

5 Exemplos

Foram desenvolvidas duas aplicações que exemplificam o uso e a utilidade das mudanças feitas no método SHDM e do ambiente de desenvolvimento Synth, apresentadas nas seções a seguir.

5.1. Produção bibliográfica da PUC-Rio

Nesta seção apresentamos a aplicação que chamaremos de “Lattes”, cujo objetivo é fornecer um catálogo eletrônico de toda a produção bibliográfica do Departamento de Informática da PUC-Rio. Esta aplicação demonstra um cenário no qual já existe uma fonte de dados em um formato próprio e se deseja publicar estes dados por meio de uma aplicação hipermídia modelada com o SHDM.

Os dados originais utilizados pela aplicação “Lattes” foram extraídos dos currículos dos membros do Departamento de Informática da PUC-Rio cadastrados na Plataforma Lattes⁸³, uma base de dados de currículos, instituições e grupos de pesquisa das áreas de Ciência e Tecnologia mantido pelo CNPq⁸⁴ (Conselho Nacional de Desenvolvimento Científico e Tecnológico). Os dados da Plataforma Lattes são disponibilizados no formato XML e descritos segundo um vocabulário próprio. Esses dados foram convertidos para RDF e descritos segundo a ontologia utilizada para o modelo de domínio da aplicação. Foi utilizada uma ontologia conhecida da comunidade da Web Semântica que foi estendida para atender os requisitos identificados para a aplicação. O passo de conversão dos dados da Plataforma Lattes de XML para RDF não será apresentado aqui por não ter relação direta com o método SHDM.

⁸³ <http://lattes.cnpq.br>

⁸⁴ <http://www.cnpq.br>

5.1.1. Modelagem de domínio

Os conceitos do domínio identificados para esta aplicação já estavam disponíveis no seu vocabulário original: pessoas, organizações, eventos, artigos, livros e outros tipos de documentos do meio acadêmico. Para construir o modelo de domínio da aplicação em RDF, foi selecionada a ontologia Bibliontology⁸⁵, que provê os principais conceitos e propriedades para descrever referências bibliográficas (i.e. livros, artigos etc.) e citações na Web Semântica. Esta ontologia possui vários termos em seu vocabulário e utiliza alguns termos de outros vocabulários conhecidos como FOAF(*Friend of a Friend*) e Dublin Core. Além disso, a ontologia de domínio foi estendida com novos termos para representar informações da fonte de dados original que não possuíam termos equivalentes nos vocabulários da Web Semântica.

A Figura 71 apresenta um diagrama de classes utilizando uma notação no estilo UML que representa a ontologia utilizada para a aplicação.

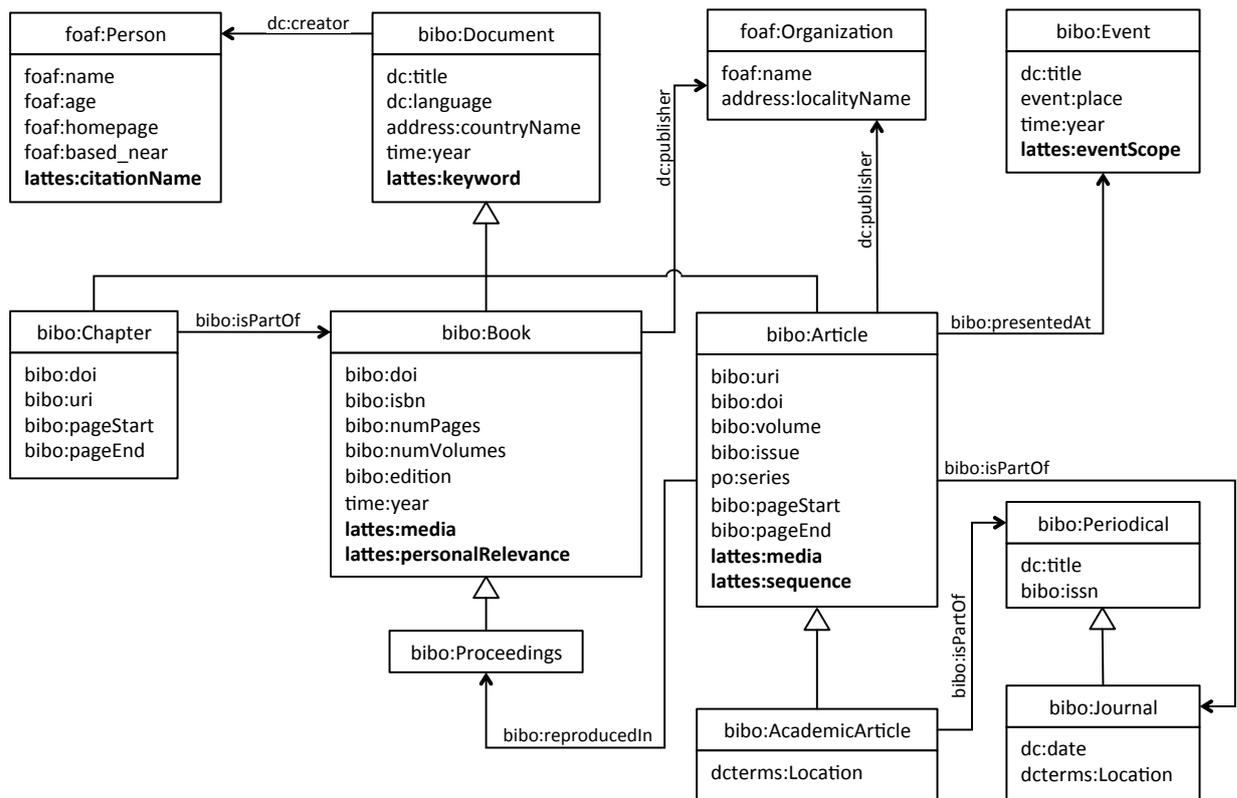


Figura 71 – Ontologia de domínio da aplicação “Lattes”

⁸⁵ <http://bibliontology.com>

A ontologia apresentada na figura anterior usa um subconjunto dos termos da Bibliontology, suficiente para representar os conceitos da aplicação “Lattes”. As duas principais classes da ontologia da aplicação são foaf:Person e bibo:Document (e suas subclasses), para os conceitos Pessoa e Documento. As propriedades apresentadas em negrito na figura acima não faziam parte da ontologia Bibliontology, foram adicionadas no *namespace* “lattes” para representar propriedades do vocabulário específico da base Lattes. Essas propriedades são descritas abaixo:

- lattes:citationName – Nome do autor em citações;
- lattes:keyword – Palavras chaves relacionadas com o documento;
- lattes:eventScope – Indica se o evento é nacional ou internacional;
- lattes:media – Indica se o documento está disponível em meio digital, impresso, magnético ou outros;
- lattes:personalRelevance – Indica se o documento é considerado relevante para o autor;
- lattes:sequence – Número sequencial que indica a ordem de produção por um autor do documento;

5.1.2. Projeto Navegacional

A Figura 72 apresenta o esquema de contextos da aplicação.

Para a classe bibo:Article, foram especificados os contextos “ArticlesAlpha”, para navegação em ordem alfabética, e “byPerson” para artigos escritos por uma pessoa cujos nós são derivados da propriedade dc:creator de bibo:Article. O contexto “byPerson” é parametrizado e espera receber como parâmetro o identificador de um recurso do tipo foaf:Person. Os índices “AllArticles” e “ArticlesByPerson” são baseados nos contextos “ArticlesAlpha” e “byPerson”, respectivamente.

Para a classe foaf:Person, foram especificados os contextos “PersonsAlpha” (navegação em ordem alfabética), “byArticle” e “Co-Authors”. Os nós do contexto “byArticle” também são derivados da propriedade dc:creator de bibo:Article que é o tipo do recurso informado como parâmetro de entrada para o

contexto, que também é parametrizado. O contexto “Co-Authors” deve relacionar uma pessoa (recurso do tipo foaf:Person) com outras pessoas através dos artigos que escreveram em conjunto. Mais adiante será apresentado como são derivados os nós do contexto “Co-Authors”. Os índices “AllPersons”, “PersonByArticle” e “Co-Authors” são baseados nos contextos “PersonAlpha”, “byArticle” e “Co-Authors”, respectivamente.

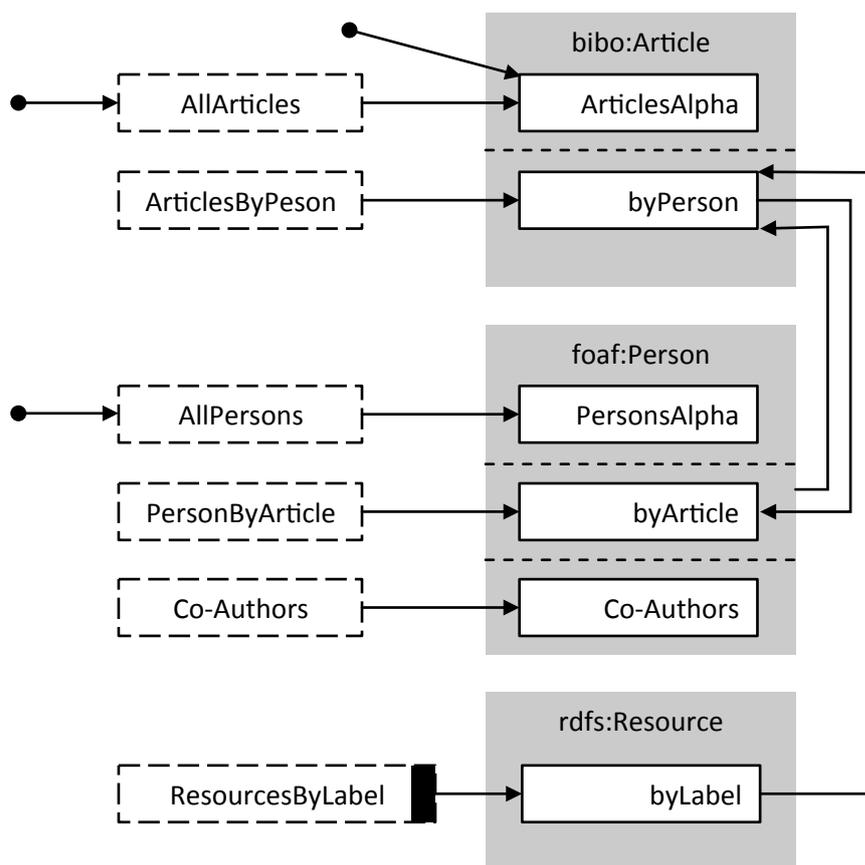


Figura 72 – Esquema de contextos da aplicação “Lattes”

O contexto “byLabel” da classe `rdfs:Resource` é um contexto parametrizado cujos nós são quaisquer recursos da base de dados que possuem como parte do valor das propriedades `rdfs:label`, `foaf:name` ou `dc:title` o valor do parâmetro de entrada passado para o contexto.

A seguir serão apresentadas as especificações dos contextos e índices da aplicação.

O Quadro 29 apresenta a especificação em RDF do contexto “ArticlesAlpha”. A expressão de consulta do contexto está descrita na linguagem SynthQL apresentada na seção 4.4.3.1.1.3, cujo significado é “Todos os recursos

do tipo `bibo:Article` classificados em ordem alfabética do valor da propriedade `dc:title`”.

```
:ArticlesAlpha a shdm:Context ;
  shdm:context_name "ArticlesAlpha";
  shdm:context_title "All Articles";
  shdm:context_query "selects { a BIBO::Article order dc::title }".
```

Quadro 29 – Especificação em RDF do contexto “ArticlesAlpha”

O Quadro 30 apresenta a especificação do contexto “byPerson”. Este contexto contém o parâmetro “person” e também tem sua expressão de consulta em SynthQL. É possível notar na expressão de consulta, propriedade `shdm:context_query`, o uso do parâmetro “person”.

```
:byPerson a shdm:Context ;
  shdm:context_name "byPerson";
  shdm:context_title "Articles Authored by a Person";
  shdm:context_query "selects { type BIBO::Article dc::creator person }";
  shdm:context_parameters [
    a shdm:ContextParameter;
    shdm:context_parameter_name "person".
  ].
```

Quadro 30 – Especificação em RDF do contexto “byPerson”

O Quadro 31 apresenta a especificação do índice “AllArticles”.

```
:AllArticles a shdm:ContextIndex ;
  shdm:index_name "AllArticles";
  shdm:index_title "All Articles";
  shdm:context_index_context :ArticlesAlpha;

  shdm:context_anchor_attributes [
    a shdm:ContextAnchorNavigationAttribute;
    shdm:navigation_attribute_name "article",
    shdm:navigation_attribute_index_position "1",
    shdm:context_anchor_label_expression "self.dc::title",
    shdm:context_anchor_target_context :ArticlesAlpha,
    shdm:context_anchor_target_node_expression "self ",
  ];

  shdm:index_index_attributes [
    a shdm:IndexNavigationAttribute;
    shdm:navigation_attribute_name "Authors",
    shdm:navigation_attribute_index_position "2",
    shdm:index_navigation_attribute_index :byArticle,
    shdm:index_navigation_attribute_index_parameters [
      a shdm:NavigationAttributeParameter,
      shdm:navigation_attribute_parameter_name "article",
      shdm:navigation_attribute_parameter_value_expression "self".
    ].
  ].
```

Quadro 31 – Especificação em RDF do índice “AllArticles”

Nesta especificação é possível observar a associação do índice com o contexto através da propriedade `shdm:context_index_context`. Também são especificados os atributos navegacionais do índice. Neste caso há um atributo do tipo âncora para contexto e outro do tipo índice. Os outros índices da aplicação tem a descrição muito parecida com a do índice “AllArticles”, variando segundo o valor do contexto associado e dos atributos navegacionais, por isso não terão suas descrições apresentadas aqui.

A especificação do contexto “PersonsAlpha” segue a mesma lógica do contexto “ArticlesAlpha” e é apresentada no quadro abaixo.

```
:PersonsAlpha a shdm:Context ;
  shdm:context_name "PersonsAlpha";
  shdm:context_title "All Persons";
  shdm:context_query "selects { a FOAF::Person order foaf::name }".
```

Quadro 32 – Especificação RDF do contexto “PersonsAlpha”

O quadro abaixo apresenta a especificação do contexto “byArticle”, cuja expressão de consulta é descrita em Ruby usando a DSL do ActiveRDF apresentada na seção 4.3.3.

```
:byArticle a shdm:Context ;
  shdm:context_name "byArticles";
  shdm:context_title "Authors by article";
  shdm:context_query "article.dc::creator";
  shdm:context_parameters [
    a shdm:ContextParameter;
    shdm:context_parameter_name "article".
  ].
```

Quadro 33 – Especificação RDF do contexto “byArticle”

O apresenta a especificação do contexto “Co-Authors”. Este contexto é diferente dos demais, porque sua expressão de consulta combina a DSL do ActiveRDF com métodos da linguagem Ruby.

```
:CoAuthors a shdm:Context ;
  shdm:context_name "Co-Authors";
  shdm:context_title "Co-Authors";
  shdm:context_query "
    BIBO::Article.find_by.dc::creator(person).execute.map
    { |a| a.dc::creator }.flatten.uniq.delete_if
    { |p| p == person }.sort_by{|c| c.foaf:name}
  ";
  shdm:context_parameters [
    a shdm:ContextParameter;
    shdm:context_parameter_name "person".
  ].
```

Quadro 34 – Especificação RDF do contexto “Co-Authors”

Como não há uma propriedade no domínio que denota diretamente a coautoria, este relacionamento foi expresso na consulta do contexto. Esta expressão de consulta é equivalente à apresentada no Quadro 21, cujo funcionamento foi explicado na seção 4.4.3.1.1.2.

O Quadro 35 apresenta a especificação do contexto parametrizado “byLabel”. Este contexto recebe como parâmetro um texto qualquer, e retorna todos os recursos RDF que possuem o texto informado como parte do valor da propriedade `rdfs:label`.

```
:byLabel a shdm:Context ;
  shdm:context_name "byLabel";
  shdm:context_title "Resources by Label";
  shdm:context_query "selects { a RDFS::Resource
                        rdfs::label like label_string }";
  shdm:context_parameters [
    a shdm:ContextParameter;
    shdm:context_parameter_name "label_string".
  ].
```

Quadro 35 – Especificação RDF do contexto “byLabel”

Embora o diagrama de classes da ontologia de domínio apresentado na Figura 71 não expresse, as propriedades `foaf:name` e `dc:title` são sub-propriedades de `rdfs:label`. Logo, a avaliação da expressão de consulta também vai retornar os artigos e pessoas de acordo com os valores de suas propriedades `foaf:name` e `dc:title` por inferência.

5.1.3. Outras etapas

Na etapa de Projeto de Interfaces, foi utilizado o modelo de interface *default* fornecido com o Synth logo, não houve tarefa de modelagem de interfaces para esta aplicação. As telas de execução da aplicação apresentadas na seção 4.4 (Figura 54, Figura 60 e Figura 70) são da aplicação “Lattes” com o modelo de interface *default*.

Na etapa de Projeto Comportamental, foi especificada apenas uma operação externa cujo objetivo é retornar a url para acesso ao índice “ResourcesByLabel” baseado no contexto parametrizado “byLabel”.

O Quadro 36 apresenta a especificação da operação “search_url”.

```
:search a shdm:Operation ;
  shdm:operation_name "search_url";
  shdm:operation_type "external"
  shdm:operation_code "SHDM::Index.find_by.index_name('ResourcesByLabel')
.execute.first.url ".
```

Quadro 36 – Especificação da operação “serach_url”

A etapa de Implementação consistiu apenas de cadastrar as especificações apresentadas nesta seção através do ambiente de autoria do Synth apresentado na seção 4.4.

A seguir são apresentadas algumas capturas de telas da aplicação “Lattes”.

The screenshot shows the 'All Persons' page in the Lattes application. The page has a header with 'PUC-Rio bibliographical production' and a search bar. Below the header, there are tabs for 'Persons', 'Articles', and 'Persons context'. The main content area is titled 'All Persons' and includes a 'Filter' section with a 'filter' input field. The list of authors is as follows:

Author Name	CoAuthors	Citation Name	Articles
29-40 1997	Co-Authors	1997, 2.	Reutilização de Projetos Em Aplicações Hipermidia
A. CALOINI	Co-Authors	CALOINI, A.	Hypertext Development Using A Model Based Approach
A. Fortier	Co-Authors	A. Fortier	Combining Meta-level and Logic-Based Constructs in Web Personalization
A. Garrido	Co-Authors	GARRIDO, A.	Reutilização de Projetos Em Aplicações Hipermidia Design Reuse In Hypermidia Application Development Pattern Systems For Hypermedia Designing Computational Hypermedia Applications Navigating Through Objects. Lessons Learned From An Object-Oriented Framework Perspective
A. L. Garrido	Co-Authors	GARRIDO, A. L.	Towards A Pattern Language For Hypermedia Applications
Adriana Pereira de Medeiros	Co-Authors	MEDEIROS, A. P. DE	Representing Design Rationale to support Reuse Kuaba Ontology: Design Rationale Representation and Reuse in Model-Based Designs

Figura 73 – Tela do índice “AllPersons”

PUC-Rio bibliographical production Search

Persons Articles Persons context

All Persons << Previous Next >>

Filter
filter

Context Index

29-40 1997

A. CALOINI

A. Fortier

A. Garrido

A. L. Garrido

Adriana Pereira de Medeiros

Adriana Sodero Rezende

Akeo Tanabe

Alberto Barbosa Raposo

Alessandro Fabricio Garcia

Alexandre Costa Ferreira...

Alexandre Ibrahim Direne

Alexandre Sieira Vilar

Alexandre de Melo Correi...

Ana Cavalli

Ana Maria Nicolaci da Costa

A. Garrido

foaf:givenname A.
foaf:familyName Garrido
lattes:citationOrder 3
foaf:name A. Garrido
lattes:citationName GARRIDO, A.
rdfs:label A. Garrido

Title

[Reutilização de Projetos Em Aplicações Hipermedia](#)
[Design Reuse In Hypermedia Application Development](#)
[Pattern Systems For Hypermedia](#)
[Designing Computational Hypermedia Applications](#)
[Navigating Through Objects: Lessons Learned From An Object-Oriented Framework Perspective](#)

articles

Figura 74 – Tela de acesso a um nó no contexto “PersonsAlpha”

PUC-Rio bibliographical production Search

Persons Articles Persons context

All Articles Um Framework JAVA para Implementação na WWW de Aplicações Hipermedia OOHD

Filter
filter

Authors

[Andrea Miranda Pizzol](#)
[Daniel Schwabe](#)

A Software Architecture for Structuring Complex Web Applications

Authors

[Daniel Schwabe](#)
[Mark Douglas Jacyntho](#)

Patterns for Designing Navigation Spaces

Authors

[Daniel Schwabe](#)
[F. Lvardet](#)
[Gustavo Rossi](#)

Patterns for Adding Search Capabilities to Web Information Systems

Authors

[Daniel Schwabe](#)
[Fernando Lvardet](#)
[Gustavo Rossi](#)

Abstraction, Composition And Lay-Out Definition Mechanisms In Oohdm

Authors

[G. Rossi](#)
[Daniel Schwabe](#)
[Simone Diniz Junqueira Barbosa](#)

Figura 75 – Tela do índice “AllArticles”

PUC-Rio bibliographical production Search

Persons Articles Persons context

All Articles << Previous Next >>

Filter

Context Index

- Um Framework JAVA para I...
- A Software Architecture ...
- Patterns for Designing N...
- Patterns for Adding Sear...
- Abstraction, Composition...
- Patterns for E-Commerce ...
- An Intelligent Tool For ...
- A Pluralistic Knowledge-...
- Real World Trust Policies
- Reutilização de Projetos...
- Design Reuse In Hypermid...
- Using Semantic Web Servi...
- Discovering And Using De...
- Experimenting with Explo...
- Application Modeling for...
- Incremental End-user Que...
- Semantic Contract Suppor...

A Software Architecture for Structuring Complex Web Applications

timeyear 2002
uri http://www2002.org/CDROM/alternate/478/index.html

list of authors
dc:title A Software Architecture for Structuring Complex Web Applications
lattes:sequence 180
lattes:personalRelevance NAO
lattes:media MEIO_DIGITAL
lattes:keyword WWW, Metodologia, Projeto, Hipermedia
biboidentifier http://www2002.org/CDROM/alternate/478/index.html
list of contributors
rdfs:label A Software Architecture for Structuring Complex Web Applications

Figura 76 – Tela de acesso a um nó no contexto “ArticlesAlpha”

PUC-Rio bibliographical production Search

Persons Articles Persons context

Search results

Parameters
label_string: semantic

Filter

Using Semantic Web Services Now
http://www.inf.puc-rio.br/publications/using_semantic_web_services_now

Application Modeling for the Semantic Web
http://www.inf.puc-rio.br/publications/application_modeling_for_the_semantic_web

Incremental End-user Query Construction for the Semantic Desktop
http://www.inf.puc-rio.br/publications/incremental_end-user_query_construction_for_the_semantic_desktop

Semantic Contract Support for E-Business Processes
http://www.inf.puc-rio.br/publications/semantic_contract_support_for_e-business_processes

Design and Implementation of Semantic Web Applications
http://www.inf.puc-rio.br/publications/design_and_implementation_of_semantic_web_applications

Exploring Semantic Web Modeling Approaches for Web Application Design
http://www.inf.puc-rio.br/publications/exploring_semantic_web_modeling_approaches_for_web_application_design

The Use of Adaptive Semantic Hypermedia for Ubiquitous Collaboration Systems
http://www.inf.puc-rio.br/publications/the_use_of_adaptive_semantic_hypermedia_for_ubiquitous_collaboration_systems

A Hybrid Approach for Searching in the Semantic Web
http://www.inf.puc-rio.br/publications/a_hybrid_approach_for_searching_in_the_semantic_web

Trust Policies for Semantic Web Repositories
http://www.inf.puc-rio.br/publications/trust_policies_for_semantic_web_repositories

A Semantic Meta-model for Adaptive Hypermedia Systems
http://www.inf.puc-rio.br/publications/a_semantic_meta-model_for_adaptive_hypermedia_systems

Modeling Applications for the Semantic Web
http://www.inf.puc-rio.br/publications/modeling_applications_for_the_semantic_web

Kuaba approach: Integrating formal semantics and design rationale representation to support design reuse
http://www.inf.puc-rio.br/publications/kuaba_approach_integrating_formal_semantics_and_design_rationale_representation_to_support_design_reuse

Figura 77 – Tela do índice “ResourcesByLabel” com label_string igual a “semantic”

The screenshot shows a web interface for PUC-Rio bibliographical production. At the top, there is a search bar and navigation tabs for 'Persons', 'Articles', and 'Persons context'. The main content area is titled 'Resources By Label' and displays a list of parameters for the selected resource, 'Using Semantic Web Services Now'. The parameters are listed in a table-like format:

Parameter	Value
list of authors	
lattes:keyword	Web Semântica, Peer to Peer, Web Services
lattes:sequence	195
dc:title	Using Semantic Web Services Now
lattes:personalRelevance	NAO
lattes:media	IMPRESSO
time:year	2003
list of contributors	
rdfs:label	Using Semantic Web Services Now

On the left side, there is a 'Context Index' section with a list of related resources, including 'Using Semantic Web Servi...', 'Application Modeling for...', 'Incremental End-user Que...', 'Semantic Contract Suppor...', 'Design and Implementatio...', 'Exploring Semantic Web M...', 'The Use of Adaptive Sema...', 'A Hybrid Approach for Se...', 'Trust Policies for Seman...', 'A Semantic Meta-model fo...', 'Modeling Applications fo...', 'Kuaba approach: Integrat...', and 'Proceedings of the 2nd S...'. A 'Filter' input field is also present.

Figura 78 – Tela de acesso a um nó contexto “byLabel”

5.2. **Semantic Issue Tracker**

Nesta seção será apresentada a aplicação “*Semantic Issue Tracker*”. Um *Issue Tracker* ou “Sistema de Rastreamento de Incidentes” é um sistema para o gerenciamento da criação, atualização e resolução de problemas (incidentes) reportados por clientes ou colaboradores, a respeito de um serviço ou produto. A aplicação “*Semantic Issue Tracker*” é um *issue tracker* cujos dados são mantidos nos padrões da Web Semântica, com o uso do RDF e de ontologias.

Neste exemplo, será demonstrado um cenário em que o projetista não tem controle sobre o domínio da aplicação. Sendo assim, os dados utilizados pela aplicação já estão disponíveis na Web em RDF e descritos segundo uma ontologia própria, combinada com vocabulários conhecidos da comunidade. Neste caso, o objetivo é manter a ontologia de domínio original, sem alterá-la, e fornecer navegabilidade, comportamentos e uma interface para interação dos usuários com os dados descritos segundo esta ontologia.

Foram utilizados os dados do sistema de *issue tracker* do Tabulator⁸⁶, um navegador para dados RDF genérico. Estes dados estão disponíveis para download no endereço <http://tabulator.org/wiki/tabulator/tracker>, em arquivos de texto simples na notação N3⁸⁷ para RDF. Os mantenedores destes dados utilizam a própria ferramenta Tabulator como interface de navegação e edição desses dados. Porém, a utilização do Tabulator nesta solução apresenta os seguintes problemas:

- A alteração de qualquer aspecto da aplicação é feita diretamente no código fonte do Tabulator (*hard coding*⁸⁸), em *javascript*;
- A execução do Tabulator ocorre completamente no navegador *web* do cliente. Da maneira como o Tabulator foi implementado, toda a base de dados da aplicação é completamente carregada na memória do navegador *web*. Por isso, mesmo que o usuário esteja visualizando apenas uma pequena parte dos dados, a quantidade de memória ocupada será do tamanho de toda a base de dados. Como resultado, a aplicação consome mais recursos do cliente do que poderia consumir com uma outra estratégia de implementação e apresenta baixo desempenho.

Nossa proposta foi modelar os aspectos navegacionais, de comportamento e de interface desta aplicação segundo o SHDM, mantendo o modelo de domínio original e implementar a aplicação no ambiente Synth.

5.2.1. Modelo de domínio

A Figura 79 apresenta um diagrama de classes utilizando uma notação no estilo UML que representa a ontologia utilizada para a aplicação. A classe `track:TabIssue` é a classe mais importante da ontologia e representa os “incidentes” reportados pelos usuários.

⁸⁶ <http://www.w3.org/2005/ajar/tab>

⁸⁷ <http://www.w3.org/DesignIssues/Notation3>

⁸⁸ http://en.wikipedia.org/wiki/Hard_coding

e categorias foi solucionada por meio do uso da *DSL* da biblioteca *ActiveRDF* e dos métodos nativos da linguagem *Ruby* na definição dos atributos navegacionais.

Ainda há as classes `user:User` e `wf:Message` que representam respectivamente usuários e mensagens.

Os dados RDF da aplicação foram importados para a base de dados local do *Synth* com auxílio da ferramenta de importação apresentada na seção 4.6, a partir dos arquivos disponíveis no endereço <http://tabulator.org/wiki/tabulator/tracker>.

5.2.2. Projeto Navegacional

A Figura 80 apresenta o esquema de contextos da aplicação.

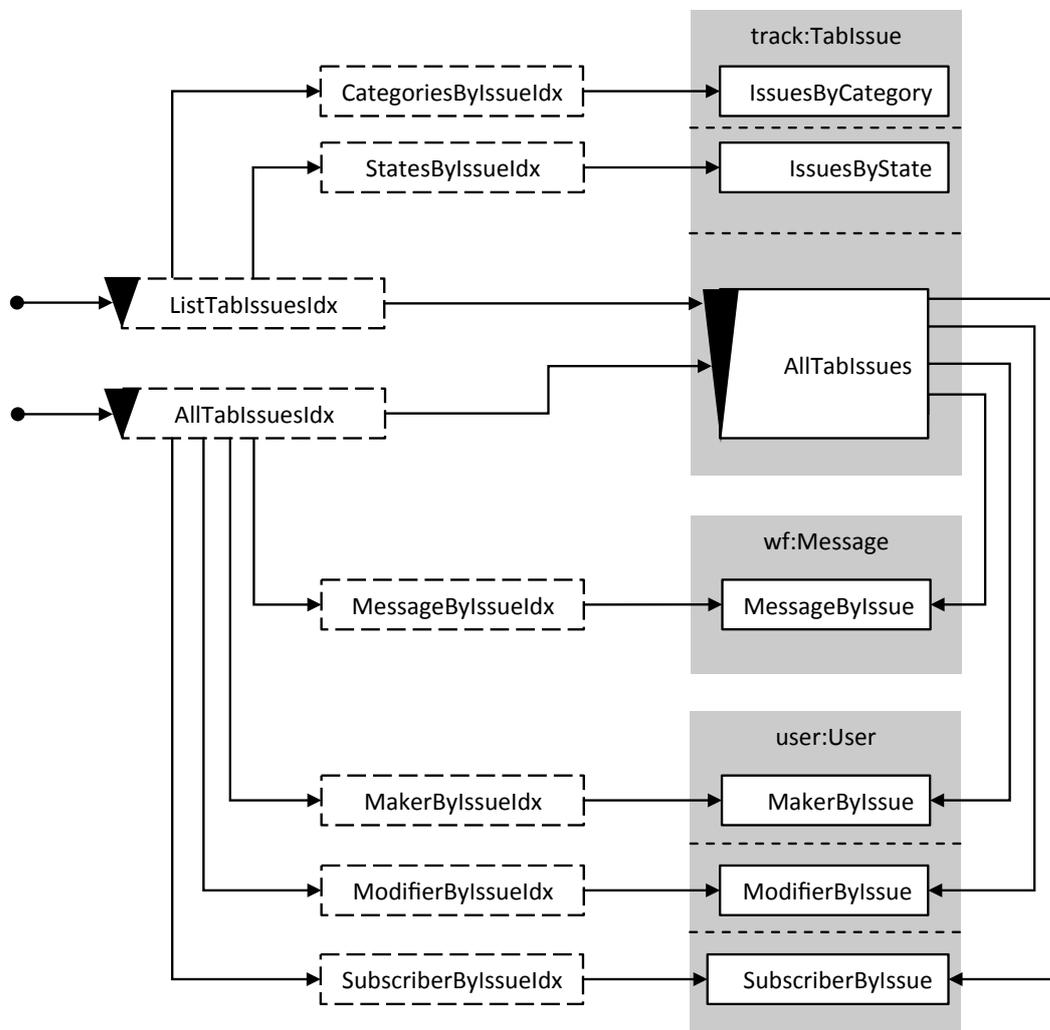


Figura 80 – Esquema de contextos da aplic. *Semantic Issue Tracker*

O principal contexto desta aplicação é o “AllTabIssues” composto pelas instâncias da classe track:TabIssue. O Quadro 37 apresenta a especificação deste contexto em RDF.

```
:AllTabIssues a shdm:Context ;
  shdm:context_name " AllTabIssues";
  shdm:context_title "All Tab Issues";
  shdm:context_query "selects { a TRACK::TabIssue }".
```

Quadro 37 – Especificação em RDF do contexto “AllTabIssues”

A expressão de consulta do contexto “AllTabIssues” está descrita na linguagem SynthQL e significa “todos os recursos do tipo track:TabIssue.

Há dois índices baseados no contexto “AllTabIssues”: “ListTabIssuesIdx” e “AllTabIssuesIdx”. Embora o conjunto de entradas destes índices seja baseado no mesmo conjunto de recursos RDF definidos pela expressão de consulta do contexto “AllTabIssues”, cada um dos índices apresentam atributos navegacionais distintos em suas entradas. O “ListTabIssueIdx” permite navegação para os índices “CategoriesByIssueIdx” e “StateByIssueIdx” que, por sua vez, permitem navegação para os contextos “IssuesByCategory” e “IssuesByState”. Estes dois contextos são compostos pelas instâncias de track:TabIssue relacionadas com as subclasses de track:TabIssue e track:TabIssueCategory que representam, respectivamente, “estados” e “categorias”.

O Quadro 38 apresenta a especificação em RDF do índice “ListTabIssueIdx”.

```
:ListTabIssueIdx a shdm:ContextIndex ;
  shdm:index_name "ListTabIssueIdx";
  shdm:index_title "Tabulator Issue List";
  shdm:context_index_context :AllTabIssues;

  shdm:context_anchor_attributes [
    a shdm:ContextAnchorNavigationAttribute;
    shdm:navigation_attribute_name "Issue";
    shdm:navigation_attribute_index_position 1;
    shdm:context_anchor_label_expression "self.dc::title";
    shdm:context_anchor_target_context :AllTabIssues;
    shdm:context_anchor_target_node_expression "self".
  ];

  shdm:context_anchor_attributes [
    a shdm:ContextAnchorNavigationAttribute;
    shdm:navigation_attribute_name "State";
    shdm:navigation_attribute_index_position 2;
    shdm:context_anchor_label_expression "
      category = self.classes.to_a & TRACK::TabIssue.subclasses
      category.first.rdfs::label
    ";
    shdm:context_anchor_target_context :IssueByState;
    shdm:context_anchor_target_node_expression "self";
```

```

shdm:context_anchor_target_parameters [
  a shdm:NavigationAttributeParameter;
  shdm:navigation_attribute_parameter_name "category";
  shdm:navigation_attribute_parameter_value_expression "
    (self.classes.to_a & TRACK::TabIssue.subclasses).first
  ".
].
];

shdm:computed_attributes [
  a shdm:ComputedNavigationAttribute;
  shdm:navigation_attribute_name "Creation";
  shdm:navigation_attribute_index_position 3;
  shdm:computed_value_expression "self.dcterms::created".
];

shdm:context_anchor_attributes [
  a shdm:ContextAnchorNavigationAttribute;
  shdm:navigation_attribute_name "Category";
  shdm:navigation_attribute_index_position 4;
  shdm:context_anchor_label_expression "
    category = self.classes.to_a & TRACK::TabIssueCategory.subclasses
category.first.rdfs::label
  ";
  shdm:context_anchor_target_context :IssueByCategory;
  shdm:context_anchor_target_node_expression "self";
  shdm:context_anchor_target_parameters [
    a shdm:NavigationAttributeParameter;
    shdm:navigation_attribute_parameter_name "category";
    shdm:navigation_attribute_parameter_value_expression "
      (self.classes.to_a & TRACK::TabIssueCategory.subclasses).first
    ".
  ].
].
];

```

Quadro 38 – Especificação em RDF do índice “ListTabIssueldx”

A especificação apresentada no quadro acima, descreve um índice com quatro atributos navegacionais a saber:

- Issue – uma âncora para o contexto “AllTabIssue”;
- State – uma âncora para o contexto parametrizado “IssueByState”;
- Creation – um atributo computado baseado na propriedade dcterms:created;
- Category – uma âncora para o contexto parametrizado “IssueByCategory”.

Os trechos destacados em negrito no quadro acima, são as expressões em Ruby com a DSL da biblioteca ActiveRDF, e mostram a estratégia para se obter separadamente os “estados” e as “categorias” das instâncias de track:TabIssue, necessários para compor o rótulo das âncoras e os parâmetros para os contextos alvo. A categoria foi obtida a partir da interseção (operador &) do vetor de classes do recurso com o vetor de subclasses de track:TabIssue. De maneira similar, para

se obter as “categorias”, foi utilizada a interseção do vetor de classes do recurso com as subclasses de track:TabIssueCategory.

A especificação do índice “AllTabIssuesIdx é parecida com a de “ListTabIssueIdx”, porém com um conjunto de atributos navegacionais distintos.

Os contextos “IssuesByCategory” e “IssuesByStates” são implementados no Synth pelo uso do contexto parametrizado “IssueByClass”, que recebe como parâmetro, em tempo de execução, uma classe RDF e retorna todas as instâncias desta classe. O quadro abaixo apresenta a especificação RDF do contexto “IssueByClass”, cuja expressão de consulta é descrita na linguagem SynthQL e significa “todas as instâncias da classe recebida no valor do parâmetro ‘klass’”.

```
:IssueByClass a shdm:Context ;
  shdm:context_name "IssueByClass";
  shdm:context_title "Issue By Class";
  shdm:context_query "selects { a klass }";

  shdm:context_parameters [
    a SHDM::ContextParameter;
    shdm:context_parameter_name 'klass'.
  ];
```

Quadro 39 – Especificação RDF do contexto “IssueByClass”

Os contextos “MessageByIssue”, “MakerByIssue”, “ModifierByIssue” e “SubscriberByIssue” são implementados no Synth com o uso do contexto parametrizado “RelatedBy”, que funciona como um meta contexto, pois permite definir qualquer contexto baseado nos relacionamentos de um dado recurso RDF. Este meta contexto recebe como parâmetros um recurso e uma propriedade RDF e retorna as instâncias de valores da propriedade informada em relação ao recurso informado. Por exemplo, se for informado um recurso denotado pelo URI ex:tim, que representa a pessoa “Tim Berners-Lee” e a propriedade foaf:knows, o meta contexto “RelatedBy” deve retornar todos os recursos relacionados com ex:tim pela propriedade foaf:knows, ou seja, os “conhecidos” de ex:tim. O Quadro 40 apresenta a especificação em RDF do meta contexto “RelatedBy”.

```
:RelatedBy a shdm:Context ;
  shdm:context_name "RelatedBy";
  shdm:context_title "Related by";
  shdm:context_query "RDF::Property.new(property, resource).to_a";

  shdm:context_parameters [
    a SHDM::ContextParameter;
    shdm:context_parameter_name 'resource'.
  ];
```

```
shdm:context_parameters [
  a SHDM::ContextParameter;
  shdm:context_parameter_name 'property'.
].
```

Quadro 40 – Especificação em RDF do meta contexto “RelatedBy”

O trecho da especificação destacado em negrito é a expressão de consulta do contexto “RelatedBy” descrita com a DSL da biblioteca ActiveRDF.

A classe `track:TabIssue` foi enriquecida com atributos navegacionais do tipo índice, que permitem a navegação, a partir do nó (instâncias de `track:TabIssue`) para os nós relacionados pelas suas propriedades. Estes atributos são índices navegacionais baseados no índice parametrizado “RelatedByIdx” que é baseado no meta contexto “RelatedBy”, a saber:

- **Subscribers** – índice dos recursos relacionados pela propriedade `wf:subscriber`;
- **Modifiers** – índice dos recursos relacionados pela propriedade `wf:modifiedBy`;
- **Maker** – índice dos recursos relacionados pela propriedade `foaf:maker`;
- **Messages** – índice dos recursos relacionados pela propriedade `wf:message`.

O quadro abaixo apresenta a especificação RDF do atributo navegacional da “Subscribers” da classe `track:TabIssue`. A especificação dos outros atributos seguem o mesmo padrão.

```
track:TabIssue a rdfs:Class ;
  rdfs:label "TabIssue" ;

  shdm:index_attributes [
    a shdm:IndexNavigationAttribute;
    shdm:navigation_attribute_name "Subscribers",
    shdm:index_navigation_attribute_index :RelatedByIdx,
    shdm:index_navigation_attribute_index_parameters [
      a shdm:NavigationAttributeParameter,
      shdm:navigation_attribute_parameter_name "resource",
      shdm:navigation_attribute_parameter_value_expression "self".
    ];
    shdm:index_navigation_attribute_index_parameters [
      a shdm:NavigationAttributeParameter,
      shdm:navigation_attribute_parameter_name "property",
      shdm:navigation_attribute_parameter_value_expression
        "WF::subscriber".
    ]
  ].
```

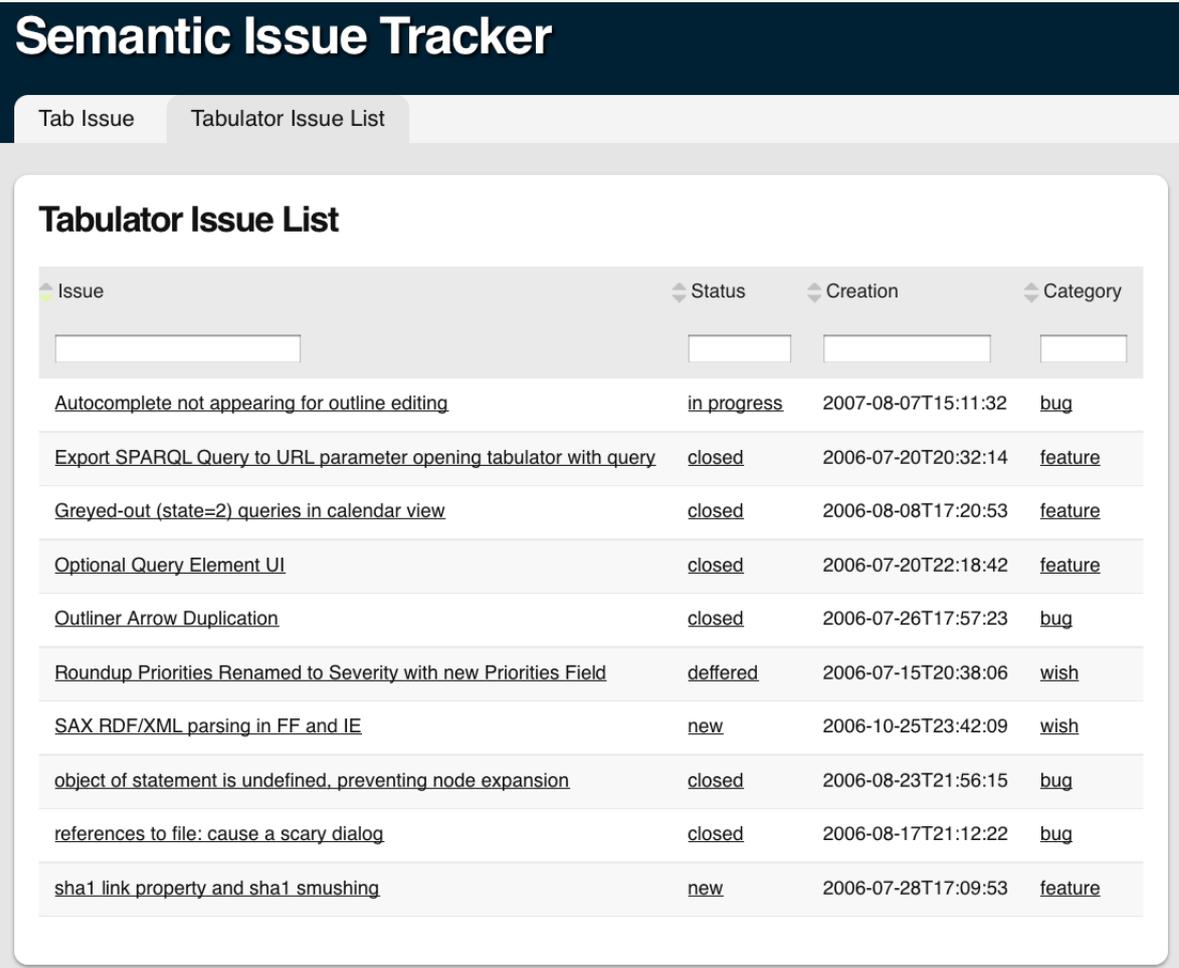
Quadro 41 – Especificação em RDF do atributo “Subscribers” da classe `track:TabIssue`

5.2.3. Outras etapas

Na etapa de Projeto de Interfaces, foi criada uma interface com descrição concreta com a linguagem de *templates* ERB suportada pelo *framework* Ruby on Rails para os índices “ListTabIssuesIdx” e “AllTabIssuesIdx”. Para os contextos foram usadas as interfaces *default* do Synth, porém estas interfaces são exibidas dentro de uma caixa de diálogo, popularmente conhecida como *lightbox*, sobre o índice a partir do qual o contexto é acessado.

Na etapa de Projeto Comportamental, foram utilizadas as operações pré-definidas do editor RDF do Synth, para a edição dos dados da aplicação. Este uso mostrou mais uma característica interessante da implementação de aplicações com o Synth, que é o uso dos recursos já implementados no ambiente como parte integrante da aplicação final.

As telas a seguir apresentam o resultado da aplicação de cada interface e do uso do editor RDF como parte da aplicação.



Issue	Status	Creation	Category
Autocomplete not appearing for outline editing	in progress	2007-08-07T15:11:32	bug
Export SPARQL Query to URL parameter opening tabulator with query	closed	2006-07-20T20:32:14	feature
Greyed-out (state=2) queries in calendar view	closed	2006-08-08T17:20:53	feature
Optional Query Element UI	closed	2006-07-20T22:18:42	feature
Outliner Arrow Duplication	closed	2006-07-26T17:57:23	bug
Roundup Priorities Renamed to Severity with new Priorities Field	deferred	2006-07-15T20:38:06	wish
SAX RDF/XML parsing in FF and IE	new	2006-10-25T23:42:09	wish
object of statement is undefined, preventing node expansion	closed	2006-08-23T21:56:15	bug
references to file: cause a scary dialog	closed	2006-08-17T21:12:22	bug
sha1 link property and sha1 smushing	new	2006-07-28T17:09:53	feature

Figura 81 – Tela do índice “ListTabIssuesIdx”

Autocomplete not appearing for outline editing

activity	2010-08-26T17:27:02Z
creation	2007-08-07T15:11:32
summary	Autocomplete not appearing for outline editing
rdfs:label	Autocomplete not appearing for outline editing
Messages	Label Autocomplete does not appear when trying to edit literals. This is not
Maker	Label Jim Hollenbach
Modifiers	Label Jim Hollenbach
Subscribers	Label KangHao Lu (Kenny) Jim Hollenbach
creator_name	Jim Hollenbach

Issue 1 of 10

Figura 82 – Tela de acesso a um nó no contexto “AllTabIssues”

Issues By Class << Previous Next >>

Parameters
klass: in progress

Context Index

- FOAF pane for social net...
- Problems with rdflib.js ...
- native GRDDL support
- Autocomplete not appeari...
- Request and undetermined...
- save, edit, update
- reflect CC license in ou...
- Handle rdf:datatype=XMLL...

Autocomplete not appearing for outline editing

activity	2010-08-26T17:27:02Z
creation	2007-08-07T15:11:32
summary	Autocomplete not appearing for outline editing
rdfs:label	Autocomplete not appearing for outline editing
Messages	Label Autocomplete does not appear when trying to edit literals. This is not
Maker	Label Jim Hollenbach
Modifiers	Label Jim Hollenbach
Subscribers	Label KangHao Lu (Kenny) Jim Hollenbach
creator_name	Jim Hollenbach

Figura 83 – Tela do contexto “IssueByClass” (implementação de “IssueByCategory” e “IssueByState”)

RelatedBy << Previous Next >>

Parameters

property: wf:message

resource: Export SPARQL Query to URL parameter opening tabulator with query

Context Index

- needs a tutorial chapter...
- Documentation also writt...
- Documentation written in...
- In the SPARQL view, ther...

messageid <1154377891.55.0.711056263944.issue28@csail.mit.edu>

creation 2006-07-31T20:31:31

date 2006-07-31.20:31:31

activity 2006-07-31T20:31:31

content Documentation also written into querytut.html

summary Documentation also written into querytut.html

rdfs:label Documentation also written into querytut.html

Label

Issue [Export SPARQL Query to URL parameter opening tabulator with query](#)

Figura 84 – Tela do Contexto “RelatedBy”, apresentando um nó no contexto “MessageByIssue”

Semantic Issue Tracker

Tab Issue Tabulator Issue List

Tabulator Issues

Summary	creation	activity	Maker	Message
Autocomplete not appearing for outline editing	2007-08-07T15:11:32	2010-08-26T17:27:02Z	Jim Hollenbach	Autocomplete does not appear when trying to edit literals. This
Export SPARQL Query to URL parameter opening tabulator with query	2006-07-20T20:32:14	2006-08-03T21:42:08	Adam Lerer	needs a tutorial chapter, maybe some tests Documentation also written into querytut.html Documentation written into Help.html. Tests??? This is a UI thi In the SPARQL view, there is an option "Get link to selected qu
Greyed-out (state=2) queries in calendar view	2006-08-08T17:20:53	2006-08-09T20:02:37	Adam Lerer	done! testing whether a query is calendarable involves checkir In the map-view, queries that are not mappable are greyed-out
object of statement is undefined, preventing node expansion	2006-08-23T21:56:15	2006-08-31T23:57:52	Joyce	Turned out to be util.js not handling multiline HTTP headers. F I've found another case: 1. http://dig.csail.mit.edu/2005/ajar/ajaw/tab.html?uri=http%3A//w/query.fcgi%3Fcmd%3DRetrieve%26db%3DOMIM%26dopt%3 Following the below instructions, somehow the request object
Optional Query Element UI	2006-07-20T22:18:42	2006-08-02T18:13:33	David Sheets	There should be a GUI feature to make a field optional The UI part is shaping up to be a little button for each of the se Resolved, plus doc added to Help.html in Rev#1819 rev 1819 seems to be irrelevant. I found relevant edits to Help, grumble... why does roundup keep reopening these things?
Outliner Arrow Duplication	2006-07-26T17:57:23	2006-08-04T19:35:55	David Sheets	http://dig.csail.mit.edu/2005/ajar/ajaw/tab.html?uri=http%3A//a/edit/urn%3Aisid%3Aabdera.watson.ibm.com%3Aentries%3A I figured out where this was going on while doing tooltip load Resolved with revision 1947. Bug: emptyNode was emptying t

Figura 85 - Tela do índice “AllTablssueidx”

The screenshot displays the 'Properties' tab of an RDF editor. At the top right, there is a 'Delete resource' button. The main area is a table of properties for the resource 'issue:issue226'. Each property has a plus sign to its left, indicating it can be expanded. The properties listed are:

- uri**: issue:issue226
- activity**: "2010-08-26T17:27:02Z"
- creation**: "2007-08-07T15:11:32"
- creator**: Jim Hollenbach
- issue:topic**: issue:keyword1, issue:keyword17
- rdf:type**: open, rdfs:Resource, track:TabIssueCategory, in progress, wf:Task, bug, Tabulator Issue
- rdfs:label**: Autocomplete not appearing for outline editing
- summary**: "Autocomplete not appearing for outline editing"
- wf:assignee**: KangHao Lu (Kenny)
- wf:message**: Autocomplete does not appear when trying to edit literals. This is not
- wf:modifiedBy**: Jim Hollenbach
- wf:subscriber**: Jim Hollenbach

At the bottom left, there are navigation arrows and the text 'Issue 1 of 10'. At the bottom right, there is a close button (X).

Figura 86 – Tela de edição de uma instância de track:TabIssue, resultado da integração da aplicação com o editor RDF do Synth