisan Studio Uno), visto ser uma licença gratuita, não disponibiliza as capacidades de simulação das suas congéneres comerciais. Ainda no campo da simulação, para a realização de testes é necessário recorrer aos IDE mais comuns que utilizem C ou C++ (neste caso o Visual Studio), integrando os modelos na sua composição e injectando dados durante o seu funcionamento.

No que concerne o modelo utilizado, para que a sua aplicação em ambientes de simulação seja possível, poderão ser necessárias alterações aos diagramas, para que sejam incluídos todos os pormenores relativos ao funcionamento interno do sistema (por exemplo, quantificar a influência das bombas no transporte de fluido, o que levaria a aplicação de cálculos matemáticos precisos para tal), visto que se optou por um funcionamento mais simples e directo, mas menos preciso.

5.14.3. Consistência do modelo

Um aspecto que poderá apresentar problemas para os menos conhecedores do SysML será a relação entre os diagramas utilizados. Apesar dos diagramas apresentados serem vistos como elementos unitários de um sistema completo, estes encontram-se interligados entre si, quer ao nível da especificação, quer ao nível das ferramentas.

5.14.3.1. Especificação da notação

No que concerne a especificação da notação, um exemplo desta garantia de consistência é apresentado pelos diagramas de actividade e os diagramas de blocos [17]. Embora as actividades sejam especificadas independentemente da estrutura dos sistemas, a relação entre os comportamentos do sistema (modelados através dos diagramas de actividades) e a estrutura desse mesmo sistema (modelada utilizando diagramas de definição de blocos) necessita de ser estabelecida.

Para tal, existem duas alternativas: recorre-se à utilização de partição de actividades ou atribui-se a responsabilidade de um bloco a uma actividade. Aplicando ao caso prático, recorreu-se à aplicação da segunda alternativa, visto que se apresentava como mais simples e directa.

Utilizando portas de fluxo, especificou-se as actividades relativas aos blocos apresentados, indicando quais os pontos de comunicação relativos a cada um deles. Um exemplo desta utilização trata-se do diagrama de blocos internos relativo ao sistema de experimentação de transferência de calor, onde se pode visualizar as ligações entre os vários blocos existentes no sistema (Tanque 1, Tanque 2 e Sistema de Aquisição de Dados).

Além disso, para cada um dos blocos foram definidos um conjunto de operações possíveis, através das quais se especifica quais as actividades possíveis para cada um deles (início e término de registo de dados para os tanques e sistema de aquisição de dados, verificação dos mesmos, entre outras operações).

5.14.3.2. Relações na ferramenta

No que concerne à utilização da ferramenta para especificação de relações entre diagramas, esta permite que a partir da modelação de um determinado diagrama, seja possível definir os diagramas adjacentes a este, de modo a criar um todo relativo a um determinado elemento do sistema.

Relativamente à ferramenta utilizada neste caso prático (Artisan Studio Uno), nos diagramas de definição de blocos, é possível criar o diagrama de blocos internos e o diagrama paramétrico relativo a um determinado bloco, logo após a criação do mesmo. Após a criação dos respectivos diagramas, ao bloco inicial são associados os diagramas criados (diagrama de blocos internos e paramétrico), definindo-se assim a sua estrutura. Além disso, é possível associar também diagramas de máquinas de estado, casos de uso e actividades, completando a definição do bloco.

A integridade é garantida pela ferramenta, levando a que alteração de um determinado aspecto do bloco seja reflectida nos restantes diagramas.

No que concerne os requisitos, a interligação com os restantes diagramas apresenta-se como essencial, visto que a sua definição irá determinar toda a elaboração e execução do sistema modelado. Desta feita, essa interligação também é assegurada pela ferramenta, que permite a utilização dos blocos definidos anteriormente nos diagramas de requisitos, garantindo a consistência entre os elementos definidos no bloco e os respectivos requisitos.

Além disso, a utilização da notação tabela nos diagramas de requisitos permite representar os requisitos, bem como as suas relações internas. Em conjunto com esta tabela, pode-se utilizar a notação de chamada (callout notation), semelhante a uma nota de chamada e que permite representar a relação entre um determinado requisito e um elemento do modelo, através de texto. Estas duas alternativas de representação de relações entre diagramas e elementos são apresentadas pela especificação, não sendo dependentes da ferramenta utilizada na modelação.

Destas três alternativas (ferramenta, notações de tabela e chamada), utilizadas para garantir a interligação entre os requisitos e os restantes diagramas, deu-se preferência às duas primeiras, não se utilizando a chamada.

Finalmente, no que concerne os diagramas de pacotes, estes permitem a definição de blocos secundários, nos quais se encontram agrupados os diagramas especificados. Tais pacotes são posteriormente agrupados num pacote geral, apresentado como uma determinada visão do sistema (cliente, por exemplo). Podem, no entanto, haver variadas visões do sistema, podendo assim existir determinados

pacotes secundários que possam estar excluídos de determinadas visões. Todas estas relações são apresentadas pela especificação do SysML, sendo resguardadas pela ferramenta de modelação.

5.14.3.3. Adaptabilidade dos modelos

No que concerne a adaptabilidade dos modelos criados a outros problemas apresentados, a alteração dos modelos apresenta-se como uma tarefa algo extensa, mas que não deverá apresentar problemas de integridade, devido às interligações entre diagramas.

Por exemplo, durante a modelação do Sistema de Experimentação de Transferência de Calor, foi necessário reformular os diagramas de requisitos em duas ocasiões, devido a falhas de interpretação dos requisitos apresentados pela responsável. Embora os requisitos representem a parte inicial da modelação do caso prático, era necessário obter essa base para a elaboração dos restantes diagramas. No entanto, chegou-se à conclusão que a interpretação inicial tinha sido incorrecta, pelo que todos os diagramas criados teriam de ser reformulados. Desta forma, diagramas como os de actividade, casos de uso, definição de blocos e blocos internos tiveram de ser substancialmente alterados através da remoção de blocos e sequências, além da reformulação de actividades e interacções possíveis.

Apesar de ter sido uma tarefa morosa, não se apresentou relativamente difícil, visto que as alterações efectuadas num determinado diagrama reflectiam-se nos restantes diagramas, sem necessidade de intervenção do utilizador. Além disso, rapidamente se adaptou o sistema a uma nova realidade, sem que fosse necessário partir do zero, tal como poderia ser o cenário noutras notações existentes.

Desta feita, partindo desta experiência pessoal, é possível obter determinados conclusões:

- Os modelos criados em SysML podem facilmente ser reaproveitados e utilizados noutras circunstâncias, sem grandes dificuldades.
- As alterações efectuadas num diagrama terão consequências nos restantes diagramas, levando a um menor esforço envolvido nesta operação.
- A reestruturação completa de um modelo é possível, sem que tal implique recomeçar do zero e envolva grandes recursos.

É possível assim concluir que a adaptabilidade dos modelos é credível, embora eventuais alterações extremas de um modelo para outro possam tornar inviáveis estas

operações. No entanto, mesma para alterações relativamente extensas, é possível reutilizar os modelos criados em SysML ou mesmo noutra notação (desde que devidamente remodelados, de acordo com a especificação da notação SysML).

5.15. Conclusão

No âmbito geral, a modelação deste sistema apresentou diversos desafios que tiveram que ser ultrapassados, concretizando os conhecimentos teóricos apresentados inicialmente na Tese.

Relativamente à eventual complexidade da notação, de referir que apesar de se ter alguma experiência de modelação em UML, a transição para o SysML apresentou algumas dificuldades, provenientes sobretudo das novidades apresentadas pela notação. Desta feita, a utilização de novos diagramas (diagrama de requisitos e blocos internos), bem como as alterações efectuadas nos restantes, levaram a que o processo de aprendizagem não fosse tão linear quanto seria expectável.

No entanto, pode-se concluir que apesar de poder parecer complicada, a notação é facilmente compreendida em pouco tempo. Tal não implica que os utilizadores consigam utilizar as suas funcionalidades mais avançadas num curto espaço de tempo, visto que o conhecimento da notação advém sobretudo da sua aplicação prática (como comprovado durante este caso prático). Apesar disso, mesmo com conhecimentos relativamente limitados e recorrendo aos elementos básicos disponibilizados, será possível a aplicação da notação nos diversos casos apresentados.

Após este caso prático, pode-se afirmar que a notação é passível de ser utilizada nas diversas áreas da engenharia utilizadas nos projectos, desde a Mecânica até à Informática, visto que a sua simplicidade de utilização incentiva à sua aplicação prática nos projectos. No entanto, tal aplicação deve ser acompanhada pela utilização de uma metodologia, de forma a garantir um elevado padrão de rigor ao longo do projecto (tal como se pôde verificar neste caso prático).

Como nota final, de referir que o modelo criado neste caso prático, bem como os seus respectivos diagramas foram devidamente validados pela responsável pelo sistema, com todos os elementos já existentes a serem correctamente reproduzidos no ambiente virtual e os novos aspectos a enquadrarem-se numa potencial actualização do sistema. O resultado foi bastante satisfatório, com a responsável a mostrar-se bastante agradada com as novas potencialidades idealizadas para o seu sistema, que poderá facilmente ser adaptado a novos âmbitos e projectos.

6. Conclusão

Concluída que está a análise teórica da notação SysML, bem como da sua implementação prática num caso real, pode-se concluir sobre a sua eventual utilização na elaboração de sistemas de sistemas. Com diversos pontos a seu favor (entre os quais a sua relativa facilidade de aprendizagem e utilização, além da sua aplicabilidade em áreas tão distintas entre si), a notação apresenta-se como uma alternativa perfeitamente credível às diversas notações próprias utilizadas pelas diversas disciplinas de engenharia.

6.1. Conclusões do Estudo

Apesar de não apresentar todos os elementos para todas as áreas de forma imediata, a notação permite a sua expansão através de mecanismos próprios para esse efeito (estereótipos e perfis), colmatando qualquer eventual falha que possa ter. Tal tarefa necessitará de algum tempo de preparação para aplicar a notação a tal área, mas permite também uma maior personalização da notação e respectivas ferramentas ao ambiente de trabalho em que se irá inserir. A sua aprendizagem também será outro factor a ter em conta, visto que embora a sua utilização seja relativamente intuitiva, necessitará de um período inicial de adaptação (que irá variar em funções dos conhecimentos prévios de outras linguagens de modelação gráfica, tal como o UML) para que seja possível utilizar devidamente a notação e as suas potencialidades.

Juntamente com a representação dos sistemas de sistemas, a notação também pode ser utilizada para efeitos de simulação, embora tal tenha de ser avaliado com as devidas precauções. Com a possibilidade de interligação entre as diversas ferramentas disponíveis (quer relacionadas com a notação, quer com a área de simulação), é possível utilizar os modelos criados com a notação SysML para previsão de eventuais comportamentos de um sistema, em determinadas circunstâncias. No entanto, a falta de documentação sobre o assunto, bem como a falta de exemplos práticos levam a que seja uma área que necessite de maior investigação, para que se possa verificar a sua real utilidade.

Desta forma, pode-se concluir que aplicabilidade da notação SysML no desenvolvimento de projectos de sistemas de sistemas é de facto possível, devendo ser acompanhada com uma metodologia de aplicação sustentada, aspecto não incluído na especificação da notação. Com isto, será possível acompanhar todas as

fases de evolução de um determinado projecto, avaliando-se o seu grau de sucesso e rigor. No caso prático apresentado nesta Tese, recorreu-se à utilização da metodologia SYSMOD, que apesar de se revelar bastante extensa, permitiu uma aplicação concreta e rigorosa da notação numa situação real, com resultados bastante satisfatórios.

De referir que a utilização da Investigação em Acção durante a modelação deste caso prático, permitiu melhorar o conhecimento obtido a partir do sistema, levando a diversas reflexões sobre a sua interpretação e apresentação. Embora a sua utilização neste caso prático tenha implicado um maior aprofundamento da modelação de cada um dos diagramas, a qualidade das soluções obtidas a partir dos problemas encontrados permite afirmar que a sua integração foi um aspecto positivo para o caso prático, devendo ser seguida em futuros projectos.

6.2. Trabalho Futuro

Relativamente ao que poderão ser as próximas fases desta Tese, de destacar as seguintes:

- Aplicação no terreno do modelo criado
- Utilização do modelo para efeitos de simulação

Relativamente à primeira hipótese, a utilização do modelo criado para actualização do sistema criado pela responsável seria o eventual passo seguinte. Com o modelo apresentado a englobar um novo elemento (sistema de experimentação de transferência de calor), o qual seria acoplado ao sistema já criado e que serviria de elemento central para todo o projecto, seria possível avançar para a experimentação remota. A criação de uma aplicação de controlo do sistema através da rede (englobada no novo elemento central) seria um passo importante, aumentando consideravelmente o leque de possibilidades de utilização do sistema criado pela responsável.

Outro aspecto que seria avaliado nesta hipótese seria a efectiva aplicação dos modelos criados através da notação SysML no terreno. Visto que seria necessário analisar o modelo criado, verificar o que teria que ser alterado e criado de raiz para o sistema e adaptá-lo fisicamente a uma nova realidade, as possibilidades de aplicação prática do SysML seriam testadas, com os dados recolhidos a apresentarem um cenário que poderia ser utilizado como exemplo para futuras utilizações da notação.

No que concerne a segunda hipótese, para que fosse possível utilizar o modelo criado para efeitos de simulação, seria necessário adquirir uma licença comercial da ferramenta Artisan Studio (neste caso, a licença Simulation), visto que esta é a única que permite a utilização de tais capacidades nos modelos.

Caso não fosse possível recorrer a esta ferramenta, seria necessária a utilização de outra que permitisse a experimentação do modelo em condições de simulação, com o Rational Rose a apresentar-se como a alternativa mais viável (a ferramenta Artisan Studio apresenta possibilidades de exportação dos modelos para essa ferramenta). Além disso, poderiam ser necessárias adições ao sistema, para apresentação de um maior detalhe em determinadas áreas (influência das bombas nos fluidos, por exemplo). No entanto, o modelo apresentado apresenta um grau de detalhe que permite averiguar o funcionamento do sistema, sem necessidade dessas mesmas adições.

Estas duas hipóteses apresentam-se como os cenários mais prováveis para aprofundar o estudo da notação SysML, quer a nível teórico, como a nível prático. Esse estudo irá complementar o trabalho realizado ao longo desta Tese, fornecendo novos dados que poderão ser utilizados para eventuais revisões das avaliações realizadas até então.

Referências

- [1] A. T. Bahill and B. Gissing. Re-evaluating Systems Engineering Concepts Using Systems Thinking. IEEE Transaction on Systems, Man and Cybernetics, Part C: Application and Reviews, 1998, 28(4): 516-527.
- [2] Alessandra Cavarra. UML Testing Profile – An Overview. Motivations and Objectives, <http://www.agedis.de/documents/TheUMLTestingProfile.pdf>, 2008.
- [3] Bond Graph Info. <http://www.bondgraph.info/>, 2005.
- [4] E. Huang, R. Ramamurthy, L.F. McGinnis. System and Simulation Modeling Using SysML, Winter Simulation Conference, 2007, 796-803.
- [5] Georg Pelz. Mechatronic Systems. Chapter 2: Principles of Modelling and Simulation, 2003, 5-37.
- [6] INCOSE. <http://www.incose.org/practice/whatisystemseng.aspx>, 2008.
- [7] JMCAD Specification. <http://jmcad.sourceforge.net/>, 2009.
- [8] M. Fonoage, I. Cardei. The Systems Modeling Language, 8 Março, 2007, Center for Systems Integration, Florida Atlantic University.
- [9] M. Grecki, Z. Geng, G. Ayvazyan, S. Simrock, H. Bahtiar. Application of SysML to design of ATCA based LLRF Control System, IEEE Nuclear Science Symposium Conference Record, 2008.
- [10] M. Soares, J. Vrancken. Requirements Specification and Modeling through SysML. IEEE International Conference on Systems, Man and Cybernetics, 2007, 1735-1740.
- [11] Mauro Nunes e Henrique O'Neill. Fundamental de UML. Capítulo 7: Desenho de Sistemas, 2004, 108
- [12] Modelica Specification. <http://www.modelica.org/documents/>, 2007.
- [13] NASA. Systems Engineering Handbook. Chapter 5: Systems Analysis and Modeling Issues, 1995, 81-103.
- [14] OMG MARTE Specification. <http://www.omgarte.org/Specification>, 2008.
- [15] OMG SysML Specification. <http://www.omgsysml.org/#Specification>, 2007.
- [16] OMG UML Superstructure Specification. <http://www.uml.org/>, 2009.
- [17] S. Friedenthal, A. Moore, R. Steiner. A Practical Guide to SysML. 2007. 3-515.
- [18] S. Popper, S. Bankes, R. Callaway, D. DeLaurentis. System-of-Systems Symposium: Report on a Summer Conversation, 21-22 Julho, 2004, Potomac Institute for Policy Studies, Arlington, VA.
- [19] T. Brannick, D. Coghlan. Doing Action Research In Your Own Organization. 2005. 3-134.
- [20] T. Grotker, S. Liao, G. Martin, S. Swan. System Design with SystemC. Chapter 1: Introduction, 2002, 1-9.
- [21] Tim Weilkiens. Systems Engineering with SysML/UML, 2006, 1-270.
- [22] Y. Grobshtein, V. Perelman, E. Safra, D. Dori. Systems Modeling Languages: OPM Versus SysML, International Conference on Systems Engineering and Modeling, 2007.

Anexo A

1. Introdução

Este anexo apresenta os diagramas existentes no SysML que não sofreram qualquer alteração relativamente ao UML. Encontra-se organizado pela seguinte ordem:

- Diagrama de Sequência
- Diagrama de Casos de Uso
- Diagrama de Máquinas de Estado

É recomendável que leia a Secção 3 para uma apresentação da notação e dos seus novos diagramas, antes de ler este anexo.

2. Diagrama de Sequência (Sequence Diagram)

Ao contrário do UML, que incluía quatro diagramas de interacção (onde se incluía o Diagrama de Sequência), o SysML apresenta apenas um único diagrama de interacção: o Diagrama de Sequência.

Não sofreu qualquer alteração relativamente ao UML, podendo ser utilizado para descrever quais as interacções possíveis num sistema, através de um determinado fluxo.

3. Diagrama de Casos de Uso (Use Case Diagram)

O Diagrama de Caso de Uso descreve os serviços fornecidos pelo sistema, bem como os utilizadores que utilizarão esses serviços. Os elementos presentes no Diagrama de Caso de Uso são definidos vagamente, dando uma grande liberdade de utilização.

O Diagrama de Caso de Uso é relativamente simples, sendo utilizado como interface de comunicação entre os engenheiros de sistemas e os futuros utilizadores do sistema (sem conhecimentos técnicos).

O Diagrama de Caso de Uso presente no SysML, tal como o existente no UML, é independente da sua execução no sistema, não contendo informação sobre a implementação física ou programática. E tal como era o caso no UML, a aplicabilidade dos Diagramas de Casos de Uso são variadas, especialmente na modelação de sistemas e a sua integração no SysML não sofreu qualquer alteração relativamente ao UML.

3.1. Caso de Uso

Descreve um serviço fornecido pelo sistema para o seu ambiente, sendo que os utilizadores deste serviço são os actores. Um caso de uso é accionado pelo actor, sendo o resultado dessa acção, uma alteração significativa para o referido actor [21].

Além deste tipo de actores, poderão haver outros actores que são requisitados pelo caso de uso, para que seja possível fornecer um serviço.

Relativamente à notação gráfica utilizada, o caso de uso é definido por uma elipse, sendo o nome deste escrito no seu interior. Os actores participantes são ligados ao caso de uso utilizando uma associação (representada por uma linha), podendo haver vários actores ligados a um único caso de uso. Um caso de uso poderá estar inserido num sistema, definido por um rectângulo.

Os casos de uso poderão ser concretos ou abstractos, sendo que no último caso, serão necessários casos de uso concretos, para definir os serviços disponibilizados pelo caso de uso inicial. Um caso de uso indica apenas que o serviço está presente, sendo a sua descrição realizada posteriormente (através de uma actividade, por exemplo).

3.2. Actor

Representa os utilizadores do sistema, podendo esse papel ser assumido por um indivíduo ou um sistema externo. Um actor encontra-se obrigatoriamente fora do sistema, podendo interagir ou ser requisitado pelo mesmo sistema [21].

Relativamente à notação gráfica utilizada, o actor é representado por um boneco humano, podendo esta representação ser alterada. A ligação entre o actor e o caso de uso em que participa é feita através de uma associação (linha).

Relativamente às ligações entre actores, visto que estão fora do âmbito da modelação do sistema, não existe a necessidade de especificá-las em detalhe. Um tipo de relação que é permitida entre os actores é a generalização, a qual estabelece uma hierarquia entre os actores. Outro tipo de relação permitido é o fluxo de informação, que permite descrever a informação que pode ser transmitida entre os actores. Ao contrário destas relações, as associações entre actores não são permitidas.

3.3. Relação Incluir (Include Relationship)

Descreve que um caso de uso encontra-se incluído noutra caso de uso. Esta relação é utilizada para a criação de fluxos, utilizados em vários casos de uso.

Relativamente à notação gráfica utilizada, a Relação Incluir apresenta uma seta com o estereótipo include, a qual aponta para o caso de uso que será incluído.

3.4. Relação de Extensão (Extend Relationship)

Permite estabelecer uma ligação entre um determinado caso de uso base e o respectivo caso de uso de extensão, através da indicação de qual parte será inserida no caso de uso, através da utilização de pontos de extensão. Podem ser utilizadas para diferenciação de comportamentos dos sistemas, apresentar sub-fluxos opcionais e distinguir possíveis segmentos de comportamento em função das interações dos utilizadores. As extensões são condicionais, dependendo das acções executadas no caso de uso base.

No que concerne a notação gráfica, a Relação de Extensão é apresentada por intermédio de uma seta, com o estereótipo extend, indicando qual o caso de uso utilizado como extensão.

3.5. Generalização (Generalization)

Utilizada quando existem dois ou mais casos de uso com aspectos em comum (tais como comportamento, estrutura e finalidade), permite a descrição das partes iguais num único caso de uso geral, posteriormente especializado através de casos de uso específicos. Embora o caso de uso geral possa ser abstracto, tal não é obrigatório. No que concerne os casos de uso de especificação, herdaram a estrutura, o comportamento e os relacionamentos do pai, introduzindo os seus detalhes individuais.

Quanto à notação gráfica, os casos de uso gerais e específicos são apresentados como casos de uso normais, os quais são interligados por intermédio de uma seta, sem qualquer estereótipo.

4. Diagrama de Máquina de Estado (State Machine Diagram)

O Diagrama de Máquina de Estados utilizado no SysML não sofreu qualquer alteração relativamente ao UML, sendo que as máquinas de estado e os seus diagramas são independentes do tipo de sistema modelado.

Não contém qualquer elemento específico da engenharia de software, pelo que a representação de um sistema de qualquer área poderá ser realizada com recurso a estados, transições entre estados, descritos através de um Diagrama de Máquina de Estados.



isep

