

## 5 Implementação do Modelo de Controle de Acesso no Synth

### 5.1. Synth

O Synth é um ambiente de desenvolvimento que dá suporte à construção de aplicações modeladas segundo o método SHDM, fornecendo um conjunto de módulos capazes de receber como entrada os modelos gerados na execução das etapas do método SHDM e produzir como saída uma aplicação hipermídia executável descrita por estes modelos [Bomfim, 2011].

O Synth é uma evolução do HyperDE (*Hypermedia Development Environment*) que também é um ambiente de desenvolvimento para a construção de aplicações da *web* semântica modeladas segundo os métodos OOHD e SHDM [Nunes, 2005]. O Synth foi implementado sobre o Ruby on Rails<sup>29</sup>, que é um framework para o desenvolvimento de aplicações *web* escrito na linguagem Ruby<sup>30</sup>. O Rails é organizado segundo o padrão de arquitetura MVC (model-view-controller).

O Synth também dispõe de um ambiente de autoria através de uma interface gráfica de formulários para facilitar a criação e edição dos modelos gerados pelo método SHDM. A arquitetura do Synth foi estendida para a inclusão do Módulo de Controle de Acesso.

### 5.2. Arquitetura do Synth

O Synth possui uma arquitetura de software composta por cinco componentes modulares: Módulo de Domínio, Módulo Navegacional, Módulo Comportamental, Módulo de Interface e o Módulo de Persistência. Cada módulo é responsável por manter e interpretar os modelos gerados em casa fase do método SHDM. Estes módulos trabalham em conjunto, interpretando os seus

---

<sup>29</sup> <http://rubyonrails.org/>

<sup>30</sup> <http://www.ruby-lang.org>

modelos e comunicando-se entre si, a fim de gerar o ambiente de execução da aplicação de acordo com as definições de cada modelo [Bomfim, 2011].

O Módulo de Controle de Acesso foi adicionado na arquitetura do Synth e é responsável por gerar as decisões de autorização da aplicação. Ele mantém e interpreta as declarações sobre o Modelo de Controle de Acesso descritas de acordo com a Ontologia de Controle de Acesso, e sobre o Modelo de Políticas representadas de acordo com a Ontologia de Política.

O Interpretador do Modelo Comportamental foi estendido para incluir o gerenciamento da ACL. Portanto, ele passou a verificar as permissões de acesso na ACL toda vez que uma operação for executada. Tal verificação é feita pela pré-condição do modelo de operações, como mostrado no capítulo 4.1.5.

Todas as operações da aplicação são tratadas pelo Módulo Comportamental. O SHDM define dois tipos de operações: as operações internas, que tratam das regras de negócio da aplicação e não podem ser invocadas por agentes externos, e as operações externas, que funcionam como um canal de comunicação entre as entidades externas e a aplicação. A navegação do usuário na aplicação está definida também como uma operação do modelo de operações, como foi visto no capítulo 4.

Neste sentido, o controle de acesso na aplicação é dado sobre uma operação definida pelo modelo de operações. Desta forma, dizemos que o Interpretador do Modelo Comportamental usa o Módulo de Controle de Acesso para saber se uma operação deve ser executada ou não, e que o Interpretador do Modelo de Controle de Acesso consulta o Modelo Comportamental para obter um conjunto de recursos RDF relacionado à operação e aos parâmetros da operação no qual foi acionada. O Interpretador do Modelo de Controle de Acesso também consulta o Modelo de Domínio para obter os dados do domínio da aplicação sobre os quais as políticas de controle de acesso são aplicadas.

A Figura 15 mostra a visão conceitual da arquitetura do Synth estendida para a inclusão do Módulo de Controle de Acesso. As caixas de cor cinza com cantos arredondados representam os módulos enquanto que as caixas brancas representam os componentes de cada módulo. A caixa de cor cinza claro é o Módulo de Controle de Acesso incluído na arquitetura. As setas não rotuladas têm o significado definido pela legenda correspondente.

O Módulo de Persistência trata do acesso e manipulação dos dados da aplicação. A Camada de armazenamento, inferências e consultas do Módulo de Persistência fornece uma interface para o acesso aos vários ambientes e

plataformas de dados RDF, tais como os *frameworks* Sesame<sup>31</sup> e Jena<sup>32</sup>. A Camada de mapeamento RDF(S)/OWL, que também pertence ao Módulo de Persistência, fornece uma visão dos dados e matadados da aplicação na forma de primitivas do ambiente de programação utilizado. O Módulo de Domínio mantém o Modelo de Domínio da aplicação. O Módulo Navegacional é responsável pela geração do ambiente de execução da navegação do usuário na aplicação. O Módulo Comportamental gera as regras de negócio e o controle da interação entre os agentes externos e a aplicação. Por fim, o Módulo de Interface é responsável pela geração das interfaces da aplicação. Mais detalhes sobre os módulos da arquitetura do Synth estão presentes na dissertação de Bomfim (2011).

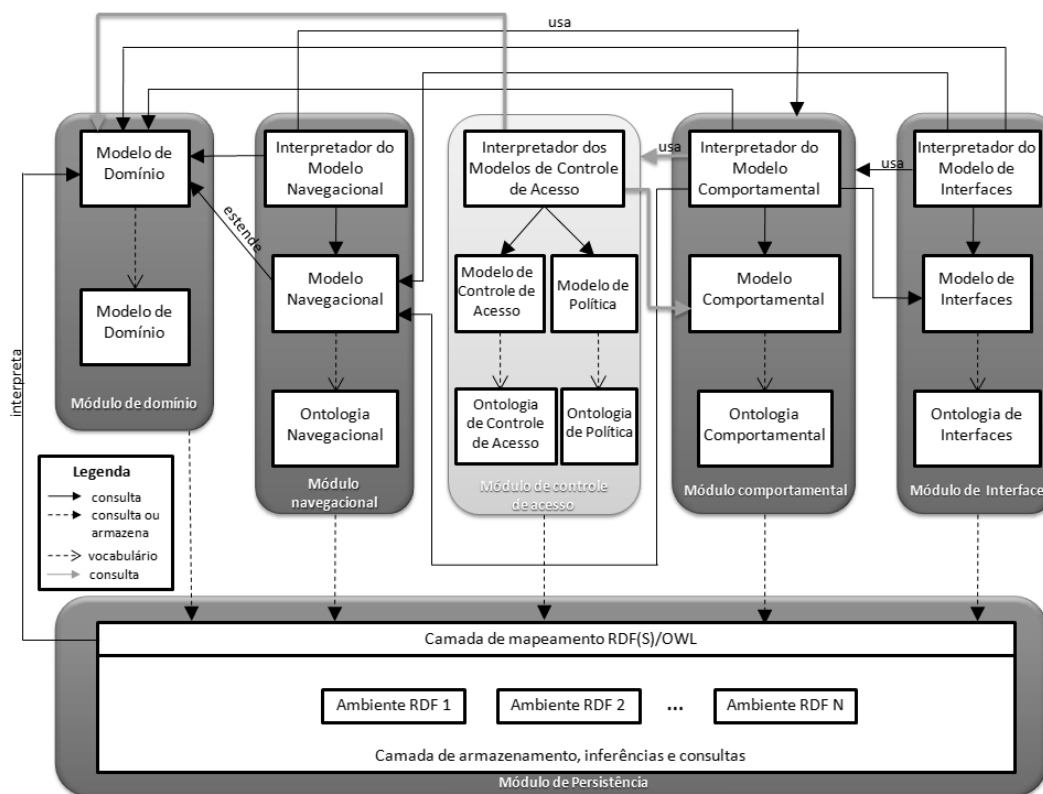


Figura 15 – Visão conceitual da arquitetura do Synth

### 5.3. Sequência de Colaboração entre os Módulos do Synth

A Figura 16 apresenta um diagrama de sequência ilustrando a sequência de colaboração entre os módulos do Synth (após a inclusão do Módulo de

<sup>31</sup> <http://www.openrdf.org>

<sup>32</sup> <http://jena.sourceforge.net>

Controle de Acesso) em um caso normal de execução de uma operação sobre os dados da aplicação. Está sendo considerado que o usuário já está autenticado na aplicação. A seguir segue a descrição dessa sequência de colaboração.

Um usuário solicita a execução de uma operação externa protegida na aplicação, acionando o método “executar” do Módulo Comportamental, e informando o nome da operação externa solicitada e um conjunto de parâmetros. Em seguida o método “obter\_permissão” do Módulo de Controle de Acesso é invocado repassando os mesmos parâmetros informados na chamada anterior. A permissão de acesso é consultada na ACL e depois retornada para o método “executar” do Módulo Comportamental. Caso a permissão de acesso tenha sido negada, a operação não será executada e uma mensagem de erro é mostrada pelo Módulo de Interface. Caso contrário, o método “obter\_contexto” do Módulo Navegacional é invocado e é passado um conjunto de parâmetros de navegação, identificado por “pn”, para a obtenção desse contexto. Em seguida, o método “obter\_dados” do Módulo de Domínio é invocado, passando como parâmetro uma expressão de consulta expressa em uma linguagem específica de domínio (DSL). Essa chamada com os mesmos parâmetros é apenas repassada para a Camada de mapeamento RDF(S)/OWL do Módulo de Persistência. Esta expressão é, portanto, convertida em uma expressão compatível com a base de dados utilizada. Em seguida, a Camada de armazenamento, inferências e consultas do Módulo de Persistência executa a consulta nos repositórios de dados e retorna um conjunto de triplas RDF para a camada anterior. Então a Camada de mapeamento RDF(S)/OWL mapeia as triplas RDF de volta para as primitivas do ambiente de programação. Esses dados mapeados são retornados para o Módulo de Domínio e repassados de volta para o método “obter\_contexto” do Módulo Navegacional. O método “gerar\_contexto” é acionado e a estrutura de dados de um contexto, baseada nos dados mapeados recebidos como parâmetro, é gerada e retornada para o método “obter\_contexto” do Módulo Navegacional. A operação externa recebe o contexto e invoca o método “gera\_interface” do Módulo de Interface que retorna a interface. For fim, a operação externa mostra o resultado final ao usuário.

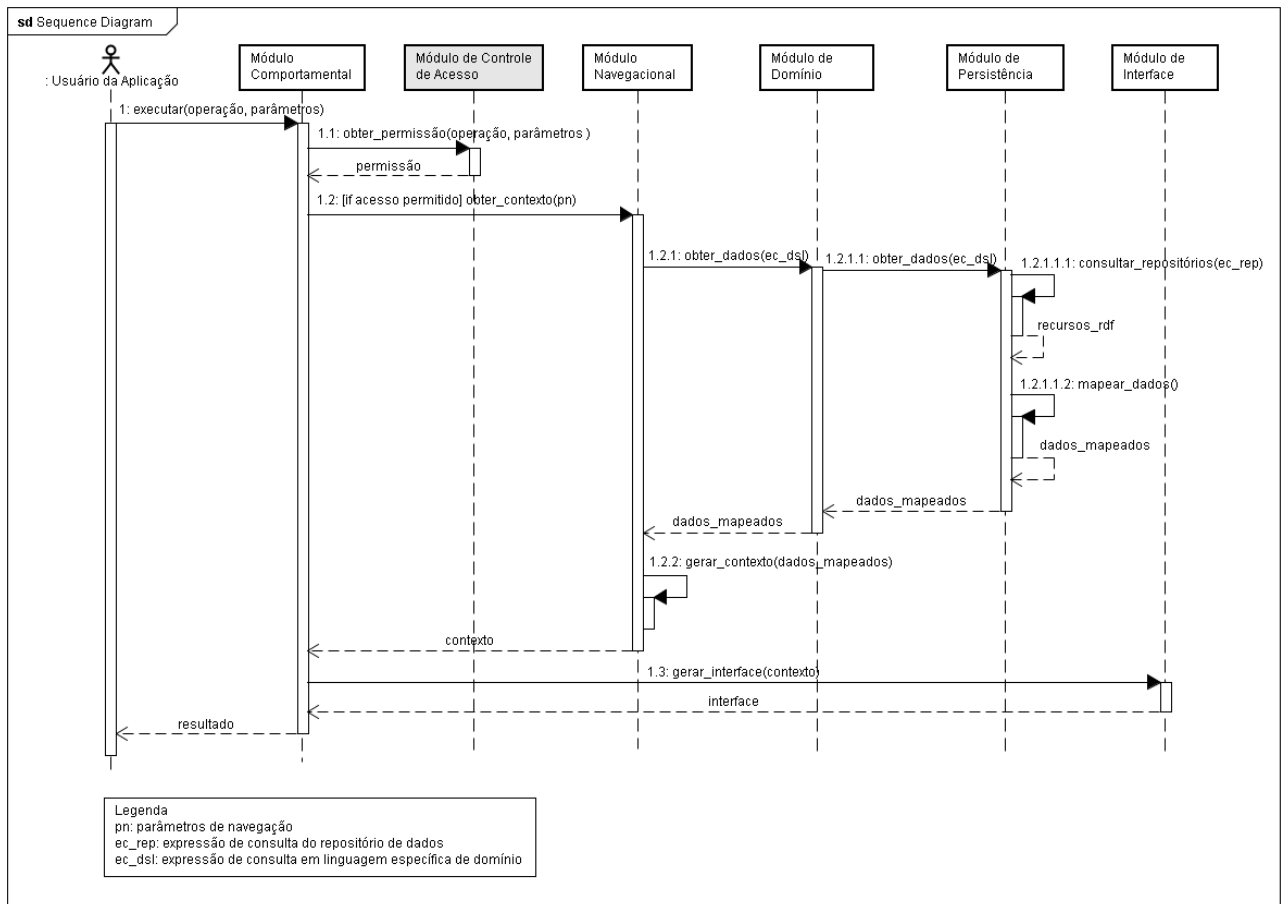


Figura 16 – Colaboração entre os módulos do Synth

#### 5.4. Ambiente de Autoria

O ambiente de autoria do Synth tem o objetivo de facilitar o projetista de aplicações nas tarefas de criação, visualização, remoção e edição das primitivas dos modelos de uma aplicação construída segundo o método SHDM através de uma interface gráfica de formulários que é acessada por meio de um navegador de internet. Através desta interface, é possível também executar a aplicação enquanto ocorre a sua construção, permitindo validá-la a cada passo de seu desenvolvimento [Bomfim, 2011].

A interface gráfica do ambiente de autoria foi organizada em dois níveis de navegação. O primeiro nível é exibido um menu horizontal de itens relacionados a cada etapa do método SHDM. Os itens de menu de primeiro nível são: “Domain” para a modelagem de domínio, “Navigation” para o projeto navegacional, “Interface” para o projeto de interface, e “Behavior” para o projeto comportamental. Existe também o item “Home” que redireciona o usuário para a tela inicial do ambiente de autoria e não está relacionado a nenhuma etapa do

método SHDM. Quando um item de menu horizontal de primeiro nível é clicado, o item de menu vertical de segundo nível é exibido. Eles representam as primitivas que compõem os artefatos que serão manipulados na etapa relacionada ao item de menu clicado.

O nome que aparece depois do rótulo “Application” no cabeçalho padrão de todas as interfaces gráficas do ambiente de autoria refere-se ao nome da aplicação. Uma aplicação Synth é um diretório com o mesmo nome da aplicação dentro do diretório “applications”, na raiz do diretório de instalação do Synth.

Uma nova aba foi adicionada no ambiente de autoria do Synth chamada “Access Control”. Ela está relacionada à etapa de projeto comportamental do método SHDM. No menu vertical de segundo nível desta aba, estão os itens de menu que representam as primitivas do modelo de controle de acesso descrito na seção 3.1 e do modelo de política mostrado na seção 3.2.

A seguir serão apresentadas as tarefas que podem ser executadas no ambiente de autoria do Synth sobre cada primitiva que compõe o modelo de controle de acesso baseado em papel e o modelo de política de autorização de uma aplicação construída segundo o método SHDM.

#### 5.4.1. Sujeitos

A Figura 17 apresenta a interface gráfica do ambiente de autoria para a manipulação dos sujeitos do modelo de controle de acesso baseado em papel.

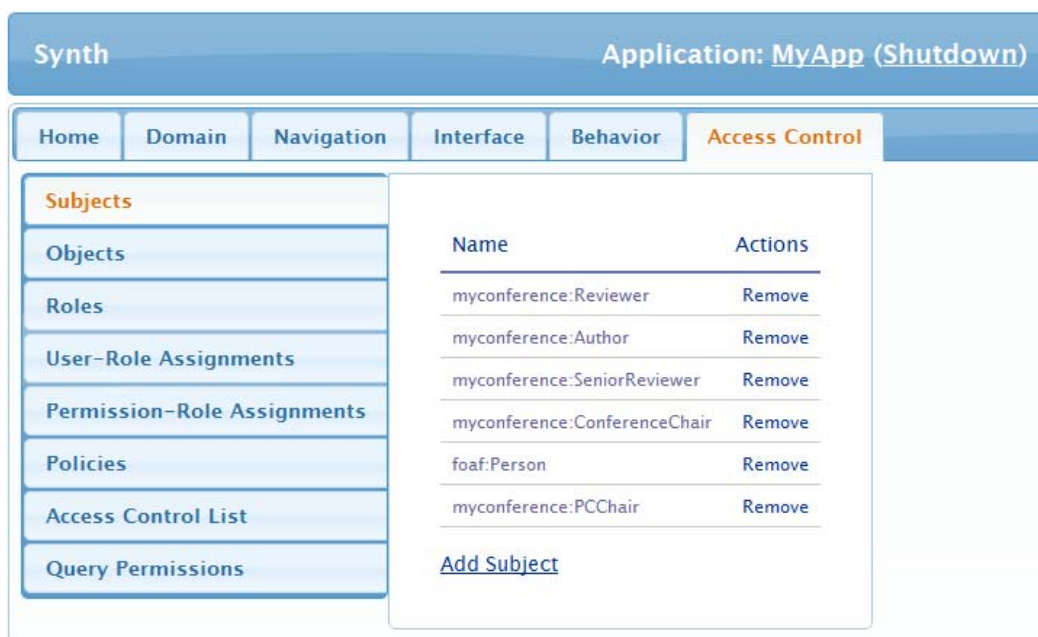


Figura 17 – Tela de listagem de sujeitos

Nela são listados todos os recursos RDF do tipo `rbac:Subject` que foram cadastrados no ambiente de autoria do Synth. Nesta tela, é possível criar ou remover os relacionamentos de subclasse entre uma classe do modelo de domínio e a classe `rbac:Subject`.

Para cadastrar um novo sujeito no modelo de controle de acesso, o projetista deve clicar na âncora “Add Subject” e então a tela de criação de um novo sujeito será mostrada, como ilustra a Figura 18. Nesta tela o projetista deve selecionar uma classe do modelo de domínio e depois clicar no botão “Create”. Vale observar que, a critério do projetista, uma meta classe do Synth pode ser também considerada como um sujeito do modelo de controle de acesso.

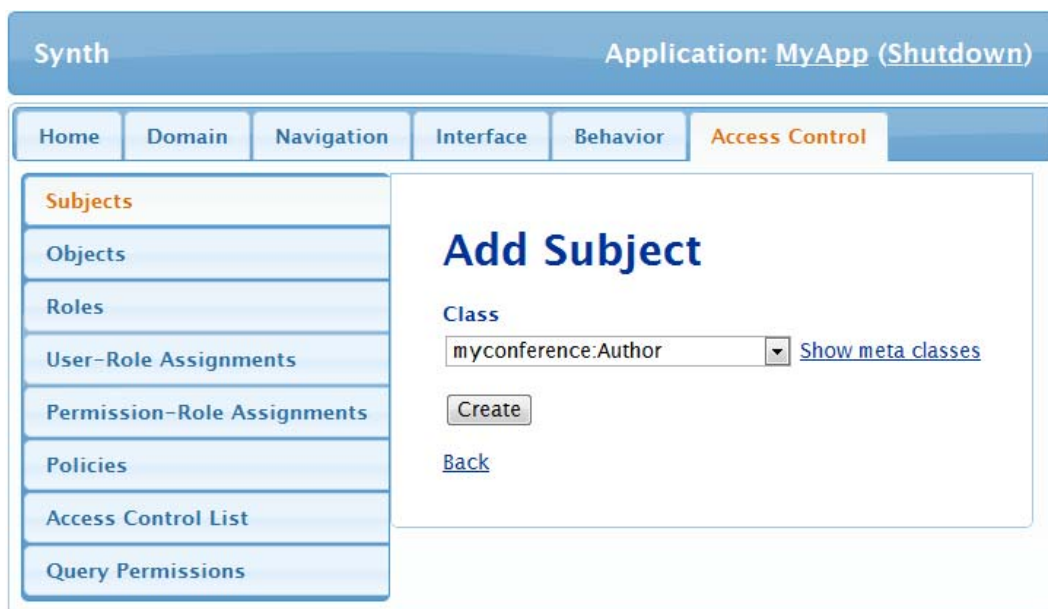


Figura 18 – Tela de criação de um sujeito

#### 5.4.2. Objetos

A Figura 19 ilustra a tela de listagem de objetos previamente cadastrados no ambiente de autoria. Essa tela é mostrada ao clicar no item de menu de segundo nível “Objects”. Nesta tela, todos os recursos RDF do tipo `rbac:Object` são listados, e é possível adicionar ou remover os relacionamentos de subclasse entre uma classe do modelo de domínio e a classe `rbac:Object` do modelo de controle de acesso baseado em papel.

The screenshot shows the Synth application interface for 'Application: MyApp (Shutdown)'. The 'Access Control' tab is active, displaying a list of objects. The left sidebar contains navigation options: Subjects, Objects (highlighted), Roles, User-Role Assignments, Permission-Role Assignments, Policies, Access Control List, and Query Permissions. The main content area shows a table of objects with columns for Name and Actions.

Name	Actions
shdm:Context	Remove
foaf:Document	Remove
myconference:Review	Remove
swc:Paper	Remove
swc:Poster	Remove
swc:SlideSet	Remove

Below the table is a link labeled [Add Object](#).

Figura 19 – Tela de listagem de objetos

Ao clicar na âncora “Add Object”, a tela de cadastro de um novo objeto no modelo é apresentada, conforme ilustrada na Figura 20. Nesta tela, o projetista deve selecionar uma classe do modelo de domínio e depois clicar no botão “Create”. Uma meta classe do Synth pode ser considerada também como um objeto do modelo de controle de acesso, ficando a critério do projetista.

The screenshot shows the Synth application interface for 'Application: MyApp (Shutdown)'. The 'Access Control' tab is active, displaying the 'Add Object' form. The left sidebar contains navigation options: Subjects, Objects (highlighted), Roles, User-Role Assignments, Permission-Role Assignments, Policies, Access Control List, and Query Permissions. The main content area shows the 'Add Object' form with a 'Class' dropdown menu set to 'swc:Paper' and a 'Create' button. There is also a link labeled [Show meta classes](#) and a [Back](#) link.

Figura 20 – Tela de criação de um objeto



### 5.4.3. Papéis e Hierarquia de Papéis

Os papéis no modelo de controle de acesso são representados como instâncias da classe `rbac:Role`, e a propriedade `rbac:subRole` especifica uma relação de herança entre dois papéis e é usada para criar uma hierarquia de papéis. A tela de listagem de papéis e os *superRoles* de cada papel cadastrado previamente no ambiente de autoria é ilustrada na Figura 21. Esta tela é mostrada ao clicar na âncora “Roles”. A partir desta tela é possível criar novos papéis e hierarquia de papéis, editar ou remover os papéis previamente cadastrados.

Name	SuperRole	Actions
<code>rbac:reviewer_role</code>		Edit Remove
<code>rbac:senior_reviewer_role</code>	<code>rbac:reviewer_role</code>	Edit Remove
<code>rbac:pcchair_role</code>	<code>rbac:reviewer_role</code> <code>rbac:senior_reviewer_role</code>	Edit Remove
<code>rbac:conference_chair_role</code>		Edit Remove
<code>rbac:author_role</code>		Edit Remove

[Add Role](#)

Figura 21 – Tela de listagem de papéis

Ao clicar na âncora “Add Role”, a tela de cadastro de um papel é mostrada, conforme apresentada na Figura 22. Nesta tela, deve-se informar o nome do papel e selecionar um “superrole” para este papel. Caso o papel não possua um “superrole”, este campo pode ser deixado em branco. Depois deve-se apertar no botão “Create” para que o papel seja criado.

Figura 22 – Tela de criação de um papel

#### 5.4.4. User-Role Assignments

A Figura 23 mostra a tela de listagem de associações entre o papel e o sujeito, quando a âncora “*User-Role Assignments*” é clicada. Esta primitiva é responsável pela definição dos possíveis papéis que um usuário (recurso da classe `rbac:Subject`) pode ter e é dada pela propriedade `rbac:role`. A partir desta tela é possível adicionar novas associações ou remover as associações previamente cadastradas pelo ambiente de autoria.

Ao clicar na âncora “Add User-Role Assignment”, a tela de criação de uma nova associação entre um papel e o sujeito é mostrada na Figura 24. Nesta tela, deve-se selecionar um sujeito e um papel no qual ele poderá ter e depois clicar no botão “Create” para que a associação seja cadastrada. A definição dos possíveis papéis de cada sujeito é dada através de regras em N3Logic. Portanto, se o projetista tiver selecionado, por exemplo, o sujeito “`myconference:Reviewer`” e o papel “`rbac:author_role`”, então a seguinte regra (Quadro 20) será automaticamente criada:

```
1 { ?X a myconference:Reviewer .
2 } => { ?X rbac:role rbac:author_role } .
```

Quadro 20 – Regra para a definição dos possíveis papéis para os sujeitos que são recursos da classe `myconference:Reviewer`

Esta regra estará visível no item de menu de segundo nível “Policies”, que será apresentado na seção 5.4.6.

Subject	Role	Actions
ConferenceChair	conference_chair_role	Remove
Author	author_role	Remove
PCChair	pcchair_role	Remove
Reviewer	reviewer_role	Remove
Reviewer	author_role	Remove
SeniorReviewer	senior_reviewer_role	Remove

[Add User-Role Assignment](#)

Figura 23 – Tela de listagem de associações entre o papel e o sujeito

**Add User-Role Assignment**

Subject:

Role:

[Back](#)

Figura 24 – Tela de criação de associação entre um papel e o sujeito

#### 5.4.5. Permission-Role Assignments

A Figura 25 mostra a tela de listagem de associações entre o papel e a ação, quando a âncora “Permission-Role Assignments” é clicada. Esta primitiva é responsável pela definição de quais ações (recurso de rbac:Action) serão permitidas para um determinado papel (recurso de rbac:Role). Se a ação possuir

um objeto, ele ser informado. A partir desta tela é possível adicionar novas associações ou remover as associações previamente cadastradas pelo ambiente de autoria.

Role	Action	Object	Actions
rbac:reviewer_role	context	shdm:Context	Remove
rbac:reviewer_role	context	myconference:Review	Remove
rbac:reviewer_role	createReview	foaf:Document	Remove
rbac:senior_reviewer_role	context	myconference:Review	Remove
rbac:pcchair_role	context	myconference:Review	Remove
rbac:conference_chair_role	context	shdm:Context	Remove
rbac:author_role	context	shdm:Context	Remove
rbac:author_role	context	myconference:Review	Remove

[Add Permission-Role Assignments](#)

Figura 25 – Tela de listagem de associações entre o papel com a ação e o objeto

Ao clicar na âncora “Add Permission-Role Assignments”, a tela de criação de uma nova associação entre um papel com uma ação e um objeto é mostrada na Figura 26. Nesta tela, deve-se selecionar um papel, uma ação (que são recursos da classe shdm:Operations) que será permitida para este papel e um objeto desta ação, caso exista. Depois o botão “Create” deve ser clicado para que a associação seja cadastrada pelo ambiente de autoria do Synth.

The screenshot shows the Synth application interface. At the top, it says 'Synth' and 'Application: MyApp (Shutdown)'. Below that is a navigation bar with tabs: Home, Domain, Navigation, Interface, Behavior, and Access Control. The 'Access Control' tab is active. On the left is a sidebar menu with items: Subjects, Objects, Roles, User-Role Assignments, Permission-Role Assignments (highlighted in orange), Policies, Access Control List, and Query Permissions. The main content area is titled 'Add Permission-Role Assignment'. It contains three dropdown menus: 'Role' with 'rbac:reviewer\_role' selected, 'Action' with 'createReview' selected, and 'Object' with 'foaf:Document' selected. Below these is a 'Create' button and a 'Back' link.

Figura 26 – Tela de criação de associação entre um papel com uma ação e um objeto

A definição dos objetos de uma ação é dada através de regras em N3Logic. Portanto, se o projetista tiver selecionado, por exemplo, o papel “rbac:reviewer\_role”, a ação “shdm:createReview” e o objeto “foaf:Document”, então as seguintes alterações (Quadro 21) nos modelos do Synth deverão ser realizadas:

```

1  rbac:createReview a rbac:Action .
2
3  rbac:createReview rbac2:relatedOperation shdm:createReview .
4  shdm:createReview shdm:relatedAction rbac:createReview .
5
6  rbac:reviewer_role rbac:permitted rbac:createReview .
7
8  { rbac:createReview a rbac:Action .
9    ?X a foaf:Document .
10 } => { rbac:createReview rbac2:object ?X } .

```

Quadro 21 – Alterações nos modelos do Synth quando ocorre a associação entre um papel e uma ação

#### 5.4.6. Políticas

Ao clicar no item de menu de segundo nível “Policies”, a tela apresentada na Figura 27 será exibida. Nela é mostrada a listagem das políticas de controle de acesso em forma de regras, ou seja, recursos da classe rule:Rule do modelo de política cujo vocabulário foi apresentado na seção 3.2. A partir desta tela é

possível adicionar novas políticas de autorização, editar ou remover as políticas previamente cadastradas no ambiente de autoria do Synth.

The screenshot shows the Synth application interface with the 'Access Control' tab selected. On the left is a navigation menu with options: Subjects, Objects, Roles, User-Role Assignments, Permission-Role Assignments, **Policies**, Access Control List, and Query Permissions. The main area displays a table of policies with columns for Name and Actions.

Name	Actions
Política - "assigned_to" dos PCChair	Edit Remove
Política - "assigned_to" dos SeniorReviewer	Edit Remove
Política - Conflito "autor de mesmo artigo"	Edit Remove
Política - Conflito "autor que já trabalhou junto"	Edit Remove
Política - Conflito "mesma instituição"	Edit Remove
Política para author_role - 1 - context	Edit Remove
Política para author_role - 2 - visualizeReview	Edit Remove
Política para reviewer_role 1 - createReview	Edit Remove
Política para reviewer_role 2 - context	Edit Remove
Política para reviewer_role 2.1 - context	Edit Remove
Política para reviewer_role 2.2 - context	Edit Remove
Política para reviewer_role 3 - visualizeReview	Edit Remove
Política para senior_reviewer_role 1 - visualizeReview	Edit Remove
Rule: Author with author_role (user-role_assignment)	Edit Remove
Rule: ConferenceChair with conference_chair_role (user-role_assignment)	Edit Remove
Rule: PCChair with pchair_role (user-role_assignment)	Edit Remove
Rule: Reviewer with author_role (user-role_assignment)	Edit Remove
Rule: Reviewer with reviewer_role (user-role_assignment)	Edit Remove
Rule: SeniorReviewer with senior_reviewer_role (user-role_assignment)	Edit Remove
Rule: author_role with context with Context (permission-role_assignment)	Edit Remove
Rule: author_role with context with Review (permission-role_assignment)	Edit Remove
Rule: conference_chair_role with context with Context (permission-role_assignment)	Edit Remove
Rule: pchair_role with context with Review (permission-role_assignment)	Edit Remove
Rule: reviewer_role with context with Context (permission-role_assignment)	Edit Remove
Rule: reviewer_role with context with Review (permission-role_assignment)	Edit Remove
Rule: reviewer_role with createReview with Document (permission-role_assignment)	Edit Remove
Rule: senior_reviewer_role with context with Review (permission-role_assignment)	Edit Remove

[Add Policy](#)

Figura 27 – Tela de listagem de políticas

Na tela de edição de uma política de autorização, apresentada na Figura 28, deve-se informar o nome da política, um título para ela e um código usando a mesma sintaxe da linguagem de regras N3Logic. Os namespaces presentes no código da regra devem estar cadastrados previamente na etapa de modelagem de domínio do método SHDM que é representado pelo item de menu de primeiro nível "Domain". As variáveis que representam recursos do tipo rbac:Action, rbac:Subjeto e rbac:Object devem ser obrigatoriamente representados por "?A", "?S" e "?O", respectivamente.



Synth Application: MyApp (Shutdown)

Home Domain Navigation Interface Behavior **Access Control**

Subjects  
Objects  
Roles  
User-Role Assignments  
Permission-Role Assignments  
**Policies**  
Access Control List  
Query Permissions

## Edit Policy

Name  
Politica - Conflito "mesma instituição"

Title  
Um Reviewer/SeniorReviewer/PCChair não pode revisar um artigo de um autor da mesma instituição

Permission-Role Assignment:

Role	Action	Object
rbac:reviewer_role	rbac:createReview	foaf:Document

OBS: If an "Permission-Role Assignment" is chosen, the rule's consequence must be either "?A a rbac:PermittedAction" or "?A a rbac:ProhibitedAction". An example of access control policy for an action called "visualize\_status\_review" are shown below:

```
{?A a rbac:Action ;
  rbac:subject ?S ;
  rbac:object ?O .

  ?A a rbac:visualize_status_review .
  ?S rbac:activeRole rbac:author_role .
  ?O a foaf:Document .

  ?S myconference:isAuthorOf ?O .

}> { ?A a rbac:PermittedAction } .
```

Code

```
{ ?A a rbac:Action ;
  rbac:subject ?S ;
  rbac:object ?O .

  ?A a rbac:createReview .
  ?S rbac:activeRole rbac:reviewer_role .
  ?O a foaf:Document .

  ?S myconference:assigned_to ?O .

  ?AUTHOR rbac:role rbac:author_role .
  ?AUTHOR myconference:isAuthorOf ?O .

  ?AUTHOR myconference:memberOf ?I1 .
  ?S myconference:memberOf ?I2 .

  ?I1 log:uri ?URI1 .
  ?I2 log:uri ?URI2 .

  ?URI1 log:equalTo ?URI2 .

}> { ?A a rbac:ProhibitedAction } .
```

Save

[Back](#)

Figura 28 – Tela de edição de uma política

Além disso, deve-se também informar o papel, a ação e o objeto de uma associação cadastrada a partir do item de menu de segundo nível "Permission-Role Assignments", apresentado na seção 5.4.5. Estas informações são usadas para identificar o papel, a ação e o objeto sobre os quais a regra será aplicada e serão úteis na tarefa de geração da ACL, que será apresentada na seção 5.4.7. Os campos de "Permission-Role Assignments" da tela de edição de uma política são obrigatórios e não devem ser deixados em branco, a menos que a ação não tenha objetos, então este campo deverá permanecer vazio.

O conseqüente da regra, colocado na área de código, deve obrigatoriamente concluir que a ação é permitida “?A a rbac:PermittedAction” ou que a ação é proibida “?A a rbac:ProhibitedAction”.

Nenhuma informação inferida pelas regras que foram criadas usando esta interface é persistida na base de dados do Synth. As regras são somente usadas com o objetivo de servir como entrada na tarefa de geração da ACL, que será apresentada a seguir na seção 5.4.7.

#### 5.4.7. Gerador de todas as Permissões

A tela de geração da ACL é apresentada quando a âncora “Access Control List” é clicada (Figura 29). Nesta tela deve-se selecionar o tipo de abordagem usada para a escolha da permissão quando ocorre a situação de uma ação (recurso da classe rbac:Action) ser considerada, ao mesmo tempo, tanto permitida (rbac:PermittedAction) quanto proibida (rbac:ProhibitedAction). Por conta disso, na etapa da geração das ACLs deve-se selecionar uma de duas abordagens pré-definidas: a abordagem conservadora e a abordagem liberal. A abordagem liberal dá preferência à proibição da ação (isto é, tudo é permitido a menos que declare o contrário), enquanto que a abordagem conservadora dá preferência à permissão da ação (isto é, tudo é proibido a menos que declare o contrário).

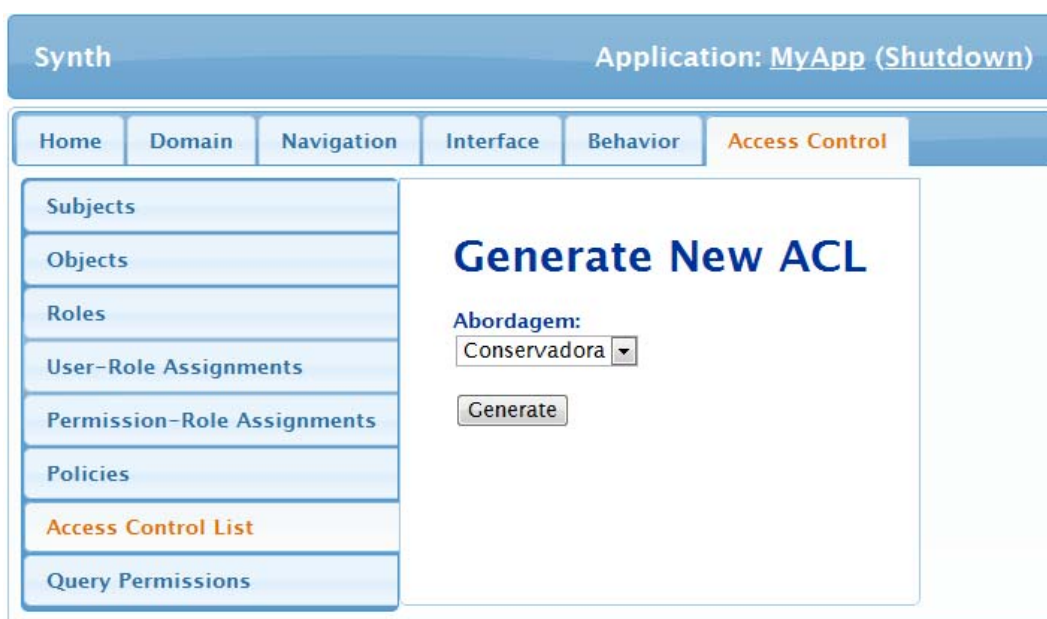


Figura 29 – Tela da geração da ACL



Para melhor entender este caso, suponha que, por exemplo, um revisor de artigos pode ser atribuído tanto ao papel de revisor quando ao papel de autor. Suponha também que exista uma política que diga que o autor não pode visualizar o contexto navegacional que liste os artigos que um revisor pode revisar, e que exista outra política que diga que os revisores podem acessar todos os contextos navegacionais da aplicação. Neste caso, quando um revisor estiver autenticado na aplicação, todos os papéis atribuídos a ele serão ativados. Sendo assim, quando ele tentar navegar para o contexto descrito acima cuja semântica de navegação entre contextos é dada pela operação `shdm:context`, tal ação será tanto `rbac:PermittedAction` quanto `rbac:ProhibitedAction` ao mesmo tempo. Desta forma, as abordagens conservadora e liberal são usadas para decidir qual permissão deve ser atribuída a ação.

A geração da lista de controle de acesso ocorre em quatro passos descritos a seguir:

1. É gerada automaticamente uma série de possíveis perguntas de autorização se um usuário tem a permissão de executar uma operação sobre um determinado objeto ao ativar todos os possíveis papéis atribuídos a este usuário. As perguntas de autorização foram apresentadas na seção 3.1.9.1;
2. Regras em N3Logic para gerar os recursos RDF descritos segundo o vocabulário da ontologia de ACL da W3C são criadas. O vocabulário da ontologia de ACL foi apresentado na seção 3.3.
3. A máquina de inferência *Euler proof engine*<sup>33</sup> em Java é executada tendo como entrada todos os dados da aplicação e todas as regras em N3Logic definidas no ambiente de autoria. É usada a opção “--query” da máquina de inferência para que as informações inferidas sejam filtradas pelas regras criadas no passo 2. Desta forma, somente as informações deduzidas pela execução destas regras serão preservadas.
4. As permissões retornadas pela máquina de inferência contendo os recursos RDF do modelo de ACL são importadas para a base do Synth.

A seguir será descrito como as regras que geram as ACLs (passo 2) são formadas.

---

<sup>33</sup> <http://eulerssharp.sourceforge.net/>

### 5.4.7.1. Definição das Regras para Gerar as ACLs

Para a geração das permissões usando a abordagem conservadora, as regras que geram as ACLs têm a estrutura semelhante ao exemplo mostrado no Quadro 22. Para cada associação “Permission-Role Assignments” definida no ambiente de autoria, uma regra seguindo essa estrutura é formada.

A sentença “<Access\_Control/ResultadoRegras.n3> log:semantics F” na linha 2 usa a propriedade lógica “log:semantics” para associar as triplas inferidas, que estão no arquivo “Access\_control/ResultadoRegras.n3”, com a variável “F”. Esse arquivo contém as triplas que são resultantes da execução da máquina de inferência (ver passo 3 da seção 5.4.7). Os arquivos temporários utilizados pela Máquina de inferência são armazenados dentro do diretório “Access Control” que está na raiz do diretório de instalação do Synth. A definição mais detalhada da propriedade “log:semantics” pode ser obtida em [Berners-lee et al., 2008].

```

1  @forall F, S, O, A, RACTION .
2  { <Access_Control/ResultadoRegras.n3> log:semantics F .
3
4      F log:includes {
5          A a rbac:Action ;
6              rbac2:subject S ;
7              rbac2:object O .
8
9          A a rbac:createReview .
10         S rbac:activeRole rbac:reviewer_role .
11         O a foaf:Document .
12
13         A a rbac:PermittedActionRBAC .
14         A rbac:action RACTION .
15
16         A a rbac:PermittedAction .
17     } .
18
19 } => { [ a acl:Authorization ;
20         acl:mode RACTION ;
21         acl:agent S ;
22         acl:accessTo O ] } .

```

Quadro 22 – Regra de geração dos recursos da ACL

As linhas 9, 10 e 11 possuem as declarações da ação, papel e o objeto da ação, respectivamente. Estes dados são obtidos através das propriedades rule:rule\_action, rule:rule\_role e rule\_object da classe rule:Rule do modelo de políticas (ver seção 3.2)

A sentença da linha 13 declara que a ação deve ser permitida pelo modelo de controle de acesso baseado em papel, que foi apresentado na seção 3.1, mudando a sintaxe para `rbac:PermittedActionRBAC` no lugar de `rbac:PermittedAction`. A sentença da linha 14 identifica a ação (recurso de `rbac:Action`) que será usada pelo modelo ACL na linha 20.

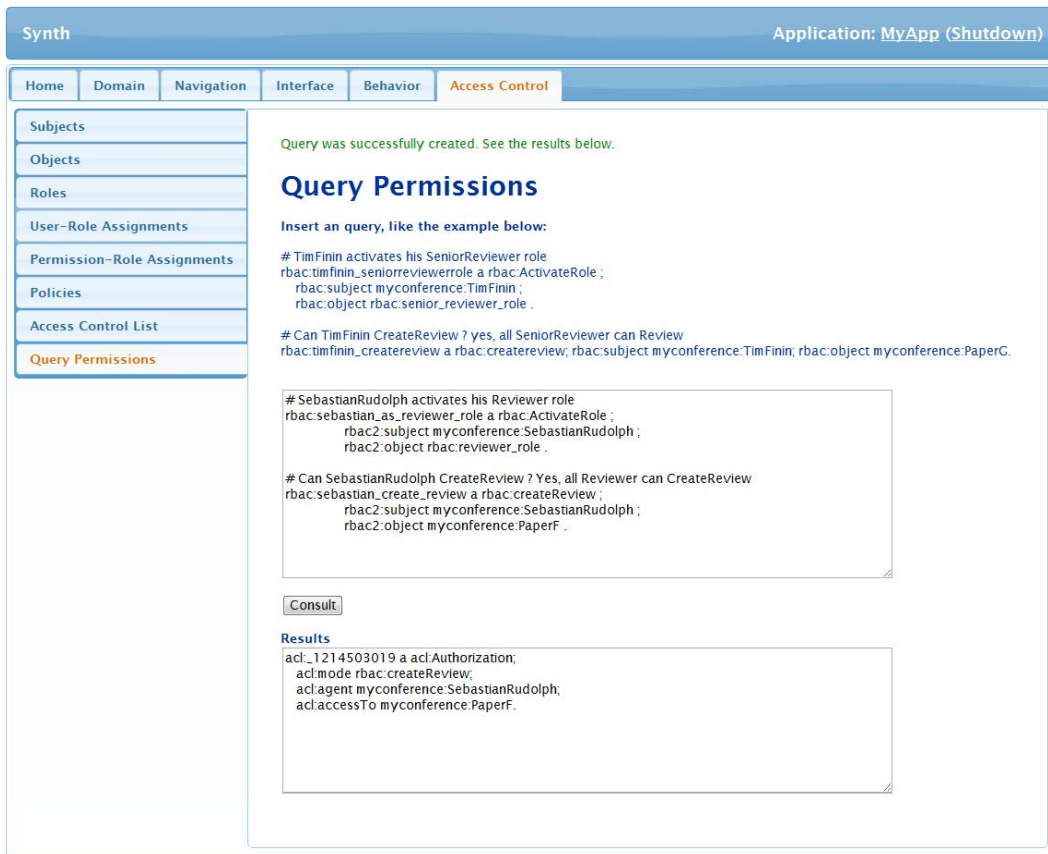
Se a associação “Permission-Role Assignments” possuir uma política em forma de regra, definida no ambiente de autoria (ver seção 5.4.6), que conclua “?A a `rbac:PermittedAction`”, então a sentença da linha 16 é adicionada. Caso contrário, ou seja, o conseqüente da regra concluiu “?A a `rbac:ProhibitedAction`”, então essa sentença é omitida.

Por outro lado, para a geração das permissões usando a abordagem liberal, as regras que geram as ACLs também têm a mesma estrutura do exemplo mostrado no Quadro 22, porém no lugar da sentença da linha 16 é colocado a sentença “A a `rbac:ProhibitedAction` .”. Ou seja, se uma ação é `rbac:ProhibitedAction`, então uma entrada na ACL é criada. É importante notar que quando a abordagem usada é a conservadora, as entradas da ACL definem permissões, enquanto que na abordagem liberal, as entradas na ACL concluem proibições.

#### **5.4.8. Consulta de Permissões**

Em muitos casos, é útil para o projetista testar políticas através da avaliação das regras, sobretudo para entender a interação entre várias regras. Para auxiliar nesta tarefa, foi incluída no ambiente uma interface que permite a consulta a permissões.

Ao clicar no item de menu de segundo nível “Query Permissions”, a tela de consulta de permissões é mostrada na Figura 30. Nesta tela é possível fazer consultas de permissão ao informar uma pergunta de autorização. As perguntas de autorização foram apresentadas na seção 3.1.9.1. O resultado da consulta são recursos da classe `acl:Authorization`. Caso a consulta não retorne resultados, a mensagem “No results” é mostrada.



Synth Application: MyApp (Shutdown)

Home Domain Navigation Interface Behavior **Access Control**

Subjects  
Objects  
Roles  
User-Role Assignments  
Permission-Role Assignments  
Policies  
Access Control List  
**Query Permissions**

Query was successfully created. See the results below.

### Query Permissions

Insert an query, like the example below:

```
# TimFinin activates his SeniorReviewer role
rbac:timfinin_seniorreviewerrole a rbac:ActivateRole ;
rbac:subject myconference:TimFinin ;
rbac:object rbac:senior_reviewer_role .

# Can TimFinin CreateReview ? yes, all SeniorReviewer can Review
rbac:timfinin_createreview a rbac:createreview; rbac:subject myconference:TimFinin; rbac:object myconference:PaperG.
```

**Results**

```
acl: 1214503019 a acl:Authorization;
acl:mode rbac:createReview;
acl:agent myconference:SebastianRudolph;
acl:accessTo myconference:PaperF.
```

Figura 30 – Tela de consulta de permissões

## 5.5. Pré-condição da Operação

Na definição de Operações no Synth é prevista a especificação de pré-condições. O controle de acesso em tempo de execução tira partido desta funcionalidade definindo automaticamente duas pré-condições. Estas pré-condições são declaradas sempre que uma operação é definida como protegida pelo modelo de controle de acesso, conforme definido na seção 4.1.5. A Figura 31 mostra as duas pré-condições criadas para uma operação protegida.

Pre Conditions		
Name	Expression	Failure handling
authentication	isAuthenticated	redirectToLogin
authorization	isAuthorized	showErrorMessage

Page 1 of 1 10 View 1 - 2 of 2

Figura 31 – Tela de edição da pré-condição de uma operação

A expressão “isAuthenticated” é uma função que verifica se o usuário está autenticado na aplicação. Caso não esteja, o código de tratamento de falhas redireciona o usuário para a tela de *login*. A expressão “isAuthorized” é outra função que verifica na lista de controle de acesso (ACL) se o usuário tem permissão de executar a operação. Caso retorne falso, o código de tratamento de falhas mostra a seguinte mensagem de erro “Access Denied”.

## 5.6. Custo Computacional

Algumas simulações de acesso foram realizadas para avaliar o custo computacional do controle de acesso no implementado Synth.

A Figura 32 mostra o tempo de consultar 100 vezes as permissões na ACL ao tentar acessar um conjunto de nós de 10 contextos diferentes de forma aleatória. A classe `acl:Authorization` do modelo de ACL tinha 217 recursos.

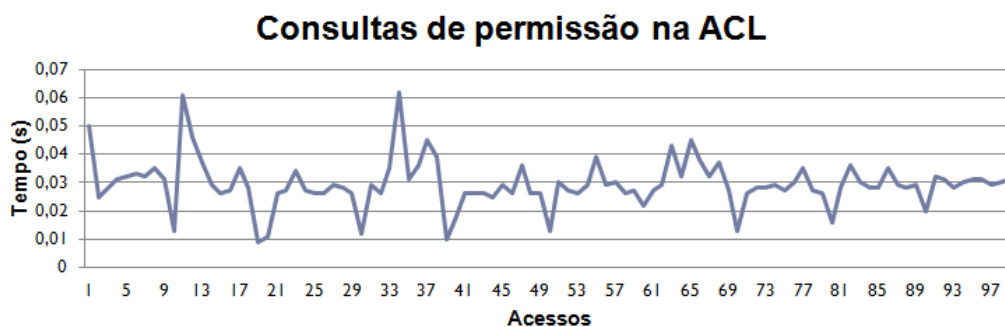


Figura 32 – Tempo para fazer consultas de permissão na ACL

A Figura 33 mostra o tempo de executar (avaliar) todas as regras definidas pela aplicação de exemplo, mostrada na seção 6, 50 vezes ao adicionar um número constante de recursos na base de dados em cada execução. Havia um total de 13 regras de controle de acesso mais aquelas regras definidas pelo modelo RBAC e as regras de geração das ACLs.

É importante perceber que quando um recurso é adicionado na base, aumenta as possibilidades de acesso. No entanto, como todas as regras estão sendo executadas, nem todas geram o resultado desejado. Uma estratégia para executar somente as regras afetadas por uma mudança na base poderia ser feita como possível trabalho futuro.

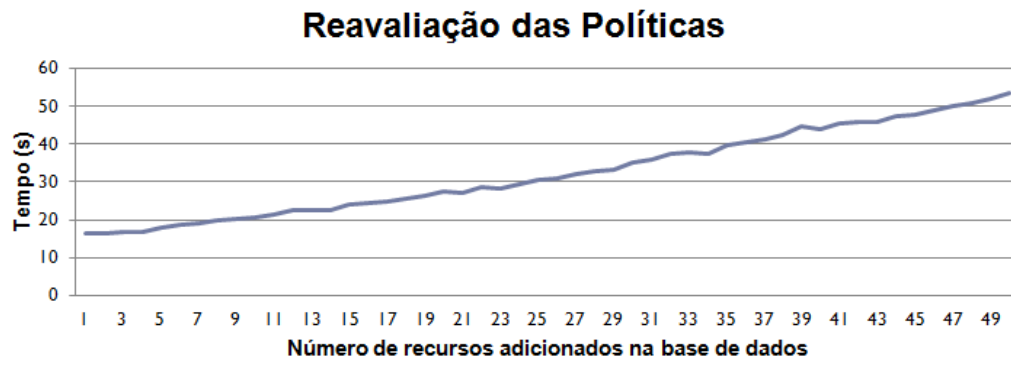


Figura 33 – Tempo de reavaliar todas as políticas quando ocorre uma mudança na base