

# Exploração de *Design Rationale* de Artefatos de Software na Web - Um Mecanismo de Busca em Documentos XML

**Lisandra C. Fumagalli**

lisandra@icmc.usp.br

**Renata P. M. Fortes**

renata@icmc.usp.br

Universidade de São Paulo – Departamento de Computação e Estatística  
São Carlos - São Paulo - Brasil CEP 13560-970

## **Abstract**

Design Rationale (DR) consist of a set of information related to the development and decision process of a project. In software projects, acquiring and making such information available are important practices for the improvement of the development. Consequently the product must be produced with higher quality. With the documentation about the artefacts produced during the software development is possible to create links with their corresponding Design Rationale. Thus, an XML document may be considered an appropriate mechanism for the documentation activity. However, the effective use of the information contained in this XML document is only possible if these information are retrieved and explored according to the developers interests, making their search and discovery easier. The DocRationale tool was developed to allow the storage and recovery of information related to software projects and their respective DR. However, as only the simple navigation was foreseen initially, searching for DR information is very onerous. In this article we present a mechanism for DR searching as a way assist the exploration of these information.

**Keywords:** Design Rationale, Software Documentation, XML Document, Searching Mechanism.

## **Resumo**

As razões de projeto (*Design Rationale* - DR) consistem em um conjunto de informações relacionadas ao processo de desenvolvimento e de tomada de decisão de um projeto. Em especial, nos projetos de software, adquirir e disponibilizar tais informações são práticas importantes para a melhoria das atividades de desenvolvimento e conseqüentemente da qualidade do produto desenvolvido. Por meio da atividade de documentação, os artefatos produzidos durante o projeto de software constituem a base para que ligações possam ser inseridas e expressem as relações com o Design Rationale correspondente. Assim, um documento XML se apresenta como mecanismo apropriado para essa atividade. No entanto, a utilização efetiva das informações nesse documento XML só é possível se elas forem recuperadas e exploradas de forma a atender às necessidades dos desenvolvedores, facilitando-se sua busca e descoberta. A ferramenta DocRationale foi desenvolvida para permitir o armazenamento e recuperação de informações de projeto de software, e respectivo DR. No entanto, para a exploração do DR armazenado, somente a navegação simples foi prevista inicialmente. Assim, a busca por informações de DR torna-se bastante custosa. Neste artigo é apresentado um mecanismo para busca de DR, de maneira a auxiliar a exploração dessas informações.

**Palavras-chave:** Design Rationale, Documentação de Software, Documento XML, Mecanismo de Busca.

## 1. INTRODUÇÃO

Na engenharia de software, o objetivo é produzir software com qualidade, reduzindo-se os custos e os esforços exigidos nas várias fases do desenvolvimento e na manutenção [30]. Nesse sentido, técnicas e métodos se consolidam e contribuem para o reaproveitamento das experiências e a retomada do processo de decisão que envolve o processo de desenvolvimento de software. Além disso, organizações de desenvolvimento de software têm empregado esforços para utilizar o conhecimento adquirido como importante recurso para o cumprimento das atividades de engenharia de software [5, 25, 35] e obter melhorias no processo de desenvolvimento. Essas ações contribuem para a melhoria da qualidade dos produtos de software.

A aquisição e recuperação de informações relacionadas às decisões de projeto se tornam, portanto, necessárias para que as experiências sejam aproveitadas em projetos futuros. Essas informações relacionadas às decisões de projeto são muito úteis para outros projetos, uma vez que erros podem ser evitados e alternativas anteriormente consideradas podem ser mais bem reaproveitadas [33], pois soluções discutidas e adotadas em um projeto podem ser relevantes a outros. Do mesmo modo, o conhecimento disponibilizado pode ser utilizado para evitar a repetição de falhas e contribuir para aquisição de familiaridade [1, 24], auxiliando os desenvolvedores na realização de suas atividades.

As razões de projeto (Design Rationale - DR) se aplicam nesse contexto, constituindo uma base de informações relacionadas às decisões tomadas e também às razões que propiciaram cada decisão, incluindo suas justificativas, alternativas consideradas, assim como o raciocínio empregado no desenvolvimento de um projeto [23, 15, 28, 31]. No caso dos projetos de software, a atividade de documentação é responsável pelo registro dos artefatos (ou seja, quaisquer tipos de documentos produzidos durante a produção do software) que são desenvolvidos. Assim, um documento XML se apresenta como mecanismo apropriado para apoiar essa atividade, propiciando que ligações sejam inseridas aos artefatos, para associar o DR correspondente.

A necessidade de armazenar as informações, tanto registradas nos diversos documentos, quanto as relacionadas a DR, e possibilitar sua efetiva disponibilização para o compartilhamento do conhecimento relacionado à execução das atividades de engenharia de software motivou o desenvolvimento da ferramenta DocRationale [13]. A DocRationale é uma ferramenta *web* que tem como finalidade permitir o armazenamento e recuperação de informações de projeto de software, e respectivo DR. No entanto, no estágio atual da ferramenta, a recuperação do DR armazenado é realizada somente através de navegação simples, tornando a busca por informações bastante custosa.

Neste artigo é apresentado um mecanismo para a busca do DR obtido e armazenado na ferramenta DocRationale de modo que a busca seja realizada de maneira mais objetiva e rápida, considerando o interesse dos desenvolvedores. Para isso são utilizadas tecnologias XML.

Na Seção 2 são apresentadas algumas estratégias investigadas como solução para a recuperação de informações de DR, considerando-se que estejam armazenadas em hiperdocumentos. As principais características da ferramenta DocRationale são apresentadas na Seção 3. O mecanismo para busca de DR adotado é descrito na Seção 4. Na Seção 5 são apresentados os trabalhos relacionados. Na Seção 6 são apresentados as conclusões e trabalhos futuros.

## 2. ESTRATÉGIAS DE RECUPERAÇÃO DE DR EM HIPERDOCUMENTOS

O esquema de representação de DR, ou seja, o modo com que as informações são organizadas para que sejam acessíveis, geralmente determina a forma de recuperação. No entanto, diferentes estratégias podem ser adotadas na recuperação, dependendo do interesse dos desenvolvedores [31]. Quando se considera a utilização de um hiperdocumento para esse esquema, entre as diferentes estratégias de recuperação se destaca, como mecanismo natural, a navegação. Assim, podem-se observar dois tipos: a navegação simples e a navegação adaptativa. Além da navegação, outras estratégias de recuperação de informações em hiperdocumento, que têm sido amplamente adotadas, são a pesquisa e abordagens híbridas.

### 2.1 Navegação Simples

A forma mais tradicional é navegação por *links*, através dos quais os nós de informação são explorados. Essa navegação pode ser realizada através de *links* entre dois ou mais documentos ou através de *links* entre partes de um mesmo documento. Existe ainda a navegação entre documentos precedentes e subsequentes, navegação através de *links* armazenados em um histórico de documentos navegados, e navegação através de *bookmarks*, que é semelhante ao anterior, mas permite ao usuário selecionar os documentos a serem armazenados [29]. Na estratégia de navegação simples, também se enquadram as interações com os *browsers* que permitem, por exemplo, a navegação de determinada página por meio da especificação direta de seu endereço (URL) e a navegação utilizando-se os botões *back* e *forward*.

### 2.2 Navegação Adaptativa

A navegação adaptativa visa auxiliar os usuários a partir da adaptação de como apresentar os *links*. Por exemplo, apresentar os *links* em uma determinada ordem de classificação, apresentar ou deixar de apresentar certos *links*, agregar informações adicionais ao *links*, dentre outras [7]. No entanto, esta estratégia requer a adoção de sistemas de hipermídia adaptativa, pois são necessárias funcionalidades específicas para o suporte a adaptatividade. Os sistemas

de hipermídia adaptativa são especialmente úteis para disponibilizar informação seletiva e contextual a usuários com diferentes objetivos e níveis de conhecimento [8].

### 2.3 Pesquisa

Essencialmente, o objetivo da pesquisa é prover resultados de busca com qualidade, de maneira eficiente [6]. Um meio de organizar páginas *web* que possibilite acesso rápido a um assunto específico e as classificar por tópicos. Nesse sentido, a utilização de máquina de busca (*search engine*) é apropriada, pois ela é capaz de relacionar em tópicos as informações pesquisadas pelos usuários [34]. A popularização e o sucesso das máquinas de busca estão relacionados ao desenvolvimento de novos algoritmos especialmente projetados para tornar mais rápida e precisa a recuperação de informações relevantes [19]. Uma máquina de busca gerencia de certa forma um índice na *web* e apresenta aos usuários as páginas relevantes relacionadas à pesquisa realizada. Para isso, é necessário algum tipo de esquema de classificação. A combinação entre pesquisa, índice e esquema de classificação, forma a estrutura básica de todas as máquinas de busca para *web* [34]. No entanto, determinar quais páginas são relevantes o suficiente para serem indexadas e ainda manter a precisão e a eficiência dessa tarefa é um enorme desafio [19]. Kobayashi e Takeda [19] classificam as pesquisas em:

- Simples: consistem na busca por uma só palavra ou frase.
- Customizadas: nessa categoria de pesquisa, a partir dos resultados obtidos em uma busca simples, procura-se refinar a mesma de modo que os melhores resultados sejam alvo de nova pesquisa.
- De diretórios: trata-se de buscas em diretórios/categorias. A habilidade de restringir a pesquisa dentro de categorias visa proporcionar melhor qualidade dos resultados.
- Por notícias atuais: nesse tipo de pesquisa, buscam-se conteúdos recentes e relevantes de fontes de alta qualidade. Os resultados podem ser apresentados eliminando ou agrupando os *links* similares. Tipicamente, essas pesquisas consideram a ordenação por data.
- Por conteúdo da *web*: nessa categoria de pesquisa, conteúdos adicionais como os de salas de bate-papo, quadros de avisos e guias regionais também são considerados. A customização também é aplicada.

Essas cinco categorias são geralmente avaliadas em termos da potencial qualidade dos dados e da facilidade de uso. No entanto, em ambientes *web*, a técnica de pesquisa por palavra-chave ainda predomina, pois é capaz de lidar efetiva e eficientemente com o imenso volume de informações contido na *web* [17].

### 2.4 Abordagens Híbridas

A combinação das estratégias apresentadas anteriormente caracteriza a abordagem híbrida. O principal objetivo é produzir melhores resultados na recuperação das informações.

Na ferramenta DocRationale, a estratégia de recuperação inicialmente adotada é a navegação simples. Outras características da ferramenta são apresentadas na próxima seção.

## 3. A FERRAMENTA DOCRATIONALE

A ferramenta DocRationale foi elaborada para permitir a aquisição, estruturação, armazenamento e recuperação de DR relacionado aos artefatos de software (todos os tipos de documentos de software, sejam na forma de diagramas, textos ou de outras mídias). Desenvolvida para a *web*, a ferramenta oferece *login* remoto e acesso controlado às informações armazenadas. Além disso, a DocRationale privilegia o fator de colaboração entre os membros das equipes de projeto para obtenção de DR.

Considerando promover uma busca satisfatória, foi adotada na DocRationale, uma combinação das perspectivas de comunicação e argumentação [32] para a aquisição e armazenamento de DR. Sob a perspectiva de comunicação, a DocRationale possibilita que diversas formas de comunicação digital (arquivos de áudio, vídeo, e-mail entre outros) sejam anexadas, complementando as informações sobre os artefatos de software. Quanto à perspectiva da argumentação, foram utilizadas características do modelo PHI (*Procedural Hierarchy of Issues*) [26] para estruturar as informações de maneira hierárquica. Na ferramenta, o modelo foi simplificado para utilizar apenas três tipos de nós: questões, posições (respostas) e argumentos. Assim, os DRs relativos aos artefatos são determinados por um conjunto de questões, posições e argumentos, obtidos na forma de anotações, que são consideradas um bom meio para aquisição de DR [33].

Com o propósito de apoiar o suporte a DR, houve a integração da DocRationale com CoTeia [3] e GroupNote [18]. CoTeia é uma ferramenta hipermídia colaborativa assíncrona de criação de páginas *web*. Possui diversas funcionalidades, sendo as mais interessantes para a DocRationale: (i) criação e edição de páginas *web*, que são feitas no próprio browser via formulário HTML (HyperText Markup Language); (ii) histórico, que permite o acesso ao conteúdo das últimas versões de cada hiperdocumento; e (iii) upload, que possibilita a submissão de arquivos de qualquer formato ao servidor CoTeia. GroupNote é um serviço aberto de anotações colaborativas na *web* implementado como uma API (*Application Programming Interface*). Suas características mais interessantes para DocRationale são: (a) compartilhamento de anotações por equipes de usuários; e (b) fornecimento da funcionalidade de hierarquia de nós.

Uma funcionalidade mais recente da DocRationale permite que um projeto de software seja criado na ferramenta através de um documento XML. Esse documento, cuja estrutura deve estar de acordo com um DTD (*Document*

*Type Definition*) previamente definido, contém dados sobre o projeto como, por exemplo, fases, atividades, artefatos e suas respectivas datas de início e de previsão de término. A DocRationale carrega esse documento XML e insere, automaticamente na base de dados da ferramenta, os dados nele contido sobre o projeto a ser criado.

No modelo desse documento XML apresentado na Figura 1, é possível saber que o projeto em questão possui as fases: **Definicao**, **Desenvolvimento** e **Finalizacao**. Além disso, também é possível notar que, por exemplo, a fase de **Desenvolvimento** é constituída das atividades de **Planejamento** e **Elaboracao**. Por sua vez, a atividade de **Planejamento** possui os artefatos: **Requisitos**, **Interface externa** e **Descricao do projeto**, e a atividade **Elaboracao** possui os artefatos **Codigo**, **Plano de Teste**, **Resultado do Teste**, **Procedimentos de Construcao**, **Relatorio de Problemas**, **Anotacoes** e **Revisao**.

<pre> &lt;?xml version="1.0" encoding="ISO-8859-1"?&gt; &lt;!DOCTYPE fases SYSTEM "projeto1.dtd"&gt; &lt;fases&gt;  &lt;fase titulo="Definicao"&gt;   &lt;atividades&gt;     &lt;atividade titulo="Especificacao do       trabalho"&gt;       &lt;artefatos&gt;         &lt;artefato titulo="Enunciado"&gt;           &lt;data_inicio&gt;03/05/2004&lt;/data_inicio&gt;           &lt;data_prevista&gt;10/05/2004&lt;/data_prevista&gt;         &lt;/artefato&gt;       &lt;/artefatos&gt;     &lt;/atividade&gt;   &lt;/atividades&gt; &lt;/fase&gt;  &lt;fase titulo="Desenvolvimento"&gt;   &lt;atividades&gt;     &lt;atividade titulo="Planejamento"&gt;       &lt;artefatos&gt;         &lt;artefato titulo="Requisitos"&gt;           &lt;data_inicio&gt;03/05/2004&lt;/data_inicio&gt;           &lt;data_prevista&gt;10/05/2004&lt;/data_prevista&gt;         &lt;/artefato&gt;         &lt;artefato titulo="Interface externa"&gt;           &lt;data_inicio&gt;10/05/2004&lt;/data_inicio&gt;           &lt;data_prevista&gt;17/05/2004&lt;/data_prevista&gt;         &lt;/artefato&gt;         &lt;artefato titulo="Descricao do projeto"&gt;           &lt;data_inicio&gt;17/05/2004&lt;/data_inicio&gt;           &lt;data_prevista&gt;24/05/2004&lt;/data_prevista&gt;         &lt;/artefato&gt;       &lt;/artefatos&gt;     &lt;/atividade&gt;     &lt;atividade titulo="Elaboracao"&gt;       &lt;artefatos&gt;         &lt;artefato titulo="Codigo"&gt;           &lt;data_inicio&gt;24/05/2004&lt;/data_inicio&gt;           &lt;data_prevista&gt;07/06/2004&lt;/data_prevista&gt;         &lt;/artefato&gt;         &lt;artefato titulo="Plano de Teste"&gt; </pre>	<pre>           &lt;data_inicio&gt;07/06/2004&lt;/data_inicio&gt;           &lt;data_prevista&gt;14/06/2004&lt;/data_prevista&gt;         &lt;/artefato&gt;         &lt;artefato titulo="Resultado do Teste"&gt;           &lt;data_inicio&gt;14/05/2004&lt;/data_inicio&gt;           &lt;data_prevista&gt;18/05/2004&lt;/data_prevista&gt;         &lt;/artefato&gt;         &lt;artefato titulo="Procedimentos de           Construcao"&gt;           &lt;data_inicio&gt;24/05/2004&lt;/data_inicio&gt;           &lt;data_prevista&gt;07/06/2004&lt;/data_prevista&gt;         &lt;/artefato&gt;         &lt;artefato titulo="Relatorio de Problemas"&gt;           &lt;data_inicio&gt;24/05/2004&lt;/data_inicio&gt;           &lt;data_prevista&gt;18/06/2004&lt;/data_prevista&gt;         &lt;/artefato&gt;         &lt;artefato titulo="Anotacoes"&gt;           &lt;data_inicio&gt;24/05/2004&lt;/data_inicio&gt;           &lt;data_prevista&gt;18/06/2004&lt;/data_prevista&gt;         &lt;/artefato&gt;         &lt;artefato titulo="Revisao"&gt;           &lt;data_inicio&gt;18/06/2004&lt;/data_inicio&gt;           &lt;data_prevista&gt;25/06/2004&lt;/data_prevista&gt;         &lt;/artefato&gt;       &lt;/artefatos&gt;     &lt;/atividade&gt;   &lt;/atividades&gt; &lt;/fase&gt;  &lt;fase titulo="Finalizacao"&gt;   &lt;atividades&gt;     &lt;atividade titulo="Verificacao"&gt;       &lt;artefatos&gt;         &lt;artefato titulo="Avaliacao do trabalho"&gt;           &lt;data_inicio&gt;25/06/2004&lt;/data_inicio&gt;           &lt;data_prevista&gt;30/06/2004&lt;/data_prevista&gt;         &lt;/artefato&gt;       &lt;/artefatos&gt;     &lt;/atividade&gt;   &lt;/atividades&gt; &lt;/fase&gt; &lt;/fases&gt; </pre>
--	--

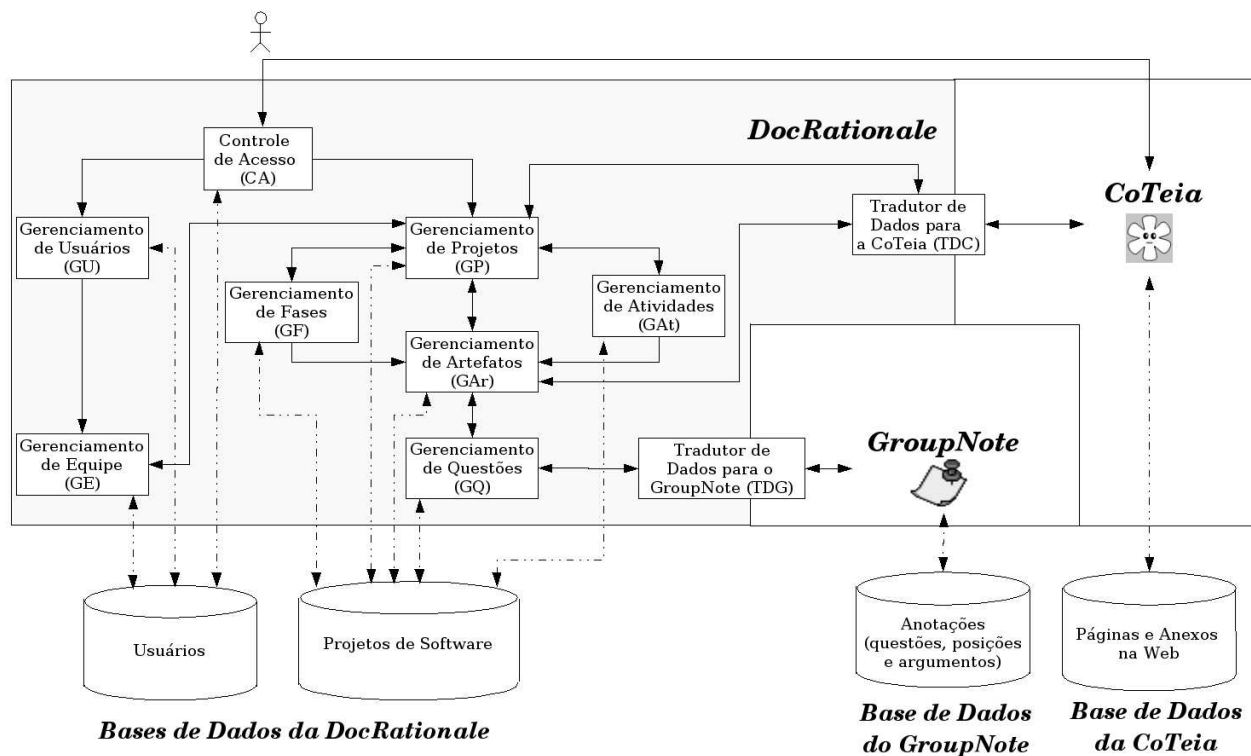
**Figura 1: Exemplo de documento XML para criação automática de projetos.**

Os dados dos projetos, assim como os DRs relacionados aos artefatos de software, são armazenados em bases de dados relacionais, distribuídas entre a DocRationale, a CoTeia e a API GroupNote como apresentado na Figura 2.

Nas bases de dados da DocRationale são armazenados os dados sobre os projetos: fases, atividades e artefatos. Os arquivos anexos são armazenados na base de dados da CoTeia e a base de dados do GroupNote armazena os DRs dos artefatos.

Observa-se, porém, que a exploração das informações de DR pelos desenvolvedores, a partir das consultas e navegações simples habilitadas na ferramenta se mostra custosa e pouco intuitiva. De fato, a maneira como estão armazenadas as informações na DocRationale dificulta a recuperação das mesmas. Associar os dados sobre os projetos com os DRs dos artefatos relativos a cada projeto, e ainda com os arquivos anexos de cada artefato, não é uma tarefa trivial.

A aquisição de DR na DocRationale é feita de maneira estruturada de modo que a recuperação desses DRs, através de navegação simples por *links*, é definida basicamente por essa estrutura. Assim, a atividade de buscar as informações desejadas se torna muito onerosa para o desenvolvedor, que deve percorrer grande quantidade de *links* para tentar encontrar a informação desejada.



**Figura 2: Arquitetura da ferramenta DocRationale.**

Dessa forma, a solução adotada para resolver essa dificuldade e promover buscas mais diretas e rápidas é apresentada a seguir.

#### 4. MECANISMO PARA A RECUPERAÇÃO DE DR

O principal objetivo é oferecer um mecanismo de busca de informações de maneira mais eficiente do que a atualmente disponível na ferramenta DocRationale, considerando os interesses dos desenvolvedores. A ideia é integrar os dados armazenados nas diferentes bases de dados de maneira a agregá-los em um único documento para cada projeto. Nesse contexto, a tecnologia *web* provê infra-estrutura apropriada, pois incorpora uma linguagem de marcação ou estruturação para documentos, a aplicação XML [36]. Uma vez criados os documentos de cada projeto, esses documentos devem ser armazenados em um banco de dados XML, favorecendo as consultas dos mesmos, com o propósito de buscar os DRs inicialmente armazenados na DocRationale.

Utilizando *tags* XML é possível desenvolver ferramentas de busca para melhorar a recuperação de informações para manutenção e reutilização dos documentos [16]. Haja vista todas essas características, optou-se por adotar os documentos XML para integrar os dados e os DRs a serem recuperados.

Para recuperar as informações de documentos XML, eles devem estar estruturados. O *XML Schema Definition* (XSD) [37] descreve a estrutura de um documento XML. Assim sendo, inicialmente, foi preciso definir o XSD. As páginas *web* dos projetos foram utilizadas para definir o XSD, de maneira semelhante ao processo descrito por Fumagalli et al. [14] para abstrair a estrutura de documentos.

Definida a estrutura do documento XML, o passo seguinte é criar os documentos. Para isso, algumas tecnologias como Cocoon [22] e XSP são utilizadas.

##### 4.1 Cocoon

O Projeto Cocoon [22, 38] foi fundado em Janeiro de 1999 por Stefano Mazzocchi como um projeto de código aberto sob o *Apache Software Foundation*. É um *framework* de desenvolvimento *web* construído em torno dos conceitos de desenvolvimento *web* baseado em componentes. Esses conceitos são implementados no Cocoon em torno da noção de *pipelines* de componentes. Cada componente no *pipeline* se especializa em uma operação particular: um documento XML é passado através de um *pipeline*, que consiste em diversos passos de transformação do documento. Todo *pipeline* inicia com um *generator*, continua com ou sem *transformers*, e termina com um *serializer*. O *generator* é o ponto inicial para o *pipeline* e o responsável por entregar os eventos SAX no *pipeline*. O *generator* é construído para ser independente do conceito de arquivo. Se for possível gerar eventos SAX a partir de uma outra fonte, isso pode utilizado sem ter que ser via arquivo temporário. O *transformer* obtém um documento XML (ou eventos SAX) e o transforma em outro documento XML (ou eventos SAX). O *serializer* é responsável por

transformar eventos SAX em um formato de apresentação. Existem diversos *serializers* que podem gerar diferentes formatos. A descrição de como o Cocoon é utilizado é descrito na próxima seção.

#### 4.2 Criação dos Arquivos XML

Para cada projeto criado na DocRationale, deve existir um documento XML correspondente. As *tags* definidas no XSD devem ser inseridas nos documentos e os dados provenientes das consultas realizadas nas bases de dados e relacionados a cada uma das *tags*, devem ser também inseridos no documento XML. Deve-se ressaltar que a estrutura definida no XSD deve ser mantida, pois ela reflete a hierarquia das informações referentes aos DRs obtidos. Resumidamente, os documentos XML são gerados pelo Cocoon através do processamento de um documento XSP.

XSP (*Extensible Server Pages*) é uma linguagem de marcação dinâmica como JSP (*Java Server Pages*) ou ASP (*Active Server Pages*) que provê diretivas de código embutido. É fortemente integrada com o Cocoon.

Um documento XSP é um documento XML com *tags* especiais XSP, que pode ser utilizado como o ponto de partida de uma *pipeline* de processamento do Cocoon. O documento XSP é processado por um *generator* especial do Cocoon chamado *ServerPagesGenerator*, cujo processador converte o documento XSP em um programa fonte para uma particular linguagem de processamento. Através da execução do programa gerado, o documento XML resultante, descrito pelo documento XSP original, é gerado e intercalado com o conteúdo dinâmico gerado pelo código extraído a partir das diretivas de *tags* XSP. Em outras palavras, o documento XSP é utilizado pelo *ServerPagesGenerator* para criar um documento XML.

Neste trabalho, o documento XSP é o responsável por acessar as bases de dados da DocRationale, CoTeia e GroupNote e realizar as consultas necessárias para recuperar os dados corretos, seguindo a estrutura definida para o documento XML. O Cocoon processa esse documento XSP e gera os documentos XML correspondentes aos projetos existentes na ferramenta, que são armazenados no banco de dados eXist [27].

#### 4.3 eXist

eXist [27] é uma base de dados XML de código aberto. O armazenamento de dados é baseado em árvores B+ e arquivos paginados. Os documentos são gerenciados em coleções hierárquicas, similar a arquivos armazenados em um diretório. eXist tem sua própria *Query Engine: XQuery*. Com extensões para dar suporte à pesquisa de documentos completos, as consultas podem transpor qualquer combinação possível de coleções ou documentos. Assim sendo, o banco de dados eXist possibilita definir diversos tipos de consultas que facilitam a busca dos DRs transcritos no documento XML de cada projeto.

#### 4.4 Recuperação de DR dos Documentos XML

Como estratégia de recuperação, adotou-se a forma de pesquisa mais comum: busca por palavra-chave. Para a busca, pode-se selecionar um ou mais projetos a serem pesquisados. Também é possível direcionar a busca através da escolha de outras opções como fases e atividades dos projetos, artefatos, pessoas. A seleção das informações, inclusive a palavra-chave, para a busca e recuperação dos DRs dos projetos é realizada a partir da DocRationale. Assim, proporciona-se que os interesses dos desenvolvedores sejam declarados.

Como a ferramenta está implementada em PHP, a busca pelos DRs dos artefatos de software é realizada via uma classe PHP (PHP API) que possibilita acessar o eXist e realizar as consultas.

O resultado das pesquisas é exibido na DocRationale como uma lista de *links* para os DRs recuperados. Prevê-se que uma classificação desses *links* pelos diversos atributos que foram estabelecidos no acionamento da busca possam auxiliar na interpretação e visualização dos DRs a serem explorados.

Além disso, possíveis arquivos anexos relacionados aos artefatos também devem ser considerados na busca pelos DRs. Para facilitar a varredura desses arquivos, que podem ser de diversos formatos (Microsoft Word, PDF, TXT, RTF), eles são convertidos em HTML através de ferramentas de código aberto: *wvWare*, *pdftohtml*, *txt2html* e *rtf-converter*.

Os arquivos em formato texto devem ser convertidos assim que seu *upload* é realizado na DocRationale. As ferramentas utilizadas diferem para cada tipo de arquivo.

- **wvWare** [21] é uma aplicação, com uma série de opções de linhas de comando, provida com a biblioteca *wv* que permite acesso a arquivos Microsoft Word, com o propósito de convertê-los em outros formatos. Dentre os diversos formatos para os quais os arquivos Word podem ser convertidos estão: HTML 4.0, Latex, dvi, PS, PDF, TXT, WML, RTF.
- **pdftohtml** [20] é uma ferramenta que converte arquivos do formato *Portable Document Format* (PDF) nos formatos HTML e XML. É baseada no XPDF, um visualizador de código aberto para arquivos pdf.
- **txt2html** [2] é uma ferramenta que converte arquivos em formato de texto para arquivos HTML. Ela oferece suporte para cabeçalhos, listas, tabelas, caracteres de marcação simples e *hyperlinks*.
- **rtf-converter** [12] é uma aplicação em linha de comando para converter arquivos de formato *Rich Text Format* (RTF) em HTML.

A varredura desses documentos, para busca por palavras-chave, será realizada por alguma ferramenta, já existente, que realize esse tipo de pesquisa.

Com esse mecanismo, graficamente esquematizado na Figura 3, busca-se recuperar os DRs obtidos de maneira mais eficiente e ágil. No entanto, deve-se ressaltar que esse mecanismo de recuperação não poupa o desenvolvedor de ter que ler as informações recuperadas para encontrar, ou não, a informação que realmente deseja. Acreditamos que a exploração a partir dessas informações pode contribuir para que o desenvolvedor tome conhecimento de outras informações também relacionadas, e para as quais não teria condições de ao simplesmente navegar, procurar por elas.

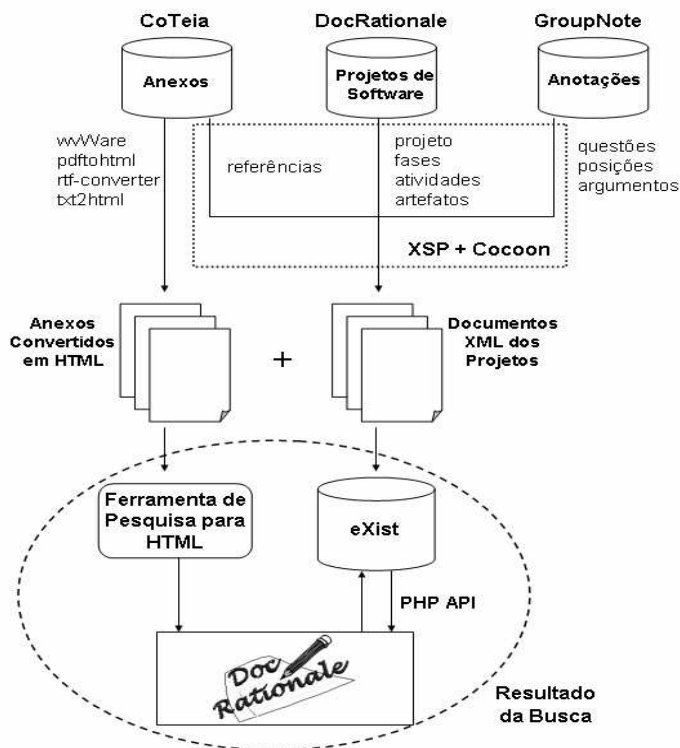


Figura 3: Mecanismo de recuperação de DR.

## 5. TRABALHOS RELACIONADOS

Uma abordagem baseada em XML para a verificação de documentos de requisitos de software é apresentada em [11]. Os requisitos de software são apresentados em XML e a linguagem XSLT é utilizada para verificar propriedades de qualidade desejadas, e para computar algumas métricas. O objetivo, nesse caso, é aplicar tecnologias XML, durante o processo de documentação de software, para verificar a qualidade da produtividade dos documentos de software e não são claramente mencionados os possíveis mecanismos para recuperação de informações relativas à DR e os próprios conteúdos dos documentos de requisitos.

Já a abordagem proposta em [4, 10] é a de mudar a representação de um código fonte para facilitar o uso de ferramentas e métodos mais poderosos de gerenciamento de documentos. É descrita uma aplicação XML que é utilizada para adicionar informação estrutural a arquivos não estruturados de código fonte. O texto pode, então, ser mais facilmente buscado, percorrido e transformado com o auxílio das *tags*. Assim, essa abordagem apresenta um conjunto mais completo de suporte à exploração de informações relacionadas com código-fonte de software. No entanto, o tratamento proposto se restringe a código-fonte, no caso, Java.

XML-GL [9] é uma linguagem gráfica de pesquisa para documentos XML. Um formalismo visual é utilizado para representar o conteúdo de documentos XML e a sintaxe para as pesquisas, permitindo que as mesmas sejam bastante complexas. Observa-se, no entanto, que a representação em XML-GL requer uma nova semântica das informações contidas nos documentos. No caso de informações referentes à documentação e ao DR de projetos de software, essa representação não é trivial.

## 6. CONCLUSÕES

As razões de projeto (*Design Rationale - DR*) constituem uma base de informações relacionadas às decisões tomadas e também às razões que propiciaram cada decisão, incluindo suas justificativas, alternativas consideradas, assim como o raciocínio empregado no desenvolvimento de um projeto. Nos projetos de software, a atividade de documentação é responsável pelo registro dos artefatos (ou seja, quaisquer tipos de documentos produzidos durante a produção do software) que são desenvolvidos. Assim, um documento XML se apresenta como mecanismo apropriado para apoiar essa atividade, propiciando que ligações sejam inseridas aos artefatos, para associar o DR correspondente.

Para possibilitar a exploração do DR registrado em documento XML, as estratégias de recuperação (navegação simples, navegação adaptativa, pesquisa e abordagem híbrida) foram analisadas.

Neste artigo foi apresentado um mecanismo para a busca mais rápida e objetiva do DR obtido e armazenado na ferramenta DocRationale, considerando o interesse dos desenvolvedores. Para isso são utilizadas tecnologias XML. Para recuperar o DR obtido, foi proposto um mecanismo, que tem como principal objetivo integrar informações sobre projetos de software, que estão distribuídas em diferentes bases de dados, em documentos XML, utilizando-se para isso tecnologias XML como o Cocoon e XSP. Esses documentos são armazenados em uma base de dados XML: eXist. A busca pelo DR armazenado é então possível a partir de consultas no eXist e como resultado, uma lista de *links* para as informações encontradas é exibida.

Embora o desenvolvedor não seja poupado de ter que navegar pela lista de *links* para encontrar a informação que realmente deseja, busca-se, com a definição desse mecanismo, estruturar as informações em um documento e tornar a recuperação de DR mais objetiva e rápida.

Como continuidade deste trabalho, diferentes alternativas de classificação dos *links* para os DR's obtidos da busca serão implementadas, possibilitando que o desenvolvedor visualize os contextos dos DR's, como por exemplo, quais os que ocorreram em determinado período de tempo, ou que envolveram um determinado membro da equipe de desenvolvimento. Assim, conforme interesse do desenvolvedor, o mecanismo de busca poderá auxiliar na exploração dos conteúdos de informação dos DRs obtidos. Finalmente, estudos experimentais serão realizados com o objetivo de avaliar com dados reais o mecanismo proposto.

## References

- [1] Adelson, B. and Soloway E. The Role of Domain Experience in Software Domain. *IEEE Transactions on Software Engineering*. Vol. 11, (1995).
- [2] Andersen K. txt2html. <http://txt2html.sourceforge.net> [February, 2004].
- [3] Arruda Jr, C. R. E., Izeki, C. A. and Pimentel, M. G. C. CoTeia: Uma Ferramenta Colaborativa de Edição Baseada na Web. Anais do Workshop de Ferramentas e Aplicações de VII SBMIDIA, (2002).
- [4] Badros, G. J. Javaml: A Markup Language for Java Source Code. *Proceedings of the International World Wide Web Conference - WWW9*, (2000).
- [5] Borges, L. S. and Falbo, R. A. Managing Software Process Knowledge. *Proceedings of the International Conference on Computer Science, Software Engineering, Information Technology, e-Business and Applications - CSITea'2002*, (2002), pp. 227-232.
- [6] Brin, S. and Page, L. The Anatomy of Large-Scale Hypertextual Web Search Engine. <http://www7.scu.edu.au/programme/fullpapers/1921/com1921.htm> [February, 2003], (2001).
- [7] Brusilovsky P. Methods and Techniques of Adaptive Hypermedia. *User Modeling and User Adapted Interaction*. Vol. 6, No. 2-3, (1996), pp. 87-129.
- [8] Brusilovsky P. Adaptive Hypermedia. *User Modeling and User Adapted Interaction*. Vol. 11, No. 1, (2001), pp. 87-110.
- [9] Ceri, S., Damiani, E., Fraternali, P., Paraboschi S. and Tanca L. XML-GL: A Graphical Language for Querying and Restructuring XML Documents. *Proceedings of the 8<sup>th</sup> International World Wide Web Conference - WWW8*, (1999).
- [10] Collard M., Maletic, J. I. and Marcus, A. Supporting Document and Data Views of Source Code. *Proceedings of the ACM Symposium on Document Engineering - DocEng'02*, (2002), pp.34-41.
- [11] Duran, A., Ruiz, A., Bernadez, B. and Toro, M. Verifying Software Requirements with XSLT. *ACM Software Engineering Notes*. Vol. 27, (2003).
- [12] Ford, G. rtf-converter. <http://freshmeat.net/projects/rtfconverter> [February, 2001], (2003).
- [13] Francisco S. D., Izeki, C. A., Paiva, D. M. B. and Fortes, R. P. M. Um Sistema de Apoio à Utilização de Design Rationale de Artefatos de Software. Anais da XXIX Conferência Latinoamericana de Informática - Clei 2003, (2003).
- [14] Fumagalli, L. C., Pansanato, L. T. E and Fortes, R. P. M Documentation Process in Interactive Systems – A Case Study to Abstract its Structure. *Proceedings of the 2<sup>nd</sup> International Information and Telecommunication Technologies Symposium - I2TS 2003*, (2003).
- [15] Gruber, T. R. and Russel, D. M. Design Knowledge and Design Rationale: A Framework for Representation, Capture and Use. *Technical Report KSL 90-45 – Knowledge Systems Laboratory*, (1991).
- [16] Harold, E. R. and Means, W. S. *XML in a Nutshell*, O'Reilly. 2002.



- [17] Hu, W. C., Chen Y., Schmalz M. S. and Ritter, G. X. An Overview of World Wide Web Search Technologies. *Proceedings of the 5<sup>th</sup> World Multi-Conference on System, Cybernetics and Informatics – SCI2001*, (2001).
- [18] Izeki, C. A., Arruda Jr, C. R. E and Pimentel, M. G. C. An XML-based Infrastructure Supporting Collaborative Annotations as First-class Hyperdocuments. *Proceedings of the VII Brazilian Symposium of Multimedia and Hypermedia Systems*, (2001), pp. 173-186.
- [19] Kobayashi M. and Takeda, K. Information Retrieval on the Web. *ACM Computing Surveys*. Vol.32, No. 2, (2002).
- [20] Kruk, M. pdfphtml. <http://pdfphtml.sourceforge.net> [February, 2004], (2003).
- [21] Lachowics, D. wwWare. <http://wwware.sourceforge.net> [February, 2003], (2000).
- [22] Langham, M. and Ziegeler C. *Cocoon: Building XML Applications*. New Riders Publishing, 2002.
- [23] Lee, J. Design Rationale Systems: Understanding the Issues. *IEEE Expert/Intelligent Systems and Their Applications*. Vol. 12, No. 3, (1997), pp.78-85.
- [24] Lindvall M. and Sandahi, K. How well do Experienced Software Developers Predict Software Change? *The Journal of Systems and Software*. No. 43, (1998), pp.19-27.
- [25] Markkula, M. Knowledge Management in Software Engineering Projects. *Proceedings of the 11<sup>th</sup> International Conference on Software Engineering and Knowledge Engineering – SEKE'99*, (1999).
- [26] McCall, R. J. PHI: A Conceptual Foundation for Design Hypermedia. *Design Studies*. Vol. 12, No. 1, (1991), pp.30-41.
- [27] Meier, W. eXist: An Open Source Native XML Database. *Revised Papers from the Web and Database-Related Workshops on Web, Web Services, and Database – NODe 2002*, (2002), pp. 169-183.
- [28] Moran, T. P. and Carroll, J. M. *Design Rationale Concepts, Techniques and Use Computers, Cognition and Work*. New Jersey: Lawrence Erlbaum Associates, 1996.
- [29] Nielsen, J. *Multimedia and Hypertext: The Internet and Beyond*. AP Professional, 1995.
- [30] Pressman, R. S. *Software Engineering: A Practitioner's Approach*. Fifth Edition, McGraw Hill, 2000.
- [31] Regli, W. C., Hu, X., Atwood, M. and Sun, W. Survey of Design Rationale Systems: Approaches, Representation, Capture and Retrieval. *Engineering with Computers: An International Journal for Simulation-Based Engineering* (Special issue on Computer Aided Engineering in honor of Professor Steven J. Fenves). Vol. 16, (2000), pp. 209-235.
- [32] Shipman F, and McCall, R. Integrating Different Perspectives on Design Rationale: Supporting the Emergence of Design Rationale from Design Communication. *Artificial Intelligence in Engineering Design, Analysis and Manufacturing*. Vol. 11, No. 2, (1997), pp. 141-154.
- [33] Souza, C., Santos, D. B., Dias, K. L. and Wainer, J. A. Model and Tool for Semi-Automatic Recording of Design Rationale in Software Diagrams. *Proceedings of the String Processing and Information Retrieval Symposium and International Workshop on Groupware*, (1999), pp.306-313.
- [34] Trevor, B., Weippl, E. and Winiwarter, W. A Modern Approach to Searching the World Wide Web: Ranking Pages by Inference over Content. *Proceedings of the 14<sup>th</sup> International Conference Applications of Prolog*, (1998).
- [35] Wangenheim, C. G., Althoffl, K. D. and Barcia, R. M. Intelligent Retrieval of Software Engineering Experienceware. *Proceedings of the 11<sup>th</sup> International Conference on Software Engineering and Knowledge Engineering – SEKE'99*, (1999).
- [36] World Wide Web Consortium. *Extensible Markup Language (XML) 1.0*. Second Edition, W3C Recommendation. <http://www.w3.org/TR/2000/REC-xml-20001006> [May, 2004], (2000).
- [37] World Wide Web Consortium. *XML Schema 1.1*. W3C Recommendation. <http://www.w3.org/XML/Schema> [May, 2004], (2001).
- [38] The Apache Software Foundation. *Cocoon 2.1*. Apache Cocoon Project. <http://cocoon.apache.org/2.1> [april, 2004], (2004).