

Uma metodologia para o desenvolvimento de produtos mecatrônicos integrando engenharia de software e engenharia de produtos

Ana Patrícia Fontes Magalhães (PPGM/UFBA) apfmag@ig.com.br

Aline Andrade (PPGM/LaSiD/DCC/UFBA) aline@ufba.br

Leila Silva (PPGM/DCCE/UFS) leila@ufs.br

Herman Lepikson (PPGM/DEM/UFBA) herman@ufba.br

Resumo

O projeto de sistemas mecatrônicos envolve um conhecimento multidisciplinar das áreas de mecânica, elétrica e computação. Este cenário de diversidade de conhecimentos impõe desafios no desenvolvimento destes sistemas, relativos a uma maior complexidade e integração da equipe de desenvolvimento. Este artigo apresenta uma metodologia de desenvolvimento de produtos mecatrônicos que reúne as técnicas da engenharia de software a técnicas de engenharia de produto, visando integrar métodos, procedimentos e ferramentas da equipe multidisciplinar num ambiente único.

Palavras chave: Metodologia de projeto de produto mecatrônico, OO, UML.

1. Introdução

Os sistemas mecatrônicos têm a particularidade de utilizar em seus projetos um conhecimento multidisciplinar, integrando as tecnologias de engenharia mecânica, eletrônica e computação (ZHANG, 2003). Este nível de integração tem induzido grandes transformações no processo de desenvolvimento de produtos mecatrônicos para que haja uma efetiva participação dos profissionais de diversas áreas de conhecimento trabalhando em conjunto na construção de uma mesma solução.

É consenso entre vários pesquisadores que a utilização das técnicas de Orientação a Objetos (OO) (RUMBAUGH, 1994) é útil para o projeto de produtos mecatrônicos. Conceitos de reutilização, modularização, encapsulamento, herança, entre outros, presentes na OO, se adequam bem ao ambiente de integração de componentes e permitem uma melhor manutenção, portabilidade e evolução destes sistemas. Também a utilização da *Unified Modeling Language* (UML) (BOOCH, 2000) está sendo adotada por oferecer modelos de fácil compreensão e padronização. (BONFE, 2003), (TRANORIS, 2003), (BONFE, 2002), (THRAMBOULIDIS, 2002), (MROZEK, 2002).

Bonfe e outros (2003) propõem uma metodologia para sistemas mecatrônicos de controle baseada em uma extensão da UML para aplicações mecatrônicas. Tranoris (2003) e Thramboulidis (2002) sugerem uma metodologia para sistemas mecatrônicos voltada a aplicações de controle distribuído, em que a UML é usada nas fases de levantamento de requisitos e análise. Após estas duas fases, regras são definidas para traduzir os modelos obtidos para padrões de engenharia, com o objetivo de facilitar a compreensão pelos engenheiros.

As metodologias mencionadas anteriormente estão voltadas para a modelagem do elemento controle do sistema. Os autores deste artigo desconhecem trabalhos que proponham metodologias que se apliquem à modelagem de todo o sistema e permitam um nível de abstração que forneça independência de tecnologia.

Nesse contexto, este artigo propõe uma metodologia para o desenvolvimento de produtos mecatrônicos que visa permitir uma boa comunicação e aproveitamento do conhecimento de cada componente da equipe e que tem como foco a modelagem do problema, abstraindo a tecnologia de construção. Assim, decisões de implementação são definidas em um estágio mais avançado dos trabalhos, com a ajuda de ferramentas que permitam selecionar a opção mais indicada. A metodologia proposta baseia-se na metodologia *Rational Unified Process* (RUP) (KRUCHTEN, 2000), a qual aplica técnicas de análise OO e utiliza como linguagem de representação de modelos a UML. Além disso, a metodologia incorpora algumas técnicas de desenvolvimento de produtos pertinentes ao domínio dos sistemas mecânicos e mecatrônicos.

Este artigo está organizado como descrito a seguir. Na seção 2 a metodologia proposta é descrita. O detalhamento das disciplinas pertinentes à metodologia é apresentado na seção 3. Finalmente, na seção 4 algumas considerações finais e direções de trabalho futuros são abordadas.

2. A metodologia proposta

O desenvolvimento de produtos mecatrônicos é composto basicamente de três etapas que englobam todo o ciclo de vida de um produto: o pré-desenvolvimento, responsável por atividades tais como estudos de mercado e plano de viabilidade; o desenvolvimento, que engloba o projeto do produto, o projeto do respectivo processo e a preparação para produção; e o pós-desenvolvimento que acontece depois que o produto já está pronto e disponibilizado ao cliente e consiste do seu acompanhamento e posterior descarte (ROSENFELD, 2006). A metodologia descrita neste artigo é centrada na segunda etapa, a de desenvolvimento do sistema, mais especificamente no projeto do produto. A especificação da metodologia a seguir apresentada usa a linguagem de padronização de especificações de processos metodológicos *Software Process Engineering Metamodel Specification* (SPEM) (OMG, 2006).

O ciclo de vida desta metodologia está dividido em quatro fases denominadas concepção, elaboração, construção e entrega. Ao final de cada uma destas fases existe um ponto de controle, onde algumas checagens precisam ser realizadas antes que a próxima fase seja iniciada (figura 1). Por exemplo, ao final da fase de elaboração espera-se que um modelo abstrato do sistema tenha sido gerado.

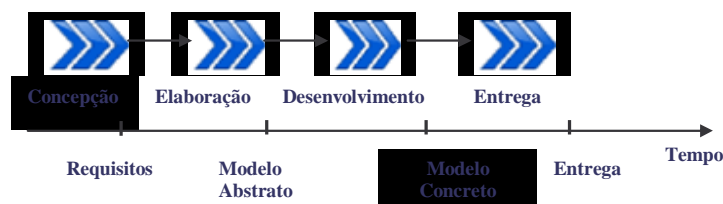


Figura 1 - Fases da Metodologia Proposta.

Assim como na metodologia RUP, cada uma destas fases engloba um conjunto de disciplinas que devem ser realizadas ao longo do processo de projeto. Para o domínio das aplicações mecatrônicas a metodologia propõe o uso de oito disciplinas, conforme apresentado na figura 2 e detalhado na seção 3. Cada uma destas disciplinas é composta por um conjunto de atividades que resultam na geração de artefatos. Artefatos representam aspectos importantes do sistema como, por exemplo, os requisitos, a estrutura estática das classes que o compõe, o comportamento, as normas e restrições a serem seguidas.

A concepção é a fase inicial de um projeto e tem como objetivo a definição dos requisitos do

produto que será construído. Para tal, realizam-se reuniões de levantamento que deverão ser multidisciplinares. O produto final desta fase é um documento que expressa o escopo do sistema que será desenvolvido. Durante a fase de concepção são desenvolvidas as atividades relativas a disciplina de Requisitos e Escopo e à disciplina de Gerenciamento.

A fase de elaboração tem como objetivo detalhar os requisitos que compõe a versão do produto que será construído. Esta fase também prevê reuniões multidisciplinares. A partir dos requisitos identificados e detalhados, são mapeados aspectos específicos do sistema tais como a sua estrutura estática e comportamental, o relacionamento entre os diversos objetos destas estruturas, a comunicação entre eles, como funcionará o controle do sistema, sua interação com os demais elementos, dentre outros aspectos. O produto final desta fase é a geração de um modelo, chamado de modelo abstrato do sistema, já simulado e particionado nos componentes e elementos de software, de hardware, mecânicos e eletrônicos que serão construídos. Durante a fase de elaboração são desenvolvidas as atividades relativas às disciplinas de Análise, Particionamento, Verificação e Gerenciamento.

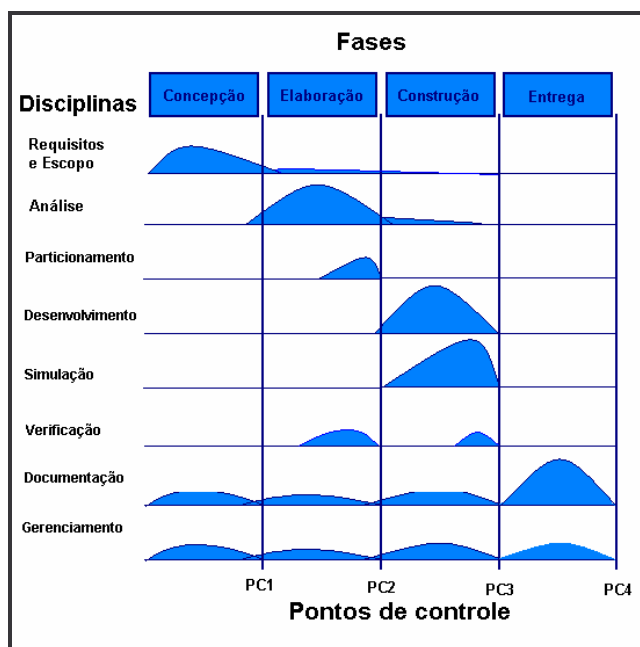


Figura 2 - Disciplinas e fases da metodologia

A fase de construção tem como entrada o modelo abstrato gerado e desenvolve os seus componentes, visando montar um produto piloto que será exaustivamente testado antes que o projeto seja enviado para a linha de produção. Um componente pode ser particionado em elementos que serão desenvolvidos por cada área específica com a técnica de desenvolvimento que julgar mais adequada, obedecendo às interfaces definidas. O produto final desta fase é um modelo concreto e sintetizado, na forma de protótipo testado e pronto para ser fabricado. Durante a fase de construção são desenvolvidas as atividades relativas às disciplinas de Desenvolvimento, Simulação e Gerenciamento.

A entrega é a fase responsável pela liberação do modelo do produto para fabricação. Toda a documentação complementar relativa ao processo de fabricação deverá ser gerada para ser entregue junto com o projeto piloto. Embora a metodologia não aborde o planejamento do processo de produção, cada área deve fornecer a documentação necessária para a fabricação dos seus elementos. Durante a fase de entrega serão desenvolvidas as atividades relativas às disciplinas de Documentação e Gerenciamento.

Considerando o ambiente multidisciplinar, a equipe de desenvolvimento deverá ser composta por um gerente, vários líderes e pode ter tantos especialistas quantos forem necessários. Serão realizadas várias reuniões no decorrer do desenvolvimento, que podem ser multidisciplinares, com a participação do gerente, de todos os líderes e se necessário de especialistas; ou específicas, com a participação do líder da área e seus especialistas.

3. As disciplinas propostas

Uma disciplina reúne um conjunto de atividades que podem ou não ser desenvolvidas, a depender da necessidade do projeto, e pode gerar um ou mais artefatos. A metodologia especifica as seguintes disciplinas: Requisitos e Escopo; Análise; Particionamento; Desenvolvimento; Simulação; Verificação; Documentação; e Gerenciamento. A seguir cada uma destas disciplinas será detalhada.

O objetivo da disciplina Requisito e Escopo é identificar as necessidades do cliente para o produto a ser construído e, a partir destas, mapear os seus requisitos e definir o respectivo escopo. A figura 3a apresenta todas as atividades desenvolvidas nesta disciplina, representado pelos símbolos na forma de seta, identifica o responsável pela disciplina, neste caso o Gerente e Líder, e os artefatos que são gerados como produto, dispostos à direita da lista de atividades.

A atividade inicial da disciplina Requisitos e Escopo consiste da descrição do produto, pois é a partir dela que se tem um primeiro contato com os conceitos do que será construído. A maior parte das atividades desenvolvidas nesta disciplina deve ser realizada em reuniões para que a equipe multidisciplinar se familiarize com o sistema como um todo.

Assim, a partir de uma lista de necessidades do cliente obtidas durante as reuniões, é elaborado um diagrama de casos de uso (BOOCH, 2000), que ilustra os requisitos funcionais. Definições importantes que tenham sido mencionadas durante as reuniões podem ser anotadas em documentos de cenários (BOOCH, 2000) para utilização em outras disciplinas.

Os requisitos não funcionais compreendem aqueles que têm impacto sobre o projeto, mas que não estão diretamente ligados à funcionalidade do produto, tais como: requisitos operacionais (desempenho, falhas e tempo); restrições (geométricas, espaciais, do processo de manufatura, fatores de risco à construção do produto); normas (segurança, uso, qualidade), entre outros. Estes requisitos devem estar devidamente documentados em uma lista de requisitos não funcionais.

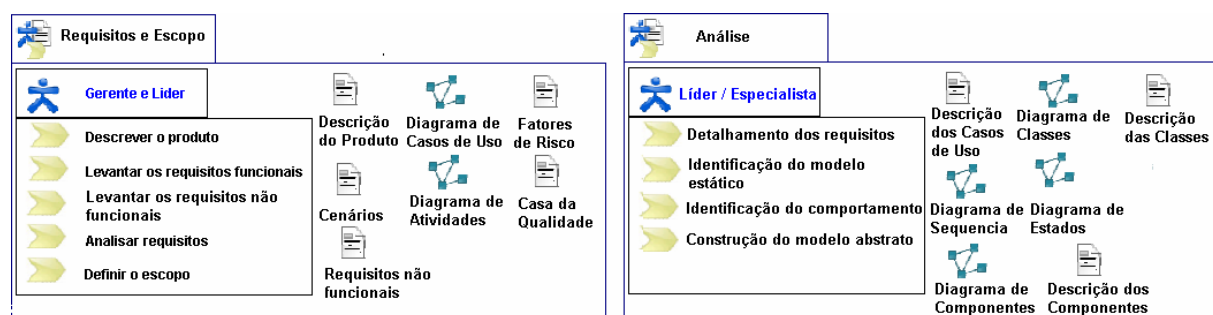


Figura 3 - Papéis, atividades e artefatos de (a) Requisitos e Escopo e (b) Análise.

Para facilitar a compreensão do funcionamento do produto pode-se desenvolver um diagrama de atividades (BOOCH, 2000) que ilustre a seqüência de ações para o fluxo normal de utilização do produto.

A atividade final desta disciplina é a definição do escopo do produto a ser construído. Para

isso, deve-se montar a matriz da casa da qualidade (ROSENFELD, 2006), onde os requisitos poderão ser analisados sob diversos aspectos diferentes, tais como: relacionamentos, conflitos, impacto e importância. Esta análise permitirá uma visão mais ampla do produto propiciando quantificar parâmetros para a seleção adequada do seu escopo.

O objetivo da disciplina Análise é detalhar os requisitos que compõe o sistema. A figura 3b apresenta todas as atividades desenvolvidas nesta disciplina, identifica o papel responsável por este desenvolvimento e os artefatos que são gerados como produto.

O detalhamento dos requisitos a partir dos casos de uso identificados serve de base para a definição do modelo estático do sistema, representado pelo diagrama de classes (BOOCH, 2000). Este modelo é uma abstração da estrutura do produto a ser construído baseado nas suas funcionalidades, independente da tecnologia que será utilizada para construí-lo. O comportamento, modelo dinâmico, do produto deve também ser especificado nesta disciplina com a construção dos diagramas de seqüência e de estados (BOOCH, 2000).

A última atividade desta disciplina é a geração do modelo abstrato do sistema que representa a estrutura do sistema em componentes, com funções específicas. Cada componente poderá ser formado por uma ou mais classes identificadas no diagrama de classes. A decisão de como agrupá-las em um único componente depende da avaliação da equipe, influenciada por fatores tais como fabricação, montagem e manutenção.

O objetivo da disciplina Particionamento (figura 4a) é identificar como serão desenvolvidos os componentes definidos no modelo abstrato do sistema. Na atividade de análise dos princípios de solução existentes definem-se, para cada componente do modelo, as diversas soluções possíveis de construção deste componente a partir da matriz morfológica (ROSENFELD, 2006). Estas soluções englobam a decisão pelo desenvolvimento do componente totalmente mecânico, totalmente eletrônica, totalmente em software ou de componentes híbridos.

A atividade de co-síntese tem como objetivo definir como cada componente será construído. Com a ajuda da matriz morfológica e a utilização de algoritmos de particionamento (THOMAS, 1993) são efetuadas simulações estatísticas que fornecem resultados relativos ao desempenho de cada solução simulada. Estes resultados provêm subsídios aos projetistas na decisão de como o componente será implementado. No entanto, questões muitas vezes subjetivas, como segurança e custo, devem também ser consideradas e podem ser decisivas na escolha da tecnologia a ser usada.

Assim, após a co-síntese, um componente híbrido será composto por um conjunto de elementos que serão desenvolvidos por cada área separadamente. Nestes casos, a interface entre estes elementos também deve ser definida.

A atividade de simulação tem como objetivo avaliar a viabilidade do modelo abstrato particionado antes que ele seja realmente implementado. Com a ajuda de um *software* de modelagem e simulação pode-se validar as interfaces entre os componentes, as interfaces internas aos componentes híbridos e se os requisitos estão sendo atendidos.

Finalmente, como última atividade desta disciplina tem-se a construção dos casos de testes que serão aplicados ao produto final prototipado. Deve-se construir casos de testes para cada componente, elemento e para o sistema como um todo.

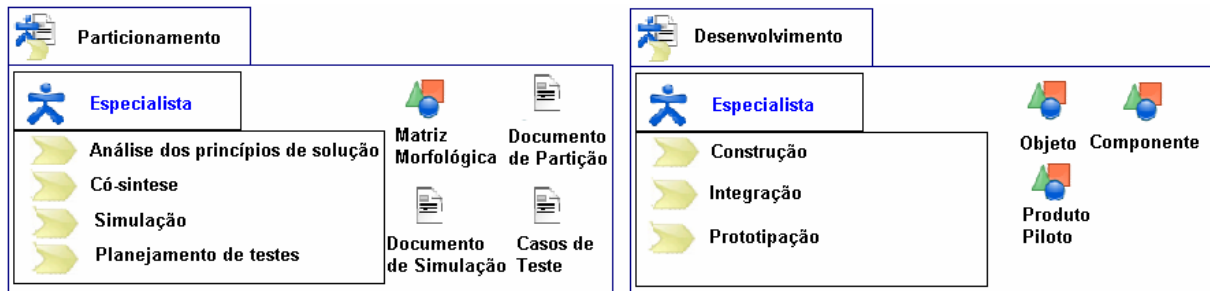


Figura 4 - Papéis, atividades e artefatos de (a) Particionamento e (b) Desenvolvimento.

O objetivo da disciplina Desenvolvimento (figura 4b) é construir um modelo concreto do produto, ou seja, construir um protótipo do produto para ser enviado à produção. O desenvolvimento dos componentes e respectivos elementos que compõe o sistema deve ser realizado pelas áreas específicas, respeitando as interfaces definidas na disciplina de Particionamento. Assim, a atividade de construção consiste no detalhamento, por cada área separadamente, da melhor solução para cada componente / elemento que lhe foi atribuído. Para isso, a área poderá utilizar a técnica que desejar desde que esta atenda às especificações definidas anteriormente.

A atividade de integração corresponde à montagem do componente híbrido reunindo os seus elementos desenvolvidos em cada área. Faz parte desta atividade validar o componente de acordo com os casos de teste definidos, deixando-o pronto para a construção do protótipo. A prototipação compreende a montagem do produto a partir dos componentes construídos.

O objetivo da disciplina Simulação (figura 5a) é testar o protótipo produzido durante o desenvolvimento para checar o seu perfeito funcionamento após a integração dos componentes. Estes testes serão realizados com base no documento de casos de testes gerado na disciplina Desenvolvimento.

A atividade de co-simulação é responsável pela validação do produto piloto sob a visão técnica, para verificar se este atende aos requisitos definidos no início do projeto. O resultado desta atividade é uma lista com os testes realizados para cada caso de teste, que indica se houve sucesso, insucesso e se foi necessário retornar a alguma fase do desenvolvimento.

A atividade de teste do cliente tem como objetivo testar o produto sob a visão do cliente, para verificar se o sistema atende às suas necessidades, definidas no início do projeto. O resultado desta atividade é uma lista com os testes realizados para cada necessidade, indicando se houve sucesso, insucesso e neste caso, se foi necessário retornar a alguma fase do desenvolvimento para rever a especificação.

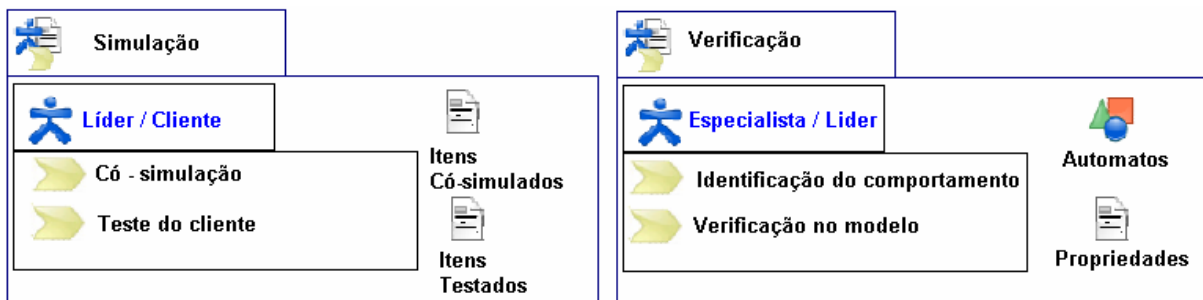


Figura 5 - Papéis, atividades e artefatos de (a) Simulação e (b) Verificação.

A disciplina Verificação (figura 5b) tem como objetivo verificar formalmente se o sistema

está correto mediante propriedades identificadas a partir dos requisitos. A primeira atividade, identificação do comportamento, realiza o mapeamento dos diagramas de estados em autômatos. Estes autômatos servirão de base para a especificação, em uma linguagem apropriada, das propriedades do sistema que deverão ser verificadas. A verificação do modelo consiste da submissão das propriedades a um verificador de modelos e posterior análise dos resultados.

A disciplina de Documentação (figura 6a) está presente em todas as fases do desenvolvimento e é responsável pela geração da documentação final do produto. Compreende os artefatos gerados nas diversas disciplinas e os documentos de implementação dos elementos específicos de cada área. Na fase de entrega do produto, esta disciplina se torna mais significativa, pois tem como objetivo reunir e revisar todos os documentos gerados ao longo do processo de desenvolvimento. Deve-se construir um documento que será enviado à linha de produção para fabricação do produto, bem como às demais áreas para geração dos demais documentos a serem criados para uso, manutenção descarte ou propaganda.

A disciplina de Gerenciamento (figura 6b) está presente em todas as fases do projeto do produto. Seu objetivo é garantir que a metodologia está sendo corretamente aplicada, além de analisar os resultados parciais do desenvolvimento de maneira a prevenir problemas que venham a comprometer o objetivo final do trabalho.

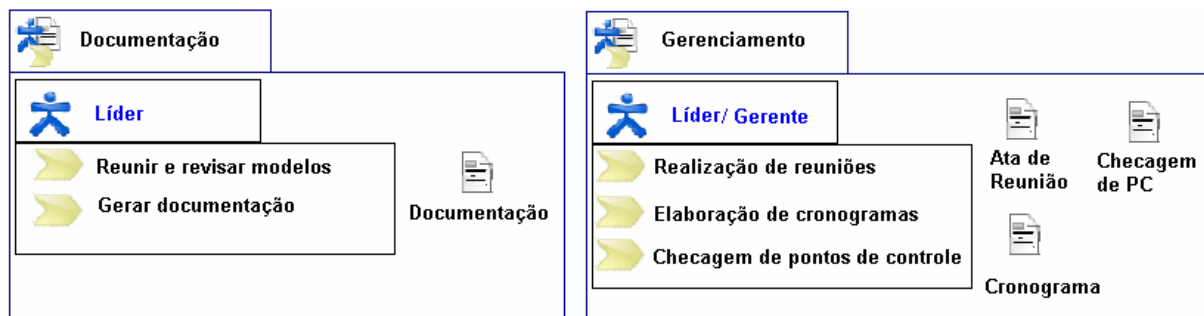


Figura 6 - Papéis, atividades e artefatos de (a) Documentação e (b) Gerenciamento.

A atividade de realização de reuniões indica que reuniões de acompanhamento devem ser realizadas periodicamente, entre o gerente e os líderes de cada área envolvida. Qualquer modificação significativa deve ser aprovada nesta reunião. A atividade de elaboração de cronogramas é importante para o acompanhamento dos trabalhos por parte do gerente e dos líderes em cada fase do projeto.

4. Conclusões

Uma característica marcante do desenvolvimento de sistemas mecatrônicos é o ambiente multidisciplinar e a complexidade que ele impõe à concepção de um produto que aproveite os benefícios deste ambiente. Cada uma das áreas envolvidas sejam elas mecânica, eletrônica ou computação, já possui seus métodos de trabalho consolidados e que são aplicados com sucesso em suas áreas restritas. Isso dificulta um processo de construção focado em um produto final único.

Este artigo propõe uma metodologia que visa integrar métodos, procedimentos e ferramentas da equipe multidisciplinar num ambiente único. A metodologia compreende todas as fases de projeto de um produto mecatrônico e foi concebida visando uma flexibilidade de forma a integrar os conhecimentos de cada área. Para facilitar a comunicação entre os membros da equipe, a metodologia sugere a adoção da linguagem de modelagem visual UML, a qual

provê alto nível de abstração.

A proposta inova nos seguintes aspectos: compreende as várias etapas do projeto do produto, desde a especificação de requisitos até a construção do protótipo que será enviado à linha de produção; é centrada na definição conceitual do produto, permitindo flexibilidade de análise quanto a pertinência de soluções híbridas; baseia-se em práticas consolidadas da Engenharia de Software, como a metodologia RUP, e da Engenharia de Produtos; define claramente as interfaces, visando facilitar a integração dos elementos construídos nos diversos domínios de conhecimento; enfatiza a necessidade de simulações, testes e verificações ao longo do processo de desenvolvimento, como meio de prover maior confiabilidade do produto gerado.

Embora a proposta geral da metodologia já esteja sedimentada, ainda é necessário um melhor detalhamento dos fluxos de atividades de cada disciplina. Além disso, pretende-se explorar aspectos de tolerância a falhas e requisitos temporais, comuns a sistemas mecatrônicos. O desenvolvimento de estudos de caso é também fundamental para validar a proposta. Todas estas tarefas são trabalhos futuros imediatos.

Referências

- BONFE, M.; DONATI, C.; FANTUZZI, C.** *An Application of Software Design Methods to Manufacturing Systems Supervision and Control*. Conference of Control Applications. IEEE, 2002.
- BONFE, M.** *Design and Verification of Mechatronic Object-Oriented Models for Industrial Control Systems*. Universita di Ferrara, Dip Ingegneria, via Saragat 1, Ferrara Italy, 2003.
- BOOCH, G.; RUMBAUGH, J.; JACOBSON, I.** *UML guia do usuário*. Rio de Janeiro. Campus, 2000.
- KRUCHTEN, P.** *The Rational Unified Process An Introduction*. Massachusetts, Addison Wesley, 2000.
- OMG.** *Software Process Engineering Metamodel Specification*. Versão 1.1. Disponível em www.omg.org, acessado em 15/02/2006.
- MROZEK, Z.** *Design of the Mechatronic Systems with Help of UML Diagrams*. 3-rd Workshop on Robot Motion and Control, Bukowy Dworek, Poland, 2002.
- ROSENFELD, H. [et al].** *Gestão de Desenvolvimento de Produtos. Uma referencia para melhoria da processo*. 1 edição, Ed. Saraiva, São Paulo, SP, 2006.
- RUMBAUGH, J. [et al].-** *Modelagem de Projetos Orientados a Objetos*. Ed. Campus, Rio de Janeiro, RJ, 1994.
- THRAMBOULIDIS, K..** *Development of Distributed Industrial Control Applications: The CORFU Framework*. IEEE, 2002.
- THRAMBOULIDIS, K.** *Using UML in Control and Automation: A Model Driven Approach*. 2. IEEE International Conference on Industrial Informatied INDIN'04, Berlin, Alemanha, 2004.
- THOMAS, D.; ADAMS, J.; SCHMIT, H.** *A Model and Methodology for Hardware-Software Codesign*. IEEE Design, 1993.
- TRANORIS, C.; THRAMBOULIDIS, K.** *Integrating. UML and the Function Block Concept for the Development of Distributed Control Applications*. IEEE, 2003.