

# Um Editor de Metadados para Documentar Padrões de Análise em uma Infraestrutura de Reuso

Douglas Alves Peixoto, Lucas Francisco da Matta Vegi, Jugurta Lisboa-Filho

Departamento de Informática – Universidade Federal de Viçosa (UFV)  
36570-000 – Viçosa – MG – Brasil

{douglasalves.ufv, lucasvegi}@gmail.com, jugurta@ufv.br

**Abstract.** *The software development process often encounters obstacles to reuse analysis patterns due to difficult access to these computational artifacts. The lack of both a computational tool to support the analysis patterns documentation process and a digital repository to store these patterns contributes negatively on the retrieval and reuse of them. This paper presents the tool DC2AP Metadata Editor. This tool is a metadata editor for analysis patterns based on the Dublin Core Application Profile for Analysis Patterns (DC2AP). To organize the analysis patterns documentation process, and facilitate their retrieval by software developers, the DC2AP Metadata Editor uses the idea of Semantic Web, and provides an environment for documentation of analysis pattern metadata as Linked Data, thus allowing the knowledge stored in these artifacts to be shared and automatically interpreted by software, facilitating the process of patterns retrieval through search engines, for instance. The system is a Web-based SaaS system developed under the concepts of Cloud Computing, and provides some of its functionality through Web services, so that these services can be used in other tools with similar purpose.*

**Resumo.** *O processo de desenvolvimento de software muitas vezes encontra obstáculos na reutilização padrões de análise devido ao difícil acesso a estes artefatos computacionais. A falta de uma ferramenta que facilite o processo de documentação de padrões de análise, e de um repositório digital para armazená-los, contribui negativamente na recuperação e reuso dos mesmos. Este trabalho apresenta a ferramenta DC2AP Metadata Editor. Esta ferramenta é um editor de metadados para padrões de análise baseada no modelo Dublin Core Application Profile for Analysis Patterns (DC2AP). Para organizar o processo de documentação dos padrões de análise, e facilitar sua recuperação por desenvolvedores de software, o DC2AP Metadata Editor utiliza da ideia da Web Semântica, e fornece um ambiente para documentação dos metadados de padrões de análise na forma de Linked Data, permitindo assim que o conhecimento armazenado nesses artefatos sejam compartilhados e automaticamente interpretados por software, facilitando o processo de recuperação de padrões através de motores de busca, por exemplo. O sistema é um SaaS desenvolvido com tecnologia Web baseado nos conceitos de Computação nas Nuvens, e fornece algumas de suas funcionalidades através de serviços Web, de forma que estes serviços possam ser utilizados em outras ferramentas com propósito semelhante.*

## 1. Introdução

Padrões de análise são artefatos computacionais reutilizáveis focados na fase de análise do processo de desenvolvimento de software. Estes artefatos auxiliam projetistas de software no processo de análise do domínio do sistema, diminuindo, desta maneira, o tempo gasto no projeto do software e, principalmente, permitindo interoperabilidade entre sistemas. Embora os padrões de análise proporcionem ganhos no processo de desenvolvimento de software por meio da reutilização de conhecimento e de ideias úteis e comprovadamente testadas, ainda assim projetistas podem encontrar dificuldades em reutilizá-los devido à carência de padrões de análise ou pela ineficiência na forma em que estes estão documentados [Blaimer *et al.*, 2010], o que dificulta sua busca por ferramentas computacionais ou mesmo por motores de busca na internet. Padrões de análise são normalmente disponibilizados em meios de acesso não processáveis por computador, como livros e documentos científicos, dificultando assim a recuperação de informações destes artefatos computacionais reutilizáveis.

A documentação de padrões de análise é uma tarefa importante para que estes artefatos possam ser facilmente compartilhados por projetistas de software. Até o momento não há nenhum modelo para documentar padrões de análise que seja amplamente aceito [Pantoquilha *et al.*, 2003][Raminhos *et al.*, 2006], sendo assim, cada conjunto destes padrões é especificado de acordo com as preferências de seus autores.

Um consenso em relação à utilização e disseminação de um mesmo padrão de documentação por parte dos autores facilitaria os processos de busca, manipulação e disseminação destes padrões, e permitiria que estes processos também se tornassem possíveis de serem realizados por meio de algoritmos e ferramentas computacionais. Para isto, também é importante que seja dada uma semântica a estas informações, este princípio baseia-se na ideia da Web Semântica, na qual é dado um significado bem definido às informações publicadas na Web, permitindo que humanos e computadores trabalhem em cooperação [Berners-Lee *et al.*, 2001]. Esta estratégia também admite a aplicação do princípio de *Linked Data* o qual permite publicar e conectar estruturas de dados na Web [Bizer *et al.*, 2009], no caso deste trabalho, criando relacionamentos entre padrões de análise que possuam alguma semelhança ou dependência entre si. Para aplicar o princípio da Web Semântica e prover *Linked Data* os metadados devem ser publicados na Web em um documento no formato RDF; estes princípios serão descritos mais adiante neste trabalho.

Visando solucionar os problemas da falta de padronização e carência de detalhes na documentação dos padrões de análise, Vegi *et al.* (2012b) propuseram o perfil de aplicação DC2AP, sendo este um perfil de metadados específico para documentar padrões de análise.

Este trabalho apresenta a ferramenta DC2AP Metadata Editor. Esta ferramenta possibilita a criação e edição de metadados para documentar padrões de análise utilizando o DC2AP como modelo. Os metadados referentes aos padrões de análise documentados com esta ferramenta podem ser exportados para o formato RDF e consequentemente transformarem-se em *Linked Data* que diminuem as barreiras de reutilização dos padrões.

O DC2AP Metadata Editor é um software livre e está sendo disponibilizado nas nuvens via Web seguindo o modelo de Software como Serviço (SaaS). Sendo assim, ele pode ser acessado de qualquer local via internet sem a necessidade de se preocupar com detalhes tecnológicos em seu terminal de acesso, necessitando apenas de um navegador Web e acesso à internet [Weiss, 2007]. Algumas das funcionalidades deste editor de metadados são também disponibilizadas através de Serviços Web, permitindo que outros sistemas possam fazer uso de algumas das funcionalidades desta ferramenta. Douglas et. al (2013) apresenta uma visão geral da ferramenta e das tecnologias utilizadas para seu desenvolvimento. Neste trabalho, adicionalmente vamos entrar nos detalhes da facilidade de utilização da ferramenta, expondo aspectos inéditos da pesquisa e do processo de desenvolvimento da ferramenta e apresentação dos detalhes dos serviços Web para utilização por desenvolvedores, bem como apresentar estudo de caso e avaliação da mesma.

O restante deste trabalho está organizado como a seguir: a Seção 2 descreve os trabalhos relacionados com a documentação de padrões de análise; Web semântica e *Linked Data*; computação em nuvem e serviços Web. A ferramenta DC2AP Metadata Editor proposta neste trabalho é apresentada na Seção 3, enquanto a Seção 4 apresenta conclusões e trabalhos futuros.

## **2. Trabalhos Relacionados**

### **2.1. Documentação de Padrões de Análise**

Os padrões de análise podem ser documentados a partir do uso de modelos e esboços predefinidos. A não utilização de modelos na documentação dos padrões de análise reduz a usabilidade destes padrões [Blaimer *et al.*, 2010]. Isso ocorre devido a maior dificuldade de recuperação de conhecimento em documentações não estruturadas.

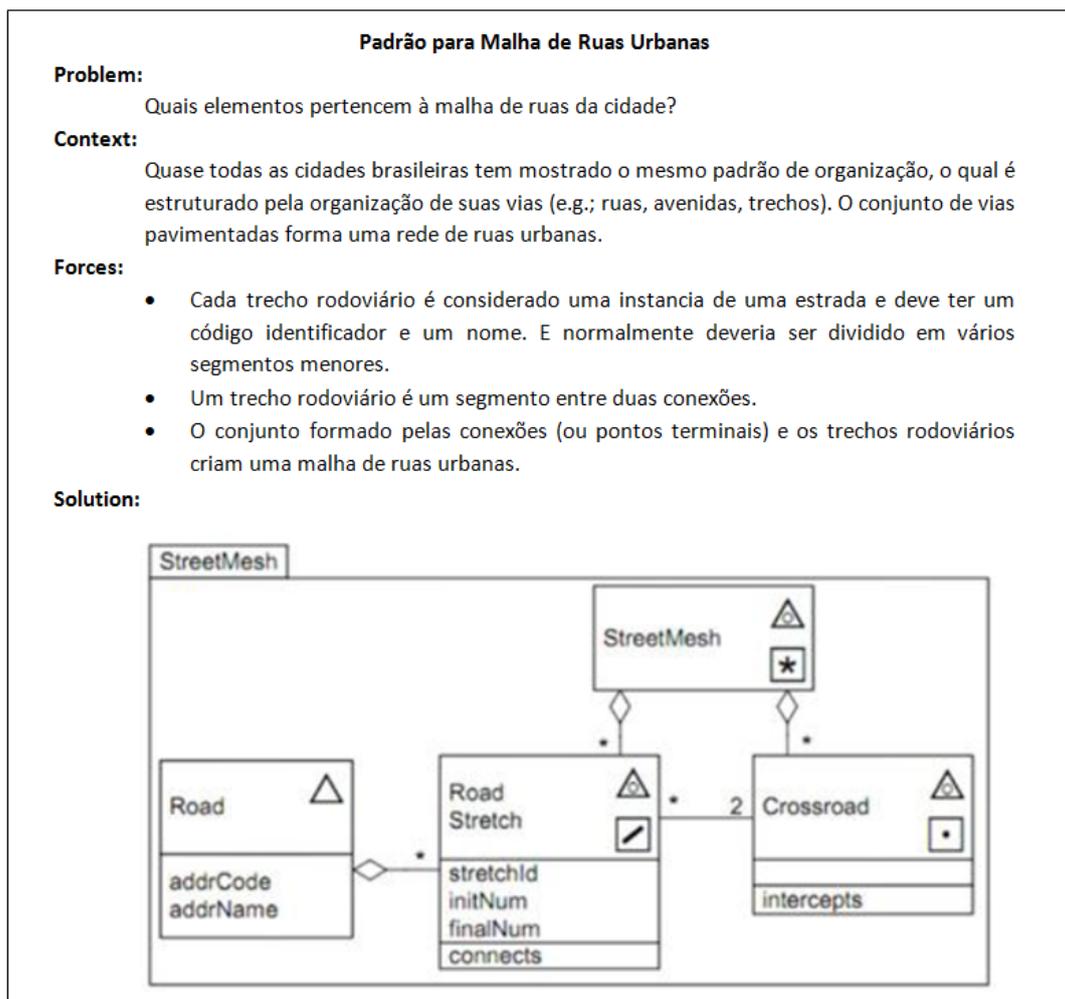
Alguns padrões de análise documentados através de modelos podem ser encontrados nos trabalhos de Lisboa-Filho *et al.* (2002) e Fernandez e Yuan (2009). Um modelo para a documentação de padrões é apresentado por Meszaros e Doble (1997), sendo este modelo composto pelos tópicos: *name, problem, context, motivation, solution, participants and related patterns*. Este modelo foi utilizado por Lisboa-Filho *et al.* (2002) como mostrado na Figura 1 para documentar um padrão de análise voltado para a definição de uma malha de ruas urbanas.

O padrão documentado na Figura 1 será apresentado mais adiante documentado com o perfil DC2AP por meio da ferramenta DC2AP Metadata Editor.

### **2.2. APRI – Analysis Patterns Reuse Infrastructure**

A fim de minimizar problemas relativos à documentação e disponibilização de padrões de análise, aumentando assim as possibilidades de reutilização desses artefatos, Vegi *et al.* (2012a) propuseram a arquitetura de uma Infraestrutura de Reutilização de Padrões de Análise (APRI). Nesta infraestrutura, que é apresentada na Figura 2, os padrões de análise são documentados usando metadados e podem ser recuperados por meio de serviços Web.

A APRI consiste dos seguintes componentes:



**Figura 1. Padrão para Malha de Ruas Urbanas [Lisboa-Filho et al., 2002]**

- **Pattern Portal:** contem um conjunto de Web sites focados na obtenção de padrões de análise e ferramentas e serviços para prover sua descoberta, catalogação e reuso.
- **Metadata Repository:** são repositórios que contem metadados em formato XML (*eXtensible Markup Language*) para a especificação de padrões de análise e serviços contidos na APRI.
- **Analysis Pattern Repository:** são repositórios que contem os padrões de análise em formato XMI (*XML Metadata Interchange*), permitindo seu uso para serviços de visualização e colaboração.
- **Portrayal Service:** são serviços que suportam a visualização dos padrões de análise da APRI.
- **Catalog Service:** estes serviços permitem a descoberta e uso dos padrões de análise e serviços da APRI baseados em seu metadados.
- **Access Service:** estes serviços permitem o acesso e obtenção de padrões de análise.

- **Collaboration Service:** este serviço permite aos projetistas experientes contribuir para a melhoria dos padrões de análise.

O DC2AP Metadata Editor, ferramenta apresentada neste trabalho, foi projetado e desenvolvido como uma das ferramentas do *Pattern Portal*, componente este que oferece serviços e ferramentas de catalogação de padrões de análise em uma APRI [Vegi *et al.*, 2012a].

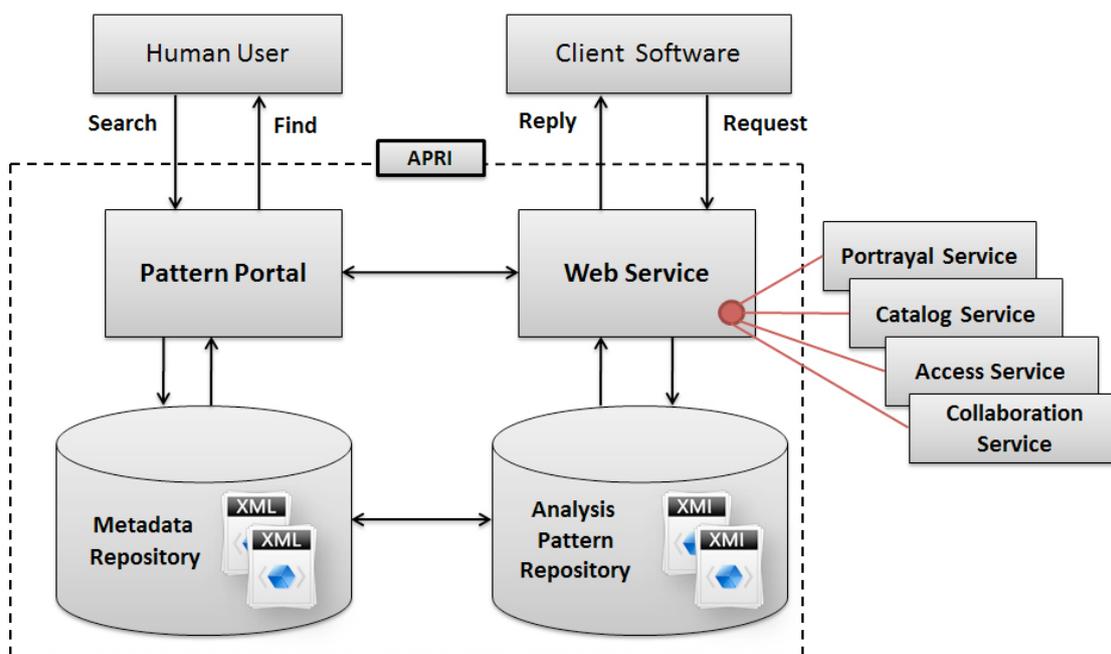


Figura 2. Analysis Patterns Reuse Infrastructure (APRI)

### 2.3. DC2AP - Dublin Core Application for Analysis Pattern

Para auxiliar autores e usuários de padrões de análise na tarefa de documentação, Vegi *et al.* (2012c) propuseram o perfil de aplicação DC2AP. O DC2AP foi desenvolvido baseado no modelo para especificação de padrões de análise proposto por Pantoquilha *et al.* (2003) e no padrão de metadados Dublin Core [DCMI, 1998]. Os principais objetivos do DC2AP são melhorar a recuperação e o reuso de padrões de análise por meio de uma documentação que permita um tratamento mais preciso das informações por um computador e, portanto, fornecer informações específicas que não poderiam ser recuperadas pelos motores de busca anteriormente.

Em contraste com o padrão de metadados Dublin Core, que é genérico e, portanto, destinado a documentar recursos de vários domínios, o modelo proposto no DC2AP foi desenvolvido especialmente para documentar padrões de análise, sendo assim rico em detalhes deste domínio específico. O DC2AP é composto por vinte e um elementos principais e quarenta e um elementos refinadores, sendo todos esses específicos para a documentação de padrões de análise.

A Tabela 1 apresenta todos os elementos que compõem o perfil DC2AP. A descrição técnica detalhada dos elementos do DC2AP pode ser encontrada em [Vegi, 2012].

Como mostra a Tabela 1, o DC2AP possui alguns elementos para controle de versão de padrões documentados e outros para compartilhamento de experiências de uso. Estas características foram adicionadas para permitir a criação de coleções dinâmicas de padrões de análise, onde novas versões melhoradas dos padrões podem ser propostas por um colaborador com experiência de uso.

**Tabela 1. DC2AP - Elementos e Regras de Aplicação [Vegi et al., 2012b]**

1. Identifier [M] [S] [UNS]			
2. Title [M] [S] [St]	2.1. Alternative Title [O] [Mu] [St]		
3. Creator [M] [Mu] [St]			
4. Subject [M] [Mu] [St]			
5. Description [M] [S] [N]	5.1. Problem [M] [S] [St]		
	5.2. Motivation [M] [Mu] [St]	5.2.1. Example [M] [Mu] [St]	
		5.2.2. Known Uses [O] [Mu] [St] *	
5.3. Context [M] [S] [St]			
6. Publisher [O] [Mu] [St]			
7. Contributor [Cd] [Mu] [St]			
8. Date [M] [S] [N]	8.1. Created [M] [S] [D]		
	8.2. Modified [Cd] [S] [D]		
9. Type [M] [S] [US]	9.1. Notation [M] [S] [St]		
10. Format [M] [Mu] [US]			
11. Source [Cd] [S] [UNS]			
12. Language [M] [S] [US]			
13. Relation [Cd] [S] [N]	13.1. Is Version of [Cd] [S] [UNS]		
	13.2. Is Replaced by [Cd] [Mu] [UNS] *		
	13.3. Replaces [Cd] [Mu] [UNS] *		
	13.4. Is Part of [O] [Mu] [UNS]		
	13.5. Has Part [O] [Mu] [UNS]		
	13.6. Is Designed With [O] [Mu] [UNS] **		
	13.7. Should Avoid [O] [Mu] [UNS] **		
	13.8. Complemented by [O] [Mu] [UNS] **		
	13.9. About [Cd] [S] [St]		
14. Coverage [O] [Mu] [St]			
15. Rights [Cd] [Mu] [US]			
16. History [M] [Mu] [N] *	16.1. Event Date [M] [S] [D]		
	16.2. Author [M] [Mu] [St]		
	16.3. Reason [M] [S] [St]		
	16.4. Changes [Cd] [S] [St]		
17. Requirements [M] [S] [N]	17.1. Functional Requirements [M] [Mu] [St]		
	17.2. Non-functional Requirements [O] [Mu] [St]		
	17.3. Dependencies and Contributions [M] [S] [St]	17.3.1. Dependency Graph [M] [S] [U]	
		17.3.2. Contribution Graph [Cd] [S] [U]	
	17.4. Conflict identification & Guidance to Resolution [Cd] [Mu] [St]		
	17.5. Priorities diagram [M] [S] [U]		
17.6. Participants [M] [Mu] [St]			
18. Modelling [M] [S] [N]	18.1. Behaviour [M] [S] [N]	18.1.1. Use Case Diagram [M] [S] [U]	
		18.1.2. Collaboration/Sequence Diagrams [M] [Mu] [U]	
		18.1.3. Activity/State Diagrams [O] [Mu] [U]	
	18.2. Structure [M] [S] [N]	18.2.1. Class Diagram [M] [S] [U]	
		18.2.2. Class descriptions [M] [S] [U]	
18.2.3. Relationship descriptions [M] [Mu] [St]			
18.3. Solution Variants [O] [Mu] [U] **			

19. Resulting Context [O] [Mu] [St] **	
20. Design Guidelines [O] [Mu] [St] **	
21. Consequences [M] [S] [N]	21.1. Positive [M] [Mu] [St] 21.2. Negative [M] [Mu] [St]

\* Elemento para controle de versão

\*\* Elemento para colaboração e experiências

Cada um dos elementos principais e refinadores do DC2AP são controlados por regras de preenchimento em relação à obrigatoriedade (*obligatoriness*), ocorrência (*occurrence*) e tipo de valor (*value type*). Estas regras buscam uniformizar a utilização deste perfil de aplicação e as documentações de padrões que o aplicam:

- **Obligatoriness:** [M] Mandatory; [O] Optional; [Cd] Conditional.
- **Occurrence:** [S] Single; [Mu] Multiple.
- **Value Type:** [St] String; [D] Date; [U] URI; [N] Null; [UNS] URI, Number and String; [US] URI and String.

Devido a limitações no tamanho do texto, a descrição semântica de cada um dos elementos que compõe o DC2AP, assim como alguns detalhes das regras para sua utilização, não serão apresentadas neste artigo. Contudo, estas informações podem ser obtidas em: <http://purl.org/dc2ap/TechnicalDescription>.

#### 2.4. Web Semântica e Linked Data

A Web Semântica é vista como uma camada da Web onde é possível publicar, obter e utilizar dados que podem ser processados direta ou indiretamente por máquinas [Berners-Lee, 2000].

Enquanto a Web Semântica objetiva transformar a Web convencional em um ambiente composto por dados processáveis por máquinas, *Linked Data* provê os meios para tornar isto possível. *Linked Data* são estruturas de dados processáveis por máquina e publicados na Web. Estes dados são bem definidos e conectados a outros dados de diferentes fontes através de relacionamentos semânticos [Bizer *et al.*, 2009].

Linked Data usa três tecnologias para dar suporte a Web Semântica. *Uniform Resource Identifiers* (URI) [Berners-Lee *et al.*, 2005] são usados para identificar os dados, o protocolo HTTP (*Hypertext Transfer Protocol*) por outro lado é usado como um mecanismo de recuperação de dados e, finalmente, o RDF (*Resource Description Framework*) é usado para descrever os dados processáveis por máquina na Web e criar relacionamentos semânticos entre eles [Morshed *et al.*, 2011].

RDF é um framework que provê uma especificação padrão para descrição de metadados processáveis por máquina [Rigo *et al.*, 2008]. Ele pode ser usado para descrever documentos Web assim como conceitos do mundo real em uma rede como, por exemplo, pessoas e empresas [Sauermaann *et al.*, 2008]. A partir do uso de RDF para descrever os dados, é possível estabelecer um relacionamento semântico entre eles devido à forma de como estes dados são codificados. Cada relacionamento entre dados no RDF consiste em uma tripla formada pelos conceitos de Sujeito, Predicado e Objeto. Nesta tripla, sujeito e objeto são identificadores URI dos dados que estão relacionados, e predicado especifica o relacionamento semântico entre eles, por exemplo, o tipo de relacionamento entre os dados [Klyne e Carroll, 2004].

O principal objetivo de *Linked Data* é construir a então chamada Web de Dados, permitindo, desta forma, que a Web seja usada como um único banco de dados global, reduzindo assim as barreiras para o reuso, integração e implementação de dados provenientes de diferentes fontes [Bizer *et al.*, 2009].

De forma a prover *Linked Data* entre padrões de análise é importante que uma ferramenta para catalogação se proponha a exportar os metadados destes artefatos para o formato RDF e associe um URI a cada um destes documentos. São objetivos da ferramenta DC2AP Metadata Editor, apresentada na Seção 3, cumprir com tais tarefas.

## 2.5. Computação em Nuvem

Computação em nuvem se refere a aplicativos fornecidos como serviços através da Internet bem como o hardware e os sistemas de software que fornecem estes serviços [Armbrust *et al.*, 2010]. A ideia é permitir que o usuário tenha acesso às mais variadas aplicações de qualquer terminal através da Internet independentemente da plataforma utilizada. Desta maneira, todo o processamento e armazenamento de dados são feitos não pelo usuário, mas por alguma central de dados na Internet tal como instâncias virtuais de servidores.

O fornecimento de computação em nuvem se faz por meio diferentes tipos de serviços. O principal deles, e que será abordado neste trabalho, é o conceito de Software como Serviço (SaaS) [Goodchild *et al.*, 2011]. Ele fornece vários recursos de software pela internet e estes podem ser utilizados por meio de Web browsers pelo usuário final [Armbrust *et al.*, 2010]. O DC2AP Metadata Editor, software apresentado neste trabalho, é um exemplo de SaaS.

Os principais motivos de se desenvolver o DC2AP Metadata Editor como um SaaS foram a facilidade no acesso à esta ferramenta e o acoplamento facilitado da mesma à uma APRI. Outras ferramentas disponíveis no mercado foram avaliadas no processo de desenvolvimento deste projeto, tais como as ferramentas CatMDEdit (2002) e xMET Metadata Editor (2011). Embora estas ferramentas sejam editores de metadados baseados no modelo Dublin Core, elas não são específicas para documentação de padrões de análise, e não foram desenvolvidas utilizando tecnologia Web, o que as impede de serem utilizadas como um SaaS em uma APRI.

## 2.6. Serviços Web

Serviços Web são muitas vezes confundidos com computação em nuvem, no entanto estas são tecnologias com definições e aplicações diferentes. Segundo o W3C (2004), um serviço Web é um sistema de software projetado para suportar a interoperabilidade entre máquinas sobre rede. Serviços Web permitem a comunicação entre aplicações através da Internet mesmo que tenham sido desenvolvidas em linguagens diferentes. Um serviço Web permite que recursos e serviços de uma aplicação estejam disponíveis na rede de uma forma que qualquer aplicação possa entender, pois qualquer aplicação disponibilizada via serviços Web é traduzida para uma linguagem universal. Um serviço Web é acessado através de protocolos e formatos de dados independentes de plataforma como o HTTP, XML e SOAP (*Simple Object Access Protocol*). A interface de um serviço Web é acessível por meio de mensagens XML padronizadas e, portanto, em formato texto [Tamae e Lima, 2005].

Utilizando a tecnologia de serviços Web, uma aplicação pode invocar outra para efetuar tarefas simples ou complexas mesmo que as duas aplicações estejam em diferentes sistemas. Em outras palavras, os serviços Web fazem com que os seus recursos estejam disponíveis para que qualquer aplicação cliente possa operar.

O DC2AP Metadata Editor fornece alguns de seus recursos por meio de serviços Web, desta maneira outras aplicações podem reutilizar estes recursos sem a necessidade de utilizar a interface gráfica da ferramenta ou mesmo de implementá-los novamente em uma outra linguagem diferente da utilizada pelo DC2AP Metadata Editor.

### 3. DC2AP Metadata Editor

O DC2AP Metadata Editor é um SaaS que permite a usuários de padrões de análise criar, editar e salvar metadados destes padrões de acordo com as regras descritas no perfil DC2AP, além de exportar estes metadados para o formato RDF fornecendo *Linked Data*.

Esta ferramenta foi desenvolvida utilizando o framework Google Web Toolkit (GWT). O GWT utiliza Java como linguagem de programação padrão e converte os códigos-fonte produzidos nesta linguagem para códigos JavaScript compatíveis com diversos navegadores. Esta tecnologia permite criar sistemas Web dinâmicos, versáteis e eficientes. O GWT é uma tecnologia para desenvolvimento Web e possui alto nível de portabilidade, permitindo, desta maneira, que o DC2AP Metadata Editor possa ser executado nos principais navegadores Web.

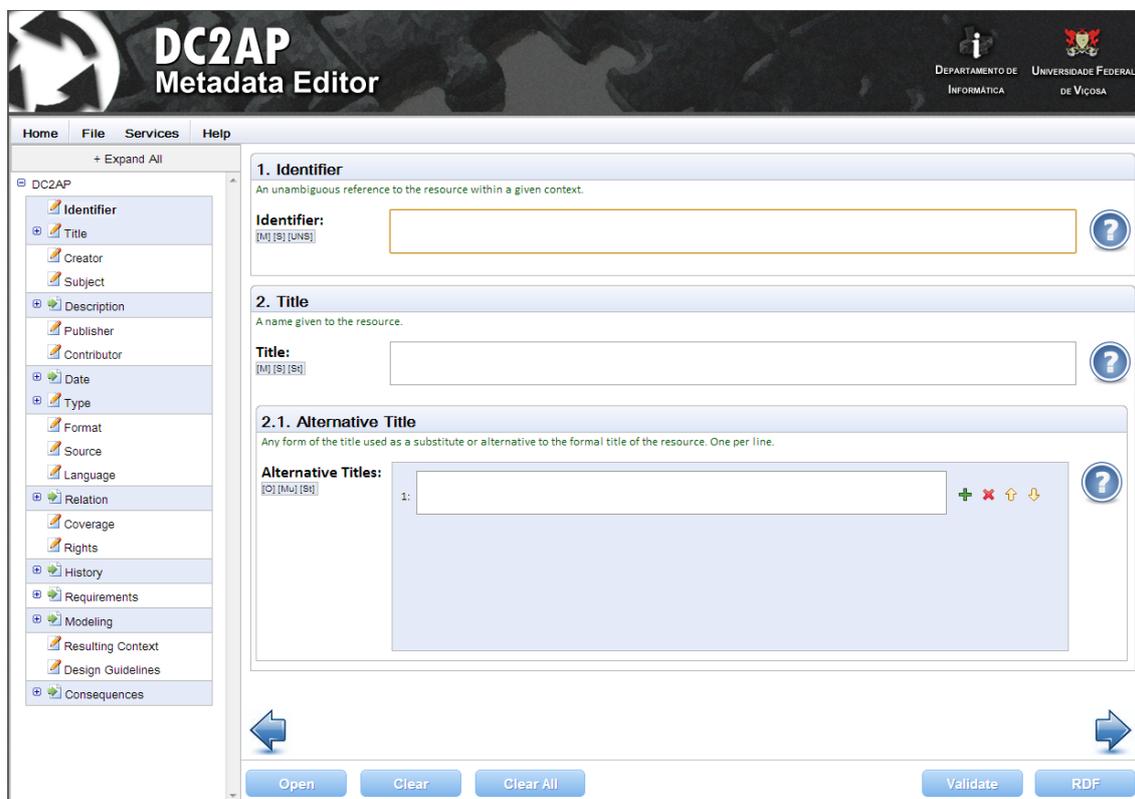


Figura 3. DC2AP Metadata Editor

A Figura 3 mostra a tela inicial do DC2AP Metadata Editor. Esta ferramenta está hospedada em um servidor nas nuvens da Universidade Federal de Viçosa que provê acesso às suas funcionalidades e Serviços Web. Ela pode ser acessada pelo URI: <http://www.purl.org/dc2ap/editor>.

Ao acessar a ferramenta, o usuário tem acesso a todos os elementos descritos no perfil DC2AP, sendo estes mostrados na árvore do lado esquerdo da ferramenta (ver Figura 3). Para cada elemento existem dois ícones que mostram detalhes do mesmo. O ícone exibido na Figura 4 (a) mostra as regras para obrigatoriedade, ocorrência e tipo de valor de um elemento. Já o ícone exibido na Figura 4 (b) provê acesso a informações detalhadas de um elemento, como é mostrado na Figura 5.

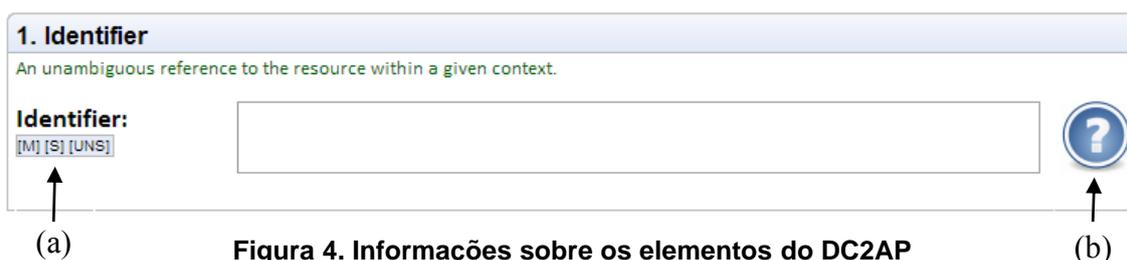


Figura 4. Informações sobre os elementos do DC2AP

O DC2AP Metadata Editor também provê campos para o preenchimento de elementos multivalorados do DC2AP, como o elemento *2.1. Alternative Title*, mostrado na Figura 3. Esta ferramenta permite que o usuário manipule e ordene a lista de valores destes campos de acordo com seus critérios utilizando os ícones presentes ao lado direito do respectivo campo.

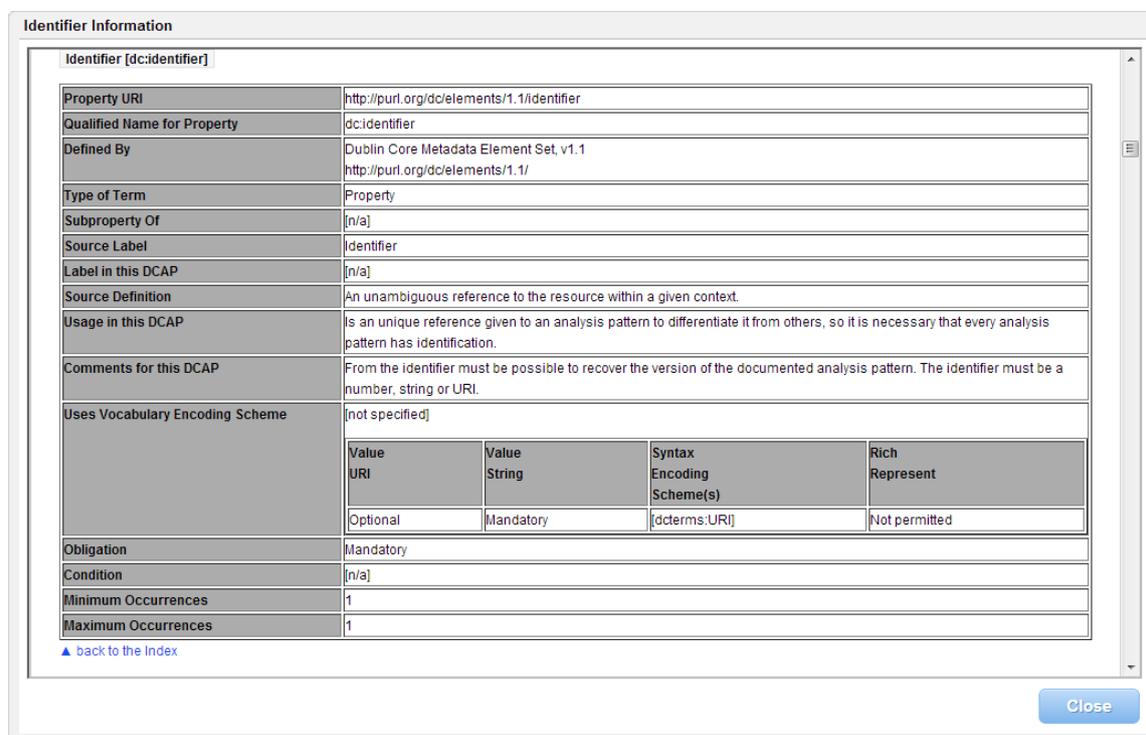


Figura 5. Informações sobre o elemento Identifier

### 3.1. Funcionalidades

Até o momento, o DC2AP Metadata Editor possui duas funcionalidades principais: 1) validação de metadados e 2) geração de documentações no formato RDF, acessíveis respectivamente pelos botões *Validate* e *RDF*, assim como pelo menu *Services*, ambos apresentados na Figura 3.

A funcionalidade *Validate* verifica se os metadados estão de acordo com as regras de obrigatoriedade do perfil DC2AP. Caso existam elementos obrigatórios que não foram informados, a ferramenta irá, então, reportar esta informação ao usuário através de uma janela que contém uma lista dos erros encontrados. Esta janela é mostrada na Figura 6.



Figura 6. Log de erros de validação

Após a validação dos metadados, estes podem ser exportados para um documento no formato RDF por meio da função *Generate RDF*. A próxima sessão apresenta um exemplo de documento RDF gerado por este sistema.

Ao utilizar o DC2AP para documentar padrões de análise em formato RDF, o repositório de metadados proposto pela arquitetura APRI, apresentada na Figura 2, torna-se uma fonte de *Linked Data* no domínio específico de padrões de análise.

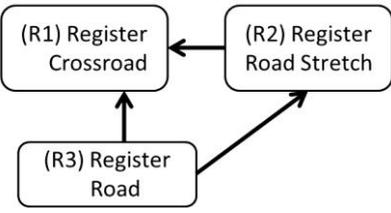
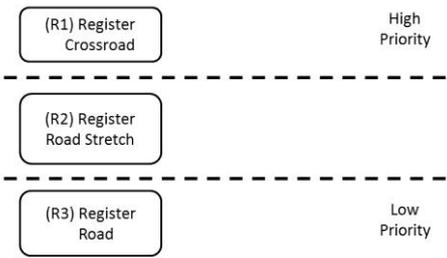
### 3.2. Exemplo

De forma a demonstrar a utilização da ferramenta DC2AP Metadata Editor, bem como do perfil DC2AP na documentação de padrões de análise, voltamos ao padrão proposto por Lisboa-Filho *et al.* (2002) apresentado anteriormente na Figura 1. A Tabela 2 mostra a mesma solução proposta na Figura 1, entretanto utilizando o perfil DC2AP.

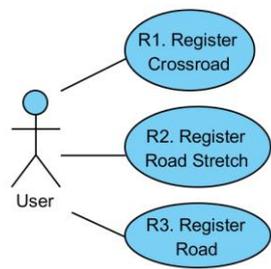
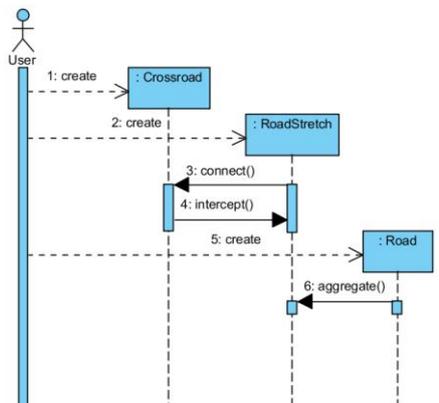
Para simplificar este exemplo, iremos utilizar apenas os campos obrigatórios (*Mandatory*) do DC2AP para descrever o padrão de análise da Figura 1. Sendo assim, qualquer padrão de análise documentado utilizando o perfil DC2AP deve ter pelo menos os campos mostrados na Tabela 2 preenchidos, pois estes representam o núcleo do perfil DC2AP.

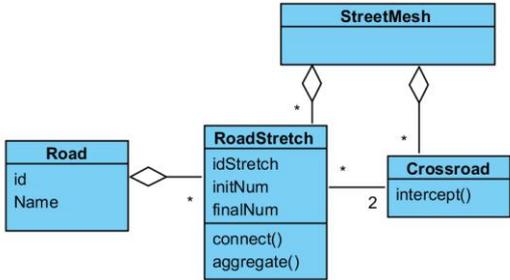
**Tabela 2. Padrão de análise documentado com o perfil DC2AP**

1. <i>Identifier</i> : <a href="http://purl.org/apri/metadata/UrbanStreetMesh-v1">http://purl.org/apri/metadata/UrbanStreetMesh-v1</a>	
2. <i>Title</i> : Urban Street Mesh	
3. <i>Creator</i> : Jugurta Lisboa-Filho, Cirano Iochpe, Karla A. V. Borges	
4. <i>Subject</i> : Sistemas de Gerenciamento Urbano, Sistemas de Informação Geográficas, Ruas	
5. <i>Description</i>	5.1. <i>Problem</i> : Quais os elementos que pertencem a malha de ruas de uma cidade?
	5.2. <i>Motivation</i> : - Cada trecho de uma via é considerado uma instância de rua e deve ter um código de identificação e um nome. Normalmente este deve ser dividido em vários segmentos. - Um trecho de rua é um segmento de percurso entre as duas conexões. - O conjunto formado pelas conexões (ou pontos terminais) e trechos de ruas cria uma malha de ruas urbana.
	5.2.1. <i>Example</i> : - Pode ser usado como base para a criação de um sistema de roteamento de ônibus urbano. - Pode ser usado como base para a criação de um sistema de posicionamento local. - Pode ser usado como base para a criação de um sistema para calcular o menor caminho entre lugares.
5.3. <i>Context</i> : Quase todas as cidades no Brasil tem mostrado o mesmo padrão de organização, que está estruturado pela sua organização percursos (por exemplo: ruas, avenidas, travessas). O conjunto de trechos de percursos gera uma rede de ruas urbanas.	
8. <i>Date</i>	8.1. <i>Created</i> : 01-01-2002
9. <i>Type</i> : Analysis Pattern	9.1. <i>Notation</i> : UML - Unified Modeling Language
10. <i>Format</i> : JPEG, XMI	
12. <i>Language</i> : Português	
16. <i>History</i>	16.1. <i>Event Date</i> : 2002
	16.2. <i>Author</i> : Jugurta Lisboa-Filho, Cirano Iochpe, Karla A. V. Borges
	16.3. <i>Reason</i> : Criação deste padrão de análise
17. <i>Requirements</i>	17.1. <i>Functional Requirements</i> : (R1) Cadastrar Cruzamentos - Usuário deve poder cadastrar todos os cruzamento de uma malha de ruas urbanas. (R2) Cadastrar Trechos de Ruas - Usuário deve poder cadastrar todos os trechos de ruas de uma malha de ruas urbanas. (R3) Cadastrar Ruas - Usuário deve poder cadastrar todas as ruas de uma malha de ruas urbanas.

	<p>17.2. <i>Dependencies and Contributions:</i>  R2(Cadastrar Trechos de Ruas) depende de R1(Cadastrar Cruzamentos) pois todo trecho inicia em um cruzamento e termina em outro. R3(Cadastrar Ruas) depende R2(Cadastrar Trechos de Ruas) e de R1(Cadastrar Cruzamentos) pois toda rua é composta de trechos de ruas, sendo que trechos de ruas possuem cruzamentos.</p>	<p>17.2.1. <i>Dependency Graph:</i>  <a href="http://purl.org/apri/patterns/UrbanStreetMesh-v1Dependency">http://purl.org/apri/patterns/UrbanStreetMesh-v1Dependency</a></p>  <pre> graph TD     R3["(R3) Register Road"] --&gt; R1["(R1) Register Crossroad"]     R3 --&gt; R2["(R2) Register Road Stretch"]     R1 --&gt; R2   </pre>
	<p>17.3. <i>Priorities Diagram:</i>  <a href="http://purl.org/apri/patterns/UrbanStreetMesh-v1Priorities">http://purl.org/apri/patterns/UrbanStreetMesh-v1Priorities</a></p>  <pre> graph TD     subgraph HighPriority [High Priority]         R1["(R1) Register Crossroad"]     end     subgraph LowPriority [Low Priority]         R3["(R3) Register Road"]     end     R2["(R2) Register Road Stretch"]   </pre> <p>17.4. <i>Participants:</i> Usuário.</p>	

**Tabela 2 (cont.). Padrão de análise documentado com o perfil DC2AP**

	<p>18.1. <i>Behaviour</i></p>	<p>18.1.1. <i>Use Case Diagram:</i>  <a href="http://purl.org/apri/patterns/UrbanStreetMesh-v1XMI">http://purl.org/apri/patterns/UrbanStreetMesh-v1XMI</a></p> 
<p>18. <i>Modelling</i></p>	<p>18.1. <i>Behaviour</i></p>	<p>18.1.2. <i>Collaboration/Sequence Diagram:</i>  <a href="http://purl.org/apri/patterns/UrbanStreetMesh-v1XMI">http://purl.org/apri/patterns/UrbanStreetMesh-v1XMI</a></p>  <pre> sequenceDiagram     actor User     participant Crossroad as : Crossroad     participant RoadStretch as : RoadStretch     participant Road as : Road      User-&gt;&gt;Crossroad: 1: create     activate Crossroad     Crossroad-&gt;&gt;RoadStretch: 2: create     activate RoadStretch     RoadStretch-&gt;&gt;Crossroad: 3: connect()     deactivate RoadStretch     Crossroad-&gt;&gt;Road: 4: intercept()     activate Road     Road-&gt;&gt;Crossroad: 5: create     deactivate Road     Crossroad-&gt;&gt;Road: 6: aggregate()     deactivate Crossroad   </pre>

	18.2. Structure	<p>18.2.1. Class Diagram:  <a href="http://purl.org/apri/patterns/UrbanStreetMesh-v1XMI">http://purl.org/apri/patterns/UrbanStreetMesh-v1XMI</a></p>  <pre> classDiagram     class StreetMesh     class Road {         id         Name     }     class RoadStretch {         idStretch         initNum         finalNum         connect()         aggregate()     }     class Crossroad {         intercept()     }     StreetMesh o-- RoadStretch     StreetMesh o-- Crossroad     Road o-- "*" RoadStretch     RoadStretch "1" -- "*" Crossroad     Crossroad "2" -- "*" RoadStretch </pre> <p>18.2.2. Class descriptions:</p> <ul style="list-style-type: none"> <li>- Crossroad: Representa os nodos da malha de ruas que permitem os trechos de ruas serem conectados.</li> <li>- RoadStretch: Um trecho da rua pode ser conectado a outros trechos através de instâncias da classe Cruzamento.</li> <li>- Raod: Esta é uma classe convencional normalmente implementada como uma tabela em um SGBD relacional. Cada classe é feita de vários trechos de rua.</li> <li>- StreetMesh: Esta classe é um fenômeno geográfico representado por um objeto espacial complexo. Nesta classe muitos atributos podem ser definidos relacionando com a malha como um todo.</li> </ul> <p>18.2.3. Relationship descriptions:</p> <ul style="list-style-type: none"> <li>- O relacionamento existente entre TrechoRua e Cruzamento representa que cada trecho de rua tem o início e fim representados por um cruzamento, sendo que um cruzamento pode conectar vários trechos de ruas.</li> <li>- O relacionamento entre Rua e TrechoRua demonstra que uma Rua é um agregado de vários trechos.</li> </ul>
--	-----------------	---

**Tabela 2 (cont.). Padrão de análise documentado com o perfil DC2AP**

21. Consequences	<p>21.1. Positive:</p> <p>- Este padrão de análise é extremamente genérico, sendo assim ele se encaixa em qualquer contexto que envolva malha de ruas urbanas, independente da região, porte da cidade ou qualquer outro fator. Este padrão é facilmente especializado para contextos mais específicos, tendo assim um grande potencial de reuso.</p> <p>21.2. Negative:</p> <p>- Por ser um padrão de análise muito genérico, muitos projetos podem ser atendidos de maneira deficiente por esse padrão de análise, sendo necessário especializá-lo para obter melhores resultados.</p>
------------------	--

A Figura 7 mostra parte do código RDF equivalente à documentação completa do padrão de análise apresentado na Tabela 2. Este código foi gerado pela ferramenta DC2AP Metadata Editor após a validação dos metadados cadastrados para o padrão de análise. Após a geração deste código, o mesmo pôde ser exportado para um arquivo RDF e armazenado em um repositório de metadados de uma APRI.

A versão completa do arquivo RDF apresentado na Figura 7, contendo os demais elementos do perfil DC2AP, pode ser acessado pelo URI contido no campo <dc:identifier> deste arquivo: <http://purl.org/apri/metadata/UrbanStreetMesh-v1>.

Para criar os URI presentes nos arquivos RDF gerados pelo DC2AP Metadata Editor, esta ferramenta foi integrada ao *Persistent Uniform Resource Locators toolkit* (PURL). Este ambiente, disponível na URL <http://purl.oclc.org/docs/index.html>, permite criar identificadores permanentes para recursos na Web que independem da localização real de um documento.

```

RDF Script for These Metadata

<rdf:Description rdf:about="">
  <dc:identifier>http://purl.org/apri/metadata/UrbanStreetMesh-v1</dc:identifier>
  <dc:title>Urban Street Mesh</dc:title>
  <dc:terms:alternative>
    <rdf:Bag>
      <rdf:li>Malha de ruas urbanas</rdf:li>
    </rdf:Bag>
  </dc:terms:alternative>
  <dc:creator>
    <rdf:Seq>
      <rdf:li>Jugurta Lisboa Filho</rdf:li>
      <rdf:li>Cirano Iochpe</rdf:li>
      <rdf:li>Karla A.V. Borges</rdf:li>
    </rdf:Seq>
  </dc:creator>
  <dc:subject xml:lang="por">
    <rdf:Bag>
      <rdf:li>Sistemas de Gerenciamento Urbano</rdf:li>
      <rdf:li>Sistemas de Informação Geográfica</rdf:li>
      <rdf:li>Vias</rdf:li>
      <rdf:li>Ruas</rdf:li>
    </rdf:Bag>
  </dc:subject>
  <dc:description xml:lang="por">
    <dc2ap:problem xml:lang="por">Quais os elementos que pertencem a malha de ruas de uma
    <dc2ap:motivation xml:lang="por">
      <rdf:Bag>
        <rdf:li>Cada trecho de uma via é considerado uma instância de rua e deve ter u
        <rdf:li>Um trecho de rua é um segmento de percurso entre as duas conexões.</rd
        <rdf:li>O conjunto formado pelas conexões (ou pontos terminais) e trechos de r
      </rdf:Bag>
    </dc2ap:motivation>
    <dc2ap:example xml:lang="por">
      <rdf:Bag>
        <rdf:li>Pode ser usado como base para a criação de um sistema de roteamento de
        <rdf:li>Pode ser usado como base para a criação de um sistema de posicionament
  </dc:description>
</rdf:Description>

```

Metadata Validation OK! Save Close

Figura 7. Script RDF para o padrão de análise de Malha de Ruas Urbanas

### 3.3. DC2AP Web Service

O DC2AP Metadata Editor fornece algumas de suas funcionalidades por meio de um serviço Web (DC2APWebService), no qual a troca de mensagens com o ambiente externo é feita através do protocolo SOAP. O objetivo do DC2APWebService é permitir que desenvolvedores de aplicações voltadas para o domínio de padrões de análise reutilizem os serviços do DC2AP Metadata Editor sem a necessidade de acessar a interface do sistema ou reimplementar estes serviços em suas aplicações. Esta abordagem permite que desenvolvedores tenham liberdade de criar sua própria aplicação utilizando recursos fornecidos pelo DC2AP Metadata Editor utilizando linguagens e tecnologias de sua preferência, uma vez que o protocolo SOAP baseia-se nas tecnologias universais XML e HTTP para troca de mensagens. O acesso ao serviço DC2APWebService pode ser feito pela URI: <http://purl.org/dc2ap/webservices>, onde se têm acesso aos seguintes métodos:

- ***generateRdfScript()*** - Criação de script RDF para os metadados de entrada.
- ***validateResource()*** - Validação de metadados segundo as regras do DC2AP.

Os cabeçalhos em Java de cada um dos métodos apresentados são mostrados nas Figuras 8 e 9.

```

1. @WebMethod
2. public String generateRdfScript(
3.     String identifier, String title, String[] alternativeTitle,
4.     String[] creator, String[] subject, String problem,
5.     String[] motivation, String[] example, String[] knownUses,
6.     String context, String[] publisher, String[] contributor,
7.     String created, String modified, String type, String notation,
8.     String[] format, String source, String languageValue,
9.     String languageLabel, String isVersionOf, String[] isReplacedBy,
10.    String[] replaces, String[] isPartOf, String[] hasPart,
11.    String[] isDesignedWith, String[] shouldAvoid,
12.    String[] complementedBy, String about, String[] coverage,
13.    String[] rights, String eventDate, String[] author,
14.    String reason, String changes, String[] functionalRequirements,
15.    String[] nonFunctionalRequirements,
16.    String dependencesAndContributions, String dependencyGraph,
17.    String contributionGraph, String[] conflictAndResolution,
18.    String prioritiesDiagram, String[] participants,
19.    String useCaseDiagram, String[] collaborationSequenceDiagrams,
20.    String[] activityStateDiagrams, String classDiagram,
21.    String[] classDescriptions, String[] relationshipDescriptions,
22.    String[] solutionVariants, String[] resultingContext,
23.    String[] designGuidelines, String[] positive, String[] negative);

```

**Figura 8. Cabeçalho do método *generateRdfScript()***

```

1. @WebMethod
2. public String[] validateResource(
3.     String identifier, String title, String[] creator,
4.     String[] subject, String problem,
5.     String[] motivation, String[] example, String[] knownUses,
6.     String context, String[] publisher, String[] contributor,
7.     String created, String modified, String type, String notation,
8.     String[] format, String source, String languageValue,
9.     String languageLabel, String isVersionOf, String[] isReplacedBy,
10.    String[] replaces, String[] isPartOf, String[] hasPart,
11.    String[] isDesignedWith, String[] shouldAvoid,
12.    String[] complementedBy, String about, String[] coverage,
13.    String[] rights, String eventDate, String[] author,
14.    String reason, String changes, String[] functionalRequirements,
15.    String[] nonFunctionalRequirements,
16.    String dependencesAndContributions, String dependencyGraph,
17.    String contributionGraph, String[] conflictAndResolution,
18.    String prioritiesDiagram, String[] participants,
19.    String useCaseDiagram, String[] collaborationSequenceDiagrams,
20.    String[] activityStateDiagrams, String classDiagram,
21.    String[] classDescriptions, String[] relationshipDescriptions,
22.    String[] solutionVariants, String[] resultingContext,
23.    String[] designGuidelines, String[] positive, String[] negative);

```

**Figura 9. Cabeçalho do método *validateResource()***

O método *generateRdfScript()* recebe como parâmetros todos os elementos do perfil DC2AP, descritos anteriormente na Tabela 1. Este método então converte esta entrada para um script no formato RDF, semelhante ao script apresentado na Figura 7, e

o retorna na forma de uma *String* ao usuário somente após a validação dos metadados pelo método *validateResource()*. Este último verifica se a entrada está de acordo com as regras de validação do DC2AP e então retorna uma lista contendo mensagens de erros de validação, ou uma lista vazia caso nenhum erro tenha sido encontrado. O método *validateResource()* recebe como parâmetros apenas os elementos obrigatórios (*Mandatory*) e condicionais (*Conditional*) do DC2AP.

#### 4. Conclusões e Trabalhos Futuros

O DC2AP Metadata Editor visa aprimorar o processo de documentação de padrões de análise por meio da aplicação do perfil DC2AP. Devido à sua funcionalidade de exportação de metadados para o formato RDF, o DC2AP Metadata Editor permite a criação de coleções digitais de padrões de análise processáveis por máquina e que podem ser encontrados de forma eficaz por motores de busca. Desta forma, a utilização dessa ferramenta em conjunto com o perfil DC2AP possibilita que padrões de análise melhores estruturados e documentados sejam recuperados mais rapidamente e de forma eficiente em repositórios contidos em APRIs distribuídas pela Web. Esta melhoria na eficiência e rapidez se faz possível pelo fato de estes artefatos serem documentados em formatos digitais, uma vez que grande parte dos padrões de análise estão catalogados de forma física; e também no fato de estes serem armazenados e distribuídos pela Web como *Linked Data* no formato RDF, o que é uma abordagem comprovadamente eficiente [Bizer et al., 2009][Hausenblas, 2009].

Quando são descritos como *Linked Data*, os padrões de análise tem o potencial de reutilização expandido e permite uma maior integração destes com outros dados de fontes heterogêneas distribuídas [Bizer et al., 2009].

Por ser um SaaS, o DC2AP Metadata Editor pode ser utilizado por qualquer usuário com acesso a internet independente da tecnologia utilizada. Esta ferramenta também fornece algumas de suas funcionalidades por meio de serviços Web, o que permite que desenvolvedores de aplicações voltadas para o domínio de padrões de análise possam utilizar os recursos do DC2AP Metadata Editor, independentemente da tecnologia de desenvolvimento utilizada e possibilitando assim a reutilização de recursos computacionais.

Como trabalhos futuros pretende-se disponibilizar novas funcionalidades no DC2AP Metadata Editor e disponibilizá-las na forma de novos serviços Web, como, por exemplo, funcionalidades para controle de versão dos documentos RDF que são gerados pela ferramenta, permitindo, desta forma, que o usuário tenha um histórico das alterações e melhorias que foram realizadas na documentação e possa gerenciar suas diferentes versões. Além disso, pretende-se adaptar esta ferramenta para a mesma se comportar também como um editor de metadados Dublin Core genérico, permitindo assim que esta nova extensão seja utilizada não apenas para documentação de padrões de análise, mas para documentar qualquer artefato ao qual o Dublin Core possa se aplicar.

Para além disso, pretende-se avaliar quantitativamente a efetividade de reuso dos padrões documentados com o DC2AP por meio da ferramenta de edição. Como apresentado neste trabalho, o DC2AP Metadata Editor faz parte de uma infraestrutura maior, a APRI, composta por diversos módulos e serviços. No entanto o nosso serviço

de recuperação de dados (Catalog Service) ainda não foi criado. Um dos próximos passos do nosso projeto é a criação deste módulo, tornando possível demonstrar melhorias na recuperação de padrões de análise, uma vez que os dados já terão sido estruturados por meio da ferramenta e do perfil de metadados.

## Agradecimentos

Projeto parcialmente financiado com recursos do CNPq/MCT e da FAPEMIG.

## Referências Bibliográficas

- Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., Lee, G., Pat-terson, D., Rabkin, A., Stoica, I. e Zaharia, M. (2010). A View Of Cloud Computing. *Communications of the ACM*, 53(4), pp. 50-58.
- Berners-Lee, T. (2000). Weaving the Web: *The Past, Present and Future of the World Wide Web by its Inventor*. London: Texere Publishing.
- Berners-Lee, T., Hendler, J. e Lassila, O. (2001). The Semantic Web. *Scientific American*, 284(5), pp. 28-37.
- Berners-Lee, T., Fielding, R. e Masinter, L. (2005). *Uniform Resource Identifier (URI): Generic syntax*, <http://tools.ietf.org/html/rfc3986>. Acesso em: 5 abr. 2013.
- Bizer, C., Heath, T. e Berners-Lee, T. (2009). Linked Data - The story so far. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 5(3), pp. 1-22. Elsevier.
- Blaimer, N., Bortfeldt, A. e Pankratz, G. (2010). *Patterns in object-oriented analysis*. Working Paper No. 451, Faculty of Business Administration and Economics, University of Hagen (Germany).
- CatMDEdit (2002). <http://catmdedit.sourceforge.net>. Acesso em: 14 jan. 2013.
- DCMI - Dublin Core Metadata Initiative. (1998). *Dublin Core metadata element set, v.1.0: Reference description*. <http://www.dublincore.org/documents/1998/09/dces/>. Acesso em: 31 jan. 2013.
- Fernandez, E. B. e Yuan, X. (2009). An Analysis Pattern for invoice processing. In *Proceedings of the 16th Conference on Pattern Languages of Programs (PLoP)*, p. 01-10.
- Hausenblas, M. (2009). Exploiting Linked Data For Building Web Applications. *IEEE Internet Computing*.
- Klyne, G. e Carroll, J. J. (2004). *Resource Description Framework (RDF): Concepts and abstract syntax*. <http://www.w3.org/TR/rdf-concepts/>. Acesso em: 30 jan. 2013.
- Lisboa-Filho, J., Iochpe, C. e Borges, K. A. V. (2002). Analysis Patterns for GIS data schema reuse on urban management applications. *CLEI Electronic Journal*, 5(2), pp. 01-15.
- Meszaros, G. e Doble, J. (1997). A pattern language for pattern writing. In R. C. Martin, D. Riehle, & F. Buschmann (Eds.), *Pattern languages of program design 3*, pp. 529-574. Addison-Wesley.

- Morshed, A., Caracciolo, C., Johannsen, G. e Keizer, J. (2011). Thesaurus alignment for Linked Data publishing. In *International Conference on Dublin Core and Metadata Applications*, p. 37-46.
- Pantoquilha, M., Raminhos, R. e Araújo, J. (2003). Analysis Patterns specifications: Filling the gaps. In *Proceedings of the 2nd Viking Conference Pattern Languages of Programs (PLoP)*, pp. 169-180.
- Peixoto, D. A., Vegi, L. F. M., Jugurta Lisboa-Filho, J. (2013). DC2AP Metadata Editor: A Metadata Editor for na Analysis Pattern Reuse Infrastructure. In *CAiSE Forum*, pp. 138-145.
- Raminhos, R., Pantoquilha, M., Araújo, J. & Ana Moreira. (2006). A systematic Analysis Patterns specification. *Proceedings of the 8th International Conference on Enterprise Information Systems (ICEIS)*, pp. 453-456.
- Rigo, S. J., Schneider, E. E. e Oliveira, J. P. M. (2008). Uma arquitetura para Hipermídia Adaptativa baseada em recursos da Web Semântica. *Revista Brasileira de Sistemas de Informação (iSys)*, (1)1, pp. 70-87.
- Sauermann, L., Cyganiak, R., Ayers, D. e Völkel, M. (2008). *Cool URIs for the Semantic Web*. <http://www.w3.org/TR/cooluris/>. Acesso em: 21 mar. 2013.
- Tamae, R. Y. e Lima, P. R. (2005). Web Services: Uma nova visão da arquitetura de aplicações distribuídas na internet. *Revista científica eletrônica de sistemas de informação*, 1(2), pp. 11-13.
- Vegi, L. F. M. (2012), *Technical description of Dublin Core application profile to Analysis Patterns (DC2AP)*. <http://purl.org/dc2ap/TechnicalDescription>. Acesso em: 05 abr. 2013.
- Vegi, L. F. M., Peixoto, D. A., Soares, L. S., Lisboa-Filho, J. e Oliveira, A. P.. (2012a). An infrastructure oriented for cataloging services and reuse of Analysis Patterns. In *Proceedings of BPM 2011 Workshops (rBPM 2011)*, LNBIP vol. 100, Part 4. Springer, p. 338-343.
- Vegi, L. F. M., Lisboa-Filho, J., Costa, G. L. S., Oliveira, A. P. e Braga, J. L.. (2012b). DC2AP: A Dublin Core application profile to analysis patterns. In *24<sup>th</sup> International Conference on Software Engineering and Knowledge Engineering (SEKE)*, KSI, Skokie, p. 511-516.
- Vegi, L. F. M., Lisboa-Filho, J., Cromptvoets, J. (2012c). A Machine-Processable Dublin Core Application Profile for Analysis Patterns to provide Linked Data. In *International Conference of Dublin Core and Metadata Applications (DC-2012)*, p. 23-32.
- Weiss, A. (2007). Computing in the clouds. *netWorker*, 11(4), pp. 16-25.
- W3C - World Wide Web Consortium (2004). *Web Services Architecture*. <http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/>. Acesso em: 04 abr. 2013.
- xMET Metadata Editor (2011). <http://www.anzlic.org.au/>. Acesso em: 04 abr. 2013.
- Yang, C., Goodchild, M., Huang, Q., Nebert, D., Raskin, R., Xu, Y., Bambacus, M. e Fay, D. (2011). Spatial cloud computing: how can the geospatial sciences use and

help shape cloud computing?, *International Journal of Digital Earth*, 4(4), pp. 305-329.