

6

Um exemplo: Controle de Acesso em um Sistema de Gestão de Trabalhos em Conferências

Foi desenvolvida uma aplicação de exemplo no domínio de um sistema de revisão de submissões para conferências, seguindo as etapas de modelagem segundo o método SHDM e construído com apoio do ambiente de desenvolvimento Synth.

Neste capítulo serão detalhadas apenas as tarefas que descrevem as primitivas de controle de acesso da aplicação, pertencentes à etapa de projeto comportamental do método SHDM.

6.1. Descrição do Cenário

Um sistema de revisão de artigos é um sistema de gerenciamento das atividades realizadas em uma conferência acadêmica. Ele fornece um conjunto de serviços tais como receber a submissão de artigos, atribuir (manualmente ou de forma automática) os artigos aos revisores, criar as informações da conferência e dos membros do comitê do programa, enviar notificações por email, etc. Alguns exemplos de um sistema de revisão de artigos são: EasyChair³⁴, Confious³⁵, ConfTool³⁶, ConfMaster³⁷, entre outros.

A seguir segue a listagem das principais tarefas de cada ator identificadas na aplicação.

a) *Author*: são aqueles que submetem artigos para uma conferência.

Suas principais funções incluem:

- Registrar-se no sistema;
- Fornecer informações pessoais;
- Submeter um artigo;
- Visualizar o status da revisão;

³⁴ <http://www.easychair.org/>

³⁵ <http://www.confious.com/>

³⁶ <http://www.conftool.net>

³⁷ <http://www.confmaster.net>

- Visualizar o resultado final da revisão;
 - Submeter a versão final do artigo aceito.
- b) *Reviewer*: são responsáveis por avaliar os artigos submetidos pelos autores. Entre suas principais funções, destacam-se:
- Informar quais tópicos de interesse é de sua preferência;
 - Fazer download dos artigos que foram atribuídos a ele (a);
 - Revisar um artigo atribuído a ele;
 - Modificar uma revisão que foi criada por ele (a);
 - Visualizar a revisão de um artigo (alocado a ele) feita por outros revisores. Isso só é permitido depois que o revisor tenha criado antes a sua própria revisão;
 - Declarar explicitamente um conflito de interesse com um dado artigo.
- c) *Senior Reviewer*: são responsáveis por supervisionar o processo de revisão de um conjunto de artigos. Os membros deste papel também podem revisar um artigo submetido. Suas principais funções incluem:
- Visualizar as revisões de quaisquer artigos que pertencem ao conjunto de artigos alocado a ele (a);
 - Criar novas revisões de quaisquer artigos que pertencem ao conjunto de artigos alocado a ele (a);
 - Todas as permissões de um *reviewer*.
- d) *Program Committee (PC) Chair*: são responsáveis por supervisionar o processo de revisão de todos os artigos submetidos. Suas principais funções são:
- Enviar convites para as pessoas participarem da fase de revisão de artigos;
 - Definir quem serão os membros de *reviewer*;
 - Definir quem serão os membros de *senior reviewer*;
 - Atribuir para revisão os artigos aos membros de *reviewer* e/ou aos membros de *senior reviewer* e/ou aos próprios membros do *PC chair*;
 - Alocar um conjunto de artigos para os membros de *senior reviewer* coordenarem;
 - Revogar uma atribuição de revisores associado a um artigo;

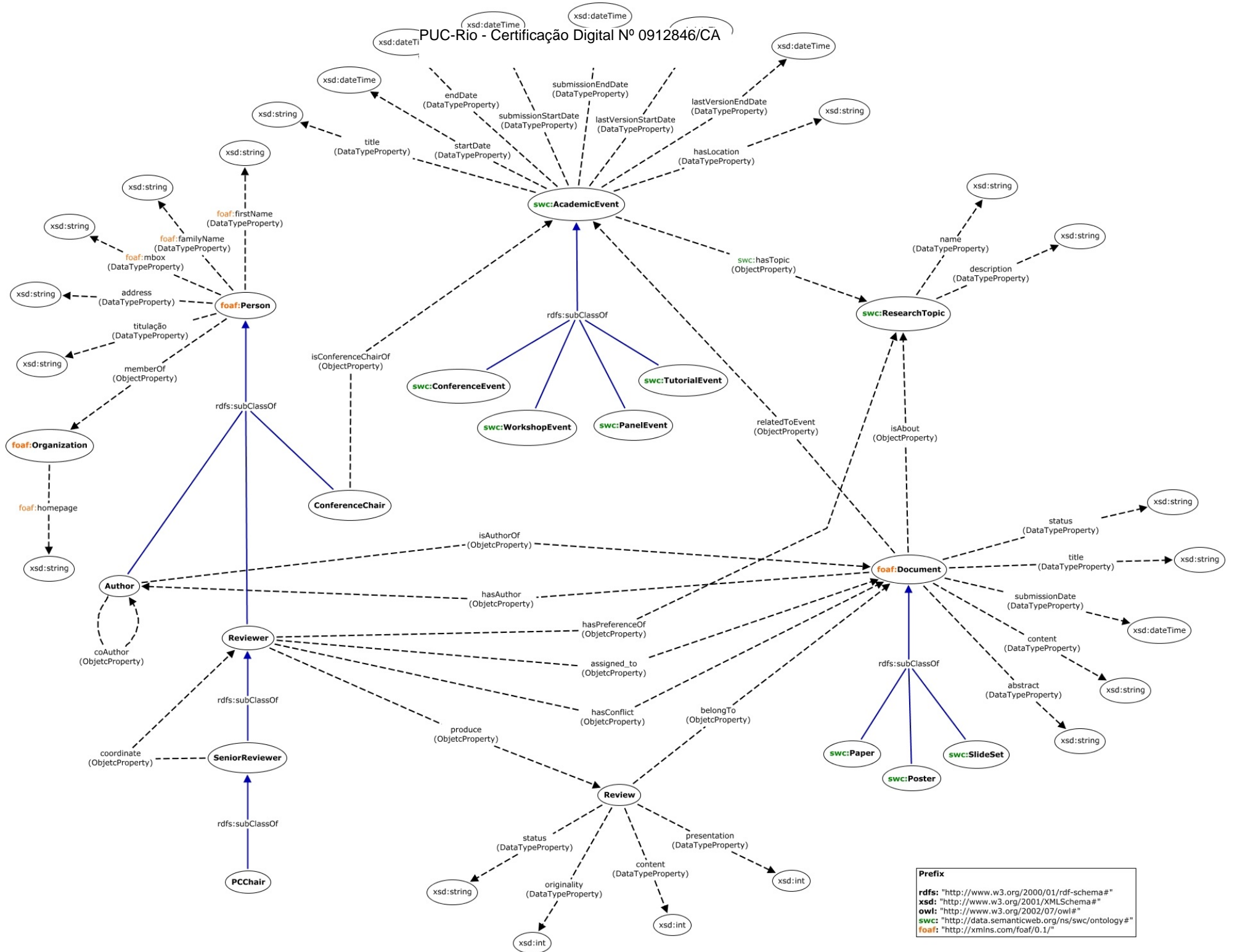
- Enviar notificações para alguns ou todos os membros de *reviewer* e/ou *senior reviewer*;
 - Liberar a divulgação das revisões para os autores;
 - Tomar a decisão final sobre a aceitação de um artigo após a sua revisão;
 - Todas as permissões de um *senior reviewer*.
- e) *Conference Chair (CC)*: é responsável por gerenciar e monitorar toda a conferência. Suas funções incluem:
- Fornecer informações sobre a conferência, como nome da conferência, prazos de submissão dos artigos, tópicos de interesse da conferência, etc.;
 - Definir quem serão os membros de *PC chair*;

6.2. Ontologia de Domínio

Foi criada uma ontologia de domínio (figura abaixo) para descrever os conceitos relacionados ao domínio da aplicação. Os vocabulários reutilizados nesta ontologia foram o FOAF³⁸ (Friend of a Friend) e o SWC³⁹ (Semantic Web Conference Ontology).

³⁸ FOAF Vocabulary Specification – <http://xmlns.com/foaf/spec/>

³⁹ http://data.semanticweb.org/ns/swc/swc_2009-05-09.html



6.3. Definição das Primitivas do Modelo de Controle de Acesso

Os seguintes sujeitos (rbac:Subject) foram definidos:

- myconference:Reviewer;
- myconference:Author;
- myconference:SeniorReviewer;
- myconference:ConferenceChair;
- foaf:Person;
- myconference:PCChair;

Os seguintes objetos (rbac:Object) foram definidos:

- shdm:Context;
- foaf:Document;
- myconference:Review;
- swc:Paper;
- swc:Poster;
- swc:SlideSet;

Os seguintes papéis (rbac:Role) foram criados:

- rbac:reviewer_role;
- rbac:senior_reviewer_role;
- rbac:pcchair_role;
- rbac:conference_chair_role;
- rbac:author_role;

A definição da hierarquia de papéis do modelo de controle de acesso baseado em papel foi apresentada na seção 2.2.2.

A Figura 34 ilustra a hierarquia de papéis para o sistema de revisão de artigos. O papel de maior poder nessa hierarquia é o rbac:pcchair_role e o de menor poder é o rbac:reviewer_role. Quando o papel rbac:pcchair_role é ativado, todas as permissões de acesso associadas aos papéis que estão abaixo dele ficarão disponíveis para ele, além das permissões associadas diretamente com seu próprio papel. Os papéis rbac:author_role e rbac:conference_chair_role não participam de uma hierarquia.

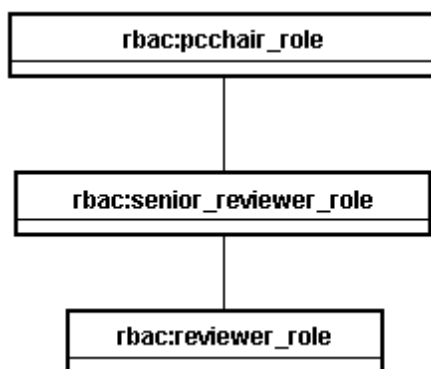


Figura 34 – Hierarquia de papéis do sistema de conferências

As associações definidas entre o sujeito e o papel são:

- Recursos da classe myconference:ConferenceChair podem ativar o papel rbac:conference_chair_role;
- Recursos da classe myconference:Author podem ativar o papel rbac:pauthor_role;
- Recursos da classe myconference:PCChair podem ativar o papel rbac:pcchair_role;
- Recursos da classe myconference:Reviewer podem ativar os papéis rbac:reviewer_role e rbac:author_role;
- Recursos da classe SeniorReviewer podem ativar o papel senior_reviewer_role.

As associações definidas entre o papel com a ação e o objeto são mostradas na Tabela 2. Observe-se que a ação rbac:context corresponde à operação de navegação em contexto que já vem definida primitivamente no ambiente Synth.

	Papéis	Ações	Objetos
1	rbac:reviewer_role	rbac:context	shdm:Context
2	rbac:reviewer_role	rbac:context	myconference:Review
3	rbac:reviewer_role	rbac:createReview	foaf:Document
4	rbac:senior_reviewer_role	rbac:context	myconference:Review
5	rbac:pcchair_role	rbac:context	myconference:Review
6	rbac:conference_chair_role	rbac:context	shdm:Context
7	rbac:author_role	rbac:context	shdm:Context
8	rbac:author_role	rbac:context	myconference:Review

Tabela 2 – Associações definidas entre o papel com a ação e o objeto

6.4. Políticas de Controle de Acesso Implementadas

A seguir apresentam-se alguns exemplos de políticas de autorização criadas para esta aplicação usando o ambiente de autoria do Synth.

1. O Quadro 23 define uma política que declara que o revisor não pode revisar um artigo cujo autor ou co-autor seja ele próprio.

```

1  {   ?A a rbac:Action ;
2      |   rbac2:subject ?S ;
3      |   rbac2:object ?O .
4
5      ?A a rbac:createReview .
6      ?S rbac:activeRole rbac:reviewer_role .
7      ?O a foaf:Document .
8
9      ?S myconference:assigned_to ?O .
10
11     ?S myconference:isAuthorOf ?O .
12
13 } => { ?A a rbac:ProhibitedAction } .

```

Quadro 23 – Exemplo de política que define um conflito ao criar uma revisão do próprio artigo

2. A política ilustrada no Quadro 24 declara que o revisor só pode criar revisões de artigos que foram alocados para ele.

```

1  {   ?A a rbac:Action ;
2      |   rbac2:subject ?S ;
3      |   rbac2:object ?O .
4
5      ?A a rbac:createReview .
6      ?S rbac:activeRole rbac:reviewer_role .
7      ?O a foaf:Document .
8
9      ?S myconference:assigned_to ?O .
10
11 } => { ?A a rbac:PermittedAction } .

```

Quadro 24 – Exemplo de política que define que o revisor só pode criar revisões de artigos que foram alocados para ele

3. A política ilustrada no Quadro 25 declara que o revisor não pode revisar um artigo de um autor que pertença a mesma instituição que a sua.

```

1  {   ?A a rbac:Action ;
2      |   rbac2:subject ?S ;
3      |   rbac2:object ?O .
4      |
5      ?A a rbac:createReview .
6      ?S rbac:activeRole rbac:reviewer_role .
7      ?O a foaf:Document .
8
9      ?S myconference:assigned_to ?O .
10
11     ?AUTHOR rbac:role rbac:author_role .
12     ?AUTHOR myconference:isAuthorOf ?O .
13
14     ?AUTHOR myconference:memberOf ?I1 .
15     ?S myconference:memberOf ?I2 .
16
17     ?I1 log:uri ?URI1 .
18     ?I2 log:uri ?URI2 .
19
20     ?URI1 log:equalTo ?URI2 .
21
22 } => { ?A a rbac:ProhibitedAction } .

```

Quadro 25 – Exemplo de política que define um conflito ao criar uma revisão de um artigo de um autor que trabalhe na mesma instituição que a sua

4. A política ilustrada no Quadro 26 define que o autor só pode visualizar os contextos "AllPublication" e "AllReviews" indicados por suas URIs.

```

1  {   ?A a rbac:Action ;
2      |   rbac:subject ?S ;
3      |   rbac:object ?O .
4      |
5      ?A a rbac:context .
6      ?S rbac:activeRole rbac:author_role .
7      ?O a shdm:Context .
8
9      ?CON a shdm:Context .
10     ?CON log:uri "http://base#41b83760-d2aa-11e0-b3d1-0017fad4bb97" .
11
12     ?CON2 a shdm:Context .
13     ?CON2 log:uri "http://base#00c3c400-d2b8-11e0-a6b6-0017fad4bb97" .
14
15     ?CON log:notEqualTo ?O .
16     ?CON2 log:notEqualTo ?O .
17
18 } => { ?A a rbac:ProhibitedAction } .

```

Quadro 26 – Exemplo de política que define que o autor só pode visualizar os contextos "AllPublication" e "AllReviews"

5. A política ilustrada no Quadro 27 define que o autor só pode visualizar as revisões dos artigos que ele mesmo submeteu, ou seja, quando ele é o autor do artigo.

```

1  {      ?A a rbac:Action ;
2          rbac2:subject ?S ;
3          rbac2:object ?O .
4
5          ?A a rbac:context .
6          ?S rbac:activeRole rbac:author_role .
7          ?O a myconference:Review .
8
9          ?D a foaf:Document .
10         ?O myconference:belongsTo ?D .
11         ?D myconference:hasAuthor ?S .
12
13     } => { ?A a rbac:PermittedAction } .

```

Quadro 27 – Exemplo de política que declara que o autor só pode visualizar as revisões dos artigos que ele mesmo submeteu

6. A política ilustrada no Quadro 28 declara que o revisor não pode revisar um artigo submetido por um autor caso ele, o revisor, seja co-autor de desse autor em algum outro artigo submetido nessa conferência.

```

1  {      ?A a rbac:Action ;
2          rbac2:subject ?S ;
3          rbac2:object ?O .
4
5          ?A a rbac:createReview .
6          ?S rbac:activeRole rbac:reviewer_role .
7          ?O a foaf:Document .
8
9          ?S myconference:assigned_to ?O .
10
11         ?Author rbac:role rbac:author_role .
12         ?Author myconference:isAuthorOf ?O .
13         ?Author myconference:isAuthorOf ?O2 .
14
15         ?O log:uri ?URI1 .
16         ?O2 log:uri ?URI2 .
17
18         ?URI1 log:notEqualTo ?URI2 .
19
20         ?S myconference:isAuthorOf ?O2 .
21
22     } =>{ ?A a rbac:ProhibitedAction } .

```

Quadro 28 – Exemplo de política que define um conflito ao criar uma revisão de um artigo de um autor que já trabalhou junto

7. A política ilustrada no Quadro 29 declara que o revisor só pode visualizar as revisões de um artigo (alocado para ele) feitas por outros revisores caso ele tenha criado antes a sua própria revisão para este artigo.

```

1  {    ?A a rbac:Action ;
2      rbac2:subject ?S ;
3      rbac2:object ?O .
4
5      ?A a rbac:context .
6      ?S rbac:activeRole rbac:reviewer_role .
7      ?O a myconference:Review .
8
9      ### Revisor produziu pelo menos uma
10     ### revisão de um paper associado a ele
11     ?REV a myconference:Review .
12     ?S myconference:produce ?REV .
13     ?REV myconference:belongsTo ?D .
14
15
16     ### Verifica se revisor foi alocado para
17     ### o paper da qual esta review se refere
18     ?D a foaf:Document .
19     ?O myconference:belongsTo ?D .
20     ?S myconference:assigned_to ?D .
21
22 } => { ?A a rbac:PermittedAction } .

```

Quadro 29 – Exemplo de política que define que o revisor só pode visualizar as revisões de artigos que ele já tenha revisado.

8. A política ilustrada no Quadro 30 declara que o revisor sênior pode visualizar as revisões de quaisquer artigos pertencentes ao conjunto de artigos que foram alocados para ele.

```

1  {   ?A a rbac:Action ;
2      rbac2:subject ?S ;
3      rbac2:object ?O .
4
5      ?A a rbac:context .
6      ?S rbac:activeRole rbac:senior_reviewer_role .
7      ?O a myconference:Review .
8
9      ?S myconference:coordinate ?R .
10     ?R rbac:role rbac:reviewer_role .
11
12     ?D a foaf:Document .
13     ?R myconference:assigned_to ?D .
14     ?O myconference:belongsTo ?D .
15
16 } => { ?A a rbac:PermittedAction } .

```

Quadro 30 – Exemplo de política que define que o revisor sênior só pode visualizar as revisões de artigos que foram alocados para ele

9. A política ilustrada no Quadro 31 declara que o revisor só pode editar uma revisão que ele tenha criado.

```

1  {   ?A a rbac:Action ;
2      rbac2:subject ?S ;
3      rbac2:object ?O .
4
5      ?A a rbac:editReview .
6      ?S rbac:role rbac:reviewer_role .
7      ?O a myconference:Review .
8
9      ?S myconference:produce ?O .
10
11 } => { ?A a rbac:PermittedAction } .

```

Quadro 31 – Exemplo de política que define que o revisor só pode editar uma revisão que ele tenha criado

10. A política ilustrada no Quadro 32 define que o autor só pode fazer o *download* de um artigo que ele mesmo submeteu, ou seja, quando ele é o autor do artigo.

```

1  {   ?A a rbac:Action ;
2      rbac2:subject ?S ;
3      rbac2:object ?O .
4
5      ?A a rbac:downloadPaper .
6      ?S rbac:role rbac:author_role .
7      ?O a foaf:Document .
8
9      ?S myconference:isAuthorOf ?O .
10
11 } => { ?A a rbac:PermittedAction } .

```

Quadro 32 – Exemplo de política que define que o autor só pode fazer o *download* de um artigo que ele mesmo submeteu

11. A política ilustrada no Quadro 33 define que o autor só pode visualizar o status da revisão dos artigos que ele submeteu, ou seja, quando ele é o autor do artigo.

```

1  {   ?A a rbac:Action ;
2      rbac2:subject ?S ;
3      rbac2:object ?O .
4
5      ?A a rbac:VisualizeStatusReview .
6      ?S rbac:role role:AuthorRole .
7      ?O a foaf:Document .
8
9      ?S myConference:isAuthorOf ?O .
10
11 } => { ?A a rbac:PermittedAction } .

```

Quadro 33 – Exemplo de política que define que o autor só pode visualizar o status da revisão dos artigos que ele submeteu

6.5. Telas da Aplicação

A seguir serão mostradas algumas capturas de telas da aplicação criada.

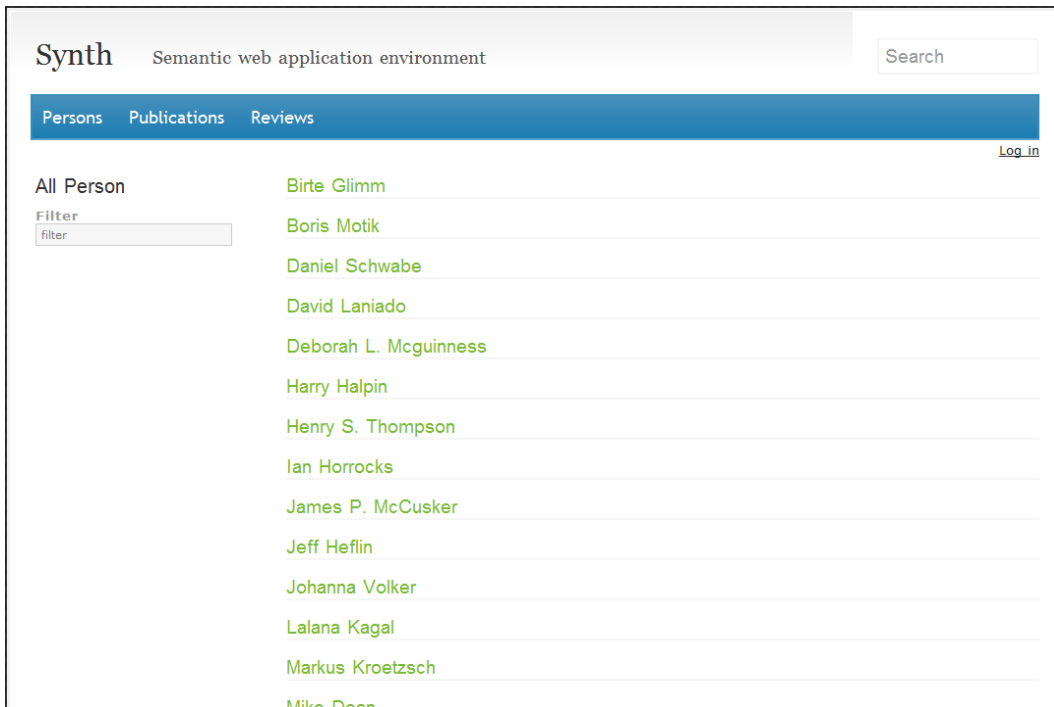


Figura 35 – Tela do índice “AllPerson”

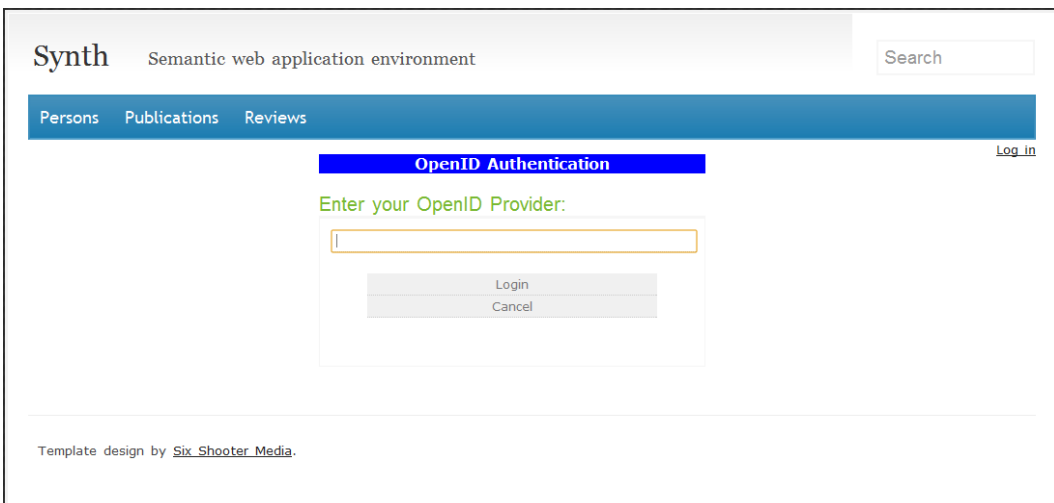


Figura 36 – Tela de *Login* através do protocolo OpenID

Synth Semantic web application environment Search

Persons Publications Reviews Hi Daniel Schwabe (danielschwabetest@email.com)! [Log out](#)

All Persons [Next >>](#)

Filter

<p>Birte Glimm</p> <p>Boris Motik</p> <p>Daniel Schwabe</p> <p>David Laniado</p> <p>Deborah L. McGuinness</p> <p>Harry Halpin</p> <p>Henry S. Thompson</p> <p>Ian Horrocks</p> <p>James P. McCusker</p> <p>Jeff Heflin</p> <p>Johanna Volker</p> <p>Lalana Kagal</p> <p>Markus Kroetzsch</p> <p>Mike Dean</p> <p>Natasha Noy</p> <p>Ora Lassila</p> <p>Patrick J. Hayes</p> <p>Peter F. Patel-Schneider</p>	<p>Birte Glimm</p> <p>rdfs:label Birte Glimm</p> <p>owl:topDataProperty Birte Glimm</p> <p>foaf:familyName Glimm</p> <p>foaf:firstName Birte</p> <p>foaf:mbox birteglimmtest@email.com</p> <p>name Birte Glimm</p> <p>openid http://birteglimmtest.myopenid.com/</p> <p>Institution Oxford University Computing Laboratory</p> <p>Name Johanna Volker</p> <p>Co-Author Markus Kroetzsch Sebastian Rudolph</p> <p>Publication Integrated Metamodeling and Diagnosis in OWL 2 SPARQL Beyond Subgraph Matching</p>
---	---

Figura 37 – Tela de acesso a um nó no contexto “AllPerson”

Synth Semantic web application environment Search

Persons Publications Reviews Hi Daniel Schwabe (danielschwabetest@email.com)! [Log out](#)

All Publications

Filter

- [Integrated Metamodeling and Diagnosis in OWL 2](#)
- [SPARQL Beyond Subgraph Matching](#)
- [Representing and Querying Validity Time in RDF and OWL: A Logic-Based Approach](#)
- [Making sense of Twitter](#)
- [When owl:sameAs isn't the Same: An Analysis of Identity in Linked Data](#)
- [Using Reformulation Trees to Optimize Queries over Distributed Heterogeneous Sources](#)
- [A Self-Policing Policy Language](#)

Template design by [Six Shooter Media](#).

Figura 38 – Tela do índice “AllPublications”

The screenshot shows the 'Add Review' form in the Synth application. The form includes fields for Name, Content, Originality, and Presentation. The Status is set to 'Approved' in a dropdown menu. A 'Create' button is at the bottom of the form. A 'Back' link is located below the form. The page header includes the Synth logo, the text 'Semantic web application environment', a search box, and navigation links for 'Persons', 'Publications', and 'Reviews'. The user is identified as 'Hi Daniel Schwabe (danielschwabetest@email.com)! Log out'.

Figura 39 – Tela de criação de uma nova revisão

The screenshot shows the 'All Publication' view in the Synth application. A red message '*** Access Denied ***' is displayed at the top. The main content area shows a list of publications, with the first one being 'Using Reformulation Trees to Optimize Queries over Distributed Heterogeneous Sources'. The 'Review' section for this publication lists 'ReviewA', 'ReviewB', and 'ReviewC'. The 'Author' section lists 'Jeff Heflin' and 'Yingjie Li'. A 'Create review:' button is visible at the bottom of the publication details. The page header includes the Synth logo, the text 'Semantic web application environment', a search box, and navigation links for 'Persons', 'Publications', and 'Reviews'. The user is identified as 'Hi Daniel Schwabe (danielschwabetest@email.com)! Log out'.

Figura 40 – Tela de acesso a um nó no contexto “AllPublication” mostrando uma mensagem “Access Denied” quando o usuário tentou visualizar uma revisão ao clicar em "ReviewA"