

## **Reuso de Processos de Software Baseado na Componentização de Processos e Conhecimento**

**André Luiz Peron Martins Lanna<sup>1</sup>**  
**Carlos Alberto Marques Pietrobon<sup>1,2</sup> (Orientador)**

<sup>1</sup> PPGEE – Programa de Pós-Graduação em Engenharia Elétrica  
PUCMG – Pontifícia Universidade Católica de Minas Gerais  
Av. Dom José Gaspar, 500 Prédio 03 - Sala 218  
Belo Horizonte – MG, CEP: 30535-901

<sup>2</sup> DECOM – Departamento de Computação  
ICEB – Instituto de Ciências Exatas e Biológicas  
UFOP – Universidade Federal de Ouro Preto  
Campus Universitário – Morro do Cruzeiro  
Ouro Preto – MG, CEP: 35400-000

andre.lanna@gmail.com, capietro@pucminas.br

**Abstract.** *Software reuse can be considered an impact factor on software quality and does not cover only work products, but also software process representation and its inherent knowledge. This paper presents the adoption of Component-Based Development (a reuse method typically applied on software artifacts) on software processes as a mean to improve software quality. This reuse method considers the technical, organizational and human aspects of a software process, but mainly the knowledge acquired on its past executions. Reusing known process that were previously executed tends to improve process quality as it become more stable. Therefore this method can be efficient for new software process definition, which has higher quality and stability.*

**Resumo.** *O reuso de processos de software pode ser considerado como um fator de impacto para a qualidade de software por utilizar maneiras de desenvolvimento já testadas e consagradas na organização desenvolvedora. Este trabalho apresenta uma proposta de reuso de processos inspirada no Desenvolvimento Baseado em Componentes de software e que faz uso de conhecimento para sua realização. Essa proposta considera os aspectos técnicos, organizacionais e humanos de um processo de software, mas também o conhecimento adquirido em suas execuções passadas. Reutilizar processos previamente executados (e portanto conhecidos) tende a aumentar a qualidade do novo processo à medida em que este se torna mais estável. Espera-se que este método de reuso seja eficaz para definições de novos processos de software.*

### **1. Introdução**

Atualmente a qualidade do produto de software tem sido um assunto de grande interesse tanto para a área acadêmica quanto para as organizações desenvolvedoras de software.

Diferentes abordagens têm sido propostas de modo a permitir que tais produtos possam atingir níveis mais elevados de qualidade. Algumas delas buscam garantir a qualidade ao focar o produto que está sendo desenvolvido, *i.e.*, o software. Dentre elas destaca-se o reuso de software, através do qual (geralmente) artefatos previamente desenvolvidos, testados e integrados em outros software são utilizados como base para o desenvolvimento do novo software. A utilização desta técnica oferece vantagens tais como diminuição dos custos de produção e a rápida entrega dos produtos [Sommerville 2007, Li et al. 2007].

Outras abordagens buscam aumentar a qualidade do software ao realizar melhorias em seu processo de desenvolvimento. Entende-se por um processo<sup>1</sup> com qualidade aquele capaz de verificar durante todas as fases do desenvolvimento se os padrões de qualidade estabelecidos pela organização serão satisfeitos [Sommerville 2007]. Nestas abordagens acredita-se que processos que possuem qualidade geram produtos com qualidade, o que indica uma clara relação entre produto e processo. Entretanto definir um processo de software é uma tarefa difícil de ser realizada, uma vez que não devem ser considerados apenas os fatores técnicos de um processo mas também os organizacionais e humanos [Lanna 2008]. Tais aspectos devem ser dispostos de modo harmônico de modo a atender às necessidades da organização. Por este motivo esta tarefa geralmente é realizada por pessoas mais experientes, com conhecimentos profundos dos processos organizacionais, na tentativa de definir processos mais estáveis e de qualidade [Barreto et al. 2008, Barreto et al. 2009]. Mesmo assim, não se pode garantir que o novo processo será de qualidade e definido dentro dos custos e prazos estabelecidos pela organização.

As duas abordagens para qualidade de software apresentadas (foco no produto e no processo) apresentam resultados positivos quando efetivamente implantadas em uma organização. Entretanto atualmente elas ocorrem de modo separado e seus resultados podem ser potencializados se forem consideradas em conjunto [Lucrédio et al 2008]. Osterweil em [Osterweil 1987, Osterweil 1997] indica similaridades entre software e processos ao afirmar que é possível definir requisitos, projetar, implementar e evoluir processos de software, tal como ocorre no desenvolvimento de software. De acordo com o autor tais semelhanças permitem que técnicas geralmente aplicadas em artefatos de software sejam adaptadas e aplicadas em processos de software, incluindo aquelas relacionadas ao reuso de software.

O objetivo principal deste trabalho é uma proposta para reutilizar processos de software junto com seus conhecimentos adquiridos em execuções passadas como uma alternativa para a definição de novos processos mais maduros e estáveis em menores tempo e custos. A abordagem foi estabelecida a partir da técnica tradicional de reuso denominada Desenvolvimento Baseado em Componentes (DBC). Para esta proposta foram estabelecidos os seguintes objetivos específicos:

- a definição de uma estrutura reutilizável (componente de processo) capaz de representar processos de software em seus diferentes aspectos e armazenar o conhecimento adquirido em suas execuções passadas,
- a descrição do processo de software e seus conhecimentos de modo a facilitar o seu entendimento e futuras reutilizações,
- a definição e descrição dos serviços prestados pelo componentes de processo, de

---

1 Neste texto os termos “processo de software”, “processo de desenvolvimento de software” e simplesmente “processo” possuem o mesmo significado.

modo a permitir que eles possam ser interligados para a composição (e consequente definição) de novos processos de software.

Este texto está estruturado como segue. Na seção 2 são apresentados os principais temas abordados neste trabalho: processos de software e reuso de software. A seção 3 apresenta a proposta de reutilização de processos de software, definindo inicialmente um modelo de processos, seguida pela definição do componente de processo reutilizável. Posteriormente é apresentado o modo como o processo de software e seu conhecimento são descritos de modo a facilitar suas futuras reutilizações. Por fim a seção demonstra o modo de reutilização destes componentes durante a definição de novos processos, bem como o apoio ferramental criado para auxiliar esta tarefa. A seção 4 apresenta as contribuições da proposta de reuso e os trabalhos a serem desenvolvidos futuramente.

## 2. Fundamentação Teórica

### 2.1. Processos de software

Um processo de software é um conjunto de políticas, tecnologias e atividades executadas por diferentes papéis em uma organização para produzir ou manter um produto de software e seus artefatos [Fuggetta 2000]. São processos intelectuais e criativos (ao contrário dos processos produtivos tradicionais) e que dependem do julgamento humano, cujo resultado é a integração de conhecimentos coletados, destilados e organizados à medida que o processo é conduzido [Sommerville 2007, Pressman 2006].

Processos devem ser considerados sob três aspectos: técnico, organizacional e humano [Lanna 2008]. Pelo aspecto técnico entende-se tudo aquilo relacionado ao modo como o software é desenvolvido por uma organização, os artefatos de software requeridos e produzidos em cada fase do desenvolvimento, ferramentas empregadas, dentre tantos outros. Este aspecto pode ser subdividido em unidades menores (em ordem decrescente de tamanho): subprocessos, atividades e tarefas. Uma unidade nesta classificação é definida com base em unidades que estão no nível logo abaixo (ex.: um processo é definido através de subprocessos, que podem ser definidos através de algumas atividades e assim sucessivamente). O aspecto organizacional descreve conceitos definidos pela organização tais como papéis desempenhados pelos *stakeholders*, restrições impostas ao projeto, políticas e metas organizacionais, entre outros. Por fim o aspecto humano está relacionado aos recursos humanos que a organização dispõe para um processo. Ele pode variar substancialmente entre organizações devido ao fato de cada equipe ser formada por diferentes membros. Um conjunto inicial para tal aspecto foi definido em [Lanna 2008] como sendo: habilidades individuais, conhecimento técnico e experiências.

Uma importante característica de qualquer processo de software é o fato de ele possuir, produzir e utilizar muito conhecimento durante sua execução. [Xu 2005] e [Borges 2002] apresentam a importância e os benefícios em utilizar o conhecimento adquirido em execuções passadas durante a definição e adaptação de novos processos de software. De acordo com [Montoni 2003] existem diferentes tipos de conhecimento como por exemplo descrições de processo, casos, lições aprendidas, idéias, dúvidas, dentre outros. Esse conhecimento possui grande valor para a organização pois pode ajudar a solucionar dúvidas ou problemas ocorridos durante a execução do processo de software, donde a importância de serem armazenados explicitamente para que estejam disponíveis aos membros de uma organização. Por fim o autor afirma que a qualidade

do software pode ser aumentada à medida em que os membros de uma organização aumentam seu conhecimento e entendimento sobre o processo de software com o qual estão envolvidos.

Diversos modelos para representação de processos de software têm sido propostos através de linguagens de modelagem de processo (do inglês *Process Modelling Language – PML*) [Fiorini 2001]. Dentre elas destaca-se o metamodelo SPEM (mantido pelo consórcio W3C) que tenta estabelecer-se como um padrão para representação de processos, o que permitiria sua utilização em qualquer ambiente e ferramenta de execução de processo [OMG 2008].

## 2.2. Reuso de software

Reuso de software é a utilização de quaisquer artefatos de software em todo os estágios de desenvolvimento de software para o desenvolvimento de novas aplicações [Kim 1998]. Seu objetivo é aumentar a qualidade do software à medida em que o novo software é produzido com base em artefatos previamente desenvolvidos, testados, integrados com sucesso em outras aplicações e possivelmente melhorados para utilizações futuras [Krueger 1992]. Organizações de software têm se interessado nesta área devido aos impactos positivos que ela acarreta tais como a redução de custos e do tempo de entrega do software além do aumento da produtividade das equipes [Almeida et al. 2007].

Dentre os métodos de reuso existentes destaca-se o Desenvolvimento Baseado em Componentes (DBC) sendo suas estruturas reutilizáveis denominadas componentes de software. De acordo com [Szyperki 2002, Wang 2009] um componente de software é definido como sendo uma unidade de composição, que representa pedaços de software com requisitos, interfaces e dependências de contexto bem definidas. O termo “unidade de composição” indica que componentes podem ser agrupados de modo a formarem unidades maiores e mais complexas. A importância de suas interfaces decorre do fato de elas serem responsáveis por especificar (informar) os serviços oferecidos pelo componente e realizar as comunicações com os demais componentes que formam o software. Por fim é importante ressaltar que um componente de software não existe por si só. É necessário satisfazer um conjunto de condições, fato conhecido como dependência de contexto, de modo a permitir que o componente possa ser reutilizado com sucesso. Uma destas condições é a existência de um ambiente técnico onde o componente será implantado em sua totalidade (e nunca parcialmente).

A escolha do componente adequado para o reuso em um software nem sempre é uma tarefa simples de ser realizada. Ainda que existam documentações e descrições disponíveis para os componentes candidatos ao reuso, é comum ter dificuldades em entender os serviços providos por eles. Para resolver este problema, propostas como [Ramadour 2008, Wang 2009] tentam aumentar a expressividade da descrição de um componente ao substituírem a utilização de palavras-chave por conceitos que são bem definidos e entendidos em um determinado contexto. Esses conceitos geralmente são estabelecidos através de ontologias<sup>2</sup>. Em [Ramadour 2008] a ontologia é utilizada para representar perfis e necessidades de usuários além de conhecimento semântico acerca do componente, enquanto [Wang 2009] utiliza ontologias para capturar relações e restrições entre

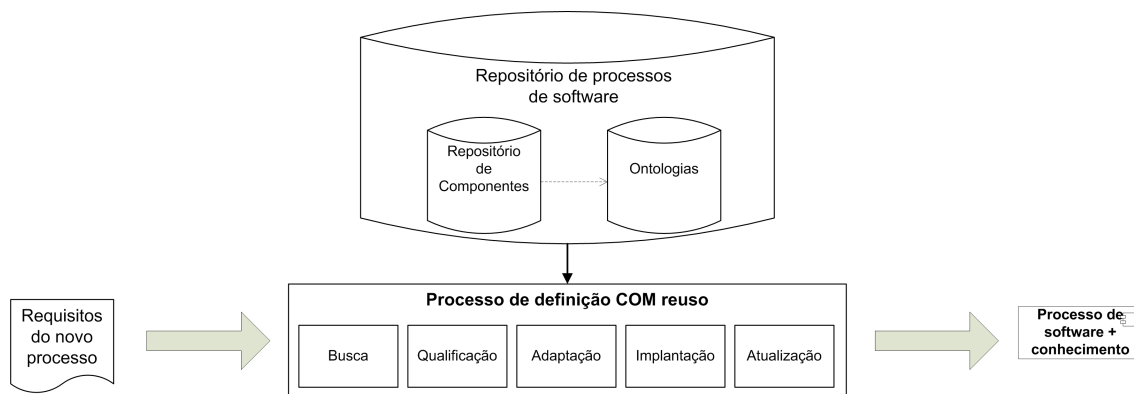
---

2 Especificação de um determinado domínio em que os elementos do mundo real são representados por conceitos. Tais conceitos estão relacionados através de associações de classificação (de modo a formarem estrutura hierárquicas) ou por relacionamentos que estabelecem restrições no entendimento acerca dos conceitos [Lanna 2009].

componentes de software, além de realizar a análise automatizada de compatibilidade do modelo de componentes. Portanto o uso de conceitos definidos por uma ontologia parecer ser uma solução viável para a descrição de componentes de software.

### 3. Abordagem para reutilização de processos de software e conhecimento

A abordagem para reutilização de processos de software foi estabelecida com base nos conceitos relativos a processos de software e no DBC tradicional. Os principais elementos constituintes desta proposta são o componente reutilizável de processo de software, o repositório de componentes de processo e a definição “com” reuso. O modo como eles se relacionam está apresentado na Figura 1.



**Figura 1. Abordagem proposta para a reutilização de processos de software.**

O desenvolvimento desta abordagem foi realizado em quatro etapas iniciando-se pela definição de um modelo de processo de software a ser considerado no restante do trabalho. Posteriormente foi definida a estrutura do componente capaz de representar processos de software aderentes ao modelo pré-estabelecido, seguida pela descrição do componente de processo de software a fim de facilitar seu entendimento e futura reutilização. Por fim foi estabelecido um processo a ser executado para guiar o modo como os componentes de processo são extraídos do repositório e reutilizados na definição do novo processo.

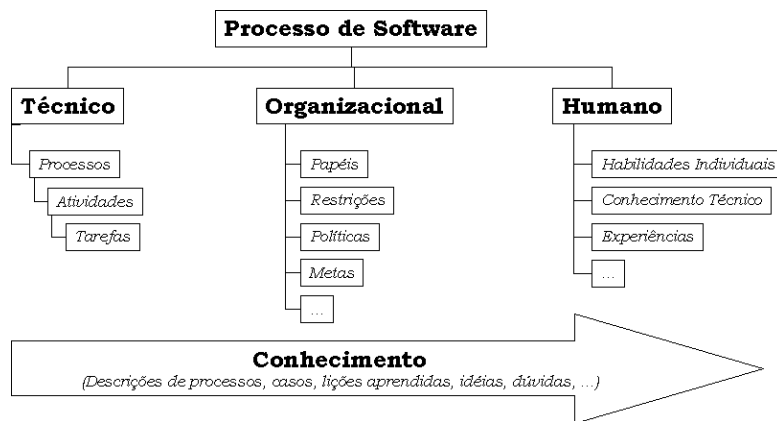
#### 3.1. Modelo de processo de software

Como apresentado na seção anterior um processo de software deve ser definido através dos aspectos técnico, organizacional e humano. Apesar de cada um destes aspectos descrever características diferentes do processo, eles devem ser considerados como sendo dependentes entre si, pois decisões em um dado aspecto podem influenciar os outros dois. Apenas como exemplo, ao definir uma restrição orçamentária para o processo (um constituinte do aspecto organizacional), podem ocorrer limitações na definição do aspecto técnico (tecnologias utilizadas, dentre outros) e nas pessoas que vão formar a equipe (aspecto humano). Portanto durante a definição de um processo de software estes três aspectos devem ser trabalhados em conjunto de modo a harmonizá-los. Além disto considera-se o aspecto técnico como sendo o nível de detalhamento (granularidade) para a representação de um processo, pois ele pode representar todo o processo ou partes dele (atividades e tarefas).

Nesta proposta considera-se que os conhecimentos de um processo de software

servem para descrevê-lo ou registrar situações ocorridas em alguma execução passada. Cada registro de conhecimento em um processo de software é chamado de item de conhecimento e pode ser de diversos tipos. Este trabalho considera os tipos de conhecimento definidos por [Montoni 2003], mas permite que as organizações definam seus próprios tipos de conhecimento. Além disso, um processo de software pode possuir vários itens de conhecimento. O conhecimento também faz parte do processo de software, mas deve ser considerado separadamente dos aspectos previamente mencionados. Essa distinção entre conhecimento e os aspectos do processo se deve ao fato de um item de conhecimento utilizar em sua descrição os elementos definidos por cada aspecto do processo, de modo que um item de conhecimento pode contemplar um, dois ou os três aspectos do processo de software.

Deste modo, estabeleceu-se como modelo de processos de software o conjunto de aspectos técnicos, organizacionais e humanos, além do conjunto de itens de conhecimento, como mostra a figura 2.



**Figura 2. Elementos constituintes do modelo de processo de software: os itens de conhecimento podem contemplar diferentes aspectos simultaneamente.**

### 3.2. Componentes de processo de software

Baseado nas características de componente de software tradicional, foi definida uma estrutura reutilizável capaz de representar um processo de software conforme o modelo de processos estabelecido para este trabalho. Tal estrutura denominada “componente de processo de software” é definida como sendo:

- Unidades de composição com interfaces e dependências de contexto bem definidas.
- Capaz de representar um processo de software em seus três aspectos (técnico, organizacional e humano) seja qual for seu nível de detalhe (processo, atividade ou tarefa).
- Capaz de armazenar os itens de conhecimento relacionados ao processo no próprio componente.

Assim como ocorre em componentes tradicionais de software, por “unidade de composição” entende-se que a partir do agrupamento de componentes de processo serão definidos novos processos de software. A ligação entre os componentes de processo de software ocorre através de suas interfaces tendo estas o papel de realizar a comunicação entre os processos componentizados, principalmente através da troca de artefatos produ-

zidos pelos processos interligados. Por fim a dependência de contexto definida nesta proposta estabelece que os requisitos dos componentes de processo devem ser atendidos para que seus processos possam ser executados. Tais requisitos são considerados em dois níveis: os requisitos do processo e os requisitos do próprio componente. Os requisitos do processo determinam as necessidades a serem supridas pela organização para que o processo possa ser implantado e executado pela sua equipe. Já os requisitos do componente determinam os fatores técnicos a serem atendidos para que o componente possa ser executado, como por exemplo a existência de um ambiente de suporte à execução do processo onde o componente será implantado em sua totalidade.

O componente de processo de software tem tamanho variável conforme o nível de detalhe da representação do seu processo. De acordo com o modelo estabelecido na Figura 1, este nível de detalhe é determinado pelo aspecto técnico. Conseqüentemente a granularidade do componente de processo é determinada pelo aspecto técnico do processo de software. Por fim, este componente é capaz de armazenar os diversos itens de conhecimento relacionados ao processo representado de modo independente dos aspectos que determinam o processo de software (conforme determina o modelo de processos).

A realização (ou implementação) do componente de processo de software consiste em representar processos de software aderentes ao modelo da Figura 1 através de uma linguagem estruturada. Como o modelo permite que cada organização defina seus próprios itens de aspecto (organizacional e humano) e tipos de conhecimento, a implementação do componente de processo deve ser flexível a ponto de conseguir representá-los. Tal implementação foi realizada através da linguagem XML, para a qual foi estabelecida uma *Data-type definition* (DTD) aderente ao modelo de processo. A estrutura genérica para representação de um processo em XML é apresentada pela figura 3.

Note que os quatro grandes conjuntos constituintes do modelo de processo estão representados pelas marcações *TechnicalAspect*, *OrganizationalAspect*, *HumanAspect* e *KnowledgeSet*, sendo cada um destes conjuntos formados por marcações que representam os elementos que, de fato, vão formar o processo de software. São as marcações *TechnicalItem*, *OrganizationalItem* e *HumanItem*, sendo que todas elas possuem *type* como atributo. Este atributo recebe como valor os elementos de cada aspecto definidos no modelo de processo (por exemplo, para o conjunto *OrganizationalAspect*, ele pode assumir os valores *roles*, *constraints*, *goals*, dentre outros). Como cada organização pode definir seus próprios elementos constituintes dos aspectos de um processo de software, a flexibilidade permitida pelo modelo ocorre no componente através da definição (por parte da organização) do valor que o atributo *type* receberá durante a definição de um novo processo.

A implementação do componente através do arquivo XML considera ainda a compatibilidade com o modelo SPEM, devido a sua larga utilização em ferramentas de modelagem e definição de processos. Tal compatibilidade decorre da utilização dos elementos definidos em SPEM nos elementos que constituem o aspecto técnico do componente de processo. Note na figura 3 que o aspecto técnico está considerando a definição de um item técnico pela organização e outro técnico através do SPEM.

Por fim, conforme a definição de componente de processo de software, é necessário que as interfaces dos componentes de processo de software sejam bem definidas de modo a permitir a ligação e comunicação entre os componentes de processo de software.

Tais interfaces devem informar os serviços prestados pelo componente de processo, bem como os recursos demandados e providos por eles. Em se tratando de processos de software tais serviços são os processos, atividade e tarefas definidas para o processo de software e os recursos são os artefatos produzidos e requeridos por tais atividades. Deste modo é necessário que também sejam informados na estrutura do componente de processo os diversos serviços que ele oferece, sendo que para cada um deles deve ser informado se é um serviço provido ou requerido bem como os artefatos utilizados em sua execução. Portanto o XML do componente de processo também especifica as interfaces do componente, seus serviços oferecidos e os artefatos requeridos e produzidos por ele (ver figura 3).

```
<?xml version="1.0" encoding="UTF-8"?>

<ProcessComponent date="" author="" collaborator="">
  <TechnicalAspect granularity="">
    <TechnicalItem id="" name="" type="">Description of item</TechnicalItem>
    <TechnicalItem id="" name="" type="">
      <spem:Process>
        <spem:documentation>Descrição de um item no SPEM
        </spem:documentation>
      </spem:Process>
    </TechnicalItem>
  </TechnicalAspect>

  <OrganizationalAspect>
    <OrganizationalItem id="" name="" type="">Description of
item</OrganizationalItem>
  </OrganizationalAspect>

  <HumanAspect>
    <HumanItem id="" name="" type="">Description of item</HumanItem>
  </HumanAspect>

  <KnowledgeSet>
    <KnowledgeItem type="">Description of item</KnowledgeItem>
  </KnowledgeSet>

  <interface id="" name="" providedRequired="">
    <service id="" name="" numberInputArtifacts="" numberOutputArtifacts="">
      <artifact id="" name="" inputOutput="">Description of
artifact</artifact>
    </service>
  </interface>
</ProcessComponent>
```

**Figura 3: Estrutura genérica definida em XML para representação do componente de processo de software.**

### 3.3. Descrição do componente de processo de software

Os componentes de processo devem ser descritos de modo a facilitar o entendimento dos processos representados e suas futuras reutilizações. Nesta proposta a descrição é realizada de duas maneiras: a descrição do componente de processo e a descrição do elementos que constituem o processo de software.

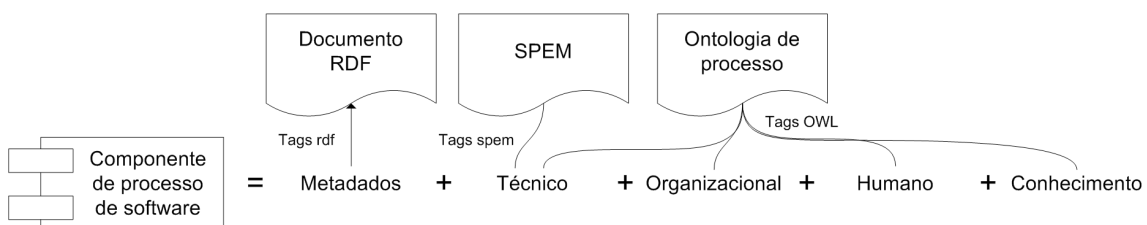
Descrever um componente de processo de software consiste em explicar as características do arquivo XML que representa o processo de software tais como nome do componente, data de criação, responsável pela criação, colaboradores, número de versões, dentre outros. Foram definidos vinte e três elementos que caracterizam um componente processo de software, sendo eles agrupados em cinco categorias:



identificação, utilização, maturidade, documentação e alteração. Cada um desses elementos descritivos é considerado um metadado do componente de processo e são representados nesta proposta através da linguagem RDF (*Resource Description Framework*).

No entanto descrever o processo de software consiste em descrever seus aspectos e conhecimentos utilizando termos e conceitos definidos por alguma ontologia. A utilização da ontologia foi considerada como forma de aumentar o nível de expressão e entendimento acerca do processo. Como ontologias podem ser definidas através da linguagem OWL<sup>3</sup> (uma extensão da linguagem XML), para descrever os aspectos e o conhecimento de processos de software nesta é necessária a utilização dos elementos OWL na descrição do processo de software. Como cada processo de software é executado em ambientes ou organizações diferentes, a descrição dos aspectos e conhecimento não deve se restringir a uma ontologia estabelecida apenas para descrever contextos de processos de software, mas sim absorver os termos definidos pela ontologia adotada pelo processo e considerá-los durante sua descrição

Como se pode notar ainda na figura 2, para cada item que compõe o processo de software (seja ele técnico, organizacional, humano ou de conhecimento), há um espaço próprio para descrição. Esta descrição é realizada ao preencher esse espaço com um texto que explique seu significado. Para aumentar o nível de expressão deste texto (e conseqüentemente o entendimento dos membros da organização acerca daquele item de processo) devem ser incluídos os termos e conceitos definidos pela ontologia adotada pelo processo. Deste modo a descrição de cada item que forma o processo não é realizada com base em palavras-chave, mas sim em conceitos que são bem definidos e entendidos pelos membros da organização, aumentando assim o nível de expressão.



**Figura 4: Descrição do componente de processo de software realizada através das linguagens RDF, OWL e SPEM sendo todas elas representadas no arquivo XML do componente.**

Nesta proposta a utilização de ontologias de processo traz ainda outra vantagem: a descrição dos elementos que compõem a interface do componente de processo (serviços e artefatos) também pode ser realizada através dos conceitos definidos pela ontologia. Deste modo os elementos que formam tais interfaces não são mais descritos apenas através de palavras-chave, mas sim em conceitos bem definidos e conhecidos acerca do processo de software. Este tipo de descrição facilita e aumenta o entendimento acerca dos serviços providos pelo componente.

### 3.4. Definição de processos de software com reuso de processos

De modo semelhante ao desenvolvimento “com” reuso estabelecido para o DBC, foi proposto um método denominado “definição com reuso”, capaz de guiar a

3 *Web Language Ontology*: linguagem para representação de ontologias mantida e recomendada pelo consórcio W3C [Cuenca et. al, 2008].

reutilização de componentes de processo. Ele foi proposto de modo a considerar a descrição do componente de processo pelos metadados, a descrição dos aspectos e conhecimentos do processo pelos termos de ontologia além das especificações das interfaces do componente. A definição com reuso é realizada através das cinco etapas apresentadas na Figura 1: busca, qualificação, adaptação, composição e atualização.

A busca de componentes tem como objetivo pesquisar no repositório os componentes de processo que possivelmente irão atender aos requisitos do processo em definição. Além de extrair os componentes do repositório ela estabelece um nível de equivalência entre a especificação de requisitos do novo processo e as especificações dos componentes vindos do repositório, através de operações denominadas *matching* (ao todo foram definidas quatro operações). O nível de equivalência destas operações é variável, contemplando a equivalência total ou parcial dos requisitos.

A etapa de qualificação de componentes tem como objetivo determinar quais componentes recuperados serão reutilizados na definição do novo processo. Portanto nesta etapa são avaliados detalhadamente os aspectos do processo de software de modo a identificar vantagens e desvantagens em sua reutilização. Essa avaliação é realizada com base nas especificações das interfaces dos componentes, nas implementações dos processos de software (através dos itens de processo que o constituem) e principalmente nos itens de conhecimento que eles possuem. Para a etapa de qualificação são estabelecidos os critérios de inclusão e de classificação. O critério de inclusão determina características que obrigatoriamente o componente de processo deve apresentar para que não seja descartado do processo de reutilização. Aqueles componentes resultantes serão ordenados pelo critério de classificação, de modo a determinar os componentes mais aptos à reutilização. A definição destes critérios fica a cargo do responsável pelo novo processo.

Os componentes resultantes da qualificação serão efetivamente reutilizados no novo processo. Entretanto nem sempre eles são compatíveis entre si de modo que adaptações são necessárias. A etapa de adaptação de componentes é responsável por identificar e resolver inconsistências existentes entre os componentes de processo qualificados, sendo que as inconsistências ocorrem por divergências entre as interfaces dos componentes que serão interligados. Duas soluções são propostas nesta abordagem. Para os casos em que os elementos das interfaces diferem apenas na nomenclatura (tipicamente casos em que os componentes de processo adotam ontologias diferentes), uma proposta de alinhamento das ontologias é utilizada para resolver as inconsistências. Entretanto, em casos que os elementos das interfaces divergentes são essencialmente diferentes, a solução ocorre pela utilização de um componente de processo intermediário (denominado adaptador) capaz de realizar a compatibilização.

Após compatibilizar os componentes que formam o novo processo de software, eles estão prontos para serem implantados no ambiente em que serão executados. Portanto é na etapa de implantação que os componentes de processo serão de fato reutilizados. Esta abordagem considera que implantar componentes de processos consiste em importar para a ferramenta de execução de processos adotada pela organização todos os elementos que representam o processo de software, incluindo as ontologias utilizadas pelo novo processo, os itens constituintes dos aspectos técnico, organizacional e humano além dos itens de conhecimento. Após todos os elementos terem sido importados, o processo está definido e implantado na organização.

Por fim, a etapa de atualização de componentes é responsável pela troca de um componente de processo por um outro de qualidade mais alta e cujos serviços apresentam mais vantagens. Para realizar esta substituição é necessário identificar os requisitos do componente a ser substituído, pesquisar no repositório os componentes que os atendem, escolher dentre os candidatos aquele que melhor irá substituí-lo, realizar as adaptações necessárias para resolver possíveis incompatibilidades e finalmente realizar a implantação do componente. Como se pode notar a atualização consiste em executar novamente as quatro etapas anteriores da definição com reuso. No entanto ela se difere por buscar níveis mais altos de equivalência entre os requisitos dos componentes de processo, de modo a minimizar os possíveis impactos na atualização, no que tange a adaptações e modificações no componente reutilizado.

### 3.5. Apoio ferramental

Um apoio ferramental para a abordagem proposta foi desenvolvido com o objetivo de auxiliar as organizações a definirem seus processos de software através da reutilização de processos já executados e conhecidos pela organização. A ferramenta é capaz de considerar termos definidos por diversas ontologias utilizadas na descrição dos diferentes aspectos e itens de conhecimento de um processo de software. Tal ferramenta consiste do repositório de componentes e ontologias de processo de software e no suporte às etapas da definição “com” reuso. A figura 5 apresenta algumas telas referentes às etapas da definição “com” reuso.

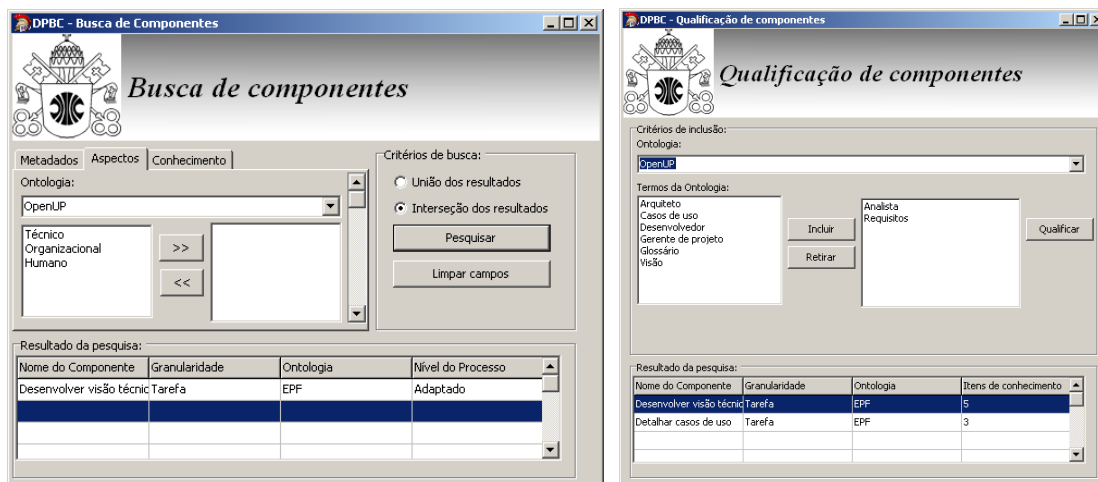


Figura 5: Ferramenta oferecendo suporte à busca e qualificação de componentes.

## 4. Conclusões e trabalhos futuros

Este trabalho apresentou uma proposta para reutilização de processos de software com base nos diferentes aspectos que formam um processo de software e no conhecimento adquirido em suas execuções passadas. Tais aspectos e conhecimentos de um processo são representados em componentes reutilizáveis, os quais são descritos através de metadados e por termos estabelecidos em ontologias, além de serem armazenados em repositório. Foi apresentado também a definição “com” reuso que estabelece um método para definição de processos de software, compreendendo desde a busca do componente no repositório até a implantação no ambiente de execução de processos. Por fim foi proposta uma ferramenta capaz de auxiliar as etapas da definição “com” reuso.

O intuito desta abordagem é aliar as vantagens do reuso de software tradicional,

tais como economia de tempo e custos além do aumento da qualidade aos benefícios oriundos de uma boa definição de processo de software. Deste modo acredita-se que o aumento da qualidade dos novos processos será consequência da reutilização de processos já executados e portanto conhecidos pela organização.

Nesta abordagem a estrutura reutilizável mostrou-se capaz de representar os diferentes aspectos e conhecimentos que formam um processo de software, indiferente da granularidade do processo representado (processo, atividade ou tarefa). Além disto esta estrutura é flexível a ponto de permitir que as organizações definam seus próprios elementos constituintes de processo ou utilizem outros modelos de representação, tal como o SPEM.

A utilização de ontologias na definição “com” reuso permitiu que processos e componentes de processo (principalmente suas interfaces) fossem descritos de um modo mais expressivo se comparados à simples utilização de palavras-chaves. Os termos definidos por uma ontologia trouxeram significados aos elementos constituintes do processo de software, de modo que a definição de novos processos agora ocorre com base na semântica dos termos que o compõem.

Isto exposto pode-se afirmar que este trabalho apresenta como contribuições:

- a definição de um modelo para representação de processos de software com base em diferentes aspectos e conhecimento;
- definição da estrutura de componente de processo reutilizável para representação dos processos;
- definição de formas de descrição voltadas ao componente de processo (metadados) e ao próprio processo (através dos termos definidos pelas ontologias e,
- definição de um método para reutilização de componentes de processo.

Como continuação deste trabalho estão previstos estabelecer a definição “para” reuso, capaz de criar componentes de processo de software “do zero” para popular o repositório de componentes de processo. Além disto a utilização de alguma forma de Gerência de Conhecimento aplicada ao repositório de componentes de processo será de grande valia, principalmente para a definição “com” reuso, de modo a disponibilizar somente os processos cujos conhecimentos possuem importância na definição de um novo processo. Portanto a implantação da Gerência de Conhecimento no repositório é considerado como um trabalho futuro.

Por fim, é necessário considerar a principal limitação deste trabalho que foi sua realização apenas em laboratório, portanto não utilizando processos adotados em projetos reais. Foram utilizados processos definidos e disponibilizados por órgãos como o W3C, e ontologias definidas em laboratório. Espera-se que tal abordagem seja utilizada em empresas desenvolvedoras de software para que seja possível avaliar sua real eficácia e viabilidade.

## 5. Bibliografia

Barreto, A. S., Murta, L. G. P., Rocha, A. R. (2008). Software process definition: a reuse-based approach. In *XXXIV CLEI - Conferencia Latinoamericana de Informatica*, páginas 409–419.

Barreto, A. S., Murta, L. G. P., Rocha, A. R. (2009). Componentizando processos legados de software visando a reutilização de processos. In *SBQS - Simpósio Brasileiro de Qualidade de Software*. Ouro Preto, MG, Brasil.

Borges, L. d. M. S. Falbo, R. d. A. (2002). Managing software process knowledge. In *International Conference on Computer Science, Software Engineering, Information Technology, e-Business, and Applications - CSITeA'02*, páginas 227–232.

Almeida, E. S., Alvaro, A., Garcia, V. C., Mascena, J. C. C. P., de Arruda Bur'egio, V. A., Nascimento, L., Lucrédio, D., Meira, S. L. (2007). *C.R.U.I.S.E. Component Reuse In Software Engineering*. C.E.S.A.R. e-books.

Grau B. C., Horrocks I., Motik B., Parsia B., Patel-Schneider P., Sattler U (2008). OWL 2: The next step for OWL. *Journal Web Semantics: Science, Services and Agents on the World Wide Web*. 6(4):309 – 322.

Fiorini, S. T. (2001). *Arquitetura para reutilização de processos de software*. Tese de Doutorado, PUCRJ, Rio de Janeiro, 2001.

Fuggetta, A. (2000). Software process: a roadmap. In *ICSE '00: Proceedings of the Conference on The Future of Software Engineering*, páginas 25–34, New York, NY, USA. ACM.

Kim, Y. and Stohr, E. A. (1998). Software reuse: survey and research directions. *J. Manage. Inf. Syst.*, 14(4):113–147.

Krueger, C. W. (1992). Software reuse. *ACM Computer. Surveys*, 24(2):131–183.

Lanna, A. L. P. M., Pietrobon, C. A. M. (2008). Proposta de reutilização de processos utilizando conhecimento e visualização. In *VI Workshop Brasileiro de Teses e Dissertações em Qualidade de Software*, páginas 1–10, Florianópolis, SC, Brasil.

Lanna, A. L. P. M. (2009). Reuso de Processos de Software baseado na componentização de Processos e Conhecimento. Dissertação de mestrado. PUCMG - Pontifícia Universidade Católica de Minas Gerais, 2009.

Li, J., Gupta, A., Arvid, J., Borretzen, and Conradi, R. (2007). The empirical studies on quality benefits of reusing software components. In *COMPSAC 2007 - Proceedings of the 31st Annual International Computer Software and Applications Conference*, volume 2, páginas 399–402.

Lucrédio, D., dos Santos Brito, K., Alvaro, A., Garcia, V. C., de Almeida, E. S., de Mattos Fortes, R. P., Meira, S. L. (2008). Software reuse: The brazilian industry scenario. *Journal of Systems and Software*, 81(6):996 – 1013.

Montoni, M. A. (2003). Aquisição de conhecimento: uma aplicação no processo de desenvolvimento de software. Dissertação de Mestrado, Universidade Federal do Rio de Janeiro, 2003.

OMG (2008). *Software & Systems Process Engineering Meta-Model Specification*. Abril, 2008.

Osterweil, L. J. (1987). Software processes are software too. *Proceedings - International Conference on Software Engineering*, páginas 2 – 13.

Osterweil, L. J. (1997). Software processes are software too, revisited: An invited talk on the most influential paper of icse 9. *Proceedings - International Conference on Software Engineering*, páginas 540 – 548.

Pressman, R. S. (2006). *Engenharia de Software*. McGraw-Hill, 6a. edição.

Ramadour, P. and Cauvet, C. (2008). An ontology-based reuse approach for information systems engineering. pages 572 –579.

Sommerville, I. (2007). *Engenharia de Software*. Pearson Addison-Wesley, São Paulo, SP, Brazil, 8a. edição.

Szyperski, C. (2002). *Component Software: Beyond Object-Oriented Programming*. Addison-Wesley Longman Publishing Co., Boston, MA, USA.

Wang, H. H. and Sun, J. (2009). A semantic web environment for components. *The Knowledge Engineering Review*, páginas 59–75.

Xu, P. (2005). Knowledge support in software process tailoring. Big Island, HI, United states.