

4

Solução proposta

Neste capítulo é apresentada a solução proposta para a resolução dos problemas apresentados nos capítulos anteriores. Para isso foi construída uma ferramenta que possibilita usuários explorarem bases RDF e montarem aplicações de consulta que podem ser compartilhadas de diversas formas e aplicadas a outras bases que sigam as mesmas ontologias. Além disso, as aplicações podem ser facilmente customizadas e evoluídas utilizando o mesmo ambiente em que elas foram criadas. Essa é uma característica importante especialmente na área da web semântica devido a dois fatores: pela grande quantidade de usuários, que pede por soluções facilmente customizáveis e pela rápida evolução das ontologias, que é prevista e suportada pelo modelo RDF.

O desenvolvimento foi realizado em cima de uma ferramenta já existente de exploração a bases RDF, chamada Explorator através de diversos novos conceitos que foram acrescentados em cima da sua arquitetura. O modelo de exploração a bases RDF do Explorator se tornou o ambiente de desenvolvimento de aplicações da nova ferramenta desenvolvida.

Os novos conceitos criados foram:

- Criação de customização de operações – Através da criação e customização de operadores, o universo de consultas possíveis de serem realizadas foi expandido. A generalização dos operadores existentes no Explorator foi necessária para permitir a criação de um gerador de aplicações.
- Reorganização da área de trabalho – A área de trabalho foi reformulada com a criação do conceito de workbenchs. Cada workbench possui uma consulta própria, considerando que uma consulta é um conjunto de operações que visa solucionar um problema. Workbenchs são utilizados para compartilhar soluções e se transformam em casos de uso na geração de uma aplicação.

- A parametrização de consultas – A parametrização de uma consulta torna possível a sua generalização. Esse conceito é a chave para o re-uso, tornando possível o compartilhamento das consultas e a também a sua utilização na geração de aplicações.
- Compartilhamento aberto de consultas – Esse conceito permite o compartilhamento de consultas entre usuários no ambiente de desenvolvimento. Com o compartilhamento aberto é possível, por exemplo, evoluir colaborativamente consultas ou importar e customizar casos de usos de aplicações construídas por outros usuários.
- Compartilhamento fechado de consultas – É o conceito que permite a geração de aplicações de consultas. Para isso são definidos índices e as consultas parametrizadas são interligadas, transformando os workbenches nos casos de uso da aplicação. Através das aplicações geradas, usuários que não possuem conhecimento do modelo RDF podem reutilizar as soluções construídas.

Esse capítulo se divide em três partes. Na seção 4.1 é apresentada uma idéia geral da solução, que foi chamada de Excelplorator, indicando a forma que ela se propõe a solucionar os problemas e desafios previamente apresentados. A seção 4.2 é dedicada à ferramenta que foi utilizada como base para a implementação. Finalmente, na seção 4.3 examinamos individualmente cada novo conceito, discutindo seu papel na solução final. Nessa parte final do capítulo será demonstrado com exemplos como a ferramenta suporta as operações apresentadas na primeira parte.

4.1 Excelplorator

Para solucionar os problemas que foram apresentados, foi desenvolvida uma ferramenta que permita a um usuário realizar diversas ações em cima de bases RDF, possibilitando que ele monte uma aplicação web para consultas a essa base.

Para exemplificar o tipo de aplicação que pode ser montada pelo Excelplorator, tomaremos como exemplo uma base RDF/RDFS que possua dados

de publicações científicas, palestras, e outros eventos que ocorrem em uma conferência de web semântica. A base em questão segue uma ontologia que descreve dados relacionados a conferências científicas⁶. Uma aplicação pode ser construída para que pessoas possam realizar consultas a eventos e publicações que ocorreram em certas conferências que possuem seus dados na base. Usando o Excelplorator para gerar tal aplicação, é possível demonstrar de uma forma mais prática a sua finalidade. A aplicação exemplo em questão possui os seguintes casos de uso:

1. Consultar publicações científicas por autor. Nesse caso de uso é exibida para o usuário uma lista com os autores de publicações disponíveis da base. Ao selecionar um autor, suas publicações seriam exibidas. Nesse momento, o usuário pode selecionar uma publicação para consultar mais dados disponíveis na base sobre a mesma.
2. Consultar co-autores de um pesquisador. O objetivo é selecionar um autor de publicações de uma lista e então visualizar outros pesquisadores que trabalharam com a pessoa selecionada em publicações. As publicações em que os autores trabalharam juntos também são exibidas.
3. Pesquisadores de uma mesma instituição. Após selecionar uma pessoa de uma lista de pesquisadores que existem na base, uma lista com outros pesquisadores da mesma instituição que o selecionado pertence é exibida. Assim como a instituição na qual os dois (pesquisador selecionado e cada um dos pesquisadores que retornaram da consulta) trabalham juntos.
4. Publicações similares a uma escolhida. Nesse caso de uso, as publicações existentes na base são exibidas e o usuário pode selecionar uma. Nesse momento é exibida uma lista com outras publicações da base que possuem semelhança com a publicação escolhida. As publicações consideradas semelhantes são aquelas que possuem palavras-chave iguais.

⁶ Semantic Web Conference Ontology - <http://data.semanticweb.org/ns/swc/ontology>

5. Busca de publicações por palavras-chave. Ao acessar o caso de uso, é exibido para o usuário um campo texto para que ele digite uma palavra. Ao submeter o formulário, é exibida uma lista com as publicações que possuem a palavra inserida pelo usuário como palavra-chave.
6. Busca facetada de publicações. O objetivo do caso de uso é encontrar publicações refinando a sua busca pelos dados existentes na base. Para começar, é exibida uma lista com as instituições de pesquisa existentes na base. Ao selecionar uma instituição, é exibida uma lista com os pesquisadores da instituição que possuem alguma publicação. É possível então selecionar um autor e visualizar todas as suas publicações.

A aplicação descrita foi gerada através do Excelplorator. Ao ser acessada por um usuário através de um navegador, a aplicação exibe uma tela inicial (Figura 1) que possui âncoras para acessar todos os casos de uso descritos (Figura 2).



Figura 1 – Tela inicial da aplicação gerada.

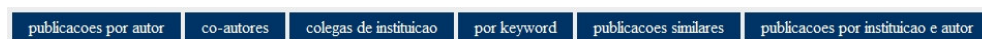


Figura 2 – As âncoras de acesso aos casos de uso.

Ao clicar na primeira âncora, é exibida uma lista de pessoas que existem na base que são autores de alguma publicação (Figura 3). Ao escolher uma pessoa, é exibida uma lista com suas publicações (Figura 4). Nesse momento é possível

clicar nas publicações para saber detalhes sobre a mesma (Figura 5), finalizando o caso de uso.



Figura 3 – Lista com os autores da base.

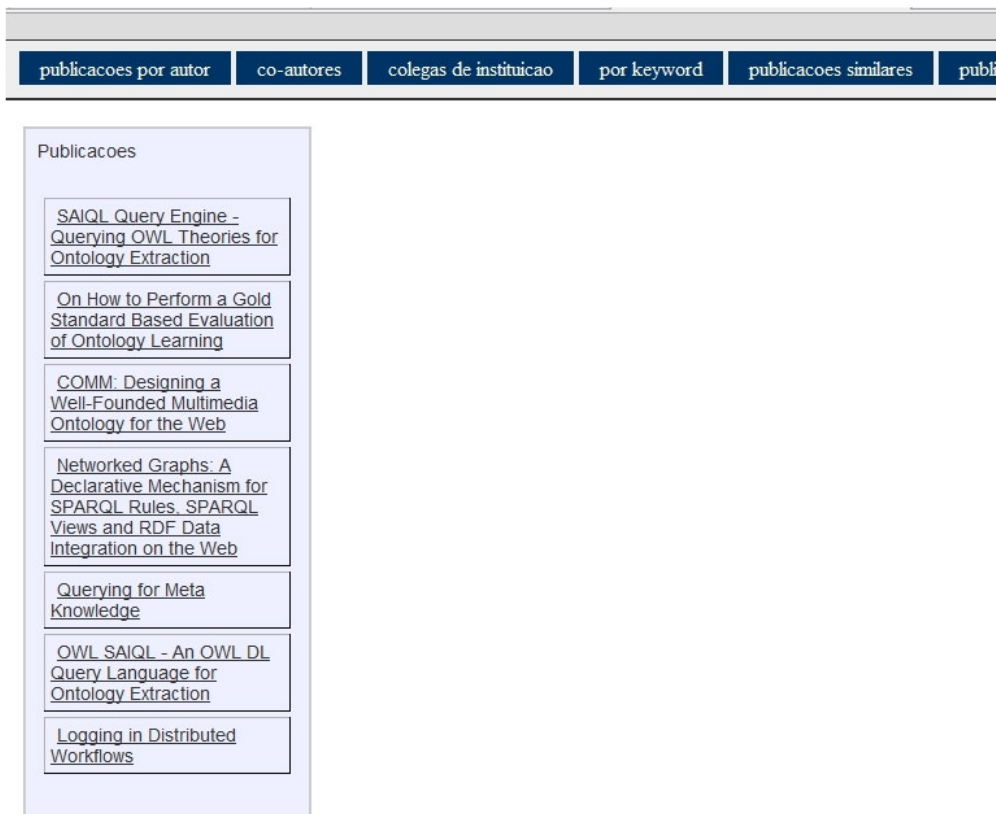


Figura 4 – Publicações do autor selecionado.

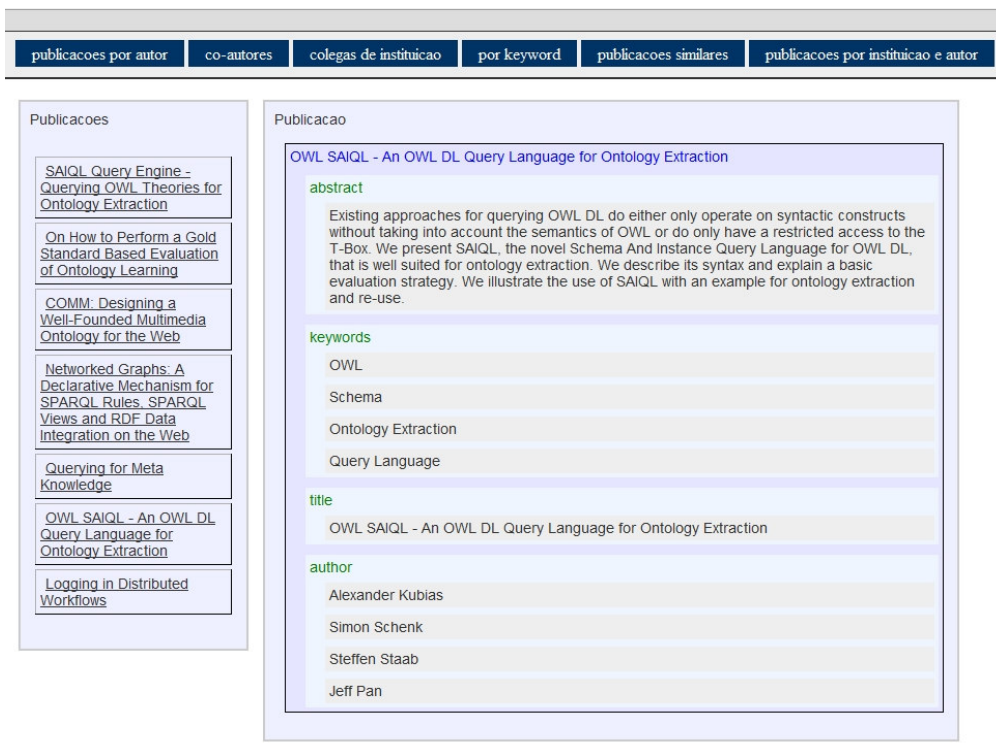
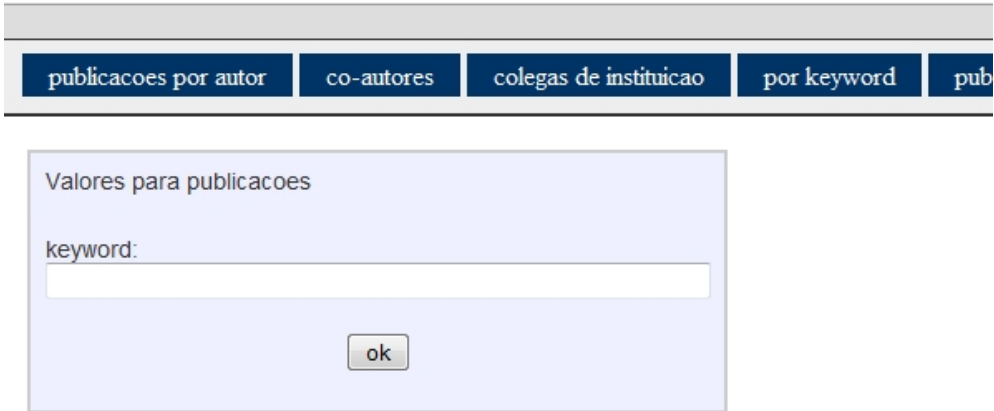


Figura 5 – Detalhes de uma publicação.

Já no quinto caso de uso (busca por palavra-chave) existe um campo de entrada de texto que é exibido quando o usuário acessa o seu link (Figura 6). No campo exibido deve ser digitada a palavra-chave desejada. Após a submissão do formulário, uma lista com as publicações encontradas é exibida. De forma semelhante ao primeiro caso de uso, o usuário pode selecionar uma das publicações para ver mais detalhes.



The image shows a web interface for searching publications. At the top, there is a horizontal navigation bar with five buttons: 'publicacoes por autor', 'co-autores', 'colegas de instituicao', 'por keyword', and 'pub'. Below this bar is a light blue dialog box titled 'Valores para publicacoes'. Inside the dialog, there is a label 'keyword:' followed by a text input field. At the bottom center of the dialog is an 'ok' button.

Figura 6 – Formulário para entrada de valores pelo usuário.

Os outros casos de uso funcionam de forma semelhante. Existem algumas observações importantes relacionadas à aplicação de consulta que foi gerada pelo Excelplorator. Como ela foi montada no ambiente de desenvolvimento da ferramenta, ela pode ser customizada e estendida facilmente no mesmo ambiente, podendo ter seus casos de uso reaproveitados em outras aplicações, como será discutido na seção 4.3 deste capítulo. Outra observação importante é que a aplicação pode ser re-utilizada com qualquer base que siga a mesma ontologia, ou seja, é possível adicionar novas bases contendo dados de outras conferências. Isso é possível porque a aplicação foi montada utilizando somente dados referentes à ontologia e não específicos da base em questão. A terceira observação é para o fato de que qualquer usuário pode utilizar a aplicação, mesmo sem nenhum conhecimento de RDF ou web-semântica. Isso ocorre porque a aplicação gerada abstrai totalmente o modelo RDF, se comportando como uma aplicação web comum. Este fato contribui para o aumento da acessibilidade à dados contidos em bases RDF.

No resto desta seção, será discutido como é possível que um usuário monte aplicações similares a do exemplo acima utilizando o Excelplorator. Os conceitos utilizados serão mais profundamente discutidos nas seções seguintes.

Seguindo uma sequência de ações, pode-se demonstrar como o Excelplorator possibilita que um usuário explore bases de dados em RDF, realize consultas, generalize suas consultas para que possam ser reutilizadas e por fim compartilhe as consultas de diversas formas, principalmente através da montagem de aplicações simples de consulta que podem ser utilizadas por outros usuários, que é o caso da aplicação discutida no exemplo acima.

O primeiro passo é permitir que o usuário consiga acessar as bases desejadas. Depois, é necessário que o usuário consiga extrair das bases RDF em questão as informações que ele deseja. Para isso, o Excelplorator permite a realização de consultas com grau de complexidade variável com relativa facilidade. É relativamente simples realizar uma consulta considerada pouco complexa, como por exemplo, consultar todas as pessoas cadastradas na base (assumindo uma base que possui uma ontologia que descreve pessoas). E também é possível realizar uma consulta complexa, que atenda a demanda de informações mais refinadas, por exemplo, encontrar publicações científicas relacionadas a assuntos específicos e de autoria de pesquisadores ligados a determinados institutos de pesquisa (assumindo bases que possuem uma ontologia que descreva publicações científicas).

Uma vez que o usuário consiga obter as respostas para as perguntas de seu interesse, a ferramenta possibilita a generalização das consultas para que seja possível sua reutilização. Para exemplificar o passo de generalização e reutilização, vamos considerar um usuário que procure todas as publicações de um autor específico em uma base RDF/RDFS de publicações científicas, de forma similar ao primeiro caso de uso da aplicação utilizada como exemplo. Para isso ele precisa explorar a ontologia utilizada na base e conseguir, a partir de um autor (que é um recurso na base) e a propriedade que possui a semântica de autoria, uma lista de recursos que representam as publicações do autor desejado. Para que o usuário possa reutilizar a consulta realizada, é necessário que o autor utilizado na consulta seja parametrizado. Dessa forma, o usuário poderia selecionar uma nova pessoa, utilizar a consulta com a nova pessoa selecionada e obter como resultado as publicações de outro autor. É importante ressaltar que, assim como na consulta

demonstrada no exemplo acima, a maioria das consultas montadas, depois de parametrizadas, só fará referência a recursos das ontologias presentes na base. Por isso, é possível, como já mencionado, reutilizá-las em outros repositórios de RDF que utilizem as mesmas ontologias. Seguindo com o exemplo de uma ontologia que descreve publicações científicas podemos construir a consulta em uma base que possui dados referentes a uma conferência específica e depois reutilizar a consulta construída em bases que possuem dados de outras conferências.

Após ter montado e generalizado consultas do seu interesse, o usuário pode optar por compartilhar suas soluções, de forma que outros usuários possam utilizar as consultas que foram montadas por ele. Existem duas formas de compartilhamento disponíveis no Excelplorator. A primeira forma é no ambiente de desenvolvimento de consultas, chamado de compartilhamento aberto. A outra forma é menos flexível, em um ambiente que abstrai o conceito de RDF, ideal para usuários com menos conhecimentos técnicos. Denominamos essa forma de compartilhamento fechado. A primeira forma de compartilhamento é a mais simples de ser realizada. O usuário, depois de montar e parametrizar sua consulta, escolhe compartilhar a mesma, e outros usuários podem fazer uma cópia da consulta em questão para a sua área de trabalho. Uma vez que a consulta esteja copiada, é possível modificá-la ou expandi-la utilizando o mesmo ambiente em que ela foi construída. Dessa forma, consultas também podem ser evoluídas colaborativamente. A segunda forma de compartilhamento, chamada de compartilhamento fechado, é um pouco mais complexa de ser realizada. O usuário deve escolher através da ferramenta exatamente quais os dados da consulta que serão exibidos e quais opções para substituição dos parâmetros que estarão disponíveis no compartilhamento. No nosso exemplo de publicações por autor, o usuário deve escolher quais atributos da publicação devem aparecer e quais os autores estão disponíveis para serem escolhidos. Esse controle mais rígido se torna necessário para que seja possível abstrair o fato que a consulta está sendo montada em cima de triplas RDF. Dessa forma um segundo usuário que deseje reutilizar a aplicação montada pelo primeiro, não precisa conhecer o ambiente do Excelplorator, a ontologia utilizada nas bases acessadas e tampouco a abstração RDF. As consultas podem ser acessadas como uma aplicação web montada para responder a perguntas previamente especificadas, como foi visto no começo desta seção na aplicação exemplo.

A forma que o Excelplorator permite que o usuário realize as ações especificadas acima será discutida nas próximas partes deste capítulo. A primeira parte, onde o usuário acessa bases RDF e monta uma consulta estática é descrita na seção 4.2. Nesse ponto faremos uma discussão sobre a ferramenta Explorator e sua arquitetura. O Explorator serviu de base para a implementação do Excelplorator e já suportava originalmente essas funcionalidades básicas. O ambiente do Explorator foi utilizado como ambiente de desenvolvimento de aplicações no Excelplorator. Consideramos que uma consulta é estática quando ela não possui nenhuma variável parametrizada, ou seja, não é possível reutilizar a consulta utilizando novos parâmetros nem recalcular os seus resultados. Mais adiante, na seção 4.3 discutiremos novos conceitos implementados que tiveram por finalidade aumentar o poder das consultas originais possibilitadas pelo Explorator e possibilitar a generalização, reutilização e o compartilhamento aberto e fechado das consultas realizadas.

4.2 O Explorator

O modelo de exploração de bases RDF do Explorator possui um conjunto de operações pré-definidas e uma interface que possibilita o usuário realizar essas operações utilizando os próprios dados da base, de uma forma que não seja necessário conhecimento prévio do domínio. As principais operações disponíveis são:

- Busca de recursos através de texto livre.
- União, interseção e diferença de um conjunto de recursos.
- Expressões SPO, onde é possível selecionar um conjunto de recursos para o papel de S (sujeito), P (predicado) ou/e O (objeto) de uma tripla RDF da base.

As três operações resultam na criação de um conjunto de recursos. Com base em conjuntos de recursos é possível realizar novas operações, resultando em consultas cada vez mais complexas em cima da base.

Pode-se exemplificar a utilização do Explorator através da realização de uma consulta simples em uma base RDF de produtos Nokia. Para saber quais

formatos de música o modelo de celular Nokia 6265 suporta, é possível começar com uma busca pelo nome do modelo (Figura 7).

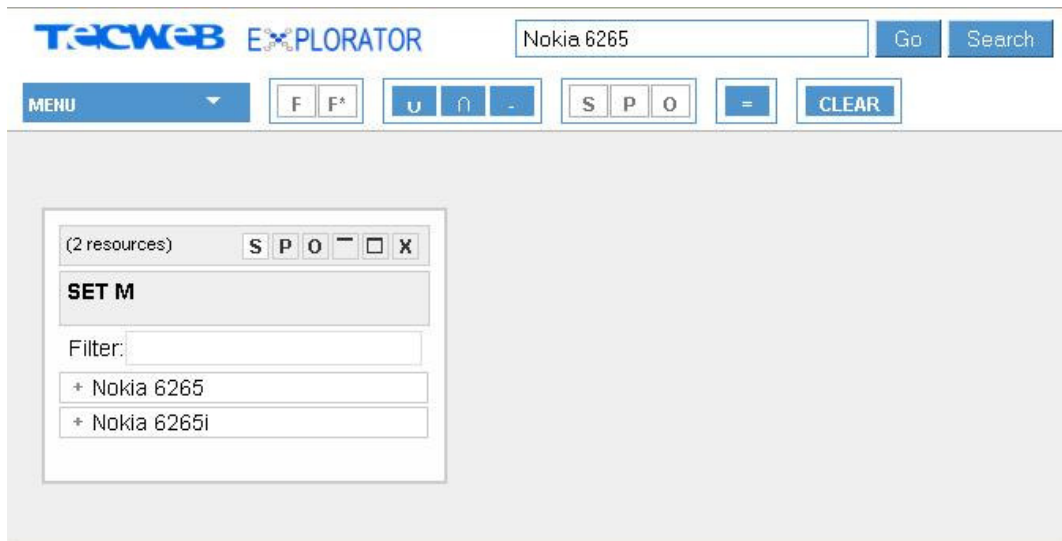


Figura 7 – Resultado de uma busca no Explorator

O conjunto “M” é o conjunto de recursos retornado pela busca. Ele possui dois elementos que representam dois modelos de celular encontrados na base. Utilizando a operação de diferença, é possível criar um novo conjunto a partir do conjunto “M” que possuirá apenas o modelo desejado. Para isso, basta selecionar o conjunto “M”, apertar a operação que realiza a diferença, selecionar o recurso que se deseja subtrair e apertar no símbolo de igual (Figura 8).

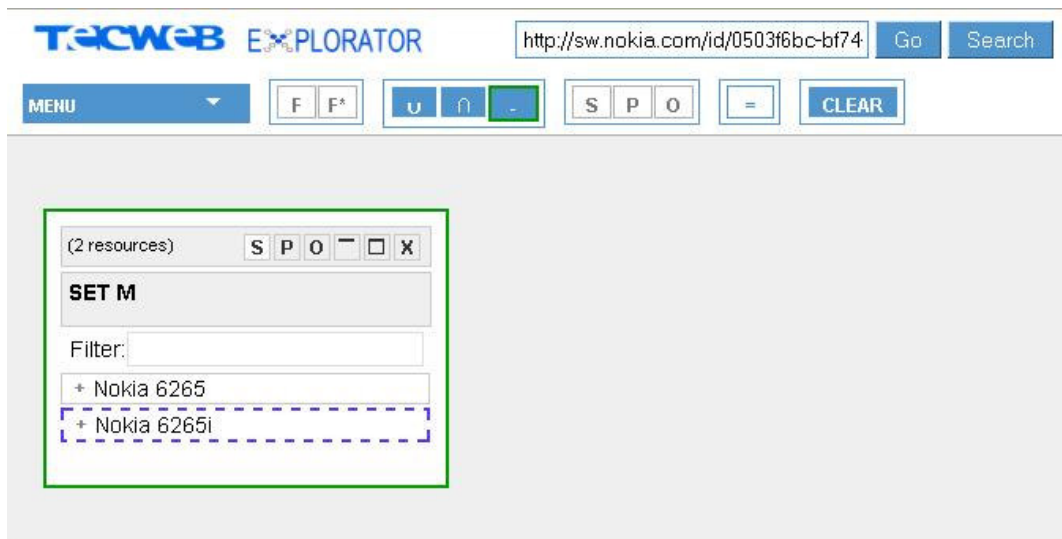


Figura 8 – Operação diferença no Explorator

O resultado da operação é o novo conjunto “B”, dessa vez somente com um elemento (Figura 9).

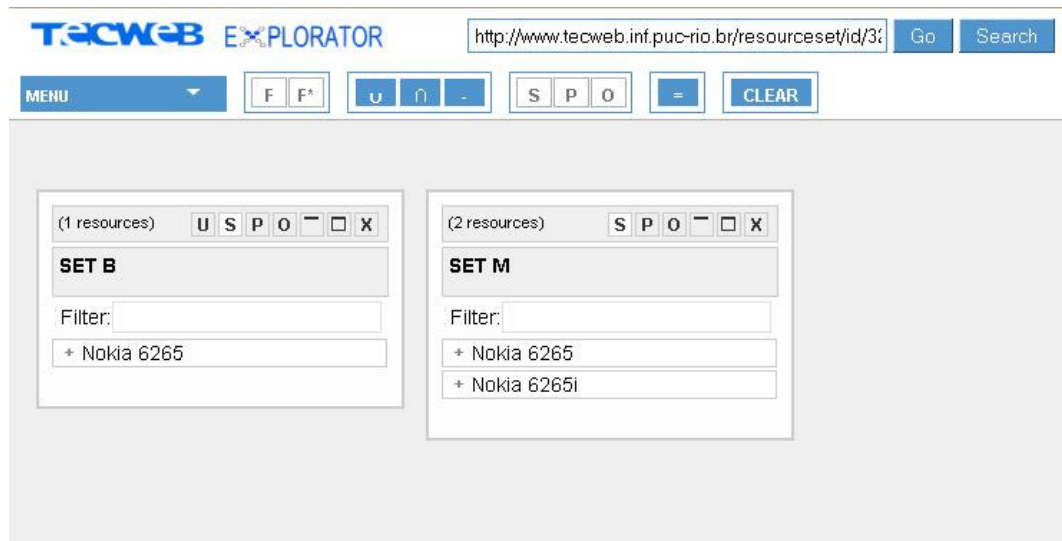


Figura 9 – Resultado da operação diferença

É importante mencionar que os conjuntos não possuem somente recursos, mas sim triplas RDF. Clicando no sinal de soma ao lado esquerdo do nome do recurso, expandimos sua visualização para exibir as triplas que fazem parte do conjunto e possuem aquele recurso como sujeito (Figura 10).

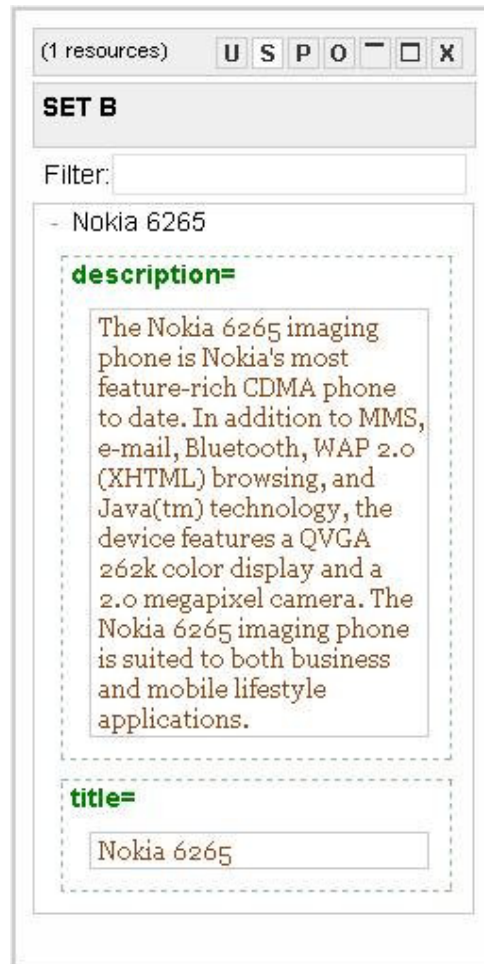


Figura 10 – Visualização de um Recurso

Para descobrir quais formatos de música o modelo suporta, precisamos da propriedade que possui essa informação. Existe uma funcionalidade no menu do Explorator que retorna todas as propriedades RDFS da base (Figura 11).

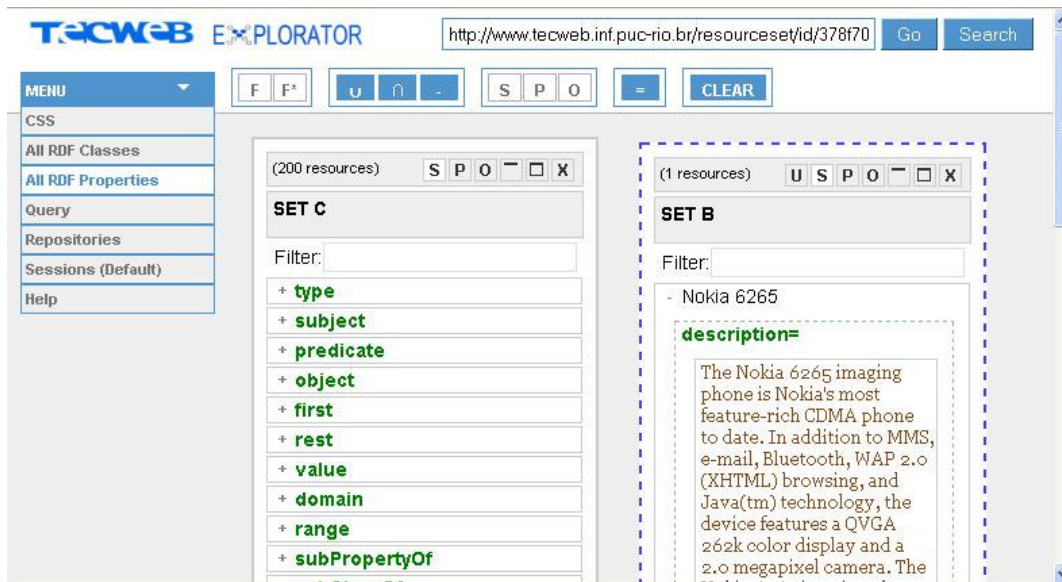


Figura 11 – Propriedades existentes na base

A funcionalidade de filtro permite uma busca nos resultados pela propriedade desejada. No nosso exemplo, “music support”. Com a propriedade desejada, podemos realizar uma operação SPO (Sujeito, Predicado e Objeto) na base, utilizando o modelo de celular desejado como S (sujeito) e a propriedade encontrada como P (predicado) (Figura 12).

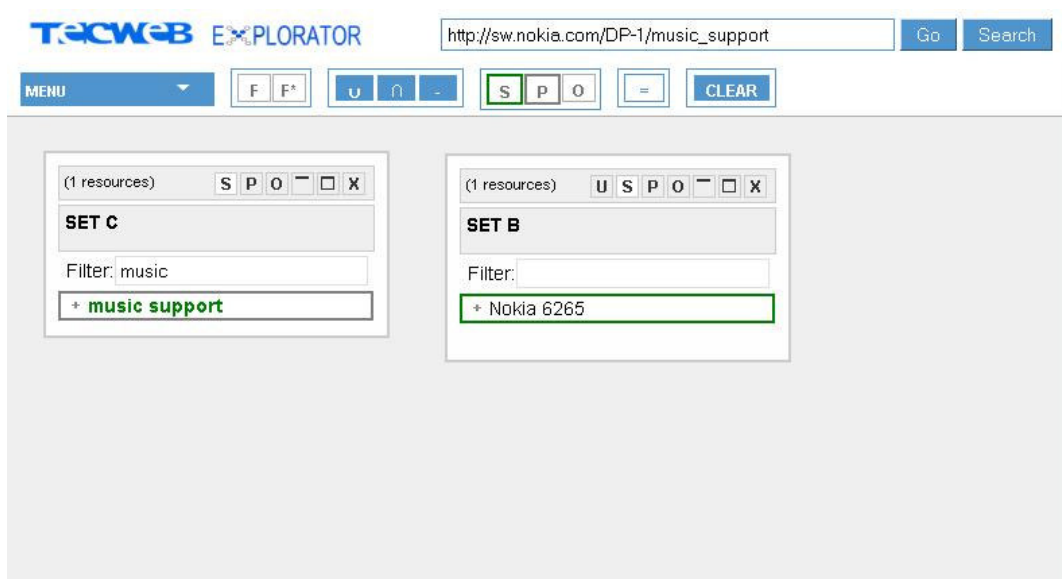


Figura 12 – A operação SPO no Explorer

O resultado fornece a informação desejada (Figura 13). O conjunto retornado exibe todas triplas que possuem o recurso selecionado como sujeito e a propriedade como predicado.



Figura 13 – Resultado da operação SPO

4.3

Novos conceitos desenvolvidos

Diversos novos conceitos foram criados para aumentar o poder das consultas realizadas no Explorator e atender as necessidades de reaproveitamento e compartilhamento de soluções. Nessa seção esses conceitos serão definidos e as funcionalidades do Excelplorator serão demonstradas passo a passo, acompanhadas de exemplos e de sua motivação dentro dos objetivos do trabalho. As funcionalidades estão divididas em cinco subseções: operações customizadas, nova organização da área de trabalho, a parametrização de consultas, o compartilhamento aberto de consultas e o compartilhamento fechado de consultas.

4.3.1 Operações customizadas

Embora o Explorator já contasse com operações que conseguiram abranger boa parte das consultas possíveis, algumas operações mais específicas não eram possíveis. Por exemplo, em uma consulta por publicações científicas, não era possível selecionar somente publicações com no máximo dez páginas. Não existia no Explorator um operador que realizasse esse tipo de filtro.

Para aumentar a utilidade do Excelplorator, seria necessário criar novos operadores. Porém, uma solução mais abrangente foi escolhida com o intuito de construir uma ferramenta genérica o bastante para atender qualquer tipo de consulta em qualquer domínio. Utilizando as vantagens da linguagem Ruby, na qual o Excelplorator foi implementado, foi criado o conceito das operações customizadas. Essa nova funcionalidade permite a criação de qualquer tipo de operação que o usuário desejar. Diversas operações podem ser criadas na própria ferramenta, em tempo de execução, e compartilhadas entre todos os usuários. Dessa forma um usuário que deseja utilizar o Excelplorator para realizar uma consulta pode escolher de um repositório de operações as que deseja utilizar. E caso nenhuma atenda as suas necessidades, ele pode criar uma nova, que passará a estar disponível para os outros usuários. Essa funcionalidade aumenta o poder de abrangência das consultas, e conseqüentemente o uso da ferramenta.

Para exemplificar a criação e o uso de uma nova operação, podemos considerar o caso de um usuário que deseja realizar consultas em uma base de aparelhos de celular. O usuário pretende buscar aparelhos de celular que possuam uma tela com uma resolução mínima de 240x240. Essa é uma consulta que não é possível de ser realizada com os operadores originais disponíveis no Explorator. Já utilizando o Excelplorator, o primeiro passo a ser realizado pelo usuário é selecionar uma coleção de triplas que possuam todos os aparelhos disponíveis na base. É possível conseguir tal conjunto partindo de um índice com todas as classes da base, obtido através do menu principal (Figura 14).

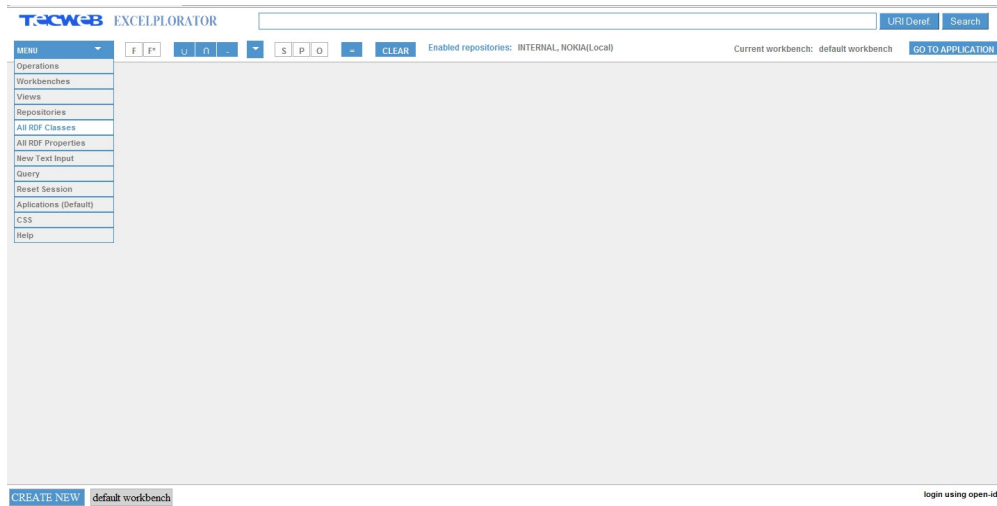


Figura 14 – Selecionando Classes na tela de desenvolvimento do Excelplorator

Uma vez com o índice de classes, basta um duplo clique na classe “Product” (o símbolo “i” indica que um duplo vai retornar as instâncias da classe e não a própria classe em si) (Figura 15).

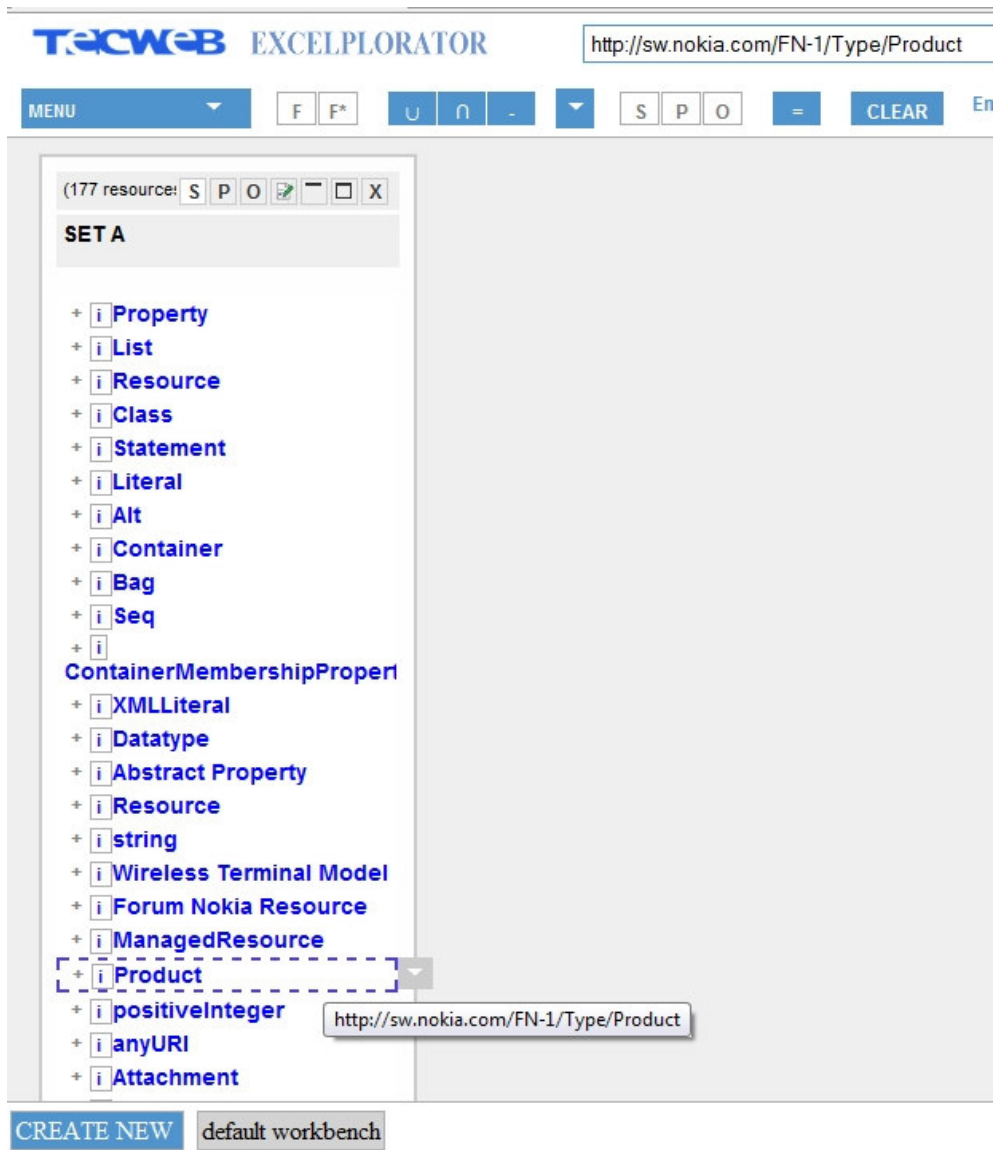


Figura 15 – Criando conjunto com todos os recursos do tipo “Product”

Gerado o conjunto que contém todos os aparelhos disponíveis na base, o usuário necessita de mais dois conjuntos, um com os aparelhos e a propriedade que indica a resolução vertical da tela e outro com os aparelhos e a propriedade que indica a resolução horizontal da tela. Para gerar esses dois conjuntos, é necessário primeiro encontrar as duas propriedades que indicam as resoluções de tela. Ou seja, existem dois recursos na base que representam essas propriedades na ontologia utilizada, e essas duas propriedades são necessárias para realizar duas operações que vão gerar os dois conjuntos que o usuário precisa. É possível encontrá-las em um conjunto de triplas que possui todas as propriedades da base (também disponível através do menu principal) (Figura 16).

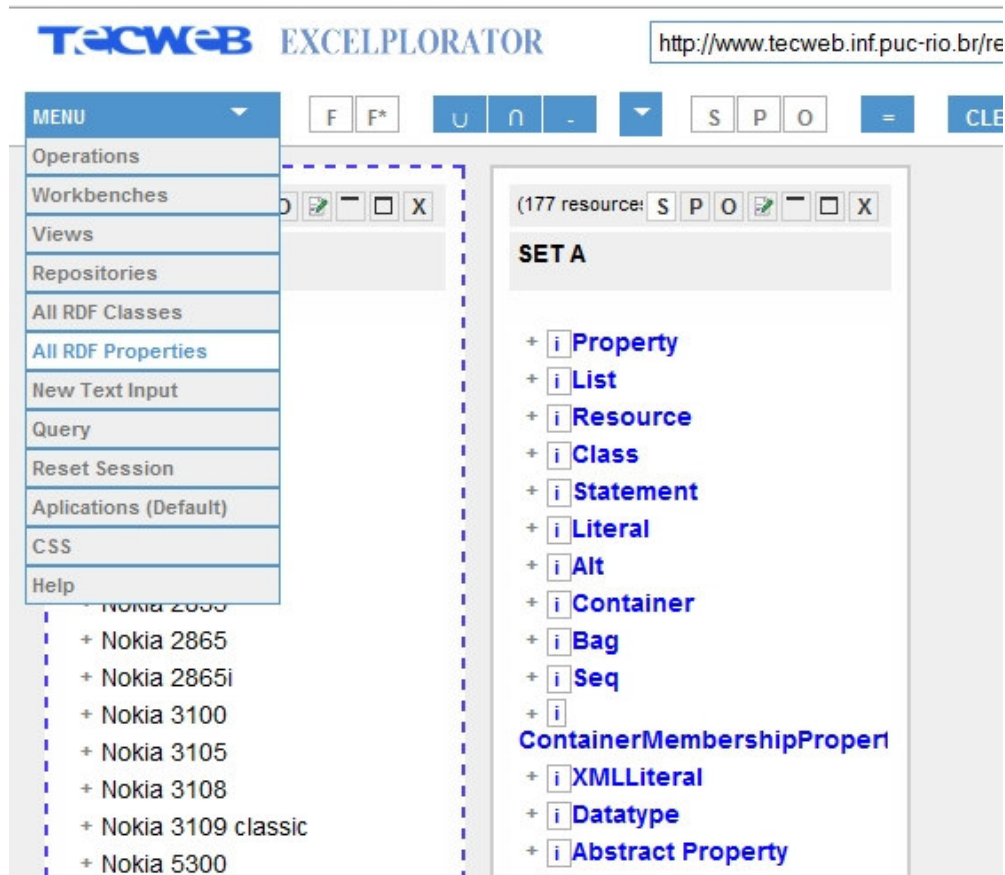


Figura 16 – Menu principal com a opção que gera o conjunto com as propriedades destacada.

Com as duas propriedades, basta a realização de uma operação SPO, fixando S como o conjunto dos aparelhos, e P como a propriedade de resolução horizontal. Esse procedimento gera um conjunto de triplas que possuem como sujeitos os aparelhos, como propriedade a resolução horizontal e como objeto a resolução horizontal de cada aparelho (Figuras 17 e 18).

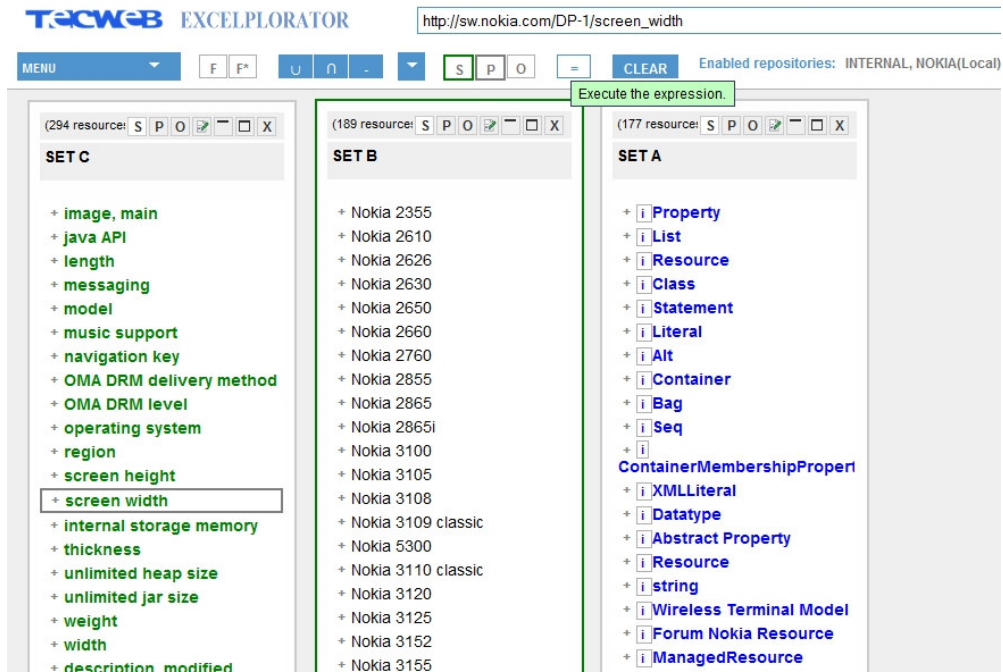


Figura 17 – Operação SPO sendo executada

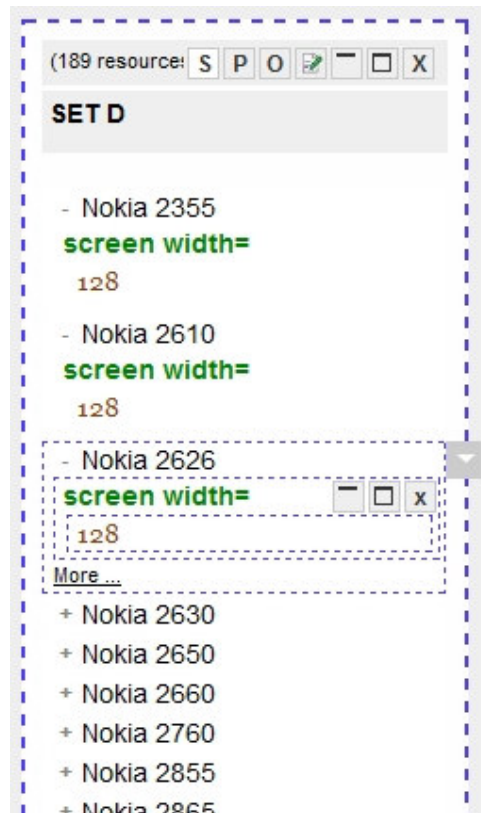


Figura 18 – Conjunto de triplas gerado pela operação SPO

O mesmo procedimento pode ser feito para gerar um conjunto de triplas com a resolução vertical de cada aparelho. Nesse momento, o usuário precisa de

um operador que filtre as triplas de um conjunto e retorne um novo conjunto somente com as triplas que possuem resolução acima ou igual a 240. Esse operador não existe originalmente na ferramenta, então é necessário criá-lo dinamicamente.

Para criar um operador, basta acessar a opção “Operations” no menu principal (Figura 19) do Excelplorator.



Figura 19 – Opção “Operations” no menu principal do Excelplorator

Um índice com todos os operadores disponíveis e um formulário de criação/edição são exibidos pela ferramenta (Figura 20).

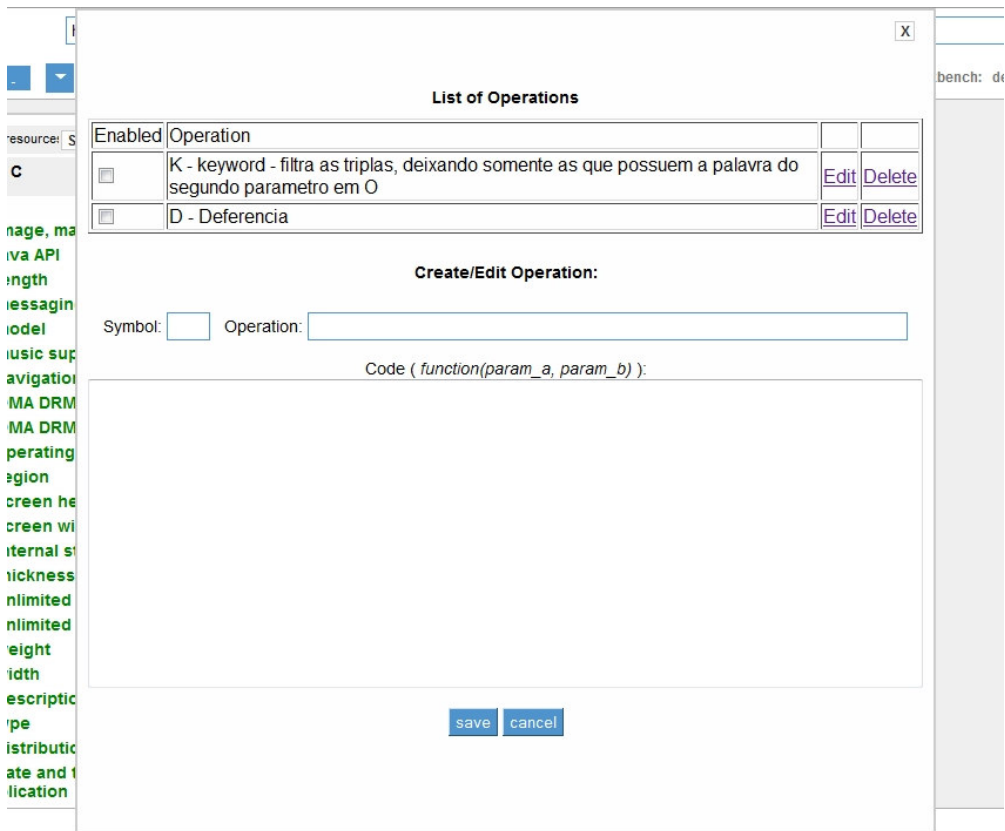


Figura 20 – Lista das Operações disponíveis e formulário de criação.

O formulário de criação possui os seguintes campos: símbolo, nome/descrição e código do operador. O símbolo deve ser preenchido arbitrariamente com qualquer caractere que simbolize o novo operador criado. Seguindo com o exemplo, será utilizada a letra F, de filtro. No nome/descrição deve ser inserida uma breve descrição que possua a semântica do operador. É importante que a descrição seja bem feita, pois qualquer operador criado por um usuário fica disponível no repositório de operadores para ser reutilizado. Portanto, uma boa descrição pode evitar re-trabalho e erros nas consultas.

O campo do código do operador define o seu comportamento. Nele deve ser inserido um trecho de código na linguagem de programação Ruby que realize a operação desejada. A execução do trecho de código deve retornar um vetor e poderá utilizar uma variável pré-existente chamada param_a e outra chamada param_b, representando respectivamente o primeiro e o segundo operando da operação. Tanto param_a quanto param_b possuem o mesmo formato, tratando-se de dois vetores. Em cada posição desses vetores, existe uma tripla RDF. A representação da tripla também é através de um vetor, esse com três posições. No

índice zero do vetor está o sujeito, na segunda posição o predicado e na terceira o objeto. Portanto, tanto `param_a` quanto `param_b` são um vetor de vetores. Esse também é o formato que é esperado o retorno da operação. As triplas RDF retornadas pelo trecho de código da operação são as triplas que estarão no conjunto de triplas resultante.

O seguinte trecho de código na linguagem Ruby pode ser utilizado para filtrar celulares com resoluções menores que 240:

```
param_a.select { |triple| triple[2].to_s.to_i >= 240 }
```

Observe que o segundo operando, `param_b`, não é utilizado. Essa é uma operação que só possui um operando, o `param_a`. Após ter preenchido todos os campos, é necessário salvar o novo operador através do botão “save” (Figura 21).

Create/Edit Operation:

Symbol:

Operation:

Code (*function(param_a, param_b)*):

```
param_a.select { |triple| triple[2].to_s.to_i >= 240 }
```

Figura 21 – Formulário de criação de novas operações preenchido.

O novo operador criado é listado no índice de operadores disponíveis, acima do formulário de criação (Figura 22).

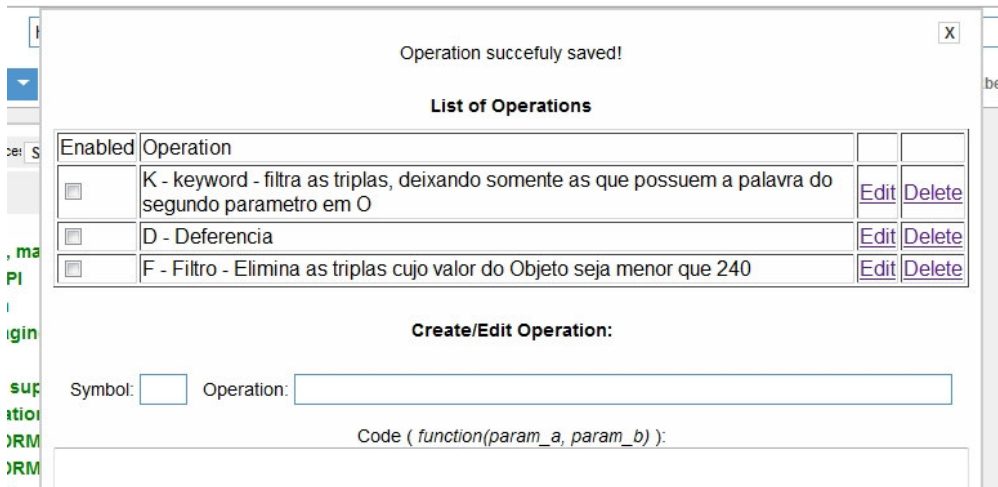


Figura 22 – Lista de operações disponíveis atualizada.

Quando um operador é criado ele entra para a o repositório de operadores da ferramenta, e pode ser utilizado por qualquer usuário. Para utilizar um operador na sua aplicação, o usuário deve habilitá-lo através do índice de operadores (Figura 23).

List of (

Enabled	Operation
<input type="checkbox"/>	K - keyword - filtra as triplas, deixando o segundo parametro em O
<input type="checkbox"/>	D - Deferencia
<input checked="" type="checkbox"/>	F - Filtro - Elimina as triplas cujo valor de

Figura 23 – Índice das operações disponíveis com o novo filtro habilitado.

Os operadores habilitados ficam disponíveis através de um “drop-down” na barra de ferramentas do Excelplorator (Figura). A operação então pode ser utilizada de forma análoga ao uso dos operadores originais.

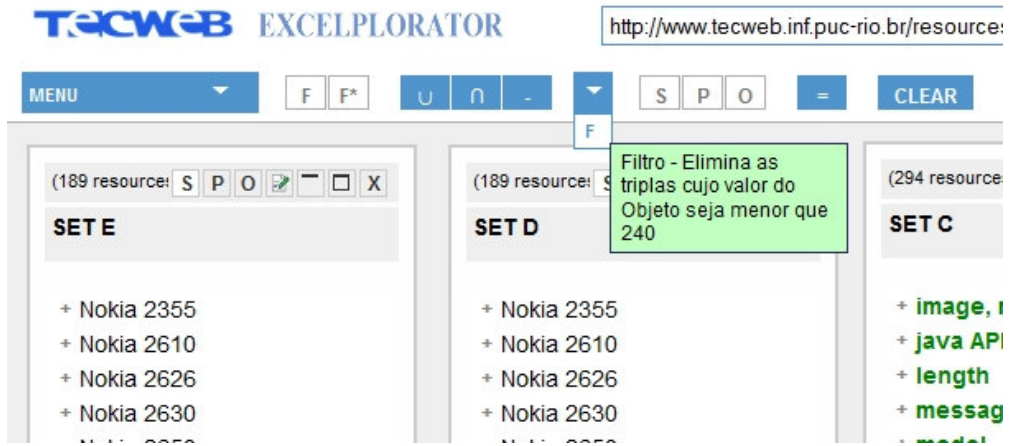


Figura 24 – Drop-down com as operações habilitadas.

Seguindo com o exemplo em que o usuário deseja os aparelhos celulares que possuam resolução acima de 240x240. É possível utilizar o novo operador criado para filtrar os dois conjuntos previamente gerados para obter dois novos conjuntos, que devem possuir somente celulares com resolução horizontal (primeiro) e vertical (segundo) acima de 240. Para isso basta selecionar um conjunto, depois o operador e finalmente o sinal de igualdade, que irá executar a operação (Figura 25).

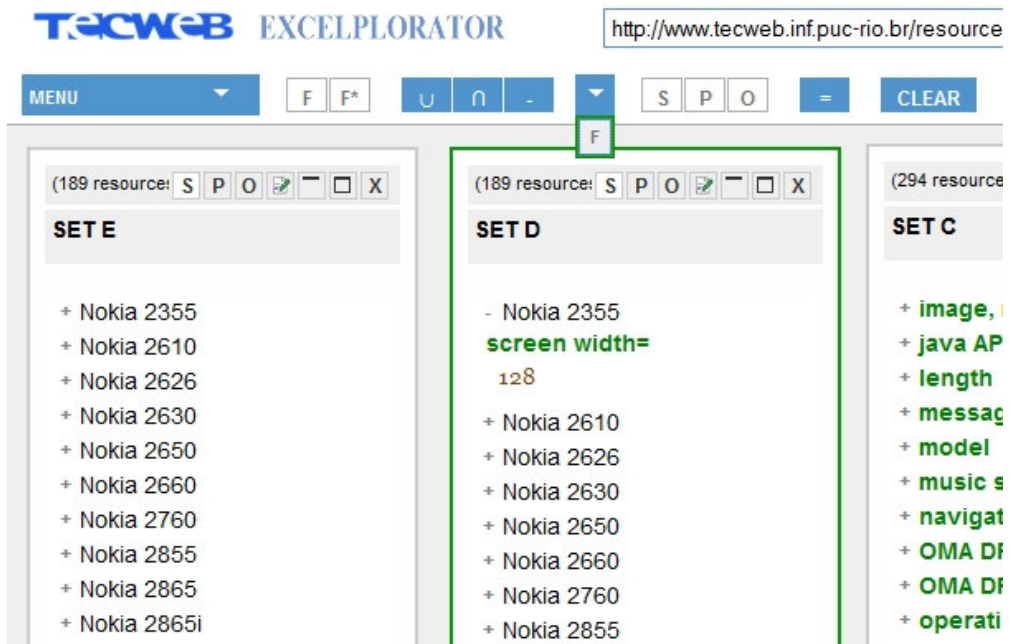


Figura 25 – Nova operação de filtro sendo utilizada para filtrar conjunto de triplas D.

Uma vez que o usuário possui os dois novos conjuntos, um com os aparelhos que possuem resolução vertical acima de 240 e outro com os aparelhos

que possuem resolução horizontal acima de 240, basta uma interseção entre os dois para conseguir o conjunto que possua todos os celulares desejados. Essa operação pode ser facilmente realizada pelo operador de interseção (Figura 26). O conjunto gerado possui a resposta para a busca do usuário.

The screenshot shows the Tecweb Explorer interface. At the top, there is a search bar with the URL `http://www.tecweb.inf.puc-rio.br/resource`. Below the search bar, there are navigation buttons: MENU, F, F*, U, \cap , -, S, P, O, =, and CLEAR. The main content area displays two sets of resources:

- SET G** (60 resources):
 - Nokia 5300
 - screen height= 320
 - Nokia 5310 XpressMusic
 - screen height= 320
 - + Nokia 5610 XpressMusic
 - + Nokia 5700 XpressMusic
 - + Nokia 6110 Navigator
 - + Nokia 6120 classic
 - + Nokia 6121 classic
 - + Nokia 6126
 - + Nokia 6131
 - + Nokia 6131 NFC
- SET F** (66 resources):
 - Nokia 5300
 - screen width= 240
 - Nokia 5310 XpressMusic
 - screen width= 240
 - + Nokia 5610 XpressMusic
 - + Nokia 5700 XpressMusic
 - + Nokia 6110 Navigator
 - + Nokia 6120 classic
 - + Nokia 6121 classic
 - + Nokia 6126
 - + Nokia 6131
 - + Nokia 6131 NFC

On the right side, there is a panel for **SET E** (189 resources) with a list of Nokia models: + Nokia 23, + Nokia 26, + Nokia 26, + Nokia 26, + Nokia 26, + Nokia 27, + Nokia 28, + Nokia 28, + Nokia 28, + Nokia 31, + Nokia 31, + Nokia 31, + Nokia 31, + Nokia 53.

Figura 26 – Criação do conjunto de triplas que possui a solução da tarefa do usuário.

Um aspecto negativo que pode ser observado na criação do operador dinâmico do exemplo acima é que o valor utilizado no filtro, no caso 240, está fixo no código da operação. Isso faz com que a reutilização da operação seja bastante limitada. Para resolver esse problema, foi criado o conceito dos transdutores, que permitem outras formas de entrada de dados para a aplicação.

4.3.1.1

Transdutores e entrada de parâmetros via texto

Para possibilitar a construção de operadores mais genéricos foi criado o conceito de transdutores, que permitem a entrada de dados na ferramenta pelo usuário. Trata-se de uma infraestrutura que permite que diversos tipos de dados sejam utilizados como entrada para utilização nos operadores. Embora a infra-

estrutura tenha sido montada pensando em qualquer forma de entrada de dados, a única forma realmente implementada foi a de entrada de texto simples. Utilizando uma opção no menu chamada “new text entry” (Figura 27), é possível criar uma caixa de entrada de texto que pode ser utilizada nos operadores (Figura 28).

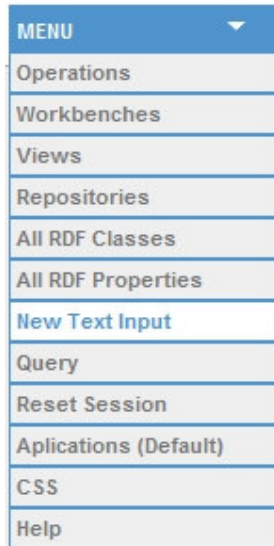


Figura 27 – Opção do menu principal para criação de uma entrada de texto.

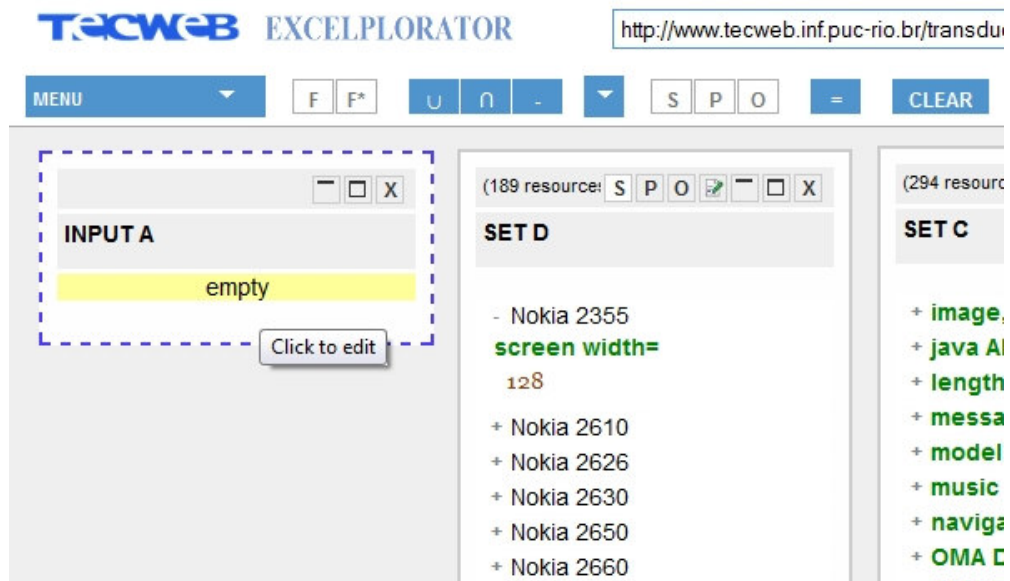


Figura 28 – Transdutor de texto criado para entrada de dados pelo usuário.

Através de uma entrada de dados que não estão contidos na base, é possível a parametrização dos operadores, tornando-os genéricos. Considere como exemplo o caso do operador que filtra celulares com resoluções abaixo de 240. Com a utilização de uma entrada de texto como segundo parâmetro, torna-se possível a reutilização do operador com outros tipos de resolução. A primeira

modificação necessária é no código do operador, que agora passa a utilizar o segundo parâmetro (param_b):

```
param_a.select { |triple| triple[2].to_s.to_i >= param_b[0].to_s.to_i }
```

Com o novo código, o valor do segundo parâmetro será utilizado no filtro. Para isso basta selecionar o conjunto a ser filtrado, selecionar o operador e depois selecionar a entrada de texto que servirá como segundo parâmetro para o filtro (Figura 29). Ao clicar no botão que possui o símbolo de igualdade, um novo conjunto será criado com os dados filtrados pelo valor corrente da entrada de texto (Figura 30).

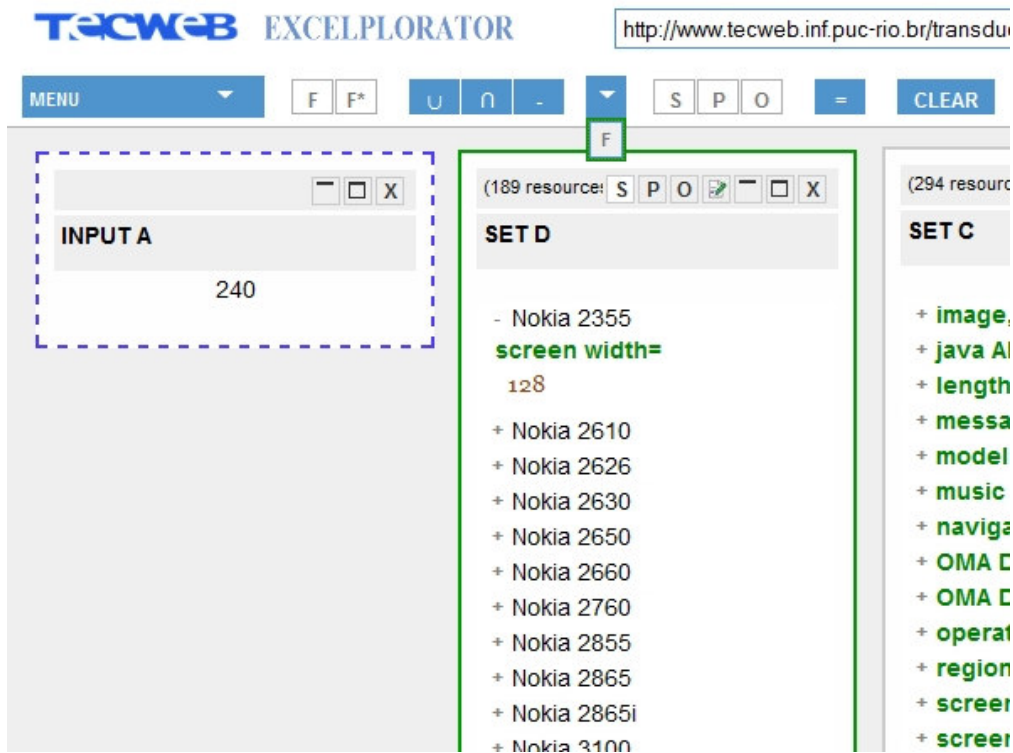


Figura 29 – Realizando a operação de filtro parametrizada com o valor 240.

The screenshot shows the TACWEB EXCELPORATOR interface. At the top, there is a URL bar with the address <http://www.tecweb.inf.puc-rio.br/resourceset/id/2e32d47b-435e-44b6>. Below the URL bar, there is a navigation menu with buttons for 'MENU', 'F', 'F*', 'U', 'n', '-', 'S', 'P', 'O', '=', and 'CLEAR'. The interface is divided into three main sections, each enclosed in a dashed blue border:

- SET H:** Contains 66 resources. The first two are Nokia 5300 and Nokia 5310 XpressMusic, both with a green 'screen width=' parameter set to 240. Other resources include Nokia 5610 XpressMusic, Nokia 5700 XpressMusic, Nokia 6110 Navigator, Nokia 6120 classic, Nokia 6121 classic, Nokia 6126, Nokia 6131, and Nokia 6131 NFC.
- INPUT A:** A central text input field containing the value '240'.
- SET D:** Contains 189 resources. The first is Nokia 2355 with a green 'screen width=' parameter set to 128. Other resources include Nokia 2610, Nokia 2626, Nokia 2630, Nokia 2650, Nokia 2660, Nokia 2760, Nokia 2855, Nokia 2865, Nokia 2865i, Nokia 3100, and Nokia 3105.

Figura 30 – Novo conjunto de triplas criado (Conjunto H) através da operação de filtro parametrizada.

É importante observar que o novo conjunto criado utiliza a caixa de entrada de texto como seu operando, portanto, quando o valor da mesma é modificado, as triplas do novo conjunto serão instantaneamente recalculadas de forma automática (Figura 31). Através de um padrão de projeto observador [Gamma et al, 1994] quando o valor de uma caixa de entrada é modificado pelo usuário, todos os conjuntos de triplas que possuem essa caixa de entrada em sua definição tem seus valores atualizados.

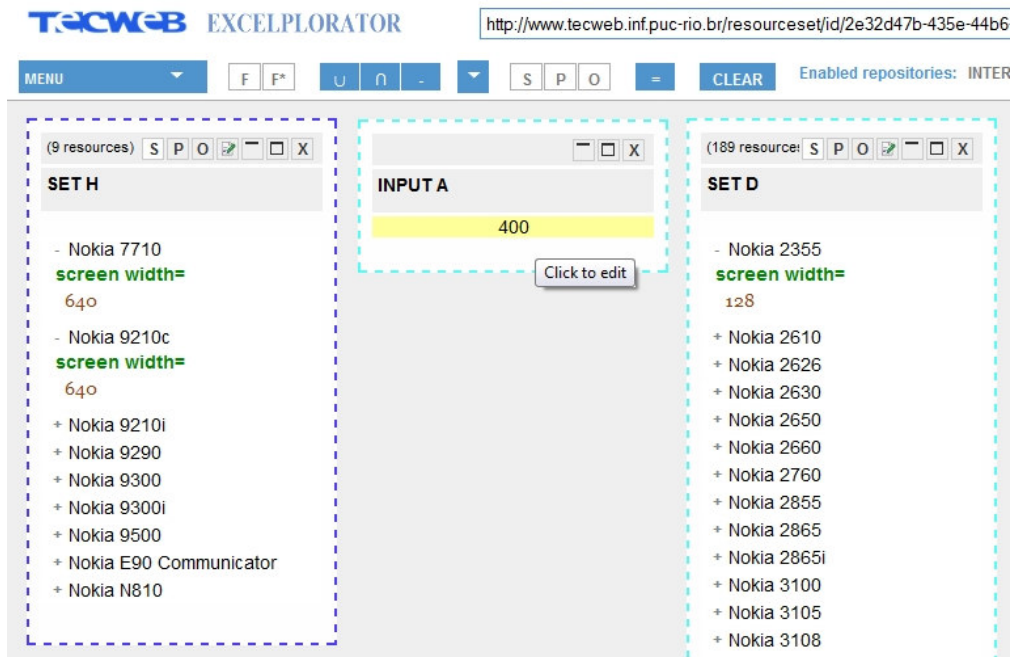


Figura 31 – Triplas do conjunto H são recalculadas quando o parâmetro “INPUT A” é atualizado.

Outra observação importante é o fato que o operador utilizado no exemplo não referencia as propriedades de resolução de tela ou qualquer propriedade específica da base utilizada. Isso significa que pode ser reutilizado em qualquer base e de qualquer domínio para filtrar um conjunto de triplas que possua um valor inteiro como objeto.

4.3.2 Nova organização da área de trabalho

Quando um usuário começa a realizar consultas no Excelplorator, ele começa a criar conjuntos de triplas RDF. Normalmente, quanto mais complexa é a consulta, mais conjuntos vão sendo criados na área de trabalho do usuário, fato que pode ser observado nos exemplos dados acima. Se um usuário montar uma consulta para descobrir, por exemplo, publicações por autor e depois quiser realizar uma nova consulta de co-autores por autor, ele terá duas opções: apagar os conjuntos criados para a consulta anterior ou criar a nova consulta e manter as duas na área de trabalho. Nenhuma das duas opções se mostrou razoável. Apagar a consulta anterior nem sempre é uma opção, pois uma boa parte das vezes o usuário pode desejar reutilizar a consulta e até mesmo relacionar os resultados

com os obtidos na nova consulta. Por outro lado, manter diversas consultas na mesma área de trabalho se mostrou inviável, pois o ambiente com muitos conjuntos de triplas se tornou muito confuso e de difícil manutenção. Para solucionar esse problema, foi criado o conceito de diversas áreas de trabalho, chamadas de workbenchs.

O objetivo de um workbench é manter uma área de trabalho para cada consulta, considerando que uma consulta seja um conjunto de operações em sequência com o objetivo de resolver um problema específico. Por exemplo: em um determinado congresso, quais são os palestrantes ligados a uma instituição específica.

Quando um usuário entra no Excelplorator pela primeira vez, um workbench padrão criado pela ferramenta se torna o workbench ativo. Ele pode então criar novos workbenchs, renomear um workbench, mudar de workbench ativo e caso deseje, apagar um workbench. Uma barra, localizada no final da tela possui a maioria dos controles relativos aos workbenchs e a indicação de qual é o workbench atual (Figura 32).

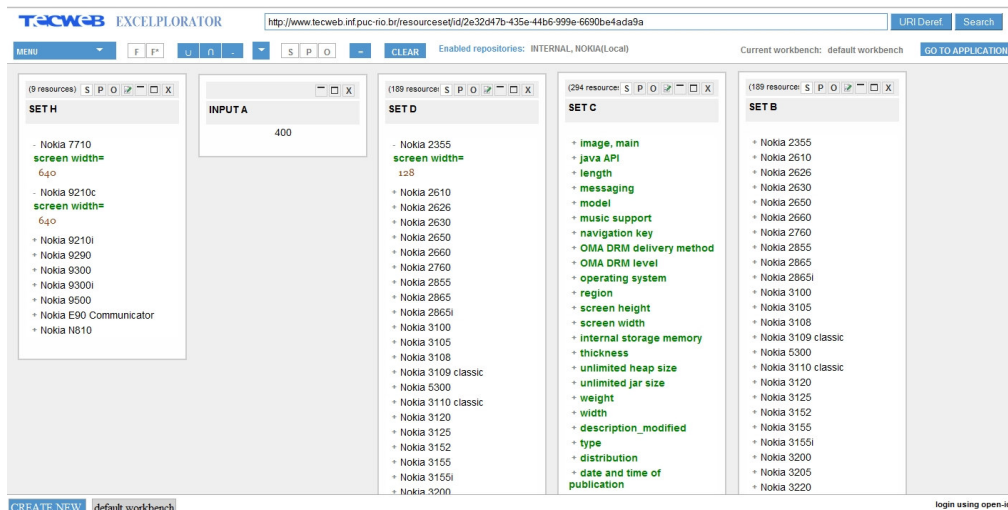


Figura 32 – A barra inferior controla os workbenchs da aplicação.

Para manter a aplicação organizada, é importante re-nomear os workbenchs para um nome que possua a semântica da consulta que ele possui. Para alterar o nome de um workbench, basta clicar no nome, à direita da barra superior. Um campo de edição irá aparecer no local (Figura 33). A atualização é feita via Ajax⁷.

⁷ Asynchronous Javascript And Xml - <http://ajaxpatterns.org/>

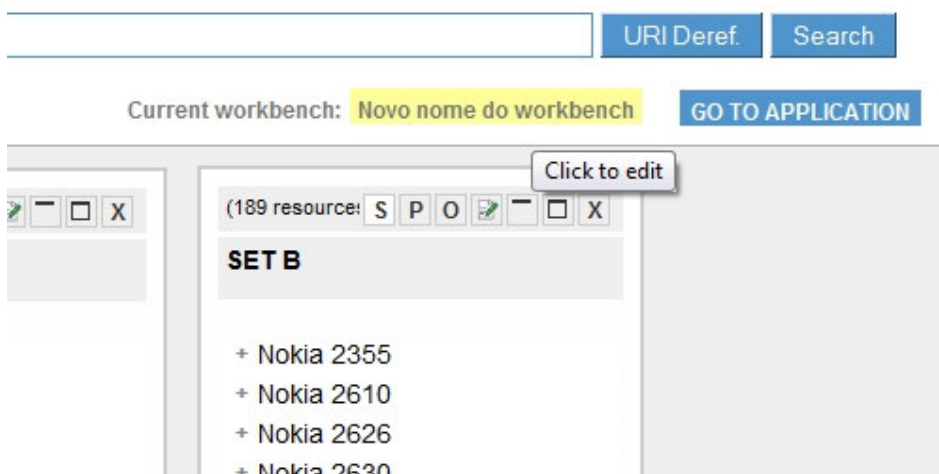


Figura 33 – Basta um clique no nome para poder atualizá-lo de forma a manter a organização do ambiente.

Para criar um novo workbench, basta um clique de mouse no botão à esquerda da barra inferior, o novo workbench criado estará vazio e com um nome padrão (Figura 34).



Figura 34 – Botão para a criação de novo workbench.

É possível alterar o workbench corrente clicando em qualquer um dos workbenchs disponíveis na barra inferior. Ao mudar de workbench, a área da tela irá se atualizar, exibindo a consulta associada ao workbench selecionado (Figura 35).

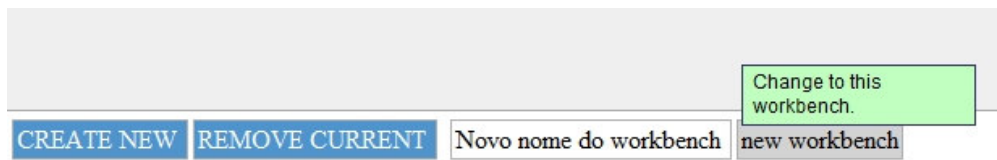


Figura 35 – Seleção de um novo workbench ativo. O preenchimento cinza indica o workbench selecionado.

Para apagar um workbench, é necessário primeiro selecioná-lo na lista de workbenchs disponíveis. Uma vez que o workbench que se deseja remover esteja selecionado, basta um clique no botão “REMOVE WORKBENCH”. (Figura 36) O workbench atual é removido, assim como a consulta que ele possui.

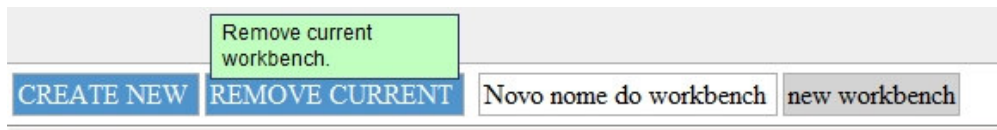


Figura 36 – Botão de remoção de workbench.

A introdução do conceito de diversas áreas de trabalho foi essencial para organizar as consultas realizadas pela aplicação. Além disso, os workbenches também possuem um papel fundamental no compartilhamento de consultas, tanto no compartilhamento aberto quanto no compartilhamento fechado (quando se deseja montar uma aplicação). Essas funcionalidades e o papel dos workbenches nelas serão discutidos respectivamente nas seções 4.3.4 e 4.3.5.

4.3.3

A parametrização de consultas

A parametrização de consultas foi um dos conceitos mais importantes introduzidos pelo Excelplorator. Com a possibilidade de parametrizar uma consulta, é possível a sua generalização, o que possibilita que a consulta seja re-utilizada de uma forma mais interessante. Sem a parametrização, seria inviável o compartilhamento de soluções e a geração de aplicações.

O objetivo da parametrização é que, depois que uma consulta seja montada utilizando os dados da base, como foi demonstrado na seção 4.2, seja possível que a mesma consulta seja re-utilizada com outros parâmetros para obter novos resultados. A forma como as consultas são montadas, através de operações sobre conjuntos de recursos, faz com que se forme um fluxo de dados que vai sendo transformado e refinado em cada operação. Caso ocorra uma alteração nos dados de entrada do fluxo, todas as operações podem ser recalculadas e novos resultados gerados. Esse tipo de arquitetura é conhecida como Dataflow Programming [Ackerman, 1982] e é bastante utilizada na criação de planilhas eletrônicas. A função da parametrização é justamente indicar quais são os dados de entrada que serão substituídos.

Para exemplificar o uso da parametrização, podemos considerar a consulta de publicações científicas de um autor específico. Essa consulta será realizada em cima da base que possui dados referentes a conferências na área da web semântica. Uma forma de começar é através de uma lista com as pessoas que

existem na base. Para isso, acessamos a opção da ferramenta que retorna um conjunto com todas as classes existentes, através de um link no menu principal (Figura 37).



Figura 37 – Opção do menu que retorna um conjunto de triplas com as Classes da base.

Como resultado da operação, um conjunto com todas as classes existentes na base é apresentado. Esse conjunto possibilita que o usuário selecione todas as instâncias de pessoas, dando um duplo clique na classe “Person” (repare que, especialmente para classes, existe um “i”, que indica que o resultado de um duplo clique será um conjunto com todas as instâncias da classe selecionada) (Figura 38).

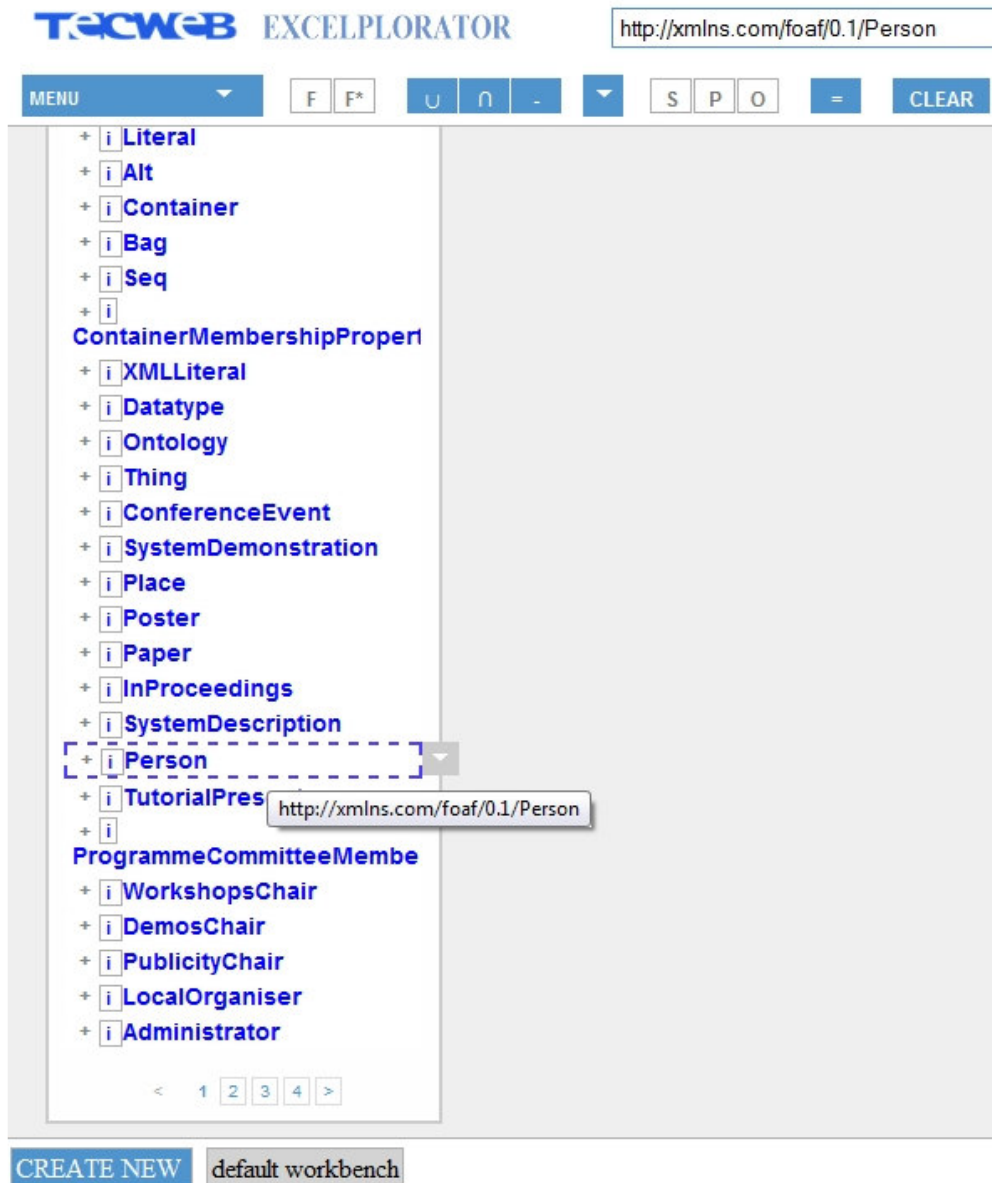


Figura 38 – Através da classe Person, é possível criar um conjunto de triplas com todas as instâncias de pessoa da base.

Uma vez que um conjunto com todas as pessoas está disponível, uma pessoa específica deve ser escolhida para a montagem da consulta. Ao selecionar uma pessoa como sujeito, deixar a propriedade e o objeto em aberto e realizar uma consulta SPO, é gerado um conjunto que possui todas as triplas da base em que a pessoa é o sujeito (Figura 39).

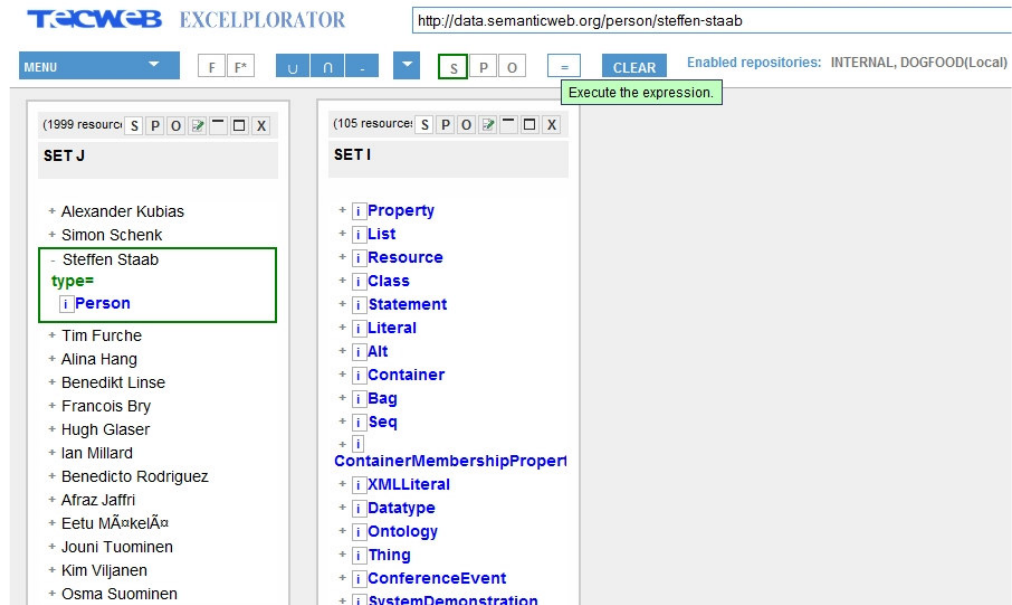


Figura 39 – Realização de uma operação SPO, com uma pessoa na posição do Sujeito.

Entre as propriedades está a propriedade “Made”, que relaciona a pessoa com os artigos científicos que ela publicou (Figura 40).

The screenshot shows the TecWeb EXCEPLORATOR interface. At the top, there is a header with the logo and the URL `http://data.semanticweb.org/person/ste`. Below the header is a navigation bar with a 'MENU' dropdown, search fields (F, F*), and a query editor with 'U', 'n', and '-' buttons, followed by 'S', 'P', 'O', '=', and 'CLEAR' buttons. The main content area displays a list of triples for a selected person. The triples are:

- holdsRole=**
 - programmechair
 - pmember
- made=** (with a tooltip showing `http://xmlns.com/foaf/0.1/made`)
 - SAIQL Query Engine - Querying OWL Theories for Ontology Extraction
 - On How to Perform a Gold Standard Based Evaluation of Ontology Learning
 - COMM: Designing a Well-Founded Multimedia Ontology for the Web
 - Networked Graphs: A Declarative Mechanism for SPARQL Rules, SPARQL Views and RDF Data Integration on the Web
 - Querying for Meta Knowledge
 - Logging in Distributed Workflows
- sameAs=**
 - Steffen_Staab
- page=**
 - Steffen_Staab

At the bottom of the interface, there are buttons for 'CREATE NEW' and 'default workbench'.

Figura 40 – Conjunto com as triplas que possuem a pessoa escolhida como sujeito. Entre as triplas, a propriedade “Made”

Com a propriedade que possui a informação desejada, é possível realizar uma consulta SPO que retorna os artigos científicos publicados por uma pessoa. Basta utilizar o conjunto que possui a pessoa selecionada como sujeito, a propriedade “Made” como propriedade e deixar o objeto em aberto (Figura 41). Nesse momento é gerado um conjunto que possui todos os artigos científicos publicados pela pessoa selecionada. Nesse momento, para descobrir artigos científicos publicados por outra pessoa, é necessário realizar uma nova consulta utilizando a outra pessoa como sujeito na operação SPO.

The screenshot displays the TACWEB EXCELPORATOR interface. At the top, the URL is `http://xmlns.com/foaf/0.1/made`. The interface features a navigation bar with buttons for 'MENU', 'F', 'F*', 'U', 'n', '-', 'S', 'P', 'O', '=', 'CLEAR', and 'Enabled repos'. The main content area is split into two panes. The left pane shows a tree view of 'Landau, Computer Science' with sub-entries for 'holdsRole=' (programmechair, pcmember), 'made=' (a list of articles), 'sameAs=' (Steffen_Staab), and 'page=' (Steffen_Staab). The right pane shows a table with a header 'Execute the expression.' and a row with values '3' and '4'. At the bottom, there is a 'CREATE NEW' button and a dropdown menu showing 'default workbench'.

Figura 41 – Realização da operação SPO, S sendo o conjunto de triplas previamente criado e P sendo a propriedade “Made”

O conceito de parametrização foi criado para possibilitar o reaproveitamento de uma consulta montada. Basta o usuário selecionar a instância que foi utilizada na criação da consulta e defini-la como um parâmetro. Após a parametrização, a instância pode ser substituída e a consulta re-avaliada.

Caso o usuário queira descobrir artigos científicos de outra pessoa, deve primeiro acessar o modo de edição do conjunto que possui a pessoa originalmente selecionada. Essa é a constante que deverá ser parametrizada e substituída. Para acessar o modo de edição do conjunto basta um clique no botão de edição (Figura 42).

The screenshot displays the TecWeb Explorer interface. At the top, the URL is <http://www.tecweb.inf.puc-rio.br/resourceset/id/e336cdce-f0d2-428c-8cb3>. The interface is divided into three panels, each representing a resource set:

- SET L** (1 resource): Contains a list of publications and a **made=** property.
- SET K** (1 resource): Contains a list of publications and several properties: **type=** (Resource, Person), **seeAlso=** (Steffen_Staab), **label=** (Steffen Staab), **based_near=** (Germany), **homepage=** (isweb.uni-koblenz.de), **name=** (Steffen Staab), **affiliation=** (University of Koblenz-Landau), **family_name=** (Staab), **firstName=** (Steffen), and **mbox_sha1sum=**.
- SET J** (1999 resources): Contains a list of names and a **type=** property (Person).

A green button labeled "Update Resource Set" is positioned above the SET K panel. At the bottom of the interface, there is a "CREATE NEW" button and a "default workbench" label.

Figura 42 – Botão de edição do conjunto de triplas, localizado no topo do conjunto, logo acima do tooltip em verde da figura.

O modo de edição não faz parte do Explorator e foi desenvolvido com dois propósitos. O primeiro é informar para o usuário qual a definição do conjunto de triplas em questão. Para isso é exibido o operador que define o conjunto, assim como o valor dos seus operandos. Essa informação é útil porque possibilita ao usuário entender exatamente qual a origem das triplas que pertencem a um conjunto. O segundo objetivo do modo de edição é possibilitar que o usuário altere a definição de um conjunto e adicione comportamento para o compartilhamento fechado (Figura 43). O compartilhamento fechado, que é o conceito que permite a criação de uma aplicação, será discutido na seção 4.3.5. A alteração da definição de um conjunto é realizada através da parametrização.

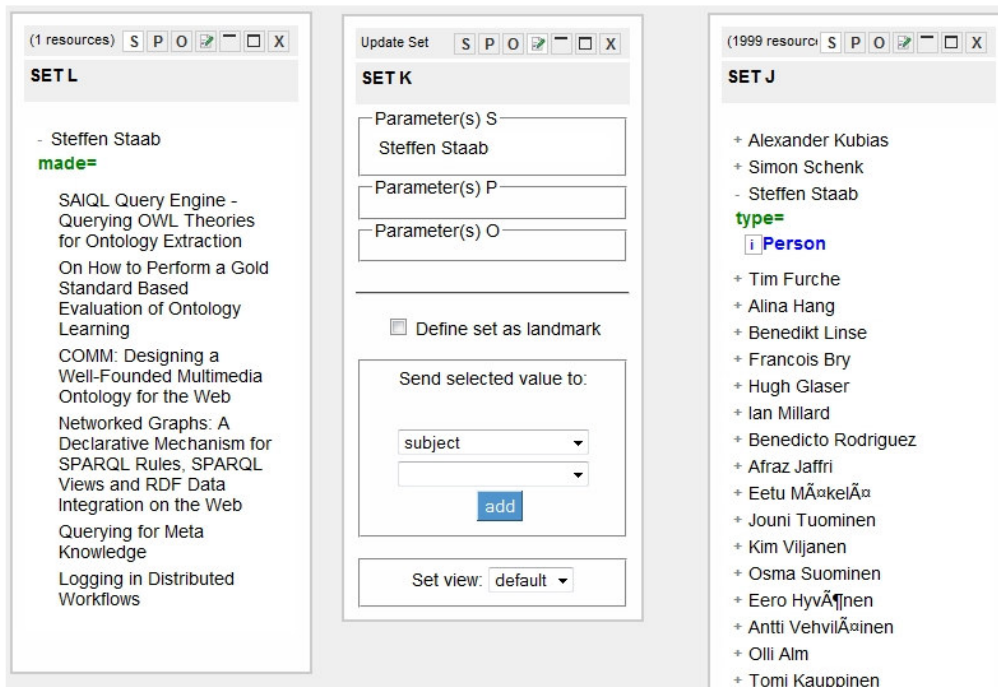


Figura 43 – Conjunto de triplas K no modo de edição.

O conjunto que deve ser parametrizado é definido por uma operação SPO que possui como único operando a pessoa que foi selecionada pelo usuário na posição do sujeito. Para parametrizar o operando basta selecionar a opção “parameterize” no *drop-down* menu⁸ (Figura 44).

⁸ http://en.wikipedia.org/wiki/Drop-down_list

Update Set S P O [refresh] [minus] [square] X

SET K

Parameter(s) S
Steffen Staab

Parameter(s) P (Parameterize)
(Remove)

Parameter(s) O

Define set as landmark

Send selected value to:

subject

add

Set view: default

Figura 44 – Opção de parametrização do conjunto de triplas.

Quando um operando é parametrizado, a ferramenta atribui a ele um nome, “p1” no caso, que pode ser alterado pelo usuário através do *drop-down* menu (Figura 45). O nome é utilizado para definir a semântica do parâmetro, de forma que a consulta fique mais organizada e assim facilitar a utilização da ferramenta pelo usuário.

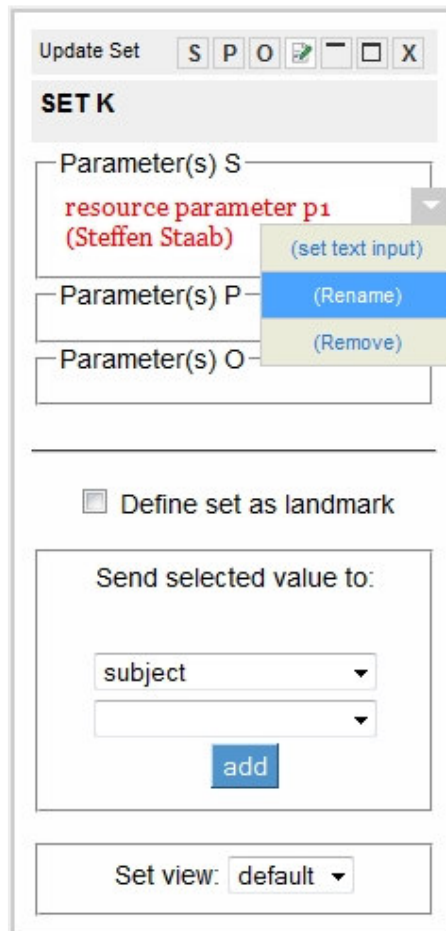


Figura 45 – Opção para re-nomear o nome do parâmetro do conjunto. Tem como objetivo facilitar o entendimento da semântica do parâmetro.

Uma vez parametrizado, o operando pode ser substituído por outros recursos, transdutores ou conjuntos através de *drag n' drop*⁹. Basta arrastar a nova seleção para a caixa do parâmetro. Nesse momento, todos os conjuntos que dependem do valor substituído serão recalculados, e os novos resultados serão exibidos (Figura 46 e 47). Dessa forma o usuário pode reavaliar a consulta com qualquer pessoa disponível na base.

⁹ <http://en.wikipedia.org/wiki/Drag-and-drop>

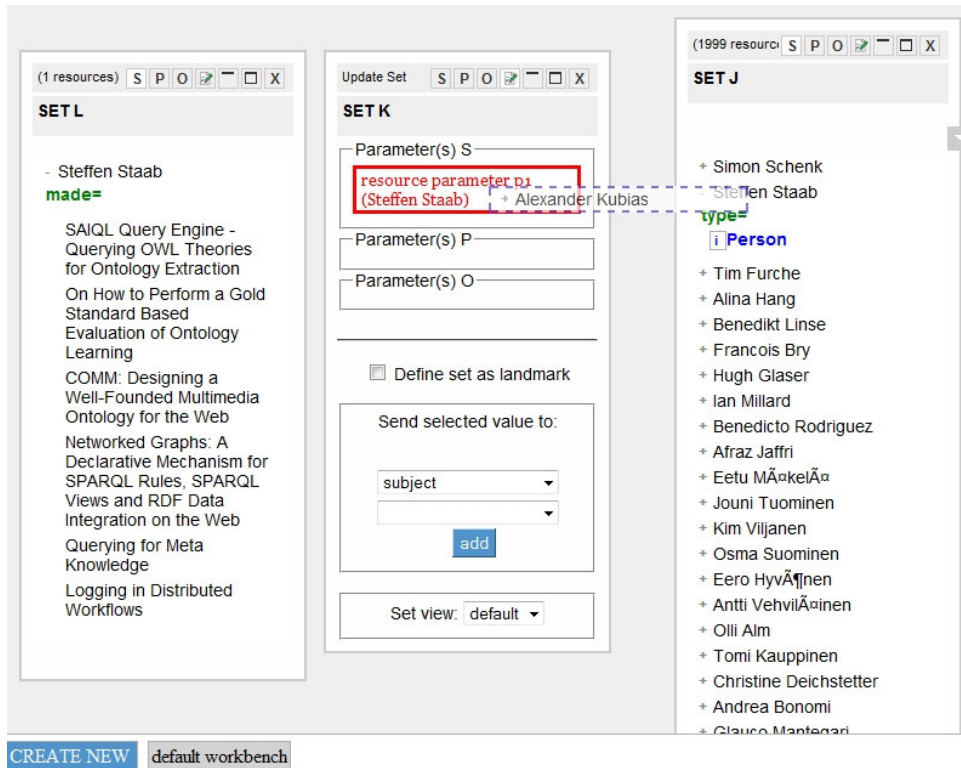


Figura 46 – Parâmetro p1, que tinha “Steffen Staab” como valor, sendo substituído por “Alexander Kubias”.

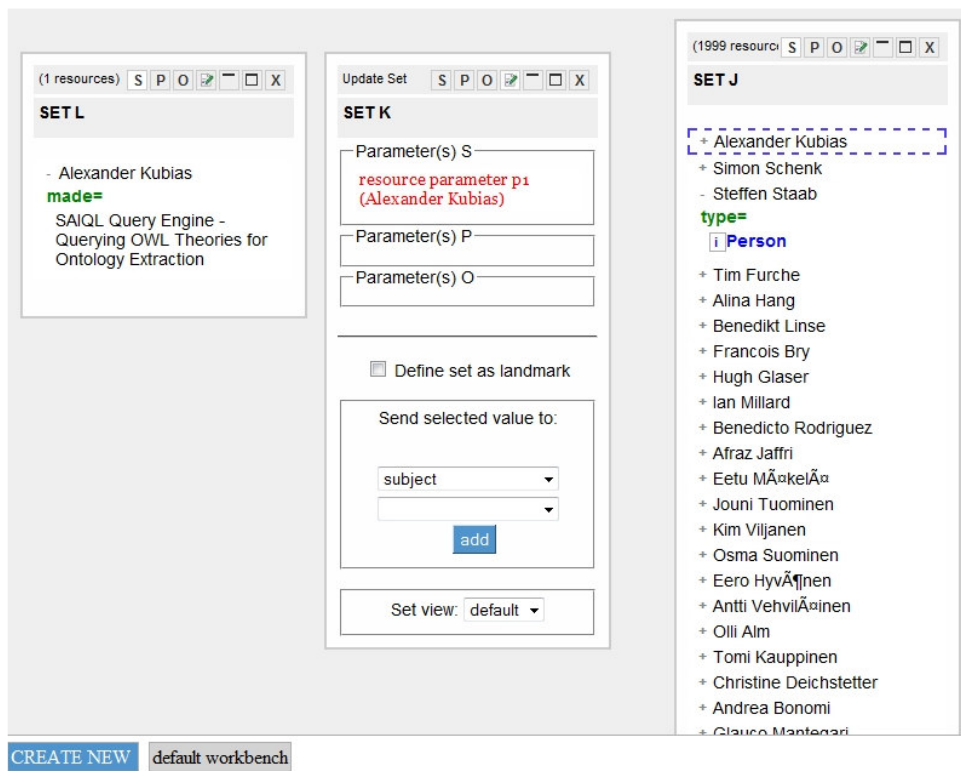


Figura 47 – A mesma tela instantes após a substituição, o conjunto de triplas L, que depende do conjunto K, teve seus valores recalculados de forma instantânea e automática.

Como os valores parametrizados também podem ser substituídos por conjuntos, existe também a possibilidade do usuário gerar um novo conjunto com uma seleção de diversas pessoas e obter uma coleção de artigos de diversos autores. Para isso basta gerar o novo conjunto (Figura 48) e depois arrastá-lo para a caixa do parâmetro (Figura 49 e 50). O conjunto que representa o resultado da consulta é atualizado automaticamente e exibe os artigos científicos produzidos pelas pessoas escolhidas. Essa é uma forma de expandir a consulta original.

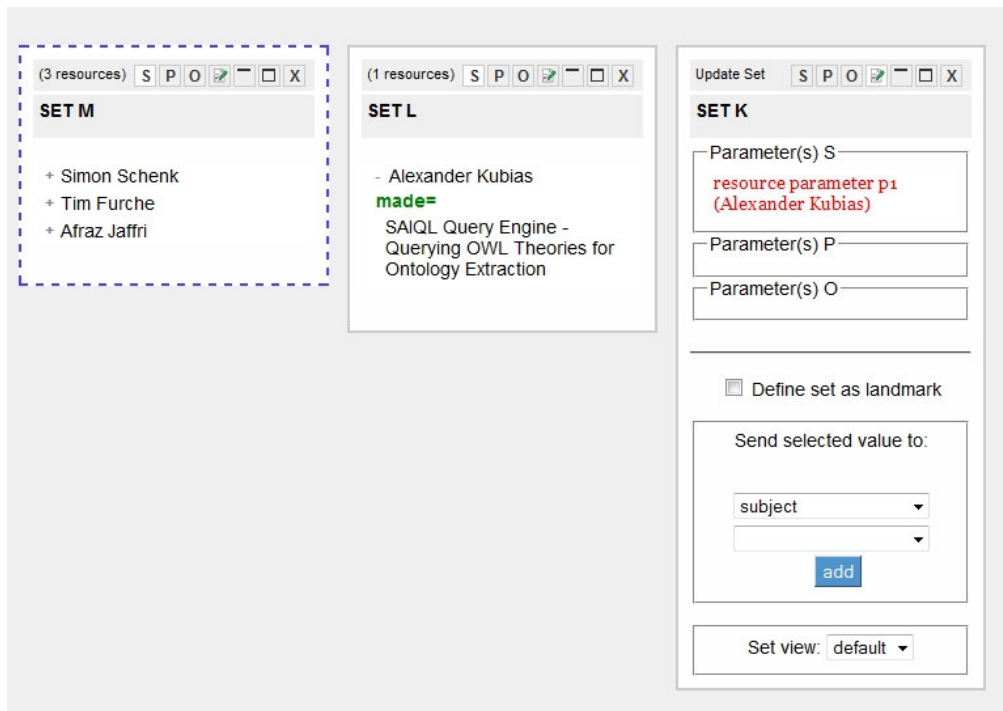


Figura 48 – O novo conjunto de triplas M, que possui três pessoas da base na posição do Sujeito.

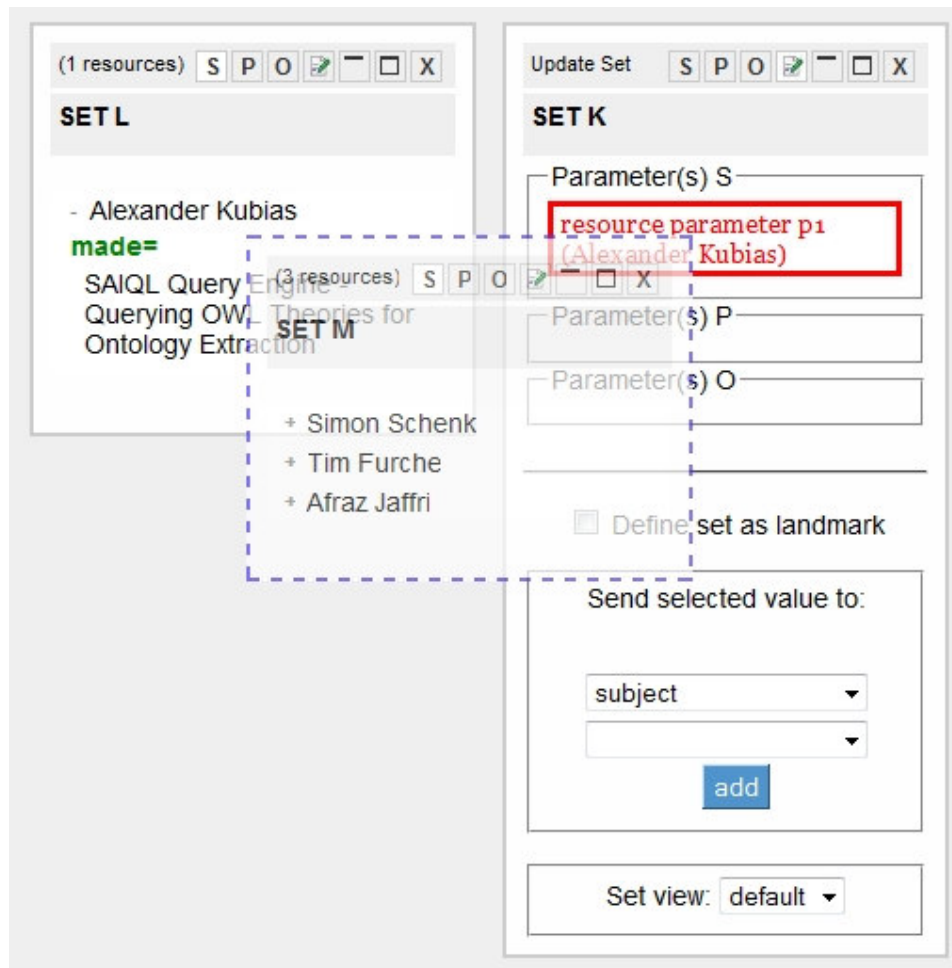


Figura 49 – Conjunto M sendo arrastado para substituir o valor do parâmetro p1, do conjunto K.

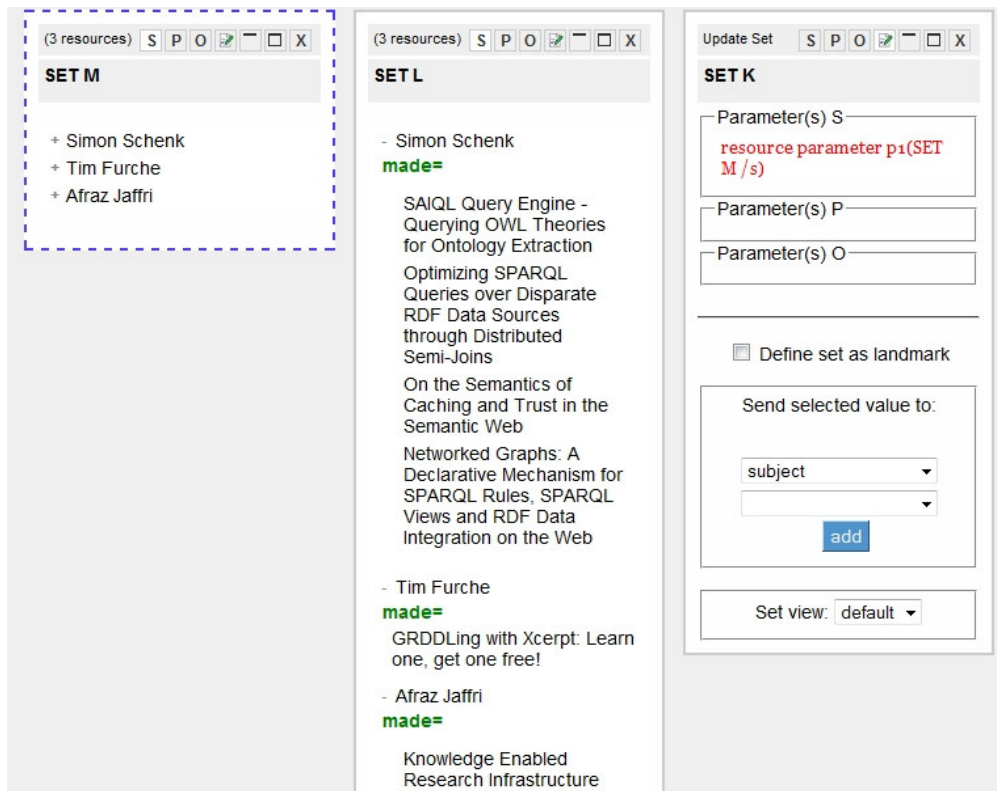


Figura 50 – Logo após a substituição, o conjunto L já conta com seus novos valores, decorrentes da atualização do valor do parâmetro p1 do conjunto K.

É importante observar que, para facilitar o entendimento, a consulta utilizada para exemplificar a parametrização é bastante simples, o que pode causar a impressão que o novo conceito apresentado não seria tão útil. Porém, em consultas mais complexas a parametrização é imprescindível. Principalmente para a reutilização e para o compartilhamento, como será demonstrado nos próximos tópicos desse capítulo.

4.3.4

Compartilhamento aberto de consultas

Um dos focos desse trabalho é permitir que usuários compartilhem consultas desenvolvidas no ambiente do Excelporator. Duas formas distintas de compartilhamento foram criadas no Excelporator, o compartilhamento aberto e o compartilhamento fechado. Nessa seção será discutido o compartilhamento aberto, e na próxima será abordado o compartilhamento fechado.

O objetivo do compartilhamento aberto é que usuários possam entrar no ambiente de criação de consultas e decidir se querem montar uma consulta

começando do zero ou se preferem importar uma já montada e compartilhada por outro usuário. Consultas compartilhadas devem ter o objetivo de resolver uma pergunta específica, por exemplo, quais são os coautores de um determinado pesquisador ou quais os celulares disponíveis na base que possuem as melhores resoluções de tela. Para o usuário, utilizar uma consulta pré-montada pode significar a resposta para a sua pergunta praticamente pronta ou também pode servir como uma base para uma consulta própria, mas que será mais facilmente realizada partindo de uma solução pré-montada. Por mais que uma consulta compartilhada não atenda completamente as necessidades do usuário, ela pode ajudar bastante, principalmente pela grande probabilidade de conter propriedades e classes, da ontologia utilizada na base, que interessam ao usuário. Uma vez que a consulta foi estendida para atender as suas necessidades específicas, o usuário pode então compartilhar essa nova versão da consulta para que possa ser reutilizada e estendida por outros usuários.

O compartilhamento aberto é feito através dos workbenches. Workbench é uma área de trabalho que deve possuir diversos conjuntos de triplas que se relacionam definindo uma consulta, como explicado no capítulo 4.3.2. Um workbench deve ter uma consulta com um objetivo claro, por exemplo, quais são os coautores de um determinado pesquisador. É possível entender um workbench como um caso de uso.

Para exemplificar como funciona o compartilhamento aberto podemos utilizar o exemplo de um usuário que acessa o Excelplorator e cria no workbench default uma consulta que identifica os coautores de um determinado autor. O primeiro passo para o compartilhamento é a parametrização dos dados de entrada que devem ser variáveis, como demonstrado no capítulo 4.3.3. Nesse caso, o operando parametrizado é o autor selecionado para a montagem da consulta. Dessa forma ela poderá ser reutilizada para descobrir quais são os co-autores de outras pessoas disponíveis na base. Uma vez que a consulta esteja pronta e parametrizada, é necessário renomear o workbench utilizado com um nome que contenha a semântica da pergunta que está sendo respondida pela consulta (Figura 51). O nome utilizado no caso do exemplo é “coautores por autor de publicação”.

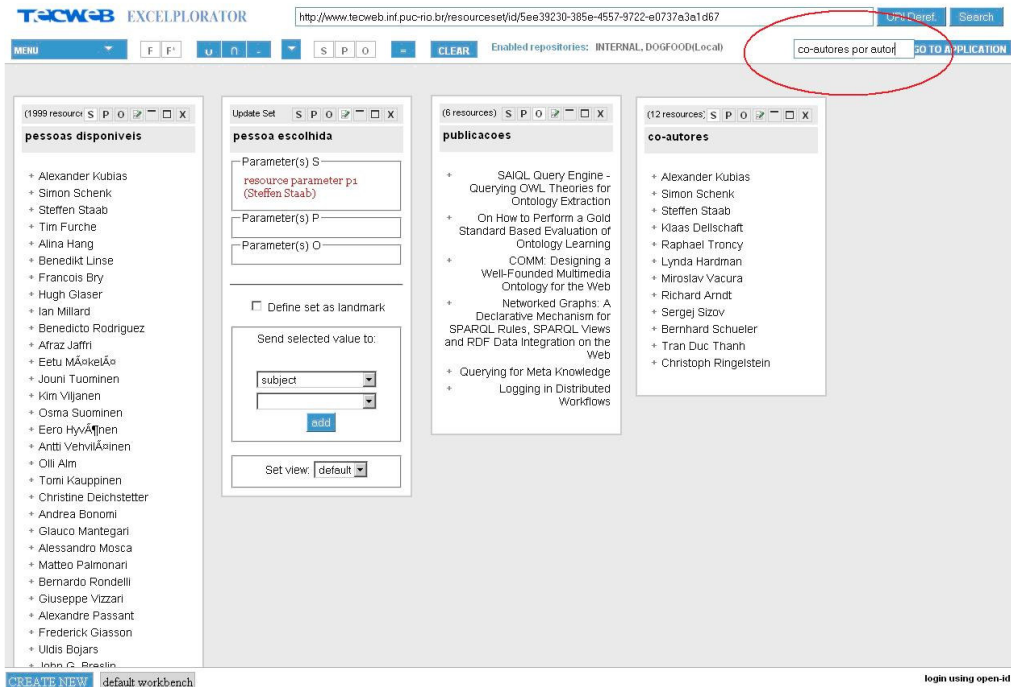


Figura 51 – Caso de uso parametrizado em um workbench que será compartilhado. No detalhe, o workbench sendo re-nomeado.

Para compartilhar o workbench, o usuário deve acessar a opção “workbenches” no menu principal (Figura 51).



Figura 52 – Opção do menu para compartilhamento e importação de workbenches.

Dois índices são exibidos pela ferramenta. O primeiro contém os workbenches existentes na aplicação atual e o segundo lista todos os workbenches compartilhados na ferramenta (Figura 53).

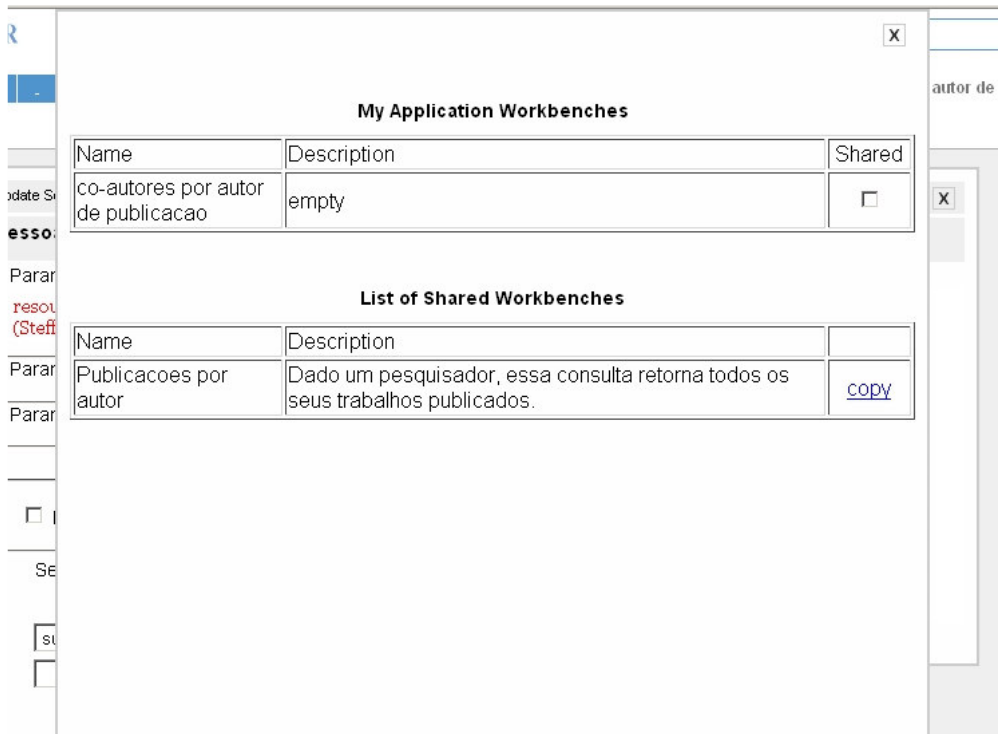


Figura 53 – Tela com a lista de workbenchs compartilhados pelo usuário e a lista de workbenchs disponibilizados por outros usuários.

O primeiro índice possui uma coluna que representa a descrição dos workbenchs da aplicação atual. Caso o usuário deseje compartilhar algum workbench é importante que ele acrescente uma descrição na coluna em questão. A descrição serve para auxiliar os usuários que pretendem reutilizar a consulta compartilhada. Para editar a descrição basta um clique na célula (Figura 54). Uma vez que o workbench já possui descrição, o usuário deve selecionar o checkbox correspondente na coluna “Shared” e o workbench estará compartilhado (Figura 55).

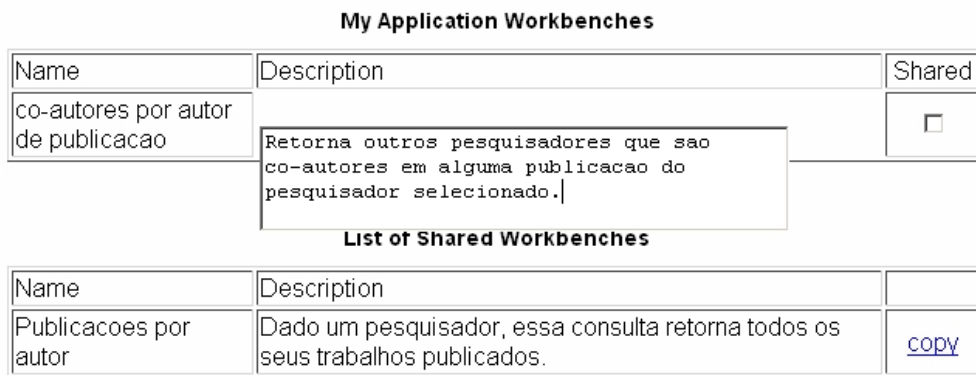


Figura 54 – Edição da descrição de um caso de uso que será compartilhado.

Name	Description	Shared
co-autores por autor de publicacao	Retorna outros pesquisadores que sao co-autores em alguma publicacao do pesquisador selecionado.	<input checked="" type="checkbox"/>

Figura 55 – A última coluna indica se o caso de uso está sendo compartilhado.

Caso usuário deseje utilizar uma consulta compartilhada por outro usuário, ele deve selecionar no índice de consultas compartilhadas uma que lhe interesse e clicar no link “Copy” correspondente (Figura 56). Nesse momento uma cópia do workbench correspondente será feita e associada à aplicação corrente (Figura 57). É importante observar que o compartilhamento é feito através de uma cópia do workbench em questão. Isso significa que o workbench original não será modificado caso ocorra alterações nas cópias de outros usuários.

List of Shared Workbenches

Name	Description	
Publicacoes por autor	Dado um pesquisador, essa consulta retorna todos os seus trabalhos publicados.	COPY

Figura 56 – Um dos casos de uso compartilhados por outros usuários.

CREATE NEW	REMOVE CURRENT	co-autores por autor...	Publicacoes por autor
----------------------------	--------------------------------	-------------------------	-----------------------

Figura 57 – Após a importação, uma cópia é criada na sessão do usuário atual.

O compartilhamento de workbenches como foi demonstrado é chamado de compartilhamento aberto porque permite aos usuários, que estão utilizando uma consulta pré-construída, visualizar e alterar os conjuntos de triplas, diferente do compartilhamento fechado, aonde uma aplicação é montada. O compartilhamento fechado será discutido a seguir.

4.3.5

O compartilhamento fechado de consultas

O objetivo do compartilhamento fechado de consultas é permitir que usuários utilizem o Excelplorator para construir aplicações de consultas a bases RDF. Essas aplicações podem ser utilizadas por outros usuários que não possuem conhecimento algum em RDF ou web semântica. Com isso, através do compartilhamento fechado é possível trazer parte dos benefícios da web semântica

para usuários comuns, que possuem somente conhecimentos básicos em computação.

A forma como é realizado o compartilhamento fechado é um pouco mais complexa do que a utilização do compartilhamento aberto, e possui um foco bem diferente. Como foi discutido nos capítulos anteriores, com o aumento da web-semântica, diversas bases RDF estão sendo disponibilizadas através da internet com todos os tipos de informações em diversos domínios. O Explorator foi desenvolvido com o objetivo de facilitar o acesso a essa gama de dados para usuários que possuem certo conhecimento técnico na área. Já o Excelplorator, desenvolvido nesse trabalho, teve como um de seus objetivos aumentar essas facilidades, principalmente através da reutilização, compartilhamento e do aumento da capacidade das consultas realizadas. No caso do Explorator, o usuário em questão necessita de conhecimentos técnicos em informática para conseguir acessar os dados e realizar consultas de forma satisfatória. O compartilhamento fechado, implementado no Excelplorator, é uma forma de tornar as consultas construídas na ferramenta disponíveis para usuários de uma forma que abstraia o modelo RDF existente. As consultas são acessadas em um ambiente diferente do qual elas foram construídas e não podem ser modificadas. A forma de acesso é realizada em uma aplicação web de consulta as informações de uma base de dados.

Utilizando o conceito de compartilhamento fechado, um usuário pode acessar o Excelplorator, montar suas consultas em diversos workbenches (manualmente ou através da importação de outras consultas já realizadas), parametrizar as consultas montadas, e depois, utilizando recursos específicos do compartilhamento fechado, montar uma aplicação para que outros usuários utilizem as soluções disponíveis.

4.3.5.1

A construção da aplicação

Para exemplificar a utilização do compartilhamento fechado podemos considerar a construção da aplicação de consulta a uma base de dados de conferências na área da web semântica demonstrada no capítulo 4.1. Como cada consulta é construída em um workbench, no compartilhamento fechado, cada

workbench da sessão do usuário se transforma em um caso de uso da aplicação montada. Como primeiro passo, o usuário deve acessar o Excelplorator e montar uma consulta que será disponibilizada na aplicação. Por exemplo, o usuário pode montar uma consulta que retorne as publicações científicas de um autor. Na aplicação gerada através do compartilhamento fechado, uma opção do menu vai dar acesso a esse caso de uso. Qualquer usuário que desejar poderá acessar a aplicação compartilhada e escolher essa opção. Nesse momento uma lista de autores da base vai aparecer e o usuário vai poder escolher qual autor deseja ver as publicações. Uma vez selecionado um autor, as publicações desse autor serão exibidas. O comportamento é o demonstrado no primeiro caso de uso da aplicação exemplo do capítulo 4.1.

Seguindo com o exemplo em que se deseja montar a aplicação, o primeiro passo do caso de uso, como mencionado acima, é montar em um workbench uma consulta que retorne as publicações de um autor. Esse será o workbench que vai conter a implementação do primeiro caso de uso. Para isso o usuário que está montando a aplicação deve realizar uma consulta que retorne todos os autores da base. Isso é possível através da propriedade “Made”, que relaciona pessoas com suas publicações. Fixando a propriedade “Made” na posição de propriedade e realizando uma consulta SPO, obtêm-se um conjunto com todas as pessoas da base que são autores de alguma publicação (Figura 58). Nesse momento é possível realizar uma nova consulta SPO, escolhendo uma pessoa na posição de objeto e a propriedade “Author” na posição de propriedade. É interessante observar que a propriedade “Author” é a inversa da propriedade “Made”. Ela relaciona a publicação com os seus autores. A consulta realizada retorna as publicações de autoria da pessoa selecionada (Figura 59).

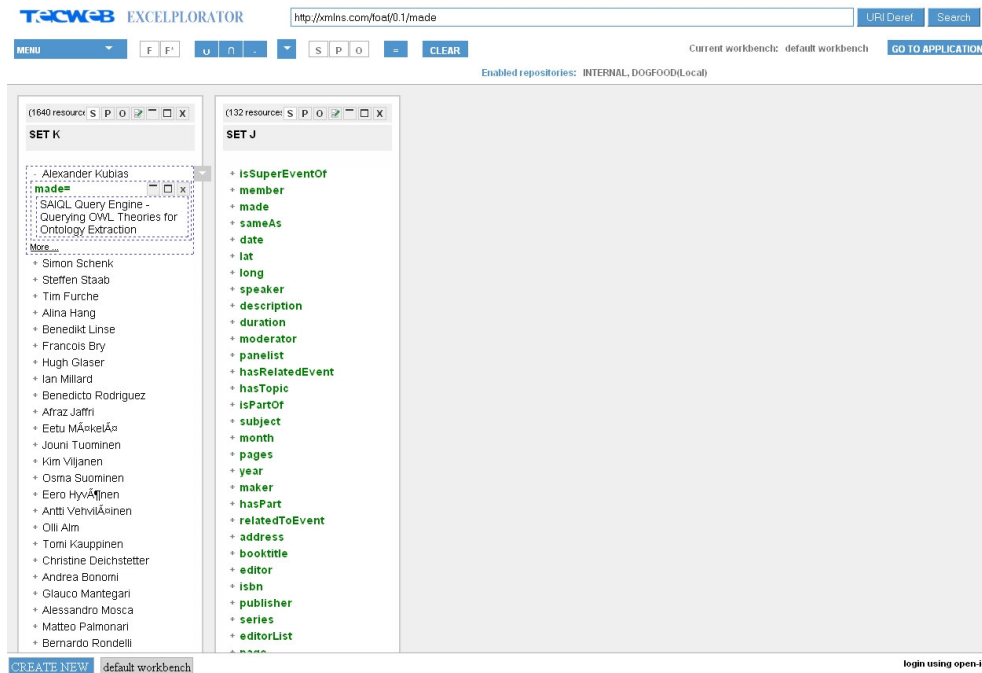


Figura 58 – Após a criação do conjunto de triplas K, que possui a propriedade “made” na posição do predicado.

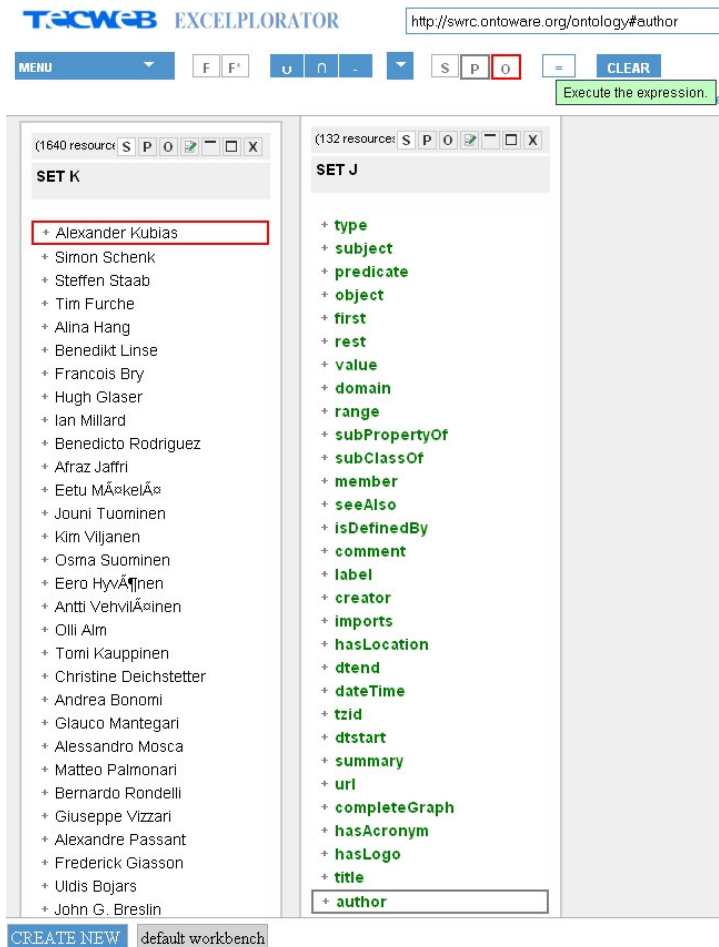


Figura 59 – Momento da execução de uma operação SPO, que criará um novo conjunto.

Para facilitar o entendimento é importante que os conjuntos gerados sejam renomeados com nomes que representem as semânticas dos conjuntos. O primeiro conjunto, por exemplo, pode ser chamado de “Autores de publicações” e o segundo de “Publicações por autor” (Figura 60). Também é importante trocar o nome do workbench que está sendo utilizado para um nome que expresse a semântica da consulta que está sendo construída. No exemplo em questão, “Publicações por autor” será utilizado (Figura 61).

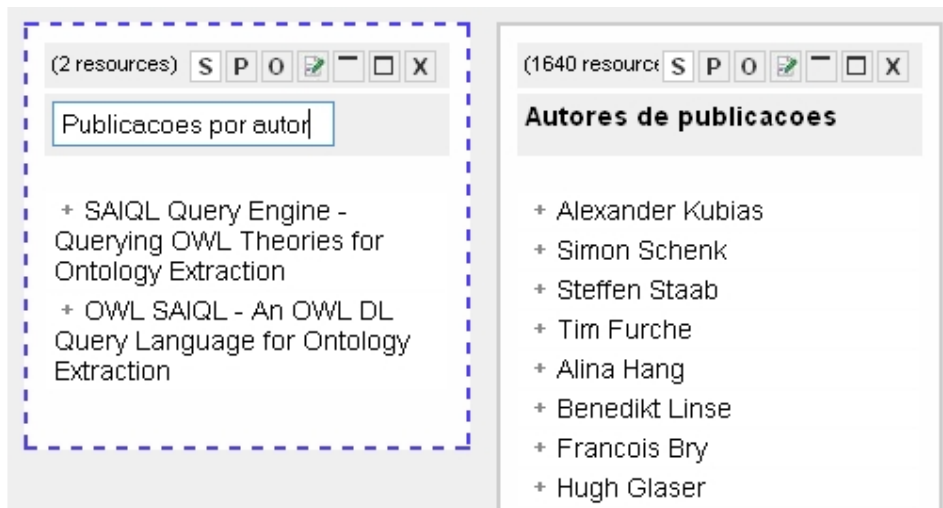


Figura 60 – Conjuntos de triplas sendo re-nomeados.

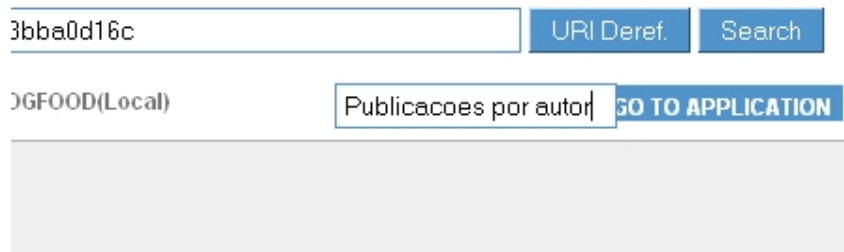


Figura 61 – Workbench sendo re-nomeado.

Para continuar a geração da aplicação, o usuário deve acessar o modo de edição dos conjuntos gerados. A primeira tarefa no modo de edição é parametrizar a pessoa selecionada no segundo conjunto (Figura 62). Dessa forma, será possível reutilizar a consulta com outros autores.

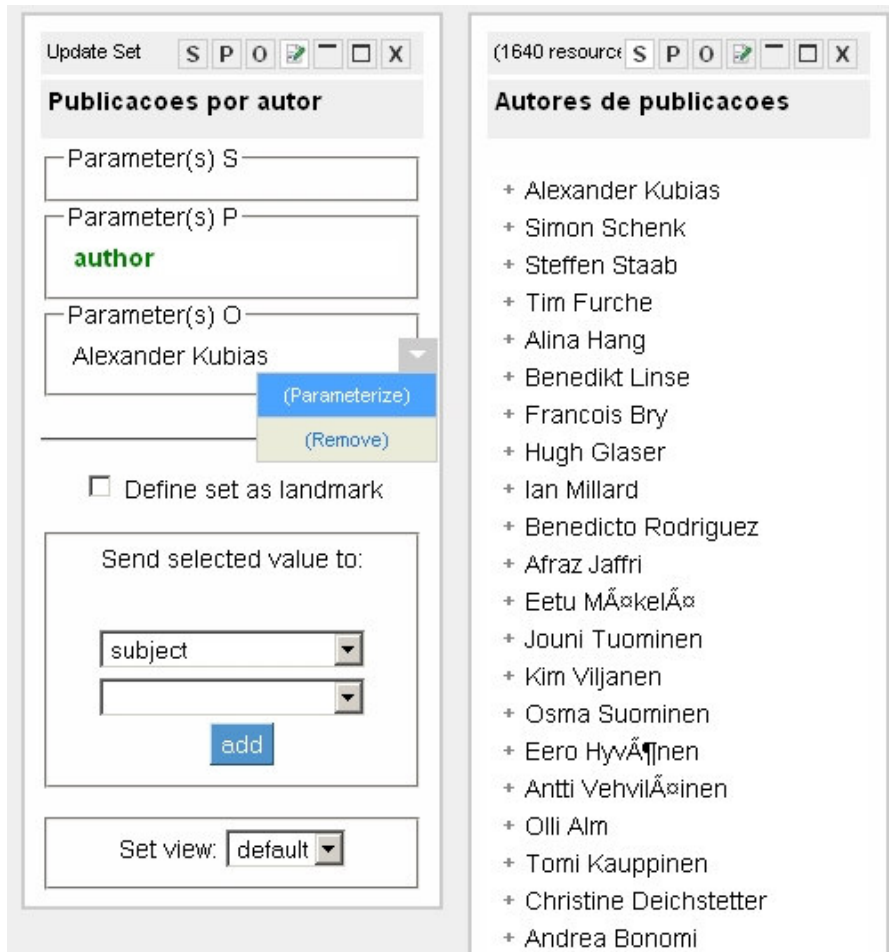


Figura 62 – Autor da publicação sendo parametrizado para tornar possível a reutilização da consulta.

Uma vez que a consulta está parametrizada, algumas funcionalidades específicas do compartilhamento fechado devem ser utilizadas para continuar a geração da aplicação. A primeira é a funcionalidade para definir os conjuntos que devem ser exibidos no momento em que a consulta em questão for acessada. É necessário que a aplicação apresente um ponto de partida para que o usuário final que estiver acessando possa alimentar os parâmetros da consulta montada. No exemplo em questão, o ideal é que o primeiro conjunto montado, uma lista com todos os autores existentes na base, sirva como esse ponto de partida, apresentando as opções disponíveis para o parâmetro da consulta. Para marcar o conjunto como ponto de partida para o caso de uso, basta selecionar a opção “Landmark” no seu modo de edição (Figura 63). Isso significa que os recursos do conjunto serão exibidos como opções quando o caso de uso em questão for selecionado do menu da aplicação gerada.

Figura 63 – Definição do conjunto de triplas como ponto de partida para o caso de uso.

O segundo passo é dar a semântica para a ação do usuário final de escolha de um autor. Quando o usuário final escolher algum autor da lista exibida, o segundo conjunto deve ser avaliado com a pessoa escolhida substituindo o valor do parâmetro e os resultados obtidos devem ser exibidos. É importante observar que o índice de autores é na verdade um conjunto de triplas, com os autores na posição do sujeito. No caso do exemplo, o comportamento de escolha deve ser associado somente ao índice formado pelos sujeitos das triplas do conjunto de triplas. A forma de adicionar esse comportamento ao conjunto que possui os autores é através de uma opção existente também no modo de edição do conjunto. Existem dois combo boxes, no primeiro é selecionado o índice no qual o comportamento está sendo adicionado (no caso do exemplo, é o índice formado pelos recursos que estão na posição do sujeito das triplas), no segundo combo box são exibidos todos os parâmetros de conjuntos de triplas existentes na aplicação. No segundo combo box, deve ser selecionado o conjunto que será avaliado e

exibido e o parâmetro que o valor selecionado pelo usuário final deve substituir (Figura 64). A posição da tripla escolhida no primeiro combo box e o nome do parâmetro e conjunto escolhidos serão exibidos em uma listagem, confirmando o comportamento desejado (Figura 65).



Figura 64 – Adicionando o comportamento de passagem de valor para parâmetro ao conjunto de triplas.

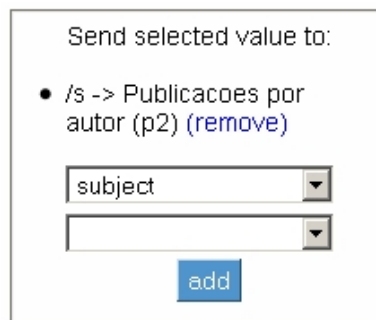


Figura 65 – Após o comportamento ter sido adicionado, uma indicação informa que o valor do recurso na posição do sujeito que for selecionado, será usado como parâmetro do outro conjunto de triplas.

Com os comportamentos definidos, quando o usuário final da aplicação gerada acessar através do menu o caso de uso do workbench em questão, será exibida para ele a listagem com todos os recursos do conjunto dos autores disponíveis. Uma vez que ele selecione algum autor, as triplas do segundo conjunto serão calculadas utilizando a pessoa escolhida no lugar do parâmetro e o resultado será exibido ao usuário.

Para finalizar o caso de uso descrito no Capítulo 4.1, é necessário que o usuário que esteja utilizando a aplicação possa selecionar uma das publicações da lista de publicações do autor selecionado e consiga visualizar alguns atributos disponíveis na base, como resumo, palavras-chaves, título e autores. Para isso é necessário criar um novo conjunto de triplas, que possua uma consulta SPO com essas propriedades na posição P e a posição S parametrizada. A publicação escolhida pelo usuário da aplicação será utilizada como valor da posição S. Uma boa forma de acessar as propriedades necessárias é visualizando todas as propriedades de uma publicação (Figura 66).

The screenshot shows the TACWEB EXCELPLORATOR interface. At the top, there is a URL bar with 'http://swrc.ontoware.org/ontology#keywords'. Below it, a navigation bar contains 'MENU', 'F F', 'u n -', and 'S P O' buttons. A green box highlights the 'Execute the expression.' button. The main area is divided into three panels:

- SET N**: Contains one resource with properties:
 - type=**: Resource, Poster
 - label=**: OWL SAIQL - An OWL DL Query Language for Ontology Extraction
 - title=**: OWL SAIQL - An OWL DL Query Language for Ontology Extraction
 - author=**: Alexander Kubias, Simon Schenk, Steffen Staab, Jeff Pan
 - abstract=**: Existing approaches for querying OWL DL do either only operate on syntactic constructs without taking into account the semantics of OWL or do only have a restricted access to the T-Box. We present SAIQL, the novel Schema And Instance Query Language for OWL DL,
- Publicacoes por autor**: Contains two resources:
 - + SAIQL Query Engine - Querying OWL Theories for Ontology Extraction
 - + OWL SAIQL - An OWL DL Query Language for Ontology Extraction
- Autores de public**: A panel for updating the set with parameters S, P, O. It includes a 'made' label and a 'Define set as' checkbox. Below it, a 'Send selected' section shows a list with a bullet point: '/s -> Publicacao autor (p2) (rem'. There is an 'add' button and a 'Set view: def' dropdown.

At the bottom, there is a 'CREATE NEW' button and a label 'Publicacoes por autor'.

PUC-Rio - Certificação Digital Nº 0721337/CA

Figura 66 – Criação de um conjunto com as propriedades desejadas através de uma operação SPO.

Uma vez que o conjunto foi criado utilizando uma publicação qualquer na posição S, é necessário parametriza-lo através do modo de edição do conjunto. Também é uma boa prática, como mencionado acima, renomear o conjunto com um nome que represente a sua semântica. Nesse caso, “Publicação” pode ser utilizado como nome (Figura 67).

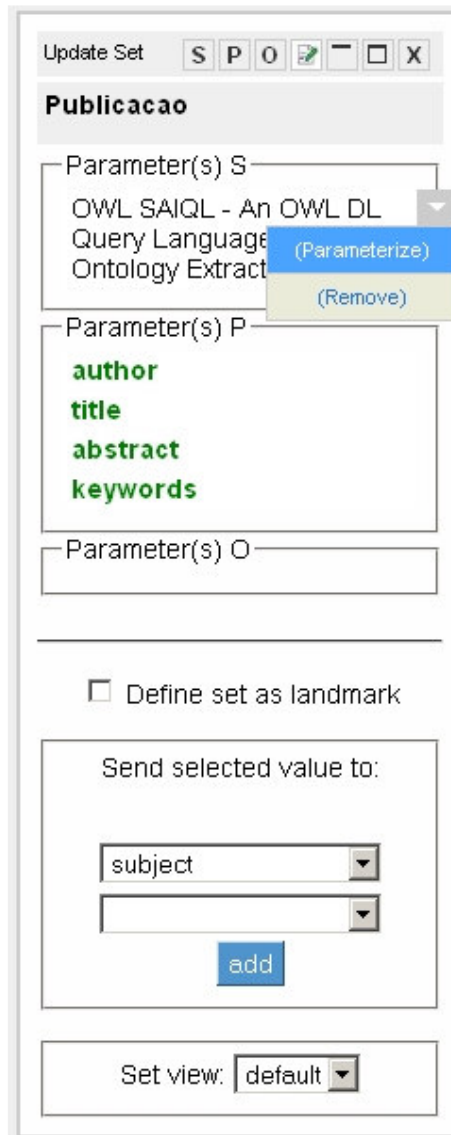


Figura 67 – Parametrizando conjunto de triplas, depois do mesmo ter sido criado e renomeado.

A última tarefa para finalizar a construção do caso de uso é adicionar o comportamento responsável por avaliar o novo conjunto utilizando a publicação selecionada pelo usuário da aplicação e exibir seus resultados. Como já mencionado acima, a forma de adicionar tal comportamento é através do modo de edição do conjunto que exibe a listagem das publicações. Deve-se selecionar a posição das triplas na qual o comportamento está sendo adicionado, no caso a posição do sujeito. E no segundo combo box, o novo conjunto criado e seu parâmetro para o envio do recurso escolhido pelo usuário (Figura 68). Nesse

momento o primeiro caso de uso está finalizado e tem seu comportamento é semelhante ao demonstrado no capítulo 4.1.

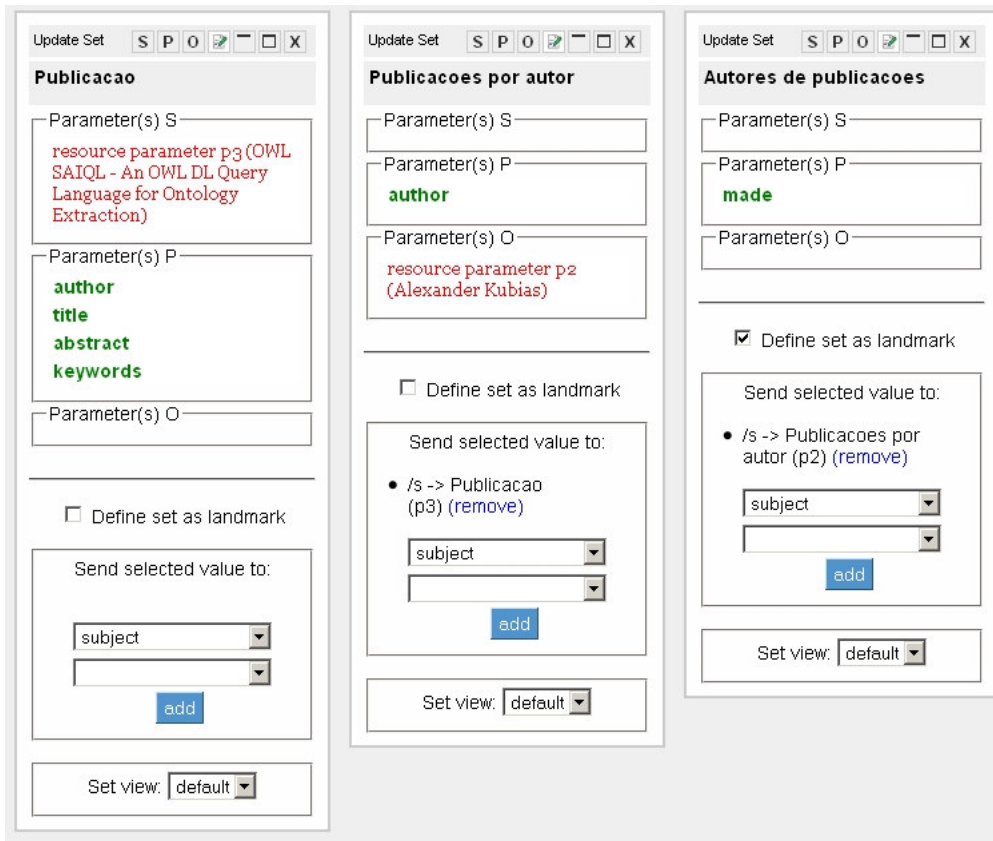


Figura 68 – Resultado final: os três conjuntos de triplas (no modo de edição) que juntos implementam o caso de uso.

Uma observação importante relacionada ao compartilhamento fechado é que o Excelplorator também permite o envio de valores para conjuntos de triplas parametrizados de outros workbenchs, ou seja, que pertencem a outro caso de uso. Essa é uma funcionalidade importante porque possibilita, por exemplo, que um caso de uso de visualização dos detalhes de certo tipo entidade seja acionado sempre que o usuário clicar em uma instância da entidade, independente de qual caso de uso ele esteja. Para isso, basta associar o comportamento desejado, de forma análoga ao envio de valores para conjuntos de triplas do mesmo workbench.

Para implementar outros casos de uso na aplicação, deve-se criar novos workbenchs. O segundo, terceiro, quarto e sexto caso de uso da aplicação exemplo podem ser montados de forma similar ao primeiro, e as suas demonstrações serão omitidas. Já o quinto possui pequenas diferenças por utilizar outros conceitos

apresentados, como o de operações customizadas e o de transdutores, e por isso será demonstrado a seguir.

No quinto caso de uso o usuário da aplicação gerada pode encontrar publicações contidas na base através de uma busca no atributo palavra-chave das publicações. As operações originais disponíveis no Explorator não possuem tratamento adequado para uma busca de string. Por exemplo, se a palavra buscada for “Ontology” e a base possuir uma publicação com a palavra-chave “ontology”, o operador SPO não retornará a tripla em questão devido à diferença de letra maiúscula e minúscula. Para realizar esse tipo de tratamento e aumentar o número de publicações encontradas, um operador customizado pode ser utilizado no caso de uso. O código do operador em questão é exibido no quadro abaixo.

```
param_a.select { |triple|  
  triple[2].to_s.strip.downcase == param_b[0].to_s.strip.downcase }
```

O código ser compacto é uma característica da linguagem Ruby. No vetor chamado de `param_a`, existem as triplas do primeiro parâmetro da operação. No vetor `param_b`, está o segundo parâmetro, que deve ser uma entrada de texto contendo a palavra-chave desejada. O método *select*, aplicado ao vetor `param_a` possui o comportamento de iterar os elementos do vetor e retornar um novo vetor que possui somente os elementos em que a passagem da iteração retornou o valor primitivo booleano verdadeiro. A variável `triple` ganha o valor da tripla corrente no passo da iteração, representada por um vetor com três posições (sujeito, predicado e objeto). A expressão avaliada em cada passo compara a segunda posição da variável `triple` (objeto da tripla) com o único valor de `param_b`, que é a palavra-chave do transdutor de texto. Um tratamento é dado tanto para o objeto da tripla quanto para a palavra-chave entrada pelo usuário com o objetivo de aumentar o número de triplas encontradas. A função *to_s* transforma os valores em string, a função *strip* retira qualquer espaçamento extra no entorno dos valores e a função *downcase* retira a caixa alta de ambos os valores. Observe que, embora o operador descrito tenha sido construído para ser utilizado em um caso de uso específico, ele é genérico o suficiente para filtrar qualquer conjunto de triplas que

possua como objeto um literal do tipo string. Portanto ele pode ser reaproveitado em diversas situações, inclusive por outros usuários do Excelplorator.

Para continuar a construção do caso de uso, é necessário um conjunto com todas as triplas da base que possuem a propriedade “Keyword”. Esse conjunto de triplas pode ser criado através de uma operação SPO (Figura 69).

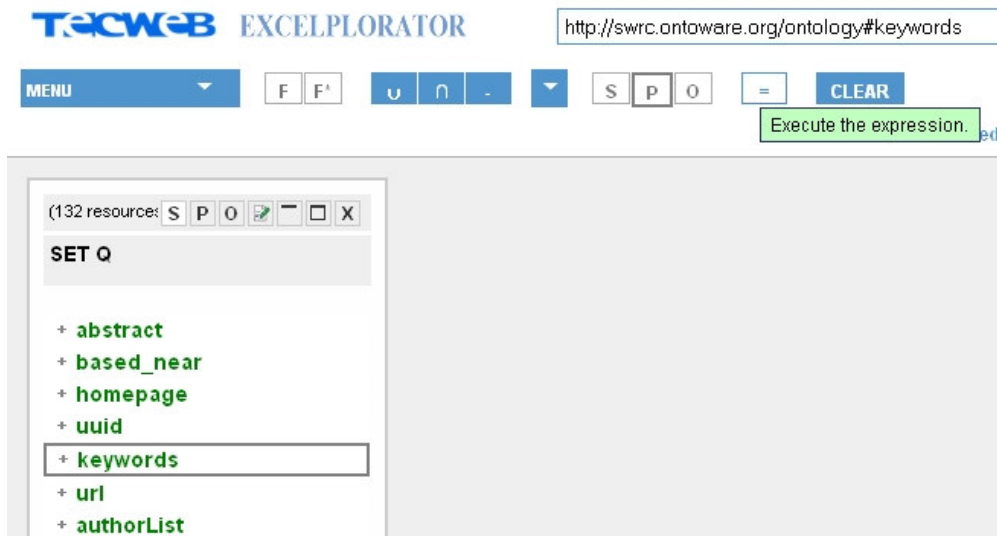


Figura 69 – Criação de um conjunto que possui as publicações na posição do sujeito e suas palavras-chave na posição de Objeto.

Com esse primeiro conjunto criado, é possível utilizar o novo operador para selecionar somente as triplas que possuem a palavra-chave desejada. Para isso é necessário um transdutor para entrada de texto, que pode ser criado através de uma opção no menu principal (Figura 27, capítulo 4.3.1.1). O novo transdutor surge no workbench e seu nome deve ser re-definido para a semântica desejada. Nesse caso, “keyword” pode ser utilizado como nome. Com o transdutor criado, é possível construir um segundo conjunto de triplas, que utiliza o primeiro conjunto como primeiro parâmetro e o transdutor de texto como segundo parâmetro (Figura 70). O novo conjunto gerado utiliza o novo operador para filtrar as triplas.

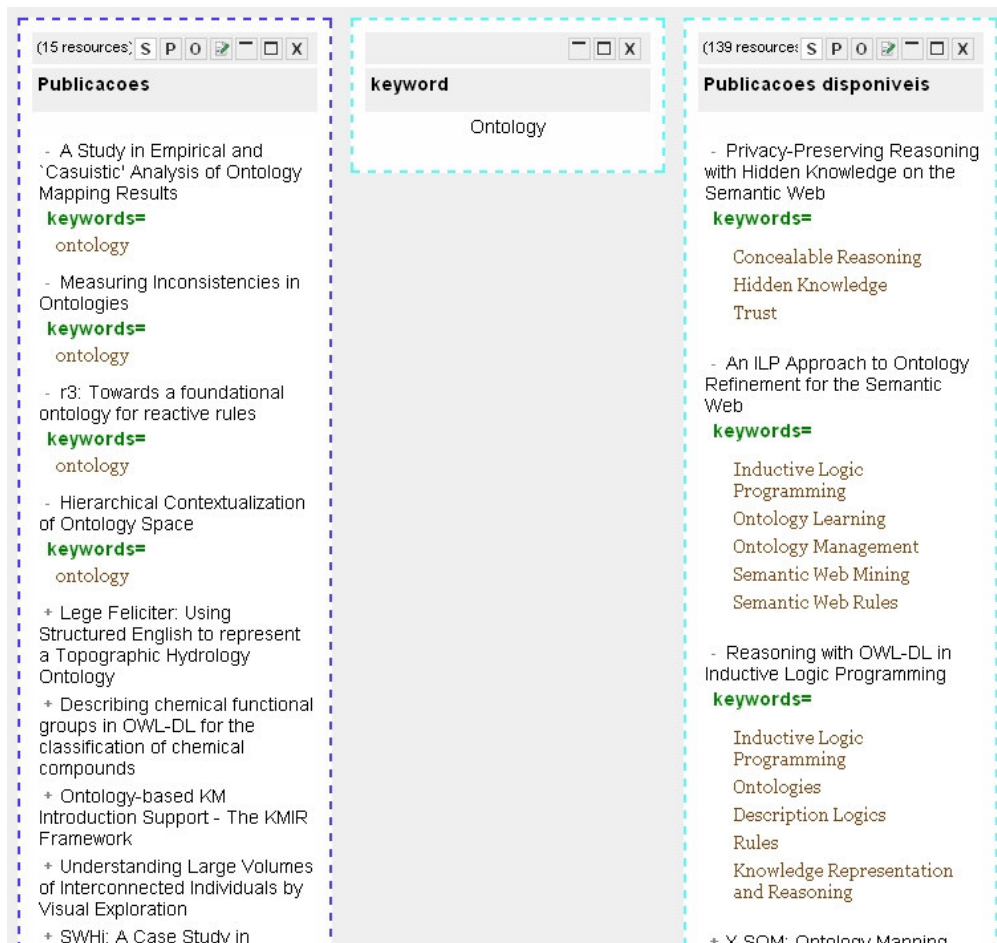


Figura 70 – O transdutor de texto e os dois conjuntos de triplas gerados.

Com todos os conjuntos necessários gerados, é preciso definir o comportamento da aplicação no caso de uso. O comportamento desejado é que no momento em que o caso de uso for acessado, o sistema exiba um campo de texto para que o usuário entre com a palavra-chave desejada. Depois que o usuário entrar com a palavra-chave, uma lista com as publicações encontradas deve ser exibida. Esse é o comportamento do Excelplorator ao tentar calcular e exibir um conjunto de triplas que só dependa de parâmetros que sejam entradas de texto. O Excelplorator exibirá um formulário para que o usuário entre com esses valores. Só então, utilizando os valores fornecidos, a ferramenta calcula os resultados e os exibe na tela. A única tarefa necessária para adicionar o comportamento desejado é, no modo de edição do ultimo conjunto criado (o filtrado pela palavra-chave), selecionar o checkbox “Landmark” (Figura 71). Uma vez que a opção esteja selecionada, quando o usuário acessar o caso de uso, o Excelplorator vai tentar

calcular o valor do conjunto e exibir o resultado. Com o seu comportamento definido, o quinto caso de uso da aplicação está pronto para ser utilizado.

Figura 71 – Conjunto de triplas gerado pela operação do filtro.

Repare que as dependências de um conjunto de triplas dependem do contexto da aplicação. Por exemplo, considere um conjunto de triplas que dependa de dois parâmetros, o primeiro sendo um recurso qualquer e o segundo uma entrada de texto. Se tal conjunto de triplas receber um valor para o primeiro parâmetro (normalmente um valor escolhido pelo usuário de alguma lista de recurso), ele passa a depender somente do segundo parâmetro, que é uma entrada de texto, e seu comportamento será como o descrito acima.

Para acessar a aplicação final montada, basta um clique no link “go to application”, que é exibido no canto superior direito da tela do Excelplorator (Figura 72). Nesse momento, o usuário é redirecionado para o ambiente de utilização das aplicações montadas pelo compartilhamento fechado.

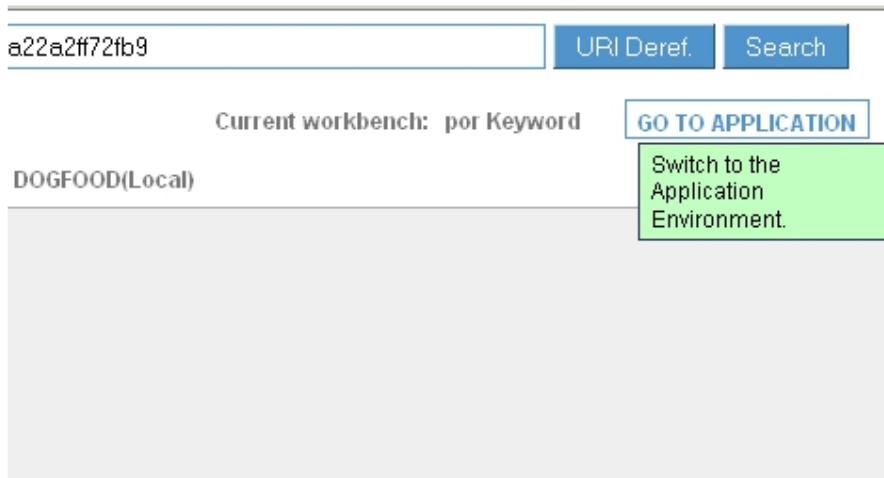


Figura 72 – Link que leva o usuário ao ambiente de utilização da aplicação montada.

A aplicação montada nos exemplos desta seção funcionará exatamente como a aplicação exemplo, demonstrada na seção 4.1. A apresentação da aplicação gerada é a padrão do Excelplorator, mas ela pode ser customizada tanto na aparência quanto em seu comportamento, como será discutido na próxima seção.

O usuário que está utilizando o Excelplorator pode salvar a aplicação que está sendo montada. O objetivo de salvar uma aplicação é que ele possa carregá-la novamente em uma nova sessão e modificá-la como desejar. Para salvar uma aplicação, o usuário deve primeiro se autenticar no Excelplorator. A autenticação é realizada através do sistema open-id [Recordon et al, 2006]. Open-id é um sistema distribuído de autenticação baseado em provedores de identidade. Basicamente, o usuário não precisa se cadastrar no Excelplorator para poder se autenticar, basta ter uma conta em um provedor de identidade que suporte o padrão open-id. Diversos serviços populares provêm tal funcionalidade, como Google¹⁰, Yahoo!¹¹ e MySpace¹². Para se autenticar, basta um clique no link localizado a direita da barra inferior do Excelplorator (Figura 73). Nesse momento, a ferramenta vai pedir o provedor de identidade que o usuário deseja utilizar (Figura 74). Uma vez autenticado, a sessão corrente pode ser salva e estará

¹⁰ Google – <http://www.google.com>

¹¹ Yahoo! – <http://www.yahoo.com>

¹² MySpace – <http://www.myspace.com>

associada à conta do usuário. Somente o mesmo usuário, após autenticação, poderá carregar e modificar a aplicação.

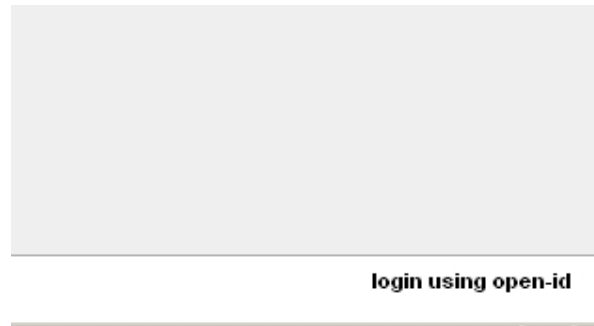


Figura 73 – Link que o usuário deve acessar para se autenticar.

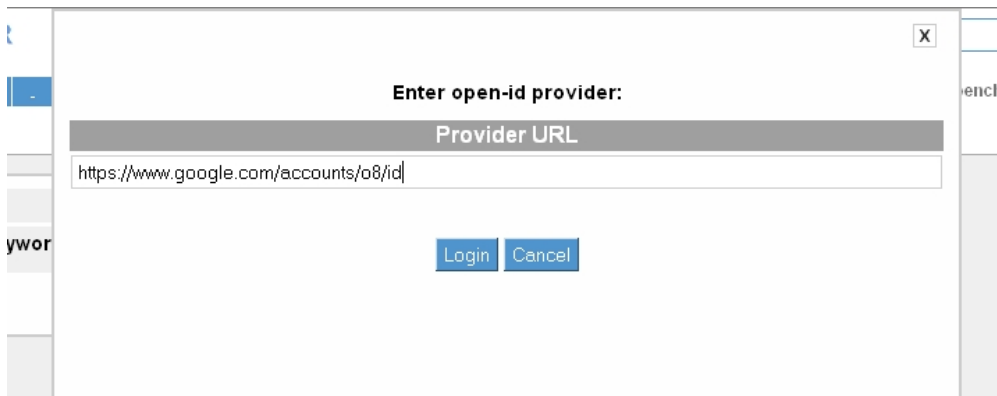


Figura 74 – Usuário deve prover o seu provedor de identidade que suporte o protocolo open-id.

Para salvar a aplicação corrente, basta acessar a opção “Applications” no menu principal do Excelplorator (Figura 75). Um índice com todas as aplicações salvas do usuário autenticado será exibido, assim como um formulário para que ele salve, caso deseje, a aplicação corrente. Para recuperar qualquer aplicação previamente salva, basta se acessar o Excelplorator, se autenticar via open-id, acessar a mesma opção do menu e selecionar do índice de aplicações disponíveis a aplicação desejada (Figura 76).



Figura 75 – Opção do menu para salvar a aplicação corrente ou carregar aplicações salvas.

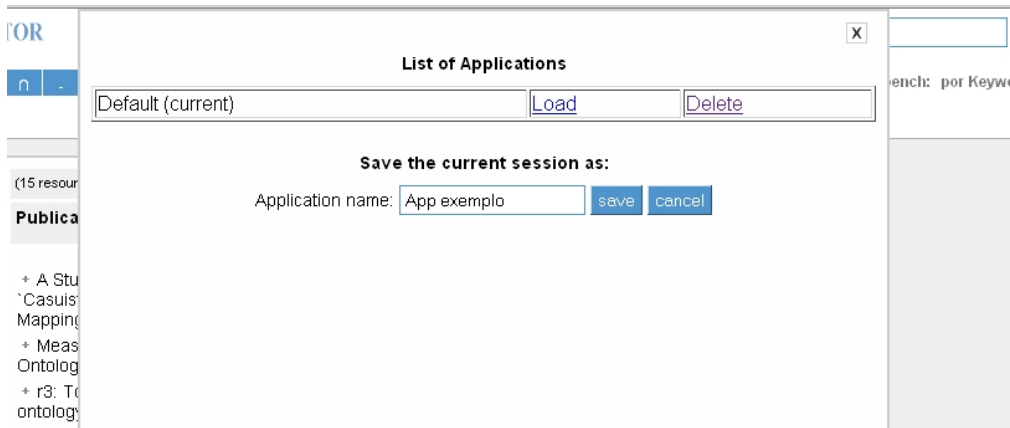


Figura 76 – Índice de aplicações salvas e formulário para salvar a aplicação atual.

Também é possível salvar uma aplicação sem estar autenticado no Excelplorator. Nesse caso, a aplicação será persistida de forma anônima e qualquer pessoa poderá carregá-la e modificá-la como desejar.

Nesta subseção foi discutida a criação de uma aplicação web para consultas a base utilizando o Excelplorator. Na próxima seção será abordada a customização da camada de apresentação das aplicações construídas. A aplicação pode ser customizada tanto na sua aparência quanto em seu comportamento.

4.3.5.2

A customização da camada de apresentação da aplicação gerada

Ao gerar uma aplicação utilizando o Excelplorator, a camada de visualização utiliza a interface padrão da ferramenta. Uma funcionalidade foi desenvolvida para tornar possível a customização da interface das aplicações construídas. A customização é realizada através da edição do código HTML [Raggett et al, 1999] e CSS [Bos et al, 2009] que define a interface. A customização da interface pode ser feita diretamente no Excelplorator, em tempo de execução, como será demonstrado ainda nesse capítulo.

Dois aspectos da apresentação podem ser customizados, a aparência e parte do comportamento. A customização da aparência diz respeito à forma como a aplicação será exibida aos usuários finais. Pode-se alterar tanto o código HTML quanto o CSS que é enviado pelo Excelplorator ao navegador do usuário. Já a customização do comportamento diz respeito a como o sistema deve se comportar nas iterações com o usuário. Por exemplo, ao selecionar uma das opções de um índice, é possível determinar em qual posição a nova janela, contendo os dados requisitados, deve se posicionar. Também se o índice deve continuar a ser exibido para que o usuário possa explorar outras opções ou se deve desaparecer da interface. Esse tipo de comportamento é de responsabilidade da camada de apresentação e pode ser customizado no Excelplorator em cada aplicação montada, como será demonstrado ainda nesse capítulo.

4.3.5.2.1

A customização da aparência da camada de apresentação

O primeiro passo para customizar a interface é acessar a opção “Views” no menu principal da ferramenta (Figura 77). Nesse momento, serão exibidos dois índices na tela (Figura 78).



Figura 77 – Opção do menu que leva ao controle de customização de visualização.

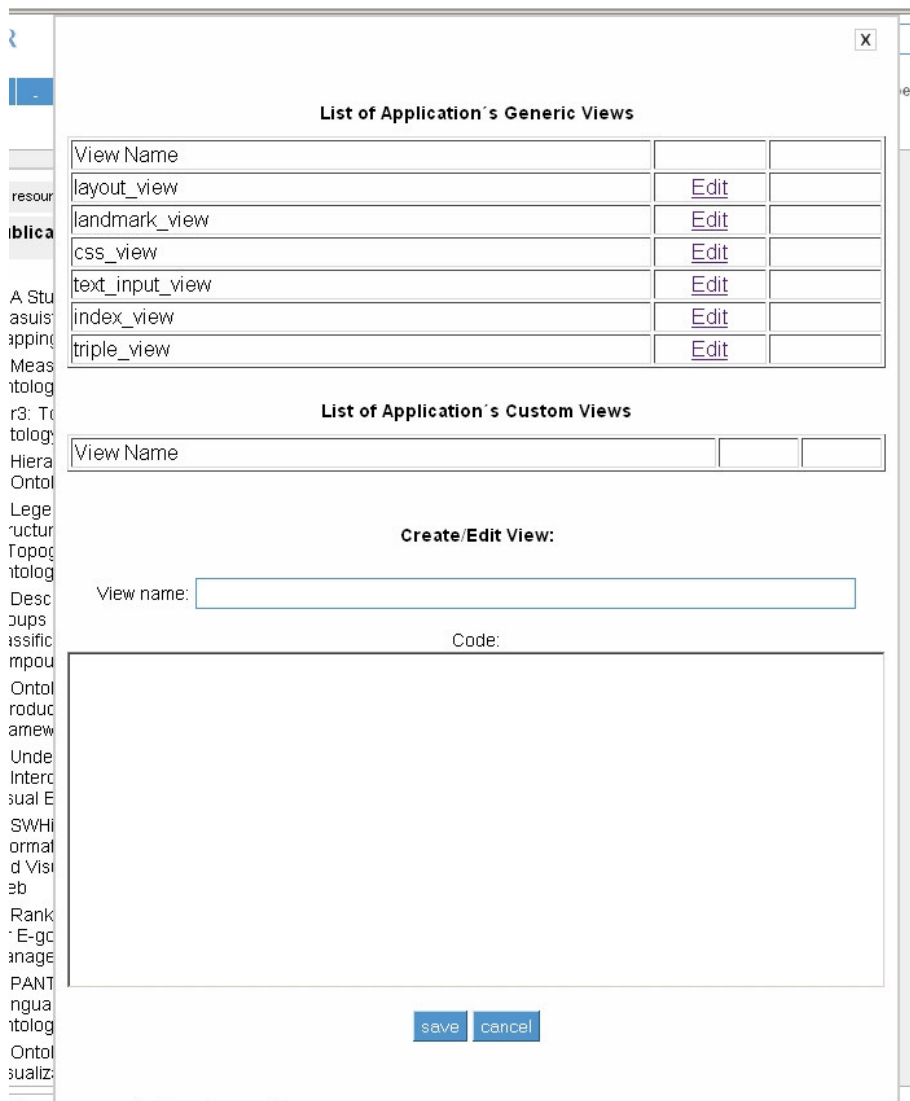


Figura 78 – Tela de customização/criação de novas interfaces.

O primeiro índice exibe as interfaces ou fragmentos de interfaces que existem na aplicação gerada. Esse índice possui seis elementos e é fixo. Cada entrada do índice representa um componente na interface da nova aplicação. Eles são:

1. `layout_view`: Código HTML que define a estrutura principal da página. Possui o local onde o menu com os links para os casos de uso será exibido, o rótulo a ser exibido no topo do navegador, entre outras coisas da estrutura principal de uma página HTML.
2. `landmark_view`: Possui o fragmento de código HTML que renderiza o menu principal da aplicação onde estão os links para os casos de uso. O local aonde esse fragmento de código deve ser incluído é definido no `layout_view`.
3. `css_view`: Possui o código CSS utilizado por todas as páginas da aplicação construída.
4. `text_input_view`: Fragmento de código HTML que exibe um formulário para entrada de texto pelo usuário. Utilizado quando existe um conjunto de triplas que depende do valor de um transdutor de texto. Também é renderizado via Ajax, dependendo dos casos de uso.
5. `index_view`: Fragmento de código HTML que renderiza um conjunto de triplas como um índice, para que o usuário selecione algum. Esse fragmento de código é carregado via Ajax e é utilizado para exibir conjuntos de triplas de acordo com as definições dos casos de uso.
6. `triple_view`: Possui o fragmento de código HTML que exibe as triplas de um conjunto de triplas. Renderizado quando necessário via Ajax, de forma análoga aos dois componentes anteriores.

Todos os componentes possuem um código padrão, que é utilizado caso não seja feita nenhuma customização. Para alterar o código de algum componente, basta clicar no link “Edit” do componente desejado. Nesse momento o formulário que existe no final da página é carregado com o código corrente do componente. Por exemplo, o fragmento de código HTML padrão do componente `index_view`:

```

<div class="set menu _set" id="<%=uri(resourceset)%>">
  <div class="menutitle">
    <span class="menutitle_span">
      <%= resourceset.explorator::name %>
    </span>
  </div>
  <div class="menuresources">
    <ul class="menuresources_ul">
      <% groupBy(:s, true).each do |resource| %>
        <li class="menuresource">
          <span class="menuresource_span">
            <a class="menuresource_a _menulink"
              id="<%=uri(resource)%>" position="s">
              <%= render_resource(resource) %>
            </a>
          </span>
        </li>
      <% end %>
    </ul>
  </div>
</div>

```

Após alterar o código do componente, as alterações devem ser salvas. O nome não deve ser alterado para que o novo código substitua o código padrão. No momento que o componente é customizado, um novo link aparece na sua linha no índice. O link “Use default” serve para descartar as customizações, sobrescrevendo o código modificado pelo padrão da ferramenta (Figura 79).

List of Application's Generic Views

View Name		
layout_view	Edit	
landmark_view	Edit	Use default
css_view	Edit	Use default
text_input_view	Edit	
index_view	Edit	
triple_view	Edit	

Figura 79 – A opção de descartar customização surge quando componente é modificado.

Os três primeiros componentes (layout_view, landmark_view e css_view) são únicos na aplicação. Ou seja, eles sempre serão utilizados da mesma forma quando um usuário qualquer acessar a aplicação. O quarto componente

(text_input_view) é utilizado quando existe a necessidade de entrada de dados pelo usuário. Já os dois últimos são utilizados para renderizar os conjuntos de triplas que fazem parte das consultas. A decisão de qual componente será utilizado para renderizar um conjunto de triplas é baseada no comportamento do caso de uso. Quando o conjunto de triplas possui o comportamento de enviar um recurso selecionado para dar valor a um parâmetro de outro conjunto, parte-se do princípio que se trata de um índice, e o componente utilizado para a renderização é o index_view. No caso contrário, quando o conjunto não passa valores para serem avaliados em parâmetros de outros conjuntos, ele é considerado o resultado final da consulta, e o componente utilizado para a sua renderização é o triple_view.

A heurística utilizada para a escolha dos componentes que renderizam os conjuntos de as triplas nem sempre é satisfatória. Além disso, em alguns casos pode não fazer sentido utilizar o mesmo componente para renderizar todos os conjuntos de triplas de uma aplicação. Algum índice ou resultado final pode necessitar de uma interface diferente das outras. Para permitir essa customização, também é possível a criação de novos componentes. Os novos componentes podem ser criados utilizando o formulário exibido no final da página. Deve-se preencher um nome que represente a semântica do novo componente de interface e o código do mesmo. Novos componentes criados aparecem no segundo índice da página. Podem ser criados tantos novos componentes quantos forem necessários para a aplicação. No índice dos novos componentes é possível editar ou apagá-los. Os novos componentes criados ficam disponíveis para serem utilizados em qualquer conjunto de triplas da aplicação (Figura 80).

List of Application's Custom Views

View Name		
person_view	Edit	Delete
work_view	Edit	Delete

Figura 80 – Lista de novas interfaces criadas para customização da aparência da aplicação.

Para modificar o componente que renderiza um conjunto de triplas basta acessar seu modo de edição e selecionar o componente desejado na listagem exibida no final das opções do conjunto (Figura 81).

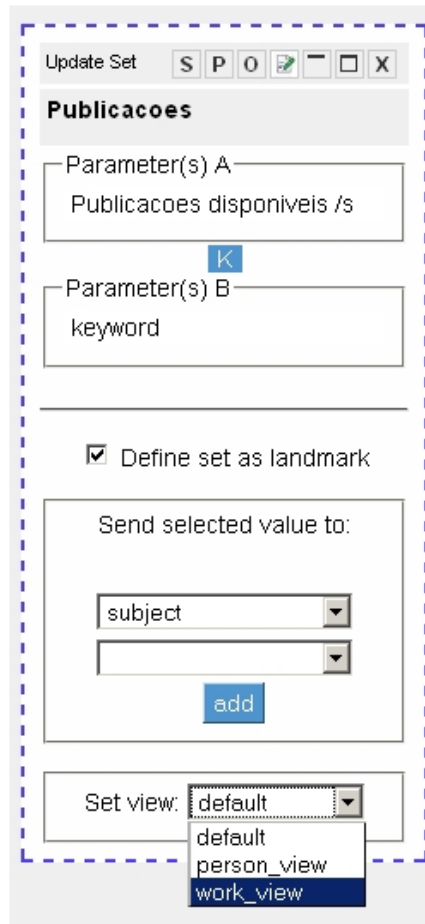


Figura 81 – Seleção do componente de interface que vai renderizar o conjunto de triplas.

Quando a aplicação vai exibir um conjunto de triplas, ela segue a seguinte ordem para decidir qual componente vai utilizar:

1. Caso exista um componente customizado escolhido no modo de edição, ele é utilizado.
2. Se no modo de edição o componente apontar para default, a aplicação descobre, baseado na heurística já apresentada, qual dos componentes padrões será utilizado: `index_view` ou `triple_view`. Nesse momento ela procura uma versão customizada desse componente e utiliza essa versão caso exista.
3. Caso não exista uma versão customizada do componente escolhido pela heurística, a versão padrão é utilizada na renderização.

Como discutido acima, a apresentação pode ser customizada em mais de um nível: primeiro modificando os componentes padrões da aplicação, e depois pela

criação de novos componentes que podem ser utilizados em conjuntos de triplas específicos.

4.3.5.2.2

A customização do comportamento da camada de apresentação

O comportamento da camada de apresentação é outro aspecto que pode ser customizado. Esse aspecto diz respeito a como o sistema deve se comportar nas interações com o usuário. Devido a restrições de tempo, somente dois pontos de customização de comportamento foram implementados, com o objetivo somente de demonstrar o conceito.

A customização do comportamento foi desenvolvida através da extensão do HTML utilizado na renderização da aplicação. Através de novos atributos criados nas marcações originais da linguagem,¹³ o comportamento da camada de apresentação da aplicação pode ser alterado.

As duas customizações de comportamento desenvolvidas são referentes à exibição dos conjuntos de triplas na aplicação gerada. A primeira define se um conjunto de triplas deve continuar a ser exibido para o usuário mesmo que um dos seus links seja clicado, ou seja, um dos recursos foi escolhido e deverá ser enviado para ser utilizado como valor de um parâmetro de outro conjunto de triplas. A segunda customização define o posicionamento que o conjunto de triplas deve ocupar quando for exibido.

O comportamento padrão de um conjunto de triplas é desaparecer da tela caso o usuário selecione algum dos seus recursos (assumindo que se trata de um índice). Para manter o índice de recursos na tela, basta adicionar o atributo “fixed” com o valor “true” à marcação “div” que define o índice na interface. A alteração deve ser realizada através da opção “Views” no menu principal da aplicação, da mesma forma em que o código da HTML da interface é customizado (Figuras 77 e 78). Utilizando o código da interface padrão como exemplo, a marcação div deverá ser alterada da seguinte forma:

- Código original:

¹³ Linguagem de marcação - http://en.wikipedia.org/wiki/Markup_language

```
<div class="set menu _set" id = "<%=uri(resourceset)%>" >
```

- Código customizado:

```
<div class="set menu _set" id = "<%=uri(resourceset)%>" fixed="true">
```

Através do primeiro caso de uso da aplicação exemplo, é possível exemplificar a mudança de comportamento obtida. Logo após o usuário acessar o caso de uso, é exibido para ele um índice contendo todos os autores da base (Figura 82). No comportamento padrão, quando o usuário escolhe um autor, o índice de autores some da tela, restando somente à lista de publicações do autor escolhido (Figura 83). Com o comportamento customizado, quando o usuário faz a sua escolha, o índice continua a ser exibido, permitindo que ele altere sua escolha sem a necessidade de recomeçar o caso de uso (Figura 84).

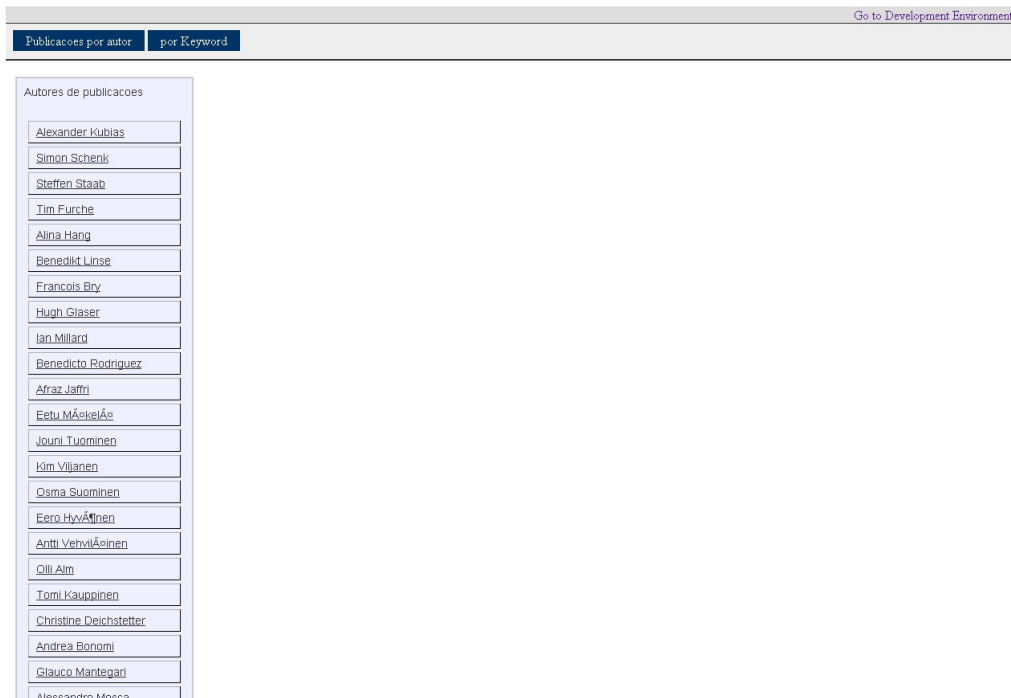


Figura 82 – Índice de pessoas que é exibido após o acesso ao caso de uso.

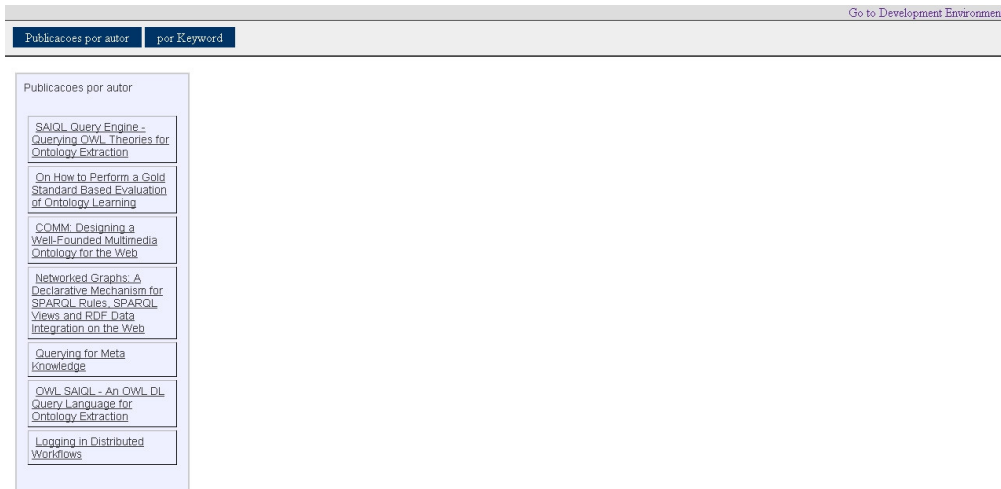


Figura 83 – No comportamento padrão, após a seleção de uma pessoa, o índice de pessoas some da tela e um novo índice com as publicações é exibido.

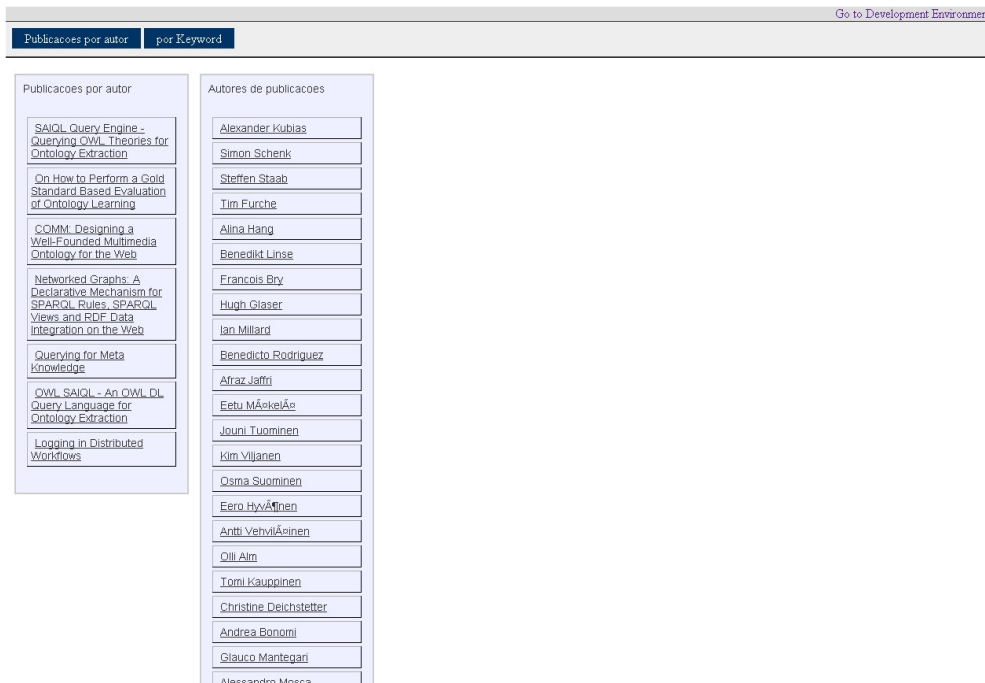


Figura 84 – Já com o comportamento customizado, o índice de pessoas permanece na tela mesmo após a seleção de uma pessoa.

O segundo ponto de customização é referente à posição que um conjunto de triplas é exibido. O comportamento padrão nesse caso é a exibição do conjunto sempre à esquerda dos outros conjuntos já existentes na tela. É possível alterar o

comportamento de forma que o novo conjunto seja exibido à direita dos conjuntos já existentes. Para isso, basta adicionar o atributo “insertPosition” com o valor “after” na marcação “div” que define o conjunto de triplas. Novamente, utilizando o código da interface padrão (dessa vez o código da interface de exibição de triplas) como exemplo, a marcação “div” deverá ser alterada da seguinte forma:

- Código original:

```
<div class="set final _set" id="<%=uri(resourceSet)%>" >
```

- Código customizado:

```
<div class="set final _set" id="<%=uri(resourceSet)%>" insertPosition="after">
```

Através do mesmo caso de uso utilizado no primeiro ponto de customização, é possível ver a diferença de comportamento observando a posição do conjunto de triplas que contém os detalhes de uma publicação. No comportamento original, o conjunto é exibido à esquerda dos índices (Figura 85). Já com o comportamento customizado, a exibição é realizada à direita dos índices existentes na tela (Figura 86).

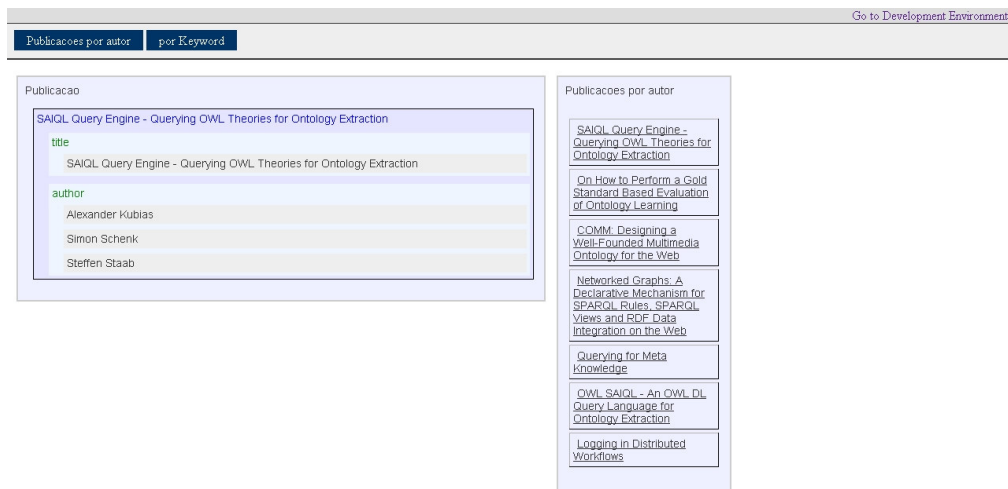


Figura 85 – No comportamento padrão, novas janelas surgem à esquerda das existentes.

The screenshot shows a web interface with a navigation bar at the top containing "Publicações por autor" and "por Keyword", and a "Go to Development Environment" link. Below the navigation bar, there are two main panels. The left panel, titled "Publicações por autor", contains a list of publication titles: "SAIQL Query Engine - Querying OWL Theories for Ontology Extraction", "On How to Perform a Gold Standard Based Evaluation of Ontology Learning", "COIM: Designing a Well-Founded Multimedia Ontology for the Web", "Networked Graphs: A Declarative Mechanism for SPARQL Rules, SPARQL Views and RDF Data Integration on the Web", "Querying for Meta Knowledge", "OWL_SAIQL - An OWL DL Query Language for Ontology Extraction", and "Logging in Distributed Workflows". The right panel, titled "Publicacao", displays the details for the selected publication "SAIQL Query Engine - Querying OWL Theories for Ontology Extraction". It shows the title, author names (Alexander Kubias, Simon Schenk, and Steffen Staab), and other metadata.

Figura 86 – O comportamento pode ser customizado para que um conjunto seja exibido à direita dos conjuntos existentes.

Nesse capítulo foi apresentada a solução proposta para os problemas discutidos no capítulo dois. A ferramenta desenvolvida para ilustrar os conceitos propostos também foi apresentada, assim como suas funcionalidades. No próximo capítulo alguns detalhes técnicos da implementação do Excelplorator serão apresentados.