

UTILIZAÇÃO DE MDA INTEGRADO AO PROCESSO UNIFICADO NA MODELAGEM DE UM SISTEMA DE GERENCIAMENTO DE COMUNICAÇÃO PARA VANTS

Christiane Barbieri De Pelegrin *

Resumo

Este artigo expõe a modelagem de um sistema de gerenciamento de comunicação para VANTs por meio do uso da Arquitetura Dirigida por Modelo Model Driven Development (MDA), um paradigma de desenvolvimento dirigido por modelo. O principal objetivo é demonstrar como o Processo Unificado foi incluído durante a modelagem MDA. Foram inseridas, durante as etapas de desenvolvimento, fases do Processo Unificado e seus artefatos. Incluiu-se, também, diagramas da SysML em substituição a alguns artefatos da UML, utilizada no Processo Unificado, com a finalidade de tornar a modelagem mais eficiente e precisa.

Palavras-chave: Arquitetura dirigida por modelo. MDA. Processo unificado. SysML.

1 INTRODUÇÃO

As aplicações dos Veículos Aéreos Não Tripulados (VANTs) no setor civil estão se desenvolvendo de modo a possibilitar a sua utilização comercial e industrial. O crescimento no desenvolvimento de aplicações civis ocorreu, principalmente, em razão das facilidades que esse tipo de aeronave proporciona em vários cenários, especialmente naqueles em que o trabalho pode ser perigoso e no qual pode haver risco de contaminação. Por esses e diversos outros fatores, os VANTs tem sido objeto de pesquisa em várias empresas e universidades do Brasil e em outros países.

A comunicação é um elemento importante na utilização de aeronaves não tripuladas, principalmente nos tipos de aeronaves que não são totalmente autônomas, necessitando ser controladas remotamente. Um sistema de comunicação é a rede por onde fluem as informações enviadas através de um ponto (emissor) a outro (receptor) por um meio (canal), cuja responsabilidade é realizar a troca de informações entre emissor e receptor de forma correta e eficaz.

Fazer a modelagem do sistema de gerenciamento de comunicação para VANTs implica a captura das informações que são relevantes e a organização delas. Para tal tarefa é necessário que sejam adotados métodos e procedimentos que possam satisfazer todas as necessidades do projeto; não apenas o *software* está envolvido, mas, também, o *hardware* e uma variedade de áreas de conhecimento.

A análise do sistema foi guiada pelo paradigma de desenvolvimento de software denominado Desenvolvimento Dirigido por Modelo Model Driven Development (MDD), em que o artefato principal do processo é o modelo. Para a utilização de tal paradigma foi utilizada uma de suas especificações, a Arquitetura Dirigida por Modelo Model Driven Architecture (MDA) desenvolvida pela Object Management Group (OMG), uma organização sem fins lucrativos e um órgão de padronização internacional, responsável por desenvolver diversos padrões para uma ampla gama de tecnologias. Este modelo foi utilizado para satisfazer as necessidades do projeto (software) e porque foi identificado como um modelo eficaz, que traz diversos benefícios para quem faz uso dele. Durante o processo

* Graduada em Bacharelado Engenharia de Computação pela Universidade do Oeste de Santa Catarina; Programadora de Softwares na Universidade do Oeste de Santa Catarina; Professora temporária no Instituto Federal Catarinense de Luzerna; christiane.bdp@gmail.com

de desenvolvimento MDA, foi introduzido e trabalhado em conjunto com o Processo Unificado, que é um processo para a modelagem de software.

Na seção 2 são apresentadas as etapas e as características da MDA. Em seguida, na seção 3, é descrito o Processo Unificado. Na seção 4, é descrita a linguagem de modelagem SysML e alguns dos seus principais diagramas. O desenvolvimento da modelagem, no qual o Processo Unificado é integrado às fases MDA, está descrito na seção 5. A seção 6 apresenta a conclusão e as perspectivas para trabalhos futuros. E, por último, a seção 7 apresenta as referências.

2 MDA – MODEL DRIVEN ARCHITECTURE

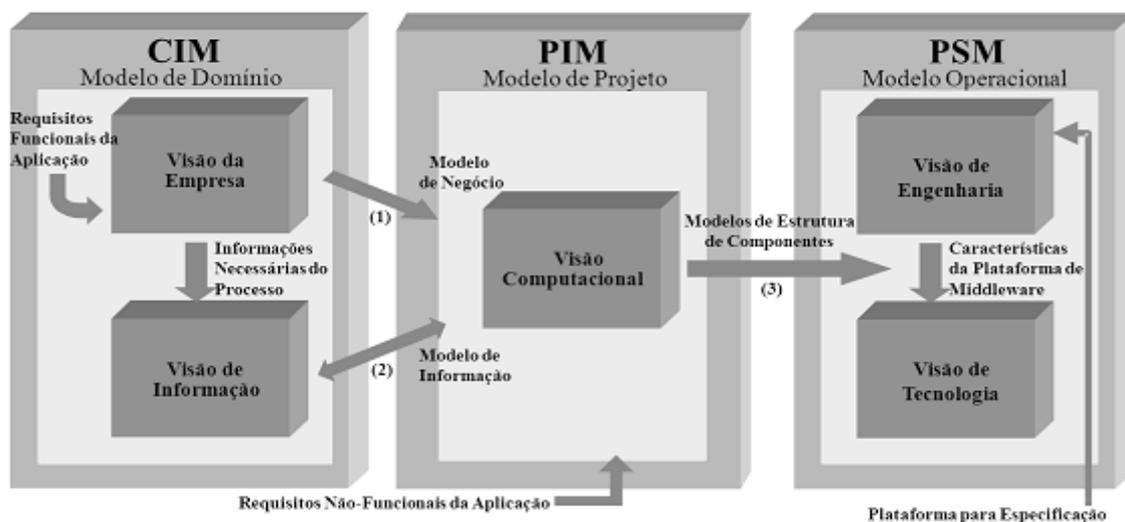
Desenvolvimento Dirigido a Modelo Model Driven Development (MDD) é um paradigma de desenvolvimento que faz uso de modelos como artefatos principais do processo de desenvolvimento de *softwares*. A OMG tem uma visão particular desse modo de elaborar sistemas e concretizou-a na Arquitetura Dirigida por Modelo (MDA), com três objetivos principais: portabilidade, interoperabilidade e reutilização por meio da separação arquitetural de interesses (*concerns*). Como na área de tecnologia não existe um padrão de plataformas de *software*, *hardware* e linguagens de programação, mas uma grande diversidade desses e outros artefatos, a MDA apresenta-se como uma solução para separar as especificações das funcionalidades de um sistema dos seus detalhes de implementação, executados em uma plataforma específica. Dessa forma, viabiliza-se a reutilização e a portabilidade de um projeto (CUNHA; LUCENA, 2004).

Para construir uma aplicação MDA deve-se seguir algumas etapas, pois cada uma trabalha em um nível de abstração diferente. Este nível de abstração é muito importante para a formulação de cada etapa do modelo, permitindo ao desenvolvedor trabalhar com sistemas mais complexos dispondo de menos esforço. A Figura 1 demonstra a visão geral do processo MDA em suas etapas.

A primeira etapa do desenvolvimento é denominada Computation Independent Model (CIM) ou Modelo Computacional Independente. O CIM não mostra detalhes da estrutura dos sistemas e constitui o Modelo de Domínio (OBJECT MANAGEMENT GROUP, 2003). Este modelo corresponde a visões mais abstratas, pois os requisitos funcionais, inicialmente estabelecidos, são utilizados, nesse momento, para delimitar o escopo da aplicação e investigar o vocabulário, as informações e as responsabilidades envolvidas (PELLEGRINI et al., 2010). Nessa etapa, estão incluídas a Visão da Empresa e a Visão da Informação.

Supõe-se que o usuário principal do CIM, o praticante de domínio, não tem conhecimento sobre os modelos ou artefatos utilizados para realizar a funcionalidade para a qual os requisitos são articulados no CIM.

Diagrama 1 – Visão geral do processo MDA



Fonte: Pellegrini et al. (2010).

A próxima etapa é a Platform Independent Model (PIM) ou Modelo Independente de Plataforma. O PIM apresenta um alto grau de abstração, pois há a independência de qualquer tecnologia de implementação. Nessa etapa, o sistema é modelado a partir do ponto de vista de como ele suporta melhor o negócio. Nesse momento devem ser definidos os serviços a serem oferecidos e suas operações, dessa forma, devem ser identificados a interface de serviços e os componentes estabelecidos no modelo de domínio, independente de uma plataforma de *middleware*¹ (PELLEGRINI et al., 2010).

A etapa seguinte é a mais complexa do processo de desenvolvimento MDA, é quando ocorre a transformação do modelo em um ou mais modelos de plataforma específica, denominado Platform Specific Model (PSM) ou Modelo Específico de Plataforma. Um PSM combina as especificações do PIM com os detalhes que definem como um sistema utiliza um tipo particular de plataforma ou tecnologia (KLEPPE; WARMER; BAST, 2003).

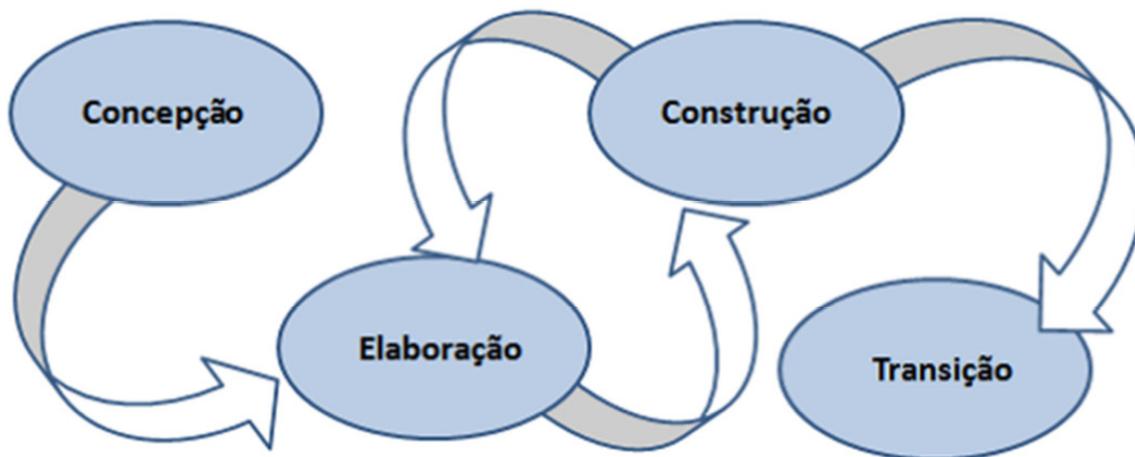
3 PROCESSO UNIFICADO

O Processo Unificado (PU) é um processo de desenvolvimento de sistemas. Propõe que a realização das atividades de análise, projeto, codificação e testes sejam alocadas em quatro fases denominadas Concepção, Elaboração, Construção e Transição (WAZLAWICK, 2011). No Diagrama 2, podem ser vistas as fases do PU e a transição que ocorre entre elas.

A primeira fase do PU é a fase de Concepção. Esta inicia-se com estudos e com a compreensão do projeto. Em seguida, é elaborado o Documento de Requisitos que compreende o levantamento dos principais requisitos, a identificação de casos de uso, de conceitos e de relatórios, bem como um cronograma de desenvolvimento baseado nos casos de uso (WAZLAWICK, 2011).

A segunda fase do PU é a fase de Elaboração que é dividida em duas partes denominadas Análise e Projeto. Nesse momento, é aprofundada a análise de requisitos, por meio da expansão dos casos de uso, e a continuação da investigação ocorre com a modelagem conceitual e com a escrita dos contratos para as operações e consultas de sistema. Em seguida, uma proposta de solução é apresentada por intermédio do projeto de domínio, realizado com o auxílio dos diagramas de colaboração e de classes, ambos da UML, com a aplicação dos *design patterns*.

Diagrama 2 – Fases do Processo Unificado e suas transições



Fonte: Wazlawick (2011).

A terceira fase é a Construção, na qual são feitas as atividades de programação e testes do sistema. A quarta fase é denominada Transição, que consiste em realizar a instalação do sistema e a utilização de dados, ambiente e usuários reais.

Como podem ser observadas no Diagrama 2, as fases de Elaboração e Construção ocorrem de forma iterativa, em ciclos. É durante os ciclos iterativos que acontece a análise detalhada do sistema e são incluídos os padrões de projeto (WAZLAWICK, 2011).

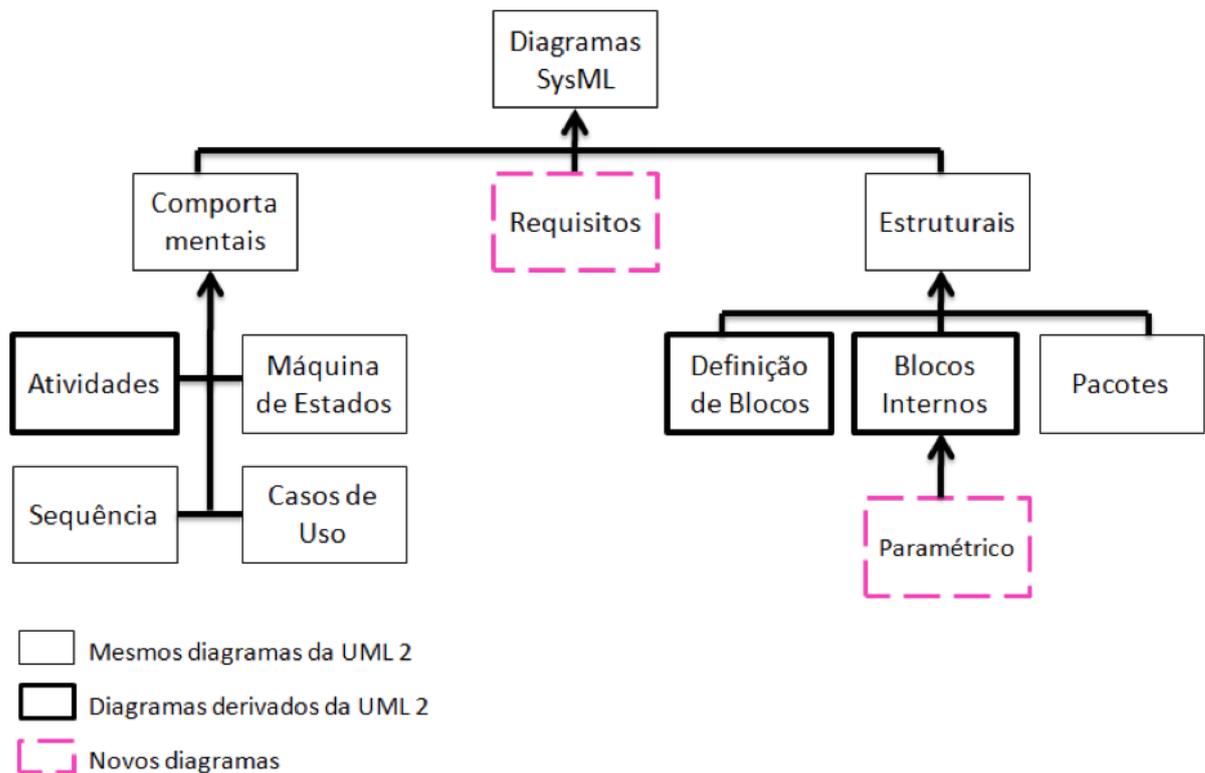
4 SYMML – SYSTEMS MODELING LANGUAGE

A SysML é uma linguagem de modelagem gráfica, de uso geral, para aplicações em engenharia de sistemas. Com ela pode-se fazer a especificação, a análise, o projeto, a verificação e a validação em uma ampla gama de sistemas complexos. Estes sistemas podem incluir *hardware*, *software*, processo, pessoas e informação (OBJECT MANAGEMENT GROUP, 2010).

A iniciativa SysML tem sua origem atribuída a uma decisão tomada pelo International Council on Systems Engineering (IncoSE), em conjunto com o Object Management Group (OMG), órgão que mantém a especificação Unified Modeling Language (UML). Juntos, criaram um grupo de trabalho em janeiro de 2001 com o objetivo de personalizar a UML para aplicações em engenharia de sistemas. Esse grupo foi quem desenvolveu os requisitos dessa nova linguagem. Estes requisitos foram, posteriormente, publicados pela OMG em março de 2003, como parte da UML (UML for Systems Engineering Request for Proposal – UML for SE RFP) (OBJECT MANAGEMENT GROUP, 2010).

A SysML é composta pelo total de nove diagramas que são divididos em três partes denominadas diagrama de comportamento, diagrama de requisito e diagrama de estrutura, como pode ser visualizado no Diagrama 3.

Diagrama 3 – Taxonomia do SysML



Fonte: adaptado de Object Management Group (2010).

O grupo denominado diagrama de comportamento é composto pelos diagramas descritos a seguir, de forma resumida:

- a) Diagrama de atividade: representa o comportamento que ocorre de forma ordenada;

as ações são executadas de acordo com suas entradas, saídas, controles, e representa como as ações transformam as entradas em saídas. É uma modificação do diagrama de atividades da UML.

- b) Diagrama de sequência: representa o comportamento de forma sequencial, em relação às mensagens trocadas entre sistemas, ou entre partes de um sistema. Foi mantido da mesma forma que na UML.
- c) Diagrama de máquina de estado: representa o comportamento de uma entidade em termos de transições entre estados, desencadeada por eventos. Foi mantido da mesma forma que na UML.
- d) Diagrama de caso de uso: representa a funcionalidade em termos de como um sistema é usado por entidades externas (atores) para realizar um conjunto de metas. Foi mantido da mesma forma que na UML.

O Diagrama de requisito é um novo tipo de diagrama utilizado na SysML, ou seja, não existe na UML. Representa requisitos e seus relacionamentos com outros requisitos e elementos da modelagem.

O grupo denominado diagrama de estrutura é composto por dois diagramas da UML modificados, um diagrama que foi mantido como o original na UML e um novo diagrama; são descritos a seguir de forma resumida:

- a) Diagrama de definição de bloco: representa elementos estruturais; sua composição e sua classificação são chamadas de blocos. É uma adaptação do diagrama de classes da UML;
- b) Diagrama de bloco interno: representa as interfaces entre as partes de um bloco e a forma como são ligadas. É uma adaptação do diagrama de estruturas compostas da UML;
- c) Diagrama de pacote: representa a organização de um modelo em forma de pacotes, que contém os elementos do modelo. Foi mantido da mesma forma que na UML;
- d) Diagrama paramétrico: representa restrições nos valores das propriedades. São utilizados para criar sistemas de equações que podem ser usados para limitar as propriedades de blocos. É um novo tipo de diagrama utilizado na SysML.

5 DESENVOLVIMENTO DO PROJETO

O início do desenvolvimento do projeto começa pelo CIM em MDA. Nesse momento, está se inserindo o início da primeira fase do Processo Unificado: a fase de Concepção. O Sistema de Gerenciamento de Comunicação tem como propósito geral enviar e receber dados do VANT. De forma mais específica às funcionalidades que o sistema de comunicação deve compreender, em relação aos dados que devem ser comunicados, são o controle de energia do VANT (que neste projeto é representado pelo nível de bateria, utilizado como combustível e alimentação do sistema embarcado), dados relativos à localização do VANT (oriundos do GPS do VANT) e dados relativos à memória interna que o VANT possui para armazenagem de informações diversas.

O escopo do projeto, ou seja, os limites do projeto podem ser visualizados por meio do diagrama de contexto do Diagrama 4. Este contexto é relatado no PU por intermédio da escrita do Sumário Executivo; já no MDA tal artefato foi substituído pelo diagrama de contexto da SysML.

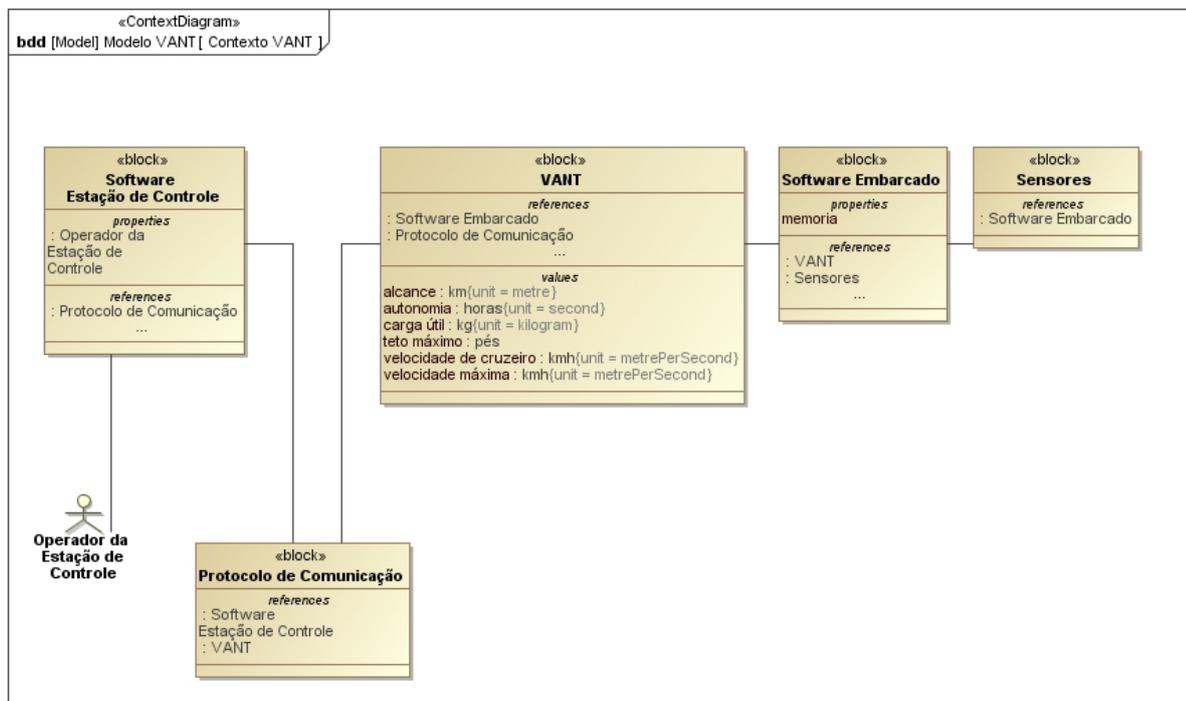
O diagrama de contexto não é um novo diagrama da SysML, mas uma utilização do diagrama de definição de blocos. A SysML também introduz o conceito de utilização do diagrama (OBJECT MANAGEMENT GROUP, 2010). Geralmente, um diagrama de contexto é elaborado por meio do uso do diagrama de definição de blocos, de um diagrama de blocos interno ou do diagrama de casos de uso. O diagrama pode ser identificado, em seu cabeçalho (no canto superior esquerdo), por intermédio da adição de um estereótipo denominado “ContextDiagram”, “diagramUsage” ou qualquer outro que o defina, como pode ser visto no diagrama do Diagrama 4. A aplicação desse estereótipo é importante, pois muitas ferramentas de modelagem são capazes de fazer validações e verificações de restrição nos diagramas e precisam dessa informação para fazer a interpretação adequada (OBJECT MANAGEMENT GROUP, 2010).

A próxima etapa do projeto é fazer o PIM do MDA. Nesta etapa, foi realizado o levantamento de requisitos. No PU os requisitos são organizados em um documento de requisitos que usa a linguagem natural em uma estrutura organizada, ao passo que em SysML os requisitos são representados por meio do diagrama de requisitos (WAZLAWICK, 2011). No Diagrama 5, constam os requisitos que foram selecionados para o Gerenciamento de Comunicação do VANT. Eles foram escolhidos por serem fundamentais para a realização de qualquer outra funcionalidade mais elaborada. Cada requisito possui duas propriedades, um *identificador (Id)* e um texto descritivo do requisito (*Text*).

Na próxima etapa definida pelo PU, os requisitos são organizados por meio da identificação dos casos de uso do sistema, visualmente representados no diagrama de casos de uso, cuja utilização é a mesma tanto na UML quanto na SysML.

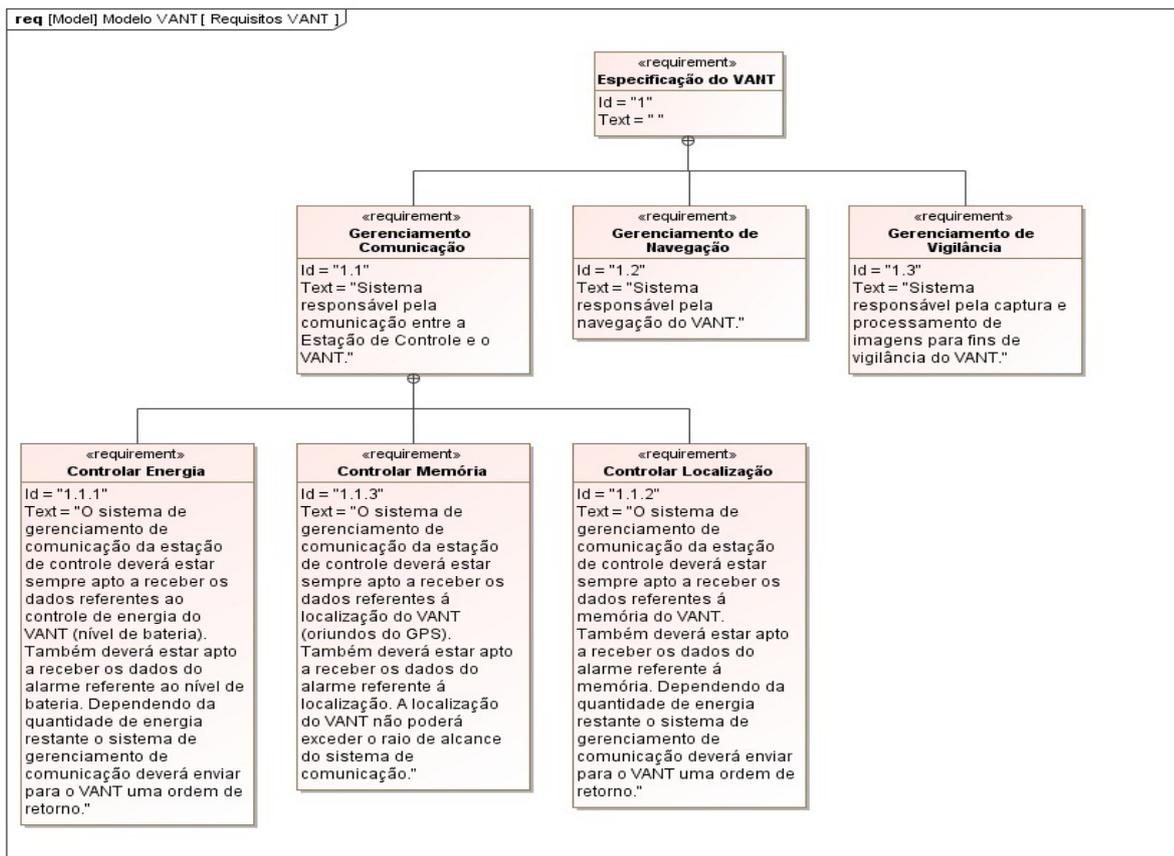
No Diagrama 6 estão modelados os casos de uso de alto nível, ou seja, em um nível alto de abstração, sem detalhamentos. Nesse momento, trabalha-se no que equivale à segunda fase do PU, a Elaboração, que é dividida em duas partes: a análise e o projeto (WAZLAWICK, 2011). No próximo diagrama são apresentados os fluxos de informação (entradas e saídas) por meio do diagrama de sequência. No Diagrama 7, pode-se ver a iteração entre os componentes do projeto (Estação de Controle, VANT, Sistema Embarcado) para o controle de energia. O fluxo de informações de entradas e saídas serve para definir quem envia uma mensagem, à quem e quando.

Diagrama 4 – Contexto do projeto



Fonte: Wazlawick (2011).

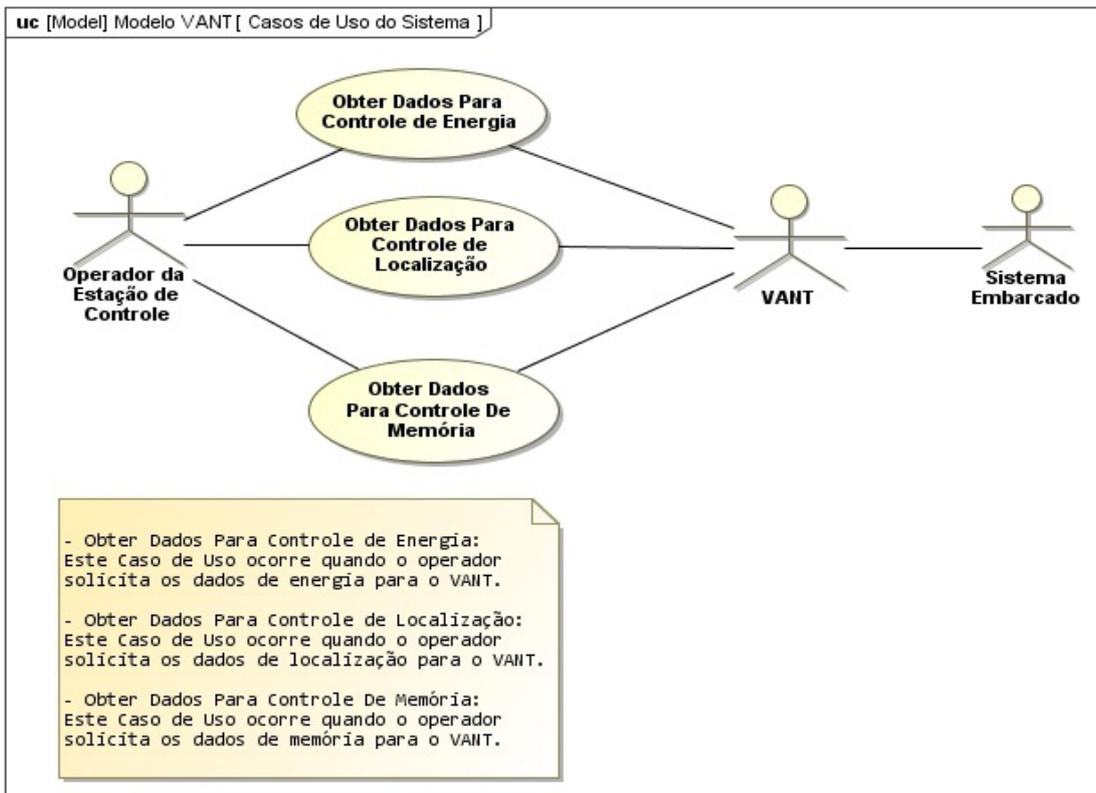
Diagrama 5 – Requisitos do Projeto



Fonte: Wazlawick (2011).

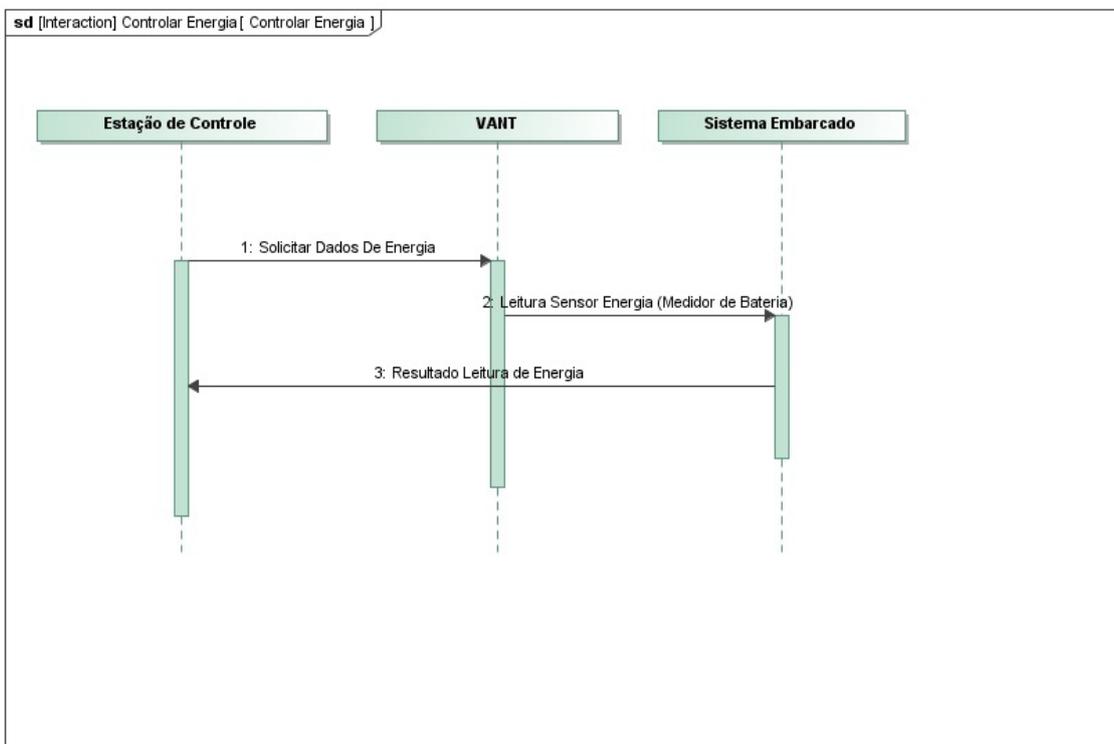
A próxima etapa do MDA, o PSM, corresponde à atividade de projeto da etapa de elaboração do PU. Esta etapa não foi realizada no projeto em função do fim dos prazos estipulados e, também, por não estar prevista a sua abordagem neste projeto de pesquisa. Tal atividade consiste em desenvolver uma proposta de solução às necessidades identificadas mediante a investigação feita na etapa de análise (PU), ou seja, na realização do PIM (MDA).

Diagrama 6 – Casos de uso do projeto



Fonte: o autor.

Diagrama 7 – Diagrama de sequência do caso de uso controlar energia



Fonte: o autor.

6 CONCLUSÃO

Em razão do tipo de modelagem necessária ao projeto, que engloba não apenas o conhecimento e a modelagem de *software*, mas também de *hardware*, percebeu-se que somente o uso de Processo Unificado não seria suficiente e nem se obteria um resultado satisfatório. Por isso, foi utilizado o MDA, descoberto durante a pesquisa.

Foi estabelecida uma relação entre as etapas do MDA e as etapas do Processo Unificado. Alguns artefatos do PU foram removidos ou substituídos, conforme verificada a forma que mais se adequava ao MDA. Trabalhos futuros poderão dar continuidade ao desenvolvimento dos requisitos que já foram modelados, completando, assim, as próximas etapas do MDA. A mais imediata é, também, a mais complexa de todas, já que elaborar o PSM envolve a definição de tecnologias e de plataformas. Novas pesquisas poderão complementar os requisitos já modelados e contemplar novos requisitos complementares a esta pesquisa.

MDA integrated use of the unified process in modeling of a management system for communication UAS

Abstract

This paper presents the modeling of a system of management communication for UAVs through the use of Model Driven Architecture (MDA), a paradigm of model driven development will. The main objective is to demonstrate how the Unified Process was included during the modeling MDA. Were entered during the stages of development phases of the Unified Process and its artifacts. Included also are diagrams SysML to replace some artifacts Process Unified in order to make more efficient modeling. Keywords: Model driven architecture. MDA. Unified process. SysML. Model.

Nota explicativa:

¹ Termo em inglês que designa alguma coisa que está no meio de outras duas. No meio da comunicação, significa um *software* encarregado de mediar dados distintos, usado para transportar dados entre programas de plataformas diferentes, por exemplo.

REFERÊNCIAS

CUNHA, L. M.; LUCENA, C. J. P. de. Proposta de um método para o desenvolvimento de aplicações para a web semântica utilizando MDA. In: WORKSHOP DE WEB SEMÂNTICA, 1., 2004, Brasília, DF. **Anais eletrônicos...** Brasília, DF, 2004. Disponível em: <http://ceur-ws.org/Vol-123/17_wss2004_final.pdf>. Acesso em: 04 dez. 2012.

FRIEDENTHAL, S.; MOORE, A.; STEINER, R. **A practical guide to sysML: the systems modeling language**. 2. ed. Waltham: Elsevier, 2011.

KLEPPE, A.; WARMER, J.; BAST, W. **MDA explained: the model driven architecture™: practice and promise**. Boston: Pearson Education, 2003.

OBJECT MANAGEMENT GROUP. **MDA guide version 1.0.1. document number:**

omg/2003-06-01. Disponível em: <<http://www.enterprise-architecture.info/Images/MDA/MDA%20Guide%20v1-0-1.pdf>>. Acesso em: 04 dez. 2011.

_____. **OMG systems modeling language (OMG sysML™) version 1.2. OMG document number:** formal/2010-06-01. Disponível em: <<http://www.omg.org/spec/SysML/1.2/PDF/>>. Acesso em: 04 dez. 2011.

PELLEGRINI, F. G. et al. Transformações de modelos para um processo MDA. In: WORKSHOP DE TRABALHOS DE INICIAÇÃO CIENTÍFICA E DE GRADUAÇÃO BAHIA-ALAGOAS-SERGIPE (WTICG-BASE), 10., 2010, Maceió. **Anais eletrônicos...** Maceió, 2010. Disponível em: <<http://www.fejal.br/erbase2010/papers/wticg/65654.pdf>>. Acesso em: 04 dez. 2012.

WAZLAWICK, R. S. **Análise e projeto de sistemas de informação orientados a objetos.** 2. ed. Rio de Janeiro: Elsevier, 2011.