

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/221359871>

Um Processo de Engenharia de Requisitos Baseado em Reutilização de Ontologias e Padrões de Análise.

Conference Paper · January 2007

Source: DBLP

CITATION

1

READS

41

6 authors, including:



Ricardo de Almeida Falbo

Universidade Federal do Espírito Santo

172 PUBLICATIONS 1,661 CITATIONS

[SEE PROFILE](#)



Aline Freitas Martins

Universidade Federal do Espírito Santo

3 PUBLICATIONS 14 CITATIONS

[SEE PROFILE](#)



Gleison Baioco

Universidade Federal do Espírito Santo

3 PUBLICATIONS 24 CITATIONS

[SEE PROFILE](#)



Rodrigo Dal Moro

Universidade Federal do Espírito Santo

7 PUBLICATIONS 68 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Knowledge Management in Software Testing [View project](#)



Standards Harmonization [View project](#)

Um Processo de Engenharia de Requisitos Baseado em Reutilização de Ontologias e Padrões de Análise

Ricardo de Almeida Falbo, Aline Freitas Martins, Bruno Marques Segrini, Gleison Baiôco, Rodrigo Dal Moro

*Departamento de Informática - Universidade Federal do Espírito Santo (UFES)
Av. Fernando Ferrari s/n, Campus de Goiabeiras-29.060-900- Vitória - ES - Brasil
falbo@inf.ufes.br, {alinefmart, bruno_segrini, gleison.baioco, rdalmoro}@yahoo.com.br*

Julio Cesar Nardi

*Centro Federal de Educação Tecnológica do Espírito Santo (CEFET-ES) - UnED/Colatina
Av. Arino Gomes Leal, 1700, 29.700-603-Colatina - ES - Brasil
julionardi@yahoo.com.br*

Abstract

Software reuse is pointed as a key factor for quality and productivity improvement. Ideally, reuse should occur in all levels of abstraction, but higher levels of abstraction tend to lead to major benefits. In this paper, we present a Reuse-based Requirements Engineering process that focus on reusing ontologies and analysis patterns. Reports about its application are also discussed.

1. Introdução

A reutilização, em diversos níveis, tem sido apontada como um fator importante para o aumento da qualidade e da produtividade no desenvolvimento de software [1]. Dado que a Engenharia de Requisitos (ER) é um dos processos cruciais para o sucesso de um projeto [2], é muito importante que elementos relacionados a requisitos de software sejam reutilizados, visando a se obter esses benefícios.

Entretanto, surpreendentemente, a reutilização no processo de ER tem sido pouco praticada, ainda que haja diversos métodos de ER que, de alguma forma, enfatizam o reúso.

Este trabalho apresenta um processo de ER baseado na reutilização de ontologias e padrões de análise, voltado para o desenvolvimento de sistemas de informação, que procura empregar métodos e técnicas atualmente bastante adotados pela comunidade de desenvolvimento de software. Assim, espera-se vencer

a inércia e efetivamente colocar em prática a reutilização na ER. A seção 2 aborda os temas Engenharia de Requisitos e Reutilização. A seção 3 apresenta o processo proposto. A seção 4 discute relatos de sua aplicação em um estudo de caso. Por fim, a seção 5 apresenta conclusões e perspectivas futuras do trabalho.

2. Engenharia de Requisitos e Reutilização

A Engenharia de Requisitos de Software é o ramo da Engenharia de Software que se preocupa com os objetivos do mundo real para as funções a serem desenvolvidas e restrições a serem levadas em conta no desenvolvimento de software [3].

Como o nome indica, o foco da ER está nos requisitos. Em relação ao tipo de informação que o requisito documenta, requisitos são classificados em funcionais (especificações de serviços que o sistema deve prover) e não funcionais (restrições impostas aos serviços ou funções oferecidos pelo sistema e ao processo de desenvolvimento) [4].

A ER é um processo complexo que envolve diversas atividades. Considerando o estado da prática, em um processo de ER convencional, geralmente, requisitos são inicialmente levantados utilizando-se técnicas como entrevistas, análise de documentos etc e documentados na forma de sentenças que descrevem as funções que o sistema deve prover (requisitos funcionais) e restrições a elas impostas (requisitos não funcionais). A seguir, modelos do sistema são

construídos, no que se convencionou chamar de Análise de Requisitos.

A modelagem é uma atividade central para o processo de ER. Modelos servem como uma base comum a todas as atividades desse processo. Por um lado, eles resultam das atividades de levantamento e análise de requisitos. Por outro, eles guiam esforços ulteriores de levantamento, análise, negociação e documentação [5]. Novamente levando-se em conta o estado da prática, os modelos mais comumente elaborados incluem diagramas de casos de uso e suas descrições, modelos conceituais da aplicação a ser desenvolvida (tais como diagramas de classes ou de entidades e relacionamentos, dependendo do paradigma adotado no desenvolvimento) e modelos comportamentais (tais como diagramas de estados e diagramas de interação).

Durante o levantamento e a análise de requisitos, os requisitos vão sendo documentados em um Documento de Especificação de Requisitos (DER) e por fim verificados e validados nas atividades de Verificação e Validação (V&V) de requisitos.

É necessário, ainda, gerenciar as mudanças nos requisitos acordados, manter uma trilha das mudanças e gerenciar os relacionamentos entre os requisitos e as dependências entre o DER e os demais artefatos produzidos no processo de software, atividades realizadas no âmbito da Gerência de Requisitos.

Analisando o processo de ER, é possível notar que a reutilização pode ser útil, sobretudo no reúso de requisitos de sistemas similares e de modelos (com destaque para os modelos conceituais). Contudo, na prática, na grande maioria das vezes, requisitos e modelos são construídos a partir do zero. Ou seja, requisitos iniciais são levantados junto aos interessados, modelos são construídos levando-se em conta esses requisitos iniciais, que são posteriormente refinados e novamente modelados, até se atingir um acordo com o cliente sobre o que o sistema deve prover. Entretanto, essa abordagem tem se mostrado insuficiente, pois desconsidera a reutilização do conhecimento já existente na organização acerca do domínio do problema.

A reutilização no desenvolvimento de software tem como objetivos principais melhorar o cumprimento de prazos, diminuir custos e obter produtos de maior qualidade [1]. Artefatos de software podem ser reutilizados, investindo-se esforço na sua adaptação a novas situações. Uma vez que esses itens tenham sido desenvolvidos para reúso, o esforço de adaptação e reutilização tende a ser menor.

No que concerne à ER, os principais esforços em direção à reutilização estão na Engenharia de Domínio [6] e no uso de padrões de análise [7] [8].

Padrões são guias. Eles são usados para descrever soluções de sucesso para problemas de software comuns, refletindo a estrutura conceitual dessas soluções [9]. Padrões de análise são geralmente definidos a partir de modelos conceituais de aplicações, sendo descobertos e não inventados [7]. Ou seja, a partir da análise de diversos eventos de negócio do mesmo tipo, um padrão de análise pode ser derivado por meio da abstração de elementos comuns [8].

A Engenharia de Domínio representa um enfoque sistemático para a produção de componentes reutilizáveis que engloba atividades de Análise, Projeto e Implementação de Domínio, as quais objetivam, respectivamente, representar requisitos comuns de uma família de aplicações por meio de modelos de domínio, disponibilizar modelos arquiteturais para aplicações a partir de um único modelo de domínio e disponibilizar implementações de componentes que representam funcionalidades básicas de aplicações relacionadas a um domínio [6].

Claramente a Análise de Domínio é a atividade diretamente ligada à reutilização na ER. Diversos são os métodos de Análise de Domínio existentes, dentre eles FODA, ODM, EDLC, FORM, RSEB e Catalysis [6]. Falbo et al. [10] propõem o uso de ontologias como modelos de domínio e apresentam um processo de Engenharia de Domínio baseado em ontologias. Neste contexto, uma ontologia é um artefato de engenharia, constituído de um vocabulário de termos organizados em uma taxonomia, suas definições e um conjunto de axiomas formais usados para criar novas relações ou para restringir as suas interpretações segundo um sentido pretendido [11].

Cima et al. [12] destacam que a Análise de Domínio pode ser considerada como uma atividade anterior ao desenvolvimento de software (e, portanto, anterior à ER), uma vez que esse se utiliza dos modelos daquela.

É inegável que a reutilização na ER vem acontecendo de alguma forma, sobretudo informalmente. Por exemplo, a partir de Documentos de Especificação de Requisitos de projetos anteriores (sobretudo os similares), restrições, listas de interessados, glossários e até requisitos podem ser reutilizados [8]. Apesar disso, parece que a inserção da prática de reutilização como parte integral do processo de ER, de fato, ainda não aconteceu. As principais razões para esse fato têm cunhos metodológicos, tecnológicos, culturais e econômicos [13].

Nuseibeh e Easterbrook [14], em seu mapa para o futuro da ER, colocaram o reuso de modelos como um dos maiores desafios da ER nos anos 2000. Eles acreditavam que, para muitos domínios de aplicação, modelos de referência para especificação de requisitos seriam desenvolvidos, de modo que o esforço de desenvolver modelos de requisitos a partir do zero fosse reduzido. De alguma forma, isso, de fato, vem ocorrendo, ainda que não propriamente na forma de modelos de referência, mas por meio da construção de diversas ontologias de domínio e pelo crescente número de padrões de análise disponíveis.

Dado o panorama atual, este trabalho apresenta um processo de Engenharia de Requisitos *com* reutilização baseado em ontologias e padrões de análise, desenvolvido para ser aplicado ao desenvolvimento de sistemas de informação. Esse processo procura empregar métodos e técnicas atualmente adotados em larga escala pela comunidade de desenvolvimento de software, visando a minimizar alguns dos fatores culturais e metodológicos que têm restringido a difusão da reutilização na ER.

3. Um Processo de Engenharia de Requisitos Baseado em Reutilização

A Figura 1 mostra as atividades do processo de ER proposto.

O processo começa com um levantamento preliminar de requisitos, cujo objetivo é definir o escopo do projeto e enumerar os principais requisitos funcionais e não-funcionais. Paralelamente, um modelo conceitual preliminar deve ser elaborado, tomando por base ontologias e padrões de análise relacionados ao domínio do problema. Como produto dessa atividade, um DER preliminar é produzido.

Com base nos requisitos iniciais e no modelo conceitual preliminar, um levantamento de requisitos detalhado é conduzido e o modelo conceitual da aplicação é refinado. Finalmente, modelos comportamentais da aplicação são construídos.

Paralelamente a esse fluxo de trabalho principal, a documentação dos requisitos vai sendo elaborada (DER Preliminar e DER), verificada, validada e gerenciada. Uma vez aprovada, passa-se para a solução técnica propriamente dita (projeto, implementação etc). Senão, há um retorno para a correspondente atividade, visando a corrigir os problemas detectados.

A seguir, cada uma dessas atividades é discutida com um mais de detalhes.

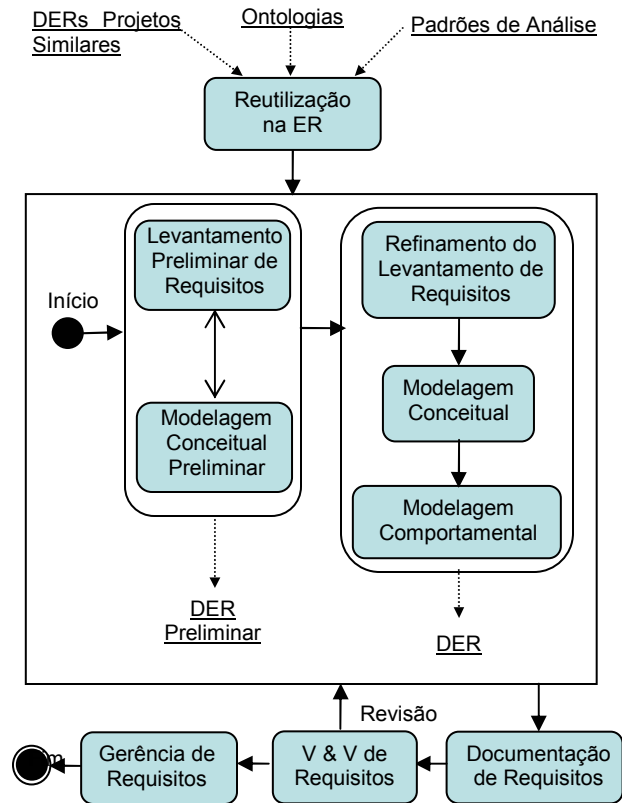


Figura 1 - Processo de Engenharia de Requisitos com reutilização

Levantamento Preliminar de Requisitos

O objetivo desta atividade é estabelecer o escopo do software a ser desenvolvido, especificando um conjunto de funcionalidades (requisitos funcionais) associado a outras características desejadas (requisitos não funcionais). Requisitos são inicialmente levantados e documentados na forma de sentenças descrevendo essas funções e restrições, como ilustra a Tabela 1. Tipicamente, diversas técnicas de levantamento de requisitos são empregadas, tais como entrevistas, investigação de documentos, observação, dinâmicas de grupo etc. Uma característica fundamental dessa atividade é o seu enfoque em uma visão do cliente/usuário. Em outras palavras, está-se olhando para o sistema a ser desenvolvido por uma perspectiva externa. Ainda não se está procurando definir a estrutura interna do sistema, mas sim procurando saber que funcionalidades o sistema deve oferecer ao usuário e que restrições elas devem satisfazer.

A reutilização nesta etapa tem um caráter mais informal. Recomenda-se a inspeção de DERs de projetos já concluídos, sobretudo aqueles similares, buscando-se reutilizar definições de requisitos, principalmente requisitos não-funcionais.

Modelagem Conceitual Preliminar

O objetivo desta atividade é definir um modelo conceitual preliminar para a aplicação a partir de ontologias e padrões de análise. A ênfase neste momento é a captura dos principais conceitos envolvidos, definindo uma estrutura interna básica do sistema, mas sem uma preocupação com a elaboração de um modelo completo, incluindo atributos e operações. Quando ontologias são reutilizadas, conceitos, relações e propriedades são mapeados, respectivamente, para classes, associações e atributos no modelo da aplicação [10]. No caso de padrões de análise, como esses são normalmente escritos na forma de diagramas convencionais (tais como diagramas de classes), eles podem ser reutilizados diretamente.

Alguns domínios podem ter vasto material disponível e, portanto, a seleção de ontologias e padrões de análise relevantes para o problema em questão é uma tarefa importante desta etapa. É útil examinar diversos modelos, procurando adaptá-los à realidade do projeto em questão, tomando por base os requisitos inicialmente levantados.

Como produto final desta etapa, um Documento de Especificação de Requisitos Preliminar é elaborado, basicamente contendo uma descrição do mini-mundo apresentando o problema e seu contexto, um conjunto de requisitos funcionais e não funcionais devidamente identificados, como ilustrado na Tabela 1, e um modelo conceitual preliminar.

Refinamento do Levantamento de Requisitos

De posse dos requisitos iniciais e do modelo preliminar, um refinamento dos requisitos deve ser realizado. As técnicas de levantamento de requisitos empregadas no levantamento preliminar de requisitos são igualmente válidas, sendo que outras, como a prototipagem e a modelagem de casos de uso, são indicadas.

No que se refere à modelagem de casos de uso, é interessante notar que casos de uso custodias (aqueles que lidam basicamente com a inclusão, exclusão, alteração e consulta de itens de informação) são mais facilmente identificados a partir do modelo conceitual preliminar, acelerando o processo. Outros, especialmente os relacionados ao negócio em si, ditos casos de uso essenciais, normalmente estão registrados de alguma forma nos requisitos funcionais iniciais. Assim, a maior parte do trabalho a ser feito neste momento está na identificação dos atores e na elaboração da descrição detalhada de cada caso de uso.

Modelagem Conceitual da Aplicação

Com os requisitos do sistema bem definidos, casos de uso especificados e um entendimento preliminar dos conceitos do domínio do problema estabelecido, é possível avançar para a proposição do modelo conceitual final da aplicação.

O modelo conceitual preliminar é, então, refinado. Tipicamente, atributos são adicionados e associações revisadas.

Atenção especial deve ser dada às multiplicidades, especialmente quando ontologias deram origem ao modelo conceitual preliminar, dado que, em geral, ontologias utilizam-se da idéia de compromissos ontológicos mínimos [15], ou seja, procuram dar uma visão mais abrangente acerca do domínio que modelam e tendem a impor menos restrições que uma aplicação específica.

Padrões de análise podem ser reutilizados, quando uma porção do modelo estiver sendo refinada e se detectar a ocorrência de um problema para o qual há um padrão de solução catalogado.

Além disso, novas classes e associações podem ser necessárias para tratar questões específicas da aplicação, descritas nos casos de uso. Um dicionário de dados do projeto deve ser elaborado e as definições dos termos da ontologia podem ser reutilizadas.

Modelagem Comportamental da Aplicação

Para finalizar a fase de modelagem de análise, assim como em qualquer processo de ER, modelos comportamentais da aplicação, tais como diagramas de estados e diagramas de interação, devem ser elaborados quando necessário.

Documentação de Requisitos

A documentação dos requisitos deve ocorrer ao longo de todo o processo de Engenharia de Requisitos. Dois documentos principais devem ser elaborados: o Documento Preliminar de Especificação de Requisitos, produto da primeira fase de levantamento e modelagem de requisitos, e o Documento de Especificação de Requisitos, que contém, além de uma descrição do problema e listas de requisitos, os diversos diagramas produzidos ao longo do processo de ER e um dicionário de dados do projeto.

Verificação e Validação de Requisitos

A atividade de Verificação e Validação (V&V) de requisitos se mantém praticamente inalterada em relação a um processo convencional de ER. A verificação se ocupa de checar se um documento de requisitos está sendo construído de forma correta, de acordo com padrões previamente definidos pela

organização, sem conter requisitos ambíguos, incompletos, incoerentes ou impossíveis de serem testados. A validação, por sua vez, avalia se os requisitos e modelos documentados atendem às reais necessidades e requisitos dos usuários / clientes.

Vale destacar a importância de se verificar se efetivamente as classes derivadas de conceitos de uma ontologia mantêm a mesma semântica em relação à conceituação por ela estabelecida, isto é, se os termos adotados são consistentes e se as restrições são respeitadas.

Gerência de Requisitos

Assim como a V&V, a gerência de requisitos se mantém com poucas alterações em relação a um processo convencional de ER. Deve-se destacar o fato de ser criado mais um nível de rastreabilidade em relação à origem de requisitos e de classes, uma vez que esses elementos podem ser derivados de ontologias e padrões de análise. Dessa maneira, devem-se manter ligações explícitas para esses artefatos.

4. Aplicação do Processo Proposto: Um Estudo de Caso

O processo proposto foi aplicado em um estudo de caso realizado com três grupos em uma disciplina no Mestrado em Informática da Universidade Federal do Espírito Santo. O problema proposto foi comum aos três grupos – alocação de recursos humanos a projetos em organizações de software – mas em cada grupo havia um representante de uma organização de software real e, assim, cada grupo tinha de levar em consideração os requisitos de um sistema para atender a essa organização específica.

Os seguintes artefatos foram disponibilizados para reuso: uma ontologia para modelagem de corporações [16], duas ontologias de organizações de software ([17] e [18]), o catálogo de padrões de análise de Fowler [7] e os padrões de análise definidos em [16].

Todos os três grupos seguiram o processo proposto, cabendo destacar que os correspondentes Documentos Preliminares de Especificação de Requisitos, nos quais são listados os requisitos funcionais e não funcionais da aplicação obtidos no Levantamento Preliminar de Requisitos, estavam bastante distintos uns dos outros.

No passo seguinte, Modelagem Conceitual Preliminar, todos os grupos reutilizaram preferencialmente ontologias. Os grupos 1 e 2 focaram basicamente na ontologia para modelagem de corporações proposta por Cota [16], enquanto o grupo 3 preferiu construir seu modelo conceitual preliminar com base na ontologia de organizações de software

proposta em [18]. Esse terceiro grupo utilizou também o padrão de análise de alocação de recursos proposto em [16].

Em todos os casos, a confecção do modelo conceitual preliminar consistiu da extração de partes de uma ontologia com a qual a aplicação mais se comprometia em relação à conceituação estabelecida, sendo feitos pequenos ajustes para adequação aos requisitos específicos da aplicação, algumas vezes usando outras ontologias e padrões de análise (como no caso do grupo 3).

Perguntados sobre a preferência pelo reuso de ontologias, todos os grupos foram enfáticos ao afirmar que as ontologias eram bastante abrangentes, davam uma visão geral do domínio e estabeleciam um vocabulário bem definido a ser usado, o que permitiu facilmente extrair um modelo preliminar de classes. A ontologia de organizações de software proposta em [17] foi preterida em relação às demais, exatamente por sua apresentação ser muito fragmentada. O mesmo foi dito em relação aos padrões de análise, ainda que todos os grupos tenham apontado que a sua análise tenha ajudado de alguma forma na concepção dos respectivos modelos conceituais preliminares e, em alguns casos, mais aplicados na modelagem conceitual final da aplicação. Ainda no que diz respeito aos padrões de análise, em relação aos propostos por Fowler [7], houve comentários apontando que o vocabulário utilizado não era muito intuitivo, o que dificultava a sua aplicação. Esse problema não foi sentido em relação aos padrões de análise propostos em [16], tendo em vista que a autora utiliza os mesmos termos empregados em sua ontologia para definir os padrões de análise.

Em relação ao Refinamento do Levantamento de Requisitos, o fato de haver disponíveis um Documento Preliminar de Especificação de Requisitos e um modelo conceitual preliminar foi apontado como útil para a identificação de casos de uso. Contudo, em relação à elaboração das descrições de casos de uso, foi relatado que praticamente não houve influência da abordagem no esforço despendido. Ou seja, o processo correu como um processo convencional, sem reutilização.

Na etapa de Modelagem Conceitual da Aplicação, segundo os relatos dos três grupos, o trabalho foi bastante simplificado, com destaque para a elaboração do dicionário de dados do projeto. O trabalho foi concentrado em ajustes de multiplicidades de associações e na definição de atributos. Alguns dos grupos, inclusive, voltaram a inspecionar as ontologias e padrões de análise, sendo que dois grupos foram buscar nos padrões de análise atributos para as classes.

Novas classes e associações foram detectadas por dois grupos (uma classe e uma associação em um deles, apenas uma classe em outro). Por fim, dado que todos os grupos se apoiaram substancialmente em ontologias para a construção de seus modelos conceituais, a elaboração dos respectivos dicionários de dados foi conduzida, basicamente, reutilizando-se definições de termos das ontologias.

Deste ponto do desenvolvimento em diante, tendo em vista que o processo segue uma abordagem bastante próxima da convencional, pouco há a relatar. Cabe registrar apenas um comentário importante acerca da verificação. O grupo 3 relatou que, durante a verificação, foi detectado que alguns conceitos estavam sendo utilizados com uma semântica diferente da proposta pela ontologia. Isso decorria de um exame superficial da ontologia (apenas o modelo gráfico estava sendo analisado em um primeiro momento).

A seguir, um relato mais detalhado do trabalho realizado pelo grupo 3 é apresentado, focando nas atividades de Levantamento Preliminar de Requisitos, Modelagem Conceitual Preliminar, Refinamento do Levantamento de Requisitos e Modelagem Conceitual da Aplicação. Um breve comentário é feito, também, sobre a Verificação e Validação de Requisitos (V&V) e sobre a Gerência de Requisitos, apresentando parcialmente o relatório de rastreabilidade do projeto.

4.1. Estudo de Caso Detalhado

O grupo 3 elaborou a especificação de requisitos para o problema da alocação de recursos no contexto da Acropolis Soluções em Tecnologia da Informação, uma organização de desenvolvimento de software que conta com aproximadamente 50 colaboradores.

A Acropolis possui diferentes equipes para atender seus clientes, sendo o principal deles, uma companhia siderúrgica. Para essa companhia siderúrgica, há uma equipe de desenvolvimento de novos sistemas e outras duas equipes responsáveis por manutenções para áreas específicas (Sistemas Administrativos e Sistemas de Produção). Membros de uma equipe podem ser alocados para projetos de outras equipes. Dessa forma, gerentes de projeto precisam conhecer a alocação dos recursos nos vários projetos.

Levantamento Preliminar de Requisitos

O levantamento preliminar de requisitos do sistema proposto foi feito com base em entrevistas com os gerentes das equipes, tendo sido identificados requisitos funcionais e não-funcionais, parcialmente descritos na Tabela 1. Vale ressaltar que o sistema é responsável apenas pela alocação dos colaboradores. O

cadastro de colaboradores e o gerenciamento dos processos são realizados por outros sistemas com os quais o sistema proposto deve ser integrado (RNF02).

Tabela 1 – Tabela Parcial de Requisitos.

RF001 - O sistema deve apoiar a alocação de colaboradores a projetos de acordo com a equipe em que cada colaborador se encontra.
RF002 - O sistema deve permitir que o gerente de uma equipe solicite a alocação de um colaborador de outra equipe.
RF003 - Um colaborador deve ser alocado com base em suas competências e papéis que pode assumir, e as alocações devem ser feitas para atividades de um projeto.
RNF01 - O sistema deve rodar em plataforma <i>Web</i> .
RNF02 - O sistema necessita de uma interface com os sistemas de cadastro de colaboradores e de controle de processos já existentes.
RNF03 - O sistema deve garantir as informações de seus usuários, controlando o acesso ao mesmo.

Modelagem Conceitual Preliminar

Levantados os requisitos iniciais do sistema, partiu-se para a inspeção dos artefatos disponíveis para reuso. O modelo de classes preliminar do sistema foi desenvolvido, baseando-se, sobretudo, na ontologia organizacional descrita em [18] e utilizando o padrão de alocação de recursos proposto em [16]. A Figura 2 apresenta parte da ontologia organizacional usada, enquanto a Figura 3 apresenta o modelo conceitual preliminar derivado. Vale destacar que essa ontologia organizacional foi desenvolvida de forma integrada a uma ontologia de processo de software [19]. Assim, os conceitos da primeira estão destacados.

Como se pode notar a partir dos modelos mostrados nas figuras 2 e 3, as classes *Atividade*, *Competencia*, *Cargo*, *Projeto*, *Equipe*, *Pessoa* e *Acumulo*, e as respectivas associações, tiveram origem nos conceitos homônimos da ontologia proposta em [18]. Diversos conceitos e relações foram desconsiderados, dado que não eram relevantes para o sistema em questão, tais como *Organização* e *Unidade Organizacional* e suas respectivas relações. No que diz respeito à hierarquia de *Processo de Software* apenas uma classe para tratar processos de projeto era relevante para o sistema em questão, dando origem à classe *Processo*. Por fim, identificou-se a necessidade da classe associativa *Alocacao* a partir da inspeção do padrão de análise “Alocação de Recursos” proposto em [16].

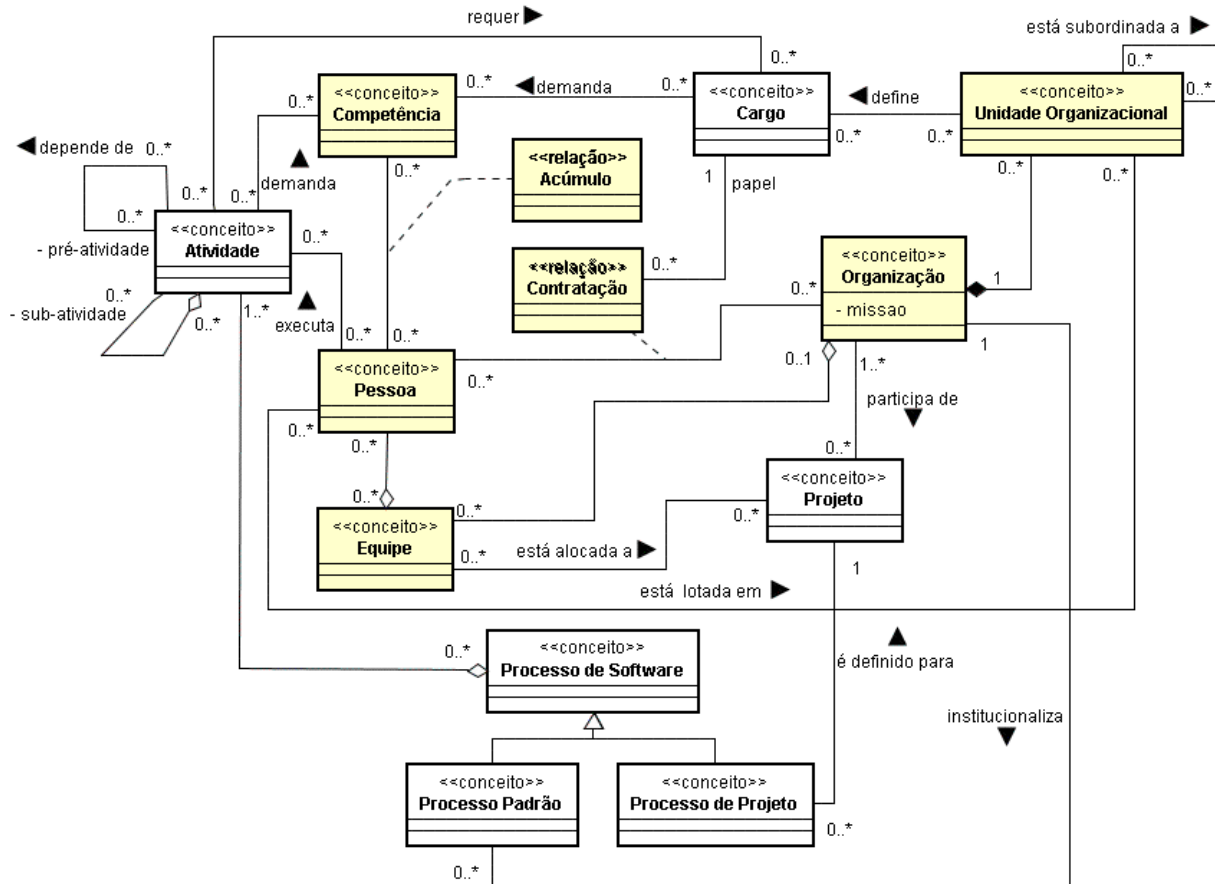


Figura 2 - Ontologia de Organizações de Software – Estrutura Organizacional [18]

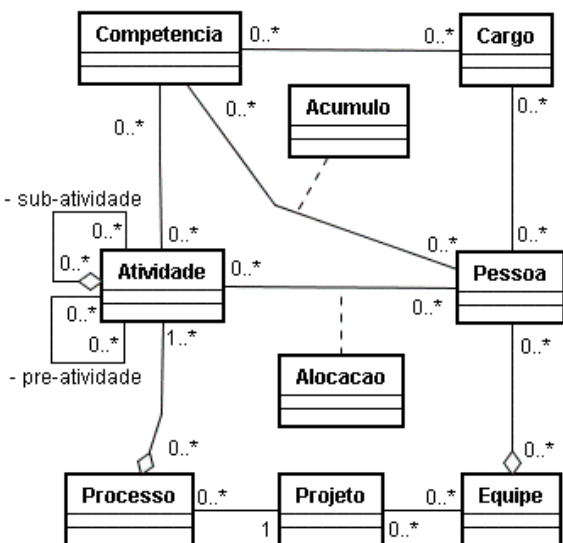


Figura 3 – Modelo conceitual preliminar

A maior parte das classes e associações foi derivada da ontologia proposta em [18], pois seus conceitos tinham definições e relações muito semelhantes aos identificados no contexto da Acropolis. Havia, ainda, similaridades com os modelos dos sistemas de cadastro de colaboradores e de gerência de projetos já existentes, aos quais o sistema proposto deverá ser integrado.

Refinamento do Levantamento de Requisitos

Com os requisitos levantados e definido um modelo preliminar de classes, a elaboração do modelo de casos de uso foi facilitada. As principais funcionalidades do sistema já estavam identificadas na lista de requisitos funcionais (Tabela 1). Além disso, o modelo de classes preliminar guiou a identificação de casos de uso custodiais. Assim, bastou identificar os atores e fazer a descrição dos casos de uso. A Figura 4 mostra o diagrama de casos de uso parcial do sistema proposto.

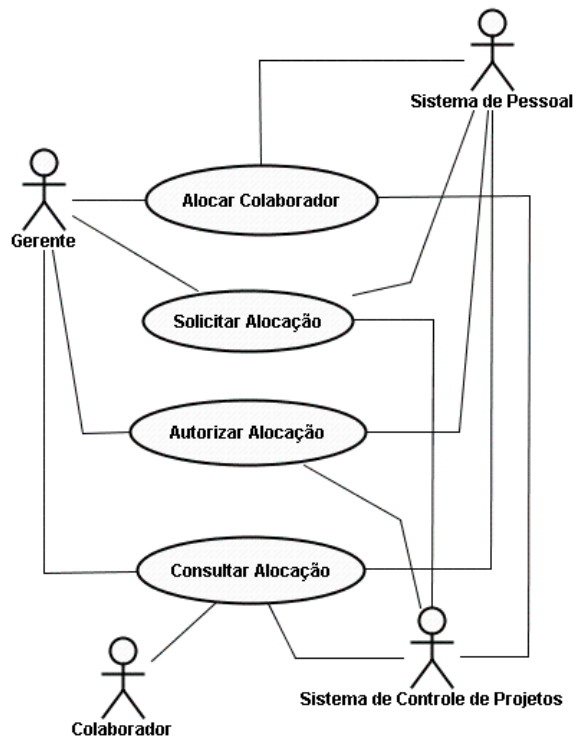


Figura 4 – Diagrama de Casos de Uso

Modelagem Conceitual da Aplicação

De posse do modelo preliminar, foi realizada uma análise mais detalhada dos requisitos, tomando por base as descrições dos casos de uso, e algumas alterações nesse modelo foram necessárias, dando origem ao modelo da Figura 5.

Confrontado requisitos e o modelo preliminar, identificou-se a necessidade de incluir mais uma classe, *Usuario*, para atender ao requisito RNF003.

Atributos foram adicionados às diversas classes, sendo que alguns deles estavam descritos na ontologia e no padrão utilizados como base, no caso deste último, os atributos da classe associativa *Alocacao*.

Algumas multiplicidades tiveram de ser alteradas para ficar em linha com as regras de negócio da Acropolis. Dado que na organização em questão cada projeto tem apenas uma única equipe associada, a multiplicidade entre *Projeto* e *Equipe* foi alterada na extremidade próxima à *Equipe*. Situações análogas ocorreram nas associações de agregação entre atividades (sub-atividade) e entre atividades e processos. Na Acropolis, uma atividade só pode ser parte de, no máximo, uma outra atividade ou processo, sendo que, neste último caso, toda atividade é parte de um processo. Além disso, ao se montar uma equipe, pelo menos um colaborador deve ser informado, o que fez com que fosse alterada a multiplicidade da

associação entre *Pessoa* e *Equipe* na extremidade próxima à *Pessoa*.

Por fim, para atender aos casos de uso relacionados à definição de alocações, cujo ator é o gerente de projeto, foi necessária a identificação do papel *gerente* de uma equipe. Para tal, houve a inclusão de uma nova associação entre as classes *Pessoa* e *Equipe*, pela qual é indicado quem é o gerente de uma determinada equipe.

Vale ressaltar, ainda, que a construção do dicionário de dados do projeto foi facilitada, pois grande parte das definições dos termos da ontologia foi diretamente utilizada para descrever classes e atributos.

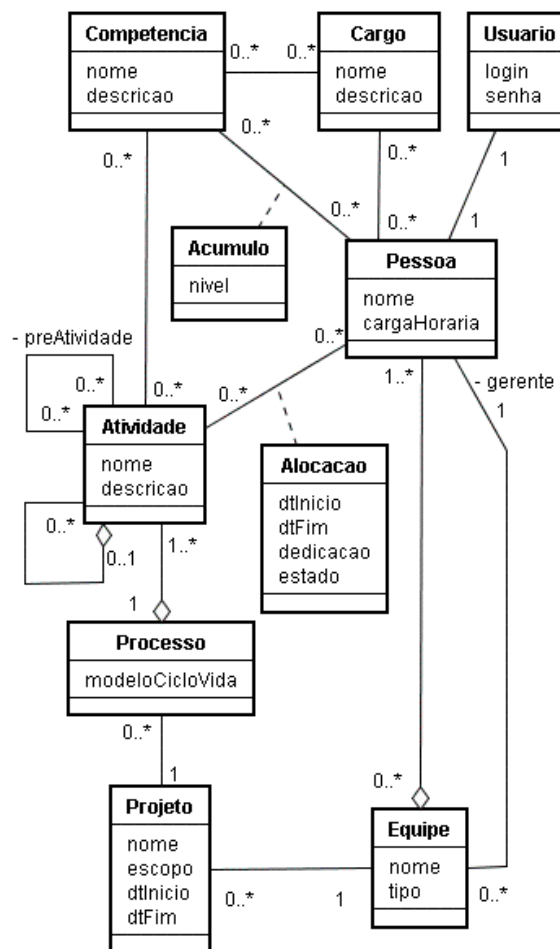


Figura 5 – Modelo conceitual da aplicação

V&V e Gerência de Requisitos

Em relação à Verificação e Validação (V&V) de Requisitos, foi muito importante verificar se efetivamente as classes derivadas de conceitos da ontologia mantinham a mesma semântica em relação à conceituação estabelecida por esta última e se os

termos adotados eram consistentes. Por ocasião da V&V do modelo preliminar de requisitos, detectou-se, por exemplo, que havia má interpretação de alguns conceitos e mudanças no modelo foram realizadas.

No que se refere à Gerência de Requisitos, mais especificamente em relação à rastreabilidade, foram adicionadas informações sobre a origem das classes do modelo de análise.

Em uma abordagem convencional, um relatório de rastreabilidade, normalmente, registra, para cada requisito, requisitos dependentes, conflitantes, casos de uso e pacotes ou classes necessários para tratar o requisito. Seguindo a abordagem de reúso adotada, foi informada, ainda, a origem de uma determinada classe, quando pertinente. A Figura 6 mostra parte do Relatório de Rastreabilidade gerado no projeto.

5. Conclusões e Perspectivas Futuras

A reutilização pode trazer diversos benefícios no desenvolvimento de software, especialmente se praticada em níveis mais altos de abstração, como no processo de Engenharia de Requisitos (ER). Contudo, diversos fatores como, por exemplo, a falta de definição de um processo visando à reutilização e de ferramentas que apoiem tal processo, têm inibido uma reutilização mais efetiva. Visando a contribuir para a institucionalização de práticas de reúso na ER, este trabalho apresentou um processo de ER baseado na reutilização de ontologias e padrões de análise.

De sua aplicação inicial, conforme relatos dos envolvidos, pode-se verificar benefícios do reúso, com destaque para a elaboração de modelos conceituais apoiados, sobretudo, por conceituações explicitadas em ontologias.

Dentre os pontos fracos detectados estão a falta de um apoio mais efetivo ao reúso na modelagem de casos de uso e a dependência de uma seleção prévia de

modelos para reúso. Em relação ao primeiro ponto, espera-se avançar na pesquisa para estabelecer meios de se utilizar ontologias de tarefas na derivação de casos de uso e suas descrições. No segundo caso, estamos cientes de que esse é um dos grandes problemas da reutilização como um todo: a seleção de itens a serem reutilizados. Acreditamos que por meio de técnicas e ferramentas de gerência de conhecimento é possível dar um passo à frente para a solução deste problema e estamos trabalhando em um ambiente integrado de apoio à ER capaz de sugerir proativamente ontologias e padrões de análise a serem reutilizados. Vale destacar que, ainda que nosso foco neste artigo esteja no desenvolvimento *com* reúso, sabemos que ele só é possível com um desenvolvimento *para* reúso. Assim, um ambiente de apoio à ER deve considerar as duas perspectivas.

6. Agradecimentos

Este trabalho foi realizado com o apoio do CNPq e da CAPES, entidades do Governo Brasileiro dedicadas ao desenvolvimento científico e tecnológico, e da FAPES, Fundação de Apoio à Ciência e Tecnologia do Espírito Santo. Agradecemos, ainda, à Acropolis Soluções em Tecnologia da Informação por participar do estudo e permitir sua divulgação.

7. Referências Bibliográficas

- [1] Gimenes, I.M.S., Huzita, E.H.M. (2005) *Desenvolvimento Baseado em Componentes: Conceitos e Técnicas*. Editora Ciência Moderna.
- [2] Hofmann, H.F., Lehner, F. (2001) "Requirements Engineering as a Success Factor in Software Projects", IEEE Software, July/August.

RF001 - O sistema deve apoiar a alocação de colaboradores a projetos de desenvolvimento e manutenção de acordo com a equipe em que cada colaborador se encontra.				
Dependências	Conflitos	Caso de Uso	Classes	Origem da Classe
RF004 RF005 RF008		Alocar Colaborador	Atividade	Conceito Atividade - Ontologia de Processo [19]
			Alocacao	Classe Associativa Alocacao - Padrão "Alocação de Recurso" [16]
			Pessoa	Conceito Pessoa - Ontologia de Organização [18]
			Projeto	Conceito Projeto - Ontologia de Organização [18]
			Processo	Conceito Processo de Projeto - Ontologia de Processo [19]
			Equipe	Conceito Equipe - Ontologia de Organização [18]

Figura 6 – Parte do Relatório de Rastreabilidade do Projeto

- [3] Zave, P. (1997) Classification of research efforts in requirements engineering. *ACM Computing Surveys Journal*, vol. 29, n. 4, p. 315-321.
- [4] Kotonya, G., Sommerville, I. (1998) *Requirements engineering: processes and techniques*. Chichester, England: John Wiley.
- [5] Lamsweerde, A. V. (2000) Requirements engineering in the year 00: a research perspective. *Proceedings of the 22nd International Conference on Software Engineering - ICSE'2000*, Limerick, Ireland. p. 5-19.
- [6] Werner, C.M.L., Braga, R.M.M. (2005) A Engenharia de Domínio e o Desenvolvimento Baseado em Componentes. In: Gimenes, I.M.S., Huzita, E.H.M. *Desenvolvimento Baseado em Componentes: Conceitos e Técnicas*. Editora Ciência Moderna.
- [7] Fowler, M. (1997) *Analysis Patterns: Reusable Object Models*. Addison-Wesley Professional Computing Series.
- [8] Robertson, S., Robertson, J. (1999), *Mastering the Requirements Process*, Addison Wesley, ACM Press.
- [9] Devedzic, V. (1999) "Ontologies: Borrowing from Software Patterns". *Intelligence*, vol. 10, issue 3 (Fall 1999), p. 14-24.
- [10] Falbo, R.A., Guizzardi, G., Duarte, K.C. (2002) "An Ontological Approach to Domain Engineering". *Proceedings of the 14th International Conference on Software Engineering and Knowledge Engineering, SEKE'2002*, Ischia, Italy, p. 351- 358.
- [11] Noy, N.F., Hafner, C.D. (1997) The State of Art in Ontology Design: A Survey and Comparative Review. *AI Magazine*.
- [12] Cima, A. M., Werner, C. M. L. (1997) A Reutilização de Conhecimento Abstrato e a Análise de Domínio. Relatório Técnico ES-432/97 – Engenharia da Sistemas e Computação, COPPE, Universidade Federal do Rio de Janeiro.
- [13] Guizzardi, G. (2000) Desenvolvimento para e com Reuso: Um Estudo de Caso no Domínio de Vídeo sob Demanda. Dissertação (Mestrado em Informática), Universidade Federal do Espírito Santo (UFES).
- [14] Nuseibeh, B., Easterbrook, S. (2000), "Requirements Engineering: A Roadmap", In: *Proc. of the Future of Software Engineering, ICSE'2000*, Ireland, p. 37-46.
- [15] Gruber, T.R. (1995) Toward principles for the design of ontologies used for knowledge sharing. *Int. Journal Human-Computer Studies*, 43(5/6), p. 907-928.
- [16] Cota, R.I. (2003) Um Estudo sobre o Uso de Ontologias e Padrões de Análise na Modelagem de Sistemas de Gestão Empresarial. Dissertação de Mestrado, Mestrado em Informática, UFES.
- [17] Lima, K.V.C. (2004) Definição e Construção de Ambientes de Desenvolvimento de Software Orientados a Organização. Tese de Doutorado, COPPE/UFRJ.
- [18] Ruy, F.B. (2006) Semântica em um Ambiente de Desenvolvimento de Software. Dissertação de Mestrado, Mestrado em Informática, UFES.
- [19] Bertollo, G. (2006) Definição de Processos em um Ambiente de Desenvolvimento de Software. Dissertação de Mestrado, Mestrado em Informática, UFES.