



UNIVERSIDADE FEDERAL DO ESTADO DO RIO DE JANEIRO
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA

Relatórios Técnicos
do Departamento de Informática Aplicada
da UNIRIO
nº 0005/2010

Estudos de Registro de Serviços e do Oracle Service Registry

Leonardo Azevedo
Flávio Faria
Flávia Santoro
Fernanda Baião

Departamento de Informática Aplicada

UNIVERSIDADE FEDERAL DO ESTADO DO RIO DE JANEIRO
Av. Pasteur, 458, Urca - CEP 22290-240
RIO DE JANEIRO – BRASIL

Projeto de Pesquisa

Grupo de Pesquisa Participante



Patrocínio



PETROBRAS

Estudos de Registro de Serviços e do Oracle Service Registry*

Leonardo Azevedo, Flávio Faria, Flávia Santoro, Fernanda Baião

Núcleo de Pesquisa e Prática em Tecnologia (NP2Tec)
Departamento de Informática Aplicada (DIA) – Universidade Federal do Estado do Rio de Janeiro (UNIRIO)

{azevedo, flavio.faria, flavia.santoro, fernanda.baiao}@uniriotec.br

Abstract. SOA is a paradigm that involves in a broader way than technology, such as, strategies and architectural model definitions, behavioral and cultural issues, indicators specifications to asses ROI etc. However, SOA deployment depends on technologies. Among the main technologies, service registry or UDDI (Universal Description, Discovery and Integration – the published pattern) is one of them as well as ESB (Enterprise Service Bus) and tools for service composition. This work presents a study of service registry concepts and the analysis of OSB (Oracle Service Bus), a UDDI implementation.

Keywords: SOA, Service Registry, UDDI, Oracle Service Registry.

Resumo. SOA é um paradigma que envolve muito mais questões do que apenas tecnologia, tais como, definições de estratégias e modelos arquiteturais, tratar questões de comportamento e cultura, especificar indicadores para medir ganhos etc. No entanto, a implantação de uma abordagem SOA depende da definição de tecnologias de apoio. Dentre estas tecnologias, o registro de serviços ou UDDI (Universal Description, Discovery and Integration - considerando o padrão), apresenta-se como uma tecnologia intrínseca para implantação de SOA, juntamente com o ESB (Enterprise Service Bus) e ferramentas de composição de serviços. Este trabalho apresenta um estudo dos conceitos de registro de serviços e a análise do Oracle Service Registry, uma implementação UDDI.

Palavras-chave: SOA, Registro de Serviços, UDDI, Oracle Service Registry.

* Trabalho patrocinado pela Petrobras.

Sumário

1	Introdução	6
2	Especificação UDDI v3	8
2.1	Nó e Registro UDDI	10
2.2	Representação de informações em um UDDI	11
2.3	Segurança no UDDI	14
2.4	Usando WSDL em um registro UDDI	15
2.5	UDDI Schemas	18
3	Oracle Service Registry	18
3.1	Instalação	20
3.2	Fluxo de aprovação	22
3.3	Subscrição e notificação	22
3.4	Classificação de serviços usando taxonomias	23
3.5	Segurança	23
3.6	Processo de aprovação	24
3.7	Consoles do OSR	25
3.8	Controle de acesso	47
3.9	Taxonomia	48
3.10	Mapeamento de recursos	49
3.11	Desenvolvimento do lado do cliente e do servidor	54
3.12	Integração do OSR com OSB	55
4	Conclusões	56
5	Referências	56
	Apêndice 1 – Roteiro de Instalação do Oracle Service Registry versão 10.3	58
	Apêndice 2 – Roteiro para gerenciamento de contas de usuários no Oracle Service Registry versão 10.3	67
	Apêndice 3 – Roteiro para gerenciamento de grupo de usuários no Oracle Service Registry versão 10.3	69
	Apêndice 4 – Roteiro para gerenciamento de permissões de usuários e grupo de usuários no Oracle Service Registry versão 10.3	71
	Apêndice 5 – Roteiro para publicação de um provedor de serviço no Oracle Service Registry versão 10.3	73
	Apêndice 6 – Roteiro para publicação de um serviço de negócio no Oracle Service Registry versão 10.3	75
	Apêndice 7 – Roteiro para controlar o acesso a um serviço de negócio no Oracle Service Registry versão 10.3	78
	Apêndice 8 – Roteiro para criar taxonomias no Oracle Service Registry versão 10.3	81

Apêndice 9 – Roteiro para associar uma categoria a um serviço de negócio no Oracle Service Registry versão 10.3	85
Apêndice 10 – Roteiro para associar um serviço de negócio a um provedor de negócio no Oracle Service Registry versão 10.3	86
Apêndice 11 – Roteiro para criar um <i>binding template</i> no Oracle Service Registry versão 10.3	88
Apêndice 12 – Utilizando taxonomia para pesquisa de serviços no Oracle Service Registry versão 10.3	89

1 Introdução

Um registro de serviços ou UDDI (Universal Description, Discovery and Integration), assim como o barramento de serviços (ESB), é uma tecnologia chave para a implantação de uma Arquitetura Orientada a Serviços (Service-Oriented Architecture). As especificações UDDI formam a fundamentação técnica necessária para publicação e descoberta de implementações de web services tanto dentro de uma empresa como entre empresas [Bellwood *et al.*, 2004].

O foco do UDDI é a definição de um conjunto de serviços suportando descrição e descoberta de:

- Negócios, organização e outros provedores de serviços;
- Web services que provedores disponibilizam;
- Interfaces técnicas que podem ser usadas para acessar estes serviços;

O registro UDDI descreve:

- Formato de dados para classificação e organização de negócios e corporações;
- Formato de dados para descrição e categorização de serviços de negócio (*web services*);
- Práticas para organização e compartilhamento de informações por *brokers* de serviços;
- Uma API-baseada em XML para interagir com *brokers* de serviços para publicar ou localizar *web services*;
- Um mecanismo para registrar serviços que é independente de plataforma.

O registro UDDI é semelhante a um catálogo de páginas amarelas, onde os serviços listados não existem dentro do livro, mas sim informações sobre os serviços (metadados).

O UDDI atua como um índice de registros (Figura 1) para serviços onde:

- Provedores de serviços (provedor) mantêm informações sobre seus serviços no registro.
- Requisitantes de serviços (consumidor) procuram nos registros por serviços.
- Uma vez encontrado, um serviço pode ser invocado através de uma comunicação ponto-a-ponto ou utilizando um intermediário.

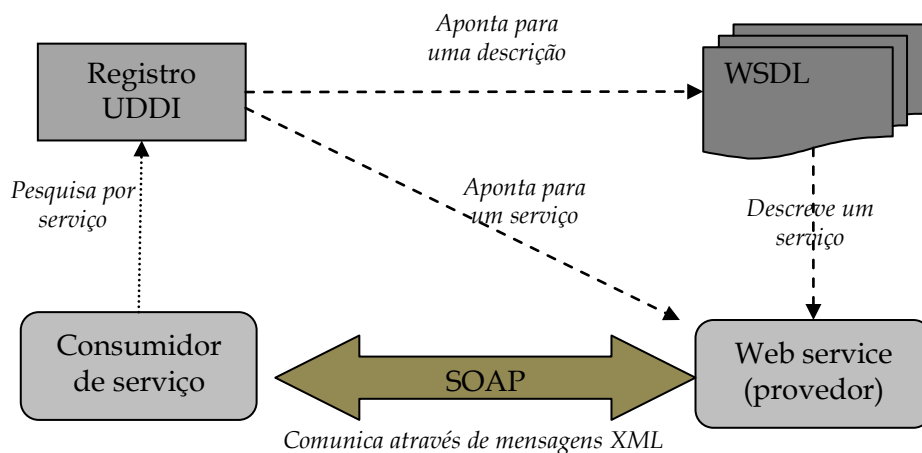


Figura 1 – Interações de web services

O registro UDDI pode ajudar a tratar questões de reuso de código, manutenção e documentação e provê os seguintes benefícios:

- **Visibilidade:** Ele provê visibilidade pela identificação de quais serviços dentro da organização podem ser reutilizados para tratar uma necessidade do negócio.
- **Reuso:** Ele promove reuso e previne recriar o mesmo serviço. Ele acelera tempo de desenvolvimento e aumenta produtividade. A habilidade do UDDI de categorizar um portfólio de serviços crescente facilita gerenciá-los.
- **Facilidades para configuração e adaptação:** Ele facilita configuração e adaptação pelo uso do princípio arquitetural orientado a serviço de independência de localização e transporte. Usuários podem dinamicamente descobrir serviços armazenados no registro UDDI.
- **Relacionamentos:** Ele ajuda a entender relacionamentos entre serviços, versões de componentes e dependências.
- **Ele facilita a gestão do ciclo de vida de serviços de negócio.** Por exemplo, o processo de mover serviços através de cada fase de desenvolvimento do código para a instalação pública.

O registro é frequentemente utilizado pelas organizações em três propósitos específicos:

- **Fase de projeto: Catálogo de serviço**
Arquitetos, desenvolvedores e administradores usam o registro para publicação e descoberta de serviços que estão disponíveis na organização para reuso. Isto é efetivo em situações onde existem muitas equipes de desenvolvimento que necessitam compartilhar serviços através dos limites departamentais.
- **Governança de TI**
Pelo uso do registro, organizações podem alcançar visibilidade dos seus serviços e garantir políticas de TI sobre os mesmos. Isto garante consistência arquitetural e visibilidade através da organização.
- **Resolução de serviço em tempo de execução**

Neste cenário, o registro é utilizado para permitir arquitetura com baixo acoplamento. Por adiar a resolução do *endpoint* do serviço até o tempo de execução, aplicações consumidoras podem garantir que elas estão conectadas com o provedor do serviço mais apropriado, ou seja, o provedor que está mais de acordo com critérios, tais como, carga do sistema atual, requisitos de QoS (*Quality of Service*) do cliente e contexto de segurança. Este tipo de abordagem melhora a resiliência e tratamento de falhas em implementações SOA.

Baseado em um conjunto de padrões comuns da indústria, incluindo HTTP, XML, XML Schema e SOAP, o UDDI provê uma infraestrutur interoperável e fundacional para ambientes de software baseados em web services. O padrão UDDI está na sua terceira versão. UDDI usa a linguagem XML Schema¹ e sua terminologia, tais como “sequence” e “choice” para formalmente descrever suas estruturas de dados. O UDDI é disponibilizado, ele mesmo, como um conjunto de web services. O comitê técnico da especificação de UDDI da OASIS de tempos em tempos publica “melhores práticas” e “notas técnicas” a respeito do padrão, os quais podem ser obtidos a partir de [OASIS, 2009a] e [OASIS, 2009b], respectivamente.

O objetivo deste relatório é apresentar as principais características do UDDI e apresentar um estudo do OSR (Oracle Service Registry). OSR é responsável pela gerência de metadados úteis durante o *runtime* de serviços para comunicação com um dado serviço. Descrições de serviços, localização e *endpoint*, operações e parâmetros de serviços, e políticas de serviços são exemplos de informação de *runtime* que o registro gerencia.

Este relatório foi produzido pelo Projeto de Pesquisa em SOA como parte das iniciativas dentro do contexto do Projeto de Pesquisa do Termo de Cooperação entre NP2Tec/UNIRIO e a Petrobras/TIC-E&P/GDIEP.

Esse relatório está organizado em 5 capítulos, sendo o capítulo 1 a presente introdução. No capítulo 2, são apresentadas as principais características de um UDDI. No capítulo 3, é apresentada, em detalhes, a arquitetura do Oracle Service Registry (OSR), que é a implementação UDDI da Oracle. Nos capítulos 4 e 5, são apresentadas as conclusões do trabalho e as referências bibliográficas, respectivamente.

2 Especificação UDDI v3

A especificação UDDI [Bellwood *et al.*, 2009] apresenta um modelo de informações composto por instâncias de estruturas de dados persistentes chamadas entidades. Entidades são descritas em XML e são armazenadas de forma persistente em nós UDDI. Cada entidade tem o tipo do seu elemento XML mais global. Um modelo de informações UDDI é composto de instâncias dos seguintes tipos de entidades (Figura 2):

- Entidade do negócio (*businessEntity*): descreve um negócio ou outra organização que provê *web services*. Cada serviço deve ser publicado dentro de um *businessEntity*.

¹ URLs para especificação da linguagem XML Schema: <http://www.w3.org/TR/xmlschema-0/>, <http://www.w3.org/TR/xmlschema-1/>, e <http://www.w3.org/TR/xmlschema-2/>.

- Serviço de negócio (*businessService*): descreve uma coleção de web services relacionados que são oferecidos por um organização descrita pela entidade do negócio (*businessEntity*).
- Template de ligação (*bindingTemplate*): descreve a informação técnica para usar um determinado web service.
- tModel: descreve um modelo técnico (“technical model”), representando um conceito, por exemplo, um tipo de web service, um protocolo usado por web services, ou um sistema de categorias.
- Assertiva de publicador (*publisherAssertion*): descreve, na visão de um *businessEntity*, o relacionamento que o *businessEntity* tem com outro *businessEntity*.
- Subscrição (*subscription*): descreve uma requisição para manter-se informado de mudanças das entidades descritas pela *subscription*. Qualquer usuário do registro pode se inscrever a um conjunto de entidades de UDDI e monitorar sua criação, modificação ou remoção. A subscrição é definida usando chamadas da API padrão *get* ou *find*. O registro UDDI notifica o usuário sempre que qualquer entidade que está de acordo com a consulta de subscrição ocorre, mesmo que as mudanças da entidade façam com que ela não atenda mais a consulta. Ele também notifica a respeito de entidades que foram alteradas de uma forma que depois da mudança fizeram com que elas estivessem de acordo com a consulta de subscrição.

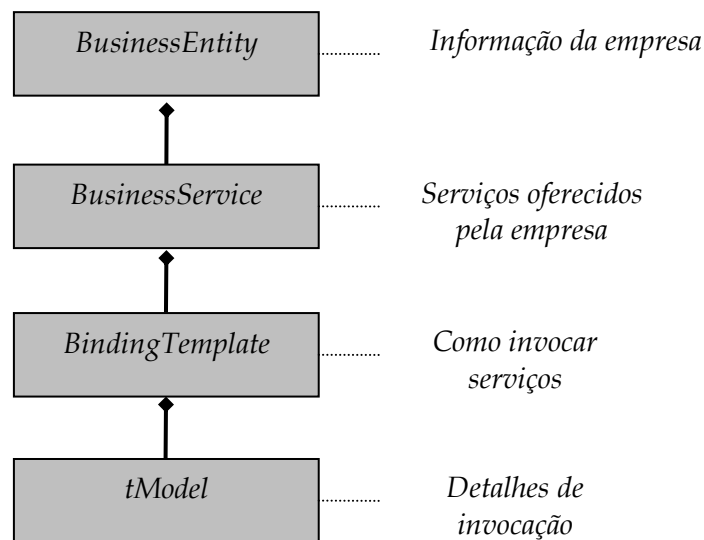


Figura 2 – Esquema do modelo de dados do UDDI

A especificação apresenta APIs que padronizam o comportamento e a comunicação com e entre implementações de UDDI com o objetivo de manipular dados de UDDI armazenados dentro destas implementações. As APIs são agrupadas nos seguintes conjuntos:

- Conjunto de APIs de nós: UDDI Inquiry; UDDI Publication; UDDI Security; UDDI Custody Transfer; UDDI Subscription; UDDI Replication.
- APIs de cliente: UDDI Subscription Listener, UDDI Value Set.

2.1 Nó e Registro UDDI

Um nó UDDI corresponde a um conjunto de web services que suporta pelo menos um conjunto de API de nó. Um nó UDDI tem as características definidoras:

- Um nó UDDI suporta interação com dados UDDI através de uma ou mais API UDDI;
- Um nó UDDI é um membro de exatamente um registro UDDI;
- Um nó UDDI conceitualmente tem acesso e manipula uma cópia lógica completa dos dados UDDI geridos pelo registro do qual ele faz parte. Além disso, estes são os dados que são manipulados por qualquer API de consulta ou de publicação suportada pelo nó. Tipicamente, a replicação de UDDI ocorre entre nós UDDI que residem em sistemas diferentes a fim de explicitar esta cópia lógica no nó.

A realização física de um nó UDDI não é mandatória pela especificação.

Um registro UDDI corresponde a um ou mais nós UDDI combinados. Os nós em um registro UDDI coletivamente gerenciam um conjunto particular de dados de UDDI. Estes dados são distintos pela visão do comportamento associada com cada entidade contida nele. Um registro UDDI tem as seguintes características definidoras:

- Um registro é composto de um ou mais nós UDDI.
- Um registro deve ter uma decisão de política para cada ponto de decisão de política. Ele pode escolher delegar decisões de políticas para nós.

A realização física de um registro UDDI não é mandatória pela especificação.

As entidades *businessEntity*, *businessService*, *bindingTemplate* e *tModel* formam as estruturas de dados principais do UDDI. Dentro do registro, cada instância de uma estrutura de dados principal é identificada por uma chave de UDDI. Pela escolha de políticas apropriadas, múltiplos registros podem formar um grupo, conhecido como uma “afiliação”, cujo propósito é permitir controlar cópias das estruturas de dados entre eles. Uma afiliação de registro UDDI tem as seguintes características definidoras:

- Os registros compartilham um *namespace* comum para as chaves de entidades;
- Os registros têm políticas compatíveis para atribuir chaves para entidades;
- As políticas de registros permitem que publicadores atribuam chaves.

Dados são replicados entre nós de um registro UDDI usando um protocolo de replicação. Cada nó tem custódia de uma parte dos dados agregados gerenciados pelo registro do qual ele é parte. Cada dado está por definição sob custódia de exatamente um nó. Um dado, neste contexto pode ser um *businessEntity*, um *businessService*, um *bindingTemplate*, um *tModel*, ou um *publisherAssertion*. Uma mudança em um dado no registro deve ser realizada no nó que tem custódia do dado. O registro define a política para custódia de dados e, se permitido, o nó que tem custódia de um dado pode ser alterado.

2.2 Representação de informações em um UDDI

Para que web services sejam realmente úteis, há a necessidade de prover informações sobre eles além das informações técnicas. O propósito central do UDDI é representar dados e metadados sobre web services. Um registro UDDI fornece mecanismos para registrar, classificar e catalogar web services, a fim de que eles possam ser descobertos e consumidos. As informações são armazenadas de uma forma padronizada que permite que web services sejam consultados em tempo de projeto ou em tempo de execução para endereçar os seguintes cenários:

- Encontrar implementações de web services que estão baseadas em uma definição de interface abstrata comum;
- Encontrar provedores de web services que estão classificados de acordo com um esquema de classificação ou sistema de identificação conhecidos;
- Determinar a segurança e os protocolos de transporte suportados por determinado web service;
- Prover a pesquisa por serviços baseado em uma palavra chave geral;
- Armazenar em *cache* a informação técnica sobre um web service e, então, atualizar esta informação em tempo de execução.

A estrutura *businessEntity* é usada para representar negócios e provedores de um UDDI. Ela contém informações descritivas sobre o negócio ou provedor e sobre os serviços que ele oferece, por exemplo, nomes e descrições em múltiplos idiomas, informação de contato (lista de pessoas que estão associadas ao *businessEntity*, incluindo nome, endereço, números de telefone e tipo de uso), classificação (por exemplo, um *businessEntity* pode estar associado à categoria Califórnia, indicando que ele está localizado nesta área geográfica), identificadores (que o identificam univocamente) e URLs para documentos que descrevem o *businessEntity*. Descrições do serviço e informações técnicas são expressas em estruturas *businessService* e *bindingTemplate*, respectivamente, dentro de um *businessEntity*. Um *businessEntity* pode ser usado para modelar provedores de serviço “pai”, tais como um departamento, uma aplicação ou mesmo um servidor. A palavra *business* não se refere ao fato de que apenas informações de negócio (no seu uso comum) devem ser modeladas.

Cada estrutura de *businessService* representa um agrupamento lógico de web services. Nenhuma informação técnica é provida. Cada *businessService* é filho de um único *businessEntity*. Cada *businessService* contém informações descritivas (nomes, descrições e classificações – por exemplo, categorias que representam disponibilidade do serviço, versão etc.) descrevendo o propósito dos web services existentes dentro do *businessService*. Por exemplo, uma estrutura de *businessService* pode conter um conjunto de web services de tratamento de pedidos (submissão, confirmação e notificação) que são providos pelo negócio. Um *businessService* pode ser usado para descrever web services que não estão ligados diretamente a um negócio, como, por exemplo, web services de um provedor de serviços. Um registro UDDI pode registrar serviços arbitrários, por exemplo, EJB, CORBA etc.

Cada estrutura de *bindingTemplate* representa um web service individualmente, provendo informações técnicas necessárias para que aplicações se conectem (*bind*) e interajam com o web service descrito. Ela deve conter as informações sobre o ponto de acesso (*access point* – que representa o *endpoint* do serviço, contendo a URI do *endpoint* e a especificação de protocolo) de um dado serviço ou um mecanismo de indireção

que levará a um ponto de acesso. Cada *bindingTemplate* é filho de um único *businessService*. O *bindingTemplate* pode ser decorado com metadados que permitam a descoberta do *bindingTemplate*, dado um conjunto de parâmetros e critérios. O *bindingTemplate* pode estar associado com categorias para referenciar características do *bindingTemplate*, por exemplo, estados de certificação (teste, em produção) ou versões. A Figura 3, apresenta um *bindingTemplate* que especifica o endereço do *endpoint* de rede (no elemento *accessPoint*) e uma pilha de *tModels* descrevendo o serviço.

```
<businessService>
  (...)
  <bindingTemplates>
    <bindingTemplate>
      (...)
      <accessPoint urlType="http"> http://www.etc.com/</accessPoint>
      <tModelInstanceDetails>
        <tModelInstanceInfo tModelKey="...">
          </tModelInstanceInfo>
          (...)
        </tModelInstanceDetails>
      </bindingTemplate>
      (...)
    </bindingTemplates>
  </businessService>
```

Figura 3 – Exemplo de *bindingTemplate* [Colgrave e Januszewski, 2002]

Modelos técnicos ou *tModels*, são usados no UDDI para representar conceitos ou construtos únicos. Eles provêm uma estrutura que permite reuso e, portanto, padronização dentro de um framework de software. Cada especificação, transporte, protocolo ou *namespace* distinto é representado por um *tModel*. Exemplos de *tModels* que permitem a interoperabilidade de web services incluem os modelos baseados em WSDL (Web Service Description Language), XSD (XML Schema Definition), e outros documentos que delineiam e especificam o contrato e comportamento (por exemplo, a interface) que um web service pode escolher para estar de acordo. Para descrever um web service que está conforme a um conjunto de especificações, transporte e protocolos particular, referências aos *tModels* que representam estes conceitos são colocados no *bindingTemplate*. Desse modo, *tModels* podem ser reutilizados por múltiplos *bindingTemplates*. Os *bindingTemplates* que referenciam precisamente o mesmo conjunto de *tModels* têm a mesma assinatura e são do mesmo tipo. Desse modo, *tModels* podem ser usados para promover a interoperabilidade entre sistemas de software.

É importante ressaltar que os documentos técnicos e a documentação de apoio necessária para um desenvolvedor usando web services não são armazenadas dentro do registro UDDI. Um *tModel* do UDDI contém os endereços onde estes documentos podem ser encontrados. Um *tModel*, entretanto, contém mais do que apenas URLs; ele também armazena metadados sobre documentos técnicos e chave de entidade que identificam o *tModel*.

Em essência, um *tModel* é algo que tem um nome e uma descrição. Para entender melhor, pode-se fazer uma associação entre um *tModel* e uma tabela de banco de dados. Primeiro, um *tModel* é uma tabela independente, significando que ele pode existir por si mesmo. Segundo, um *tModel* é uma tabela de pesquisa (*lookup table*) provendo a tradução entre uma chave e uma representação do que a chave é. Neste sentido, um *tModel* é uma tabela de referência, como um dicionário. Pode se imaginar que um *tModel* é uma tabela com as seguintes colunas: nome, descrição, URL (que pode descrever melhor a respeito do nome e da descrição), chave única (para identificar o nome) [Oracle, 2008b].

tModels podem ser utilizados para representar outros conceitos dentro do UDDI a fim de prover interoperabilidade, tais como:

- Definições de protocolo e mecanismos de transporte, tais como HTTP e SMTP.
- Conjuntos de valores incluindo sistemas de identificação, sistemas de categorização e *namespaces*.
- Estruturas de categorização usando múltiplos conjuntos de valores chamados de grupos de categorização.
- Formatos de endereços postais.
- Qualificadores de localização usados para modificar o comportamento da API UDDI *find_xx*. Por exemplo, para definir que a operação *find_business* deve ordenar seus resultados na ordem que eles foram publicados, pode-se especificar *uddi:uddi.org:findqualifier:sortbydateasc*.
- Atributos de tipos de uso que especificam o tipo de recurso sendo referenciado por uma referência de URI.

A especificação UDDI define um conjunto de *tModels* comuns que podem ser usados canonicamente para modelar informação dentro de um UDDI. Se um conceito que é necessário para modelar um cenário em particular não existe em um registro, um usuário deve introduzir o conceito pela criação de um *tModel* contendo a URL para os documentos relevantes que descrevem o conceito.

Um exemplo de uso de *tModel* seria representar o fato de que um negócio descrito por um *businessEntity* tem um identificador de taxa US específico. Um *keyedReference* é colocado no *identifierBag* do *businessEntity*. O *keyedReference* tem um valor chave que é o identificador da taxa e refere-se ao *tModel* que significa “o sistema de identificadores de código de taxas dos Estados Unidos”. Juntos, o valor chave e a referência para o *tModel* especificam um valor específico em um sistema de valores específico.

A reificação² de dados em um UDDI é o principal para sua missão de descrição, descoberta e integração. Ele alcança isto de várias maneiras:

- Primeiro, ele permite que usuários definam múltiplas taxonomias que podem ser usadas no UDDI. Taxonomias são usadas para representar características e qualidades de entidades do UDDI (tais como, tipos de produto, regiões geográficas ou departamento em uma empresa). Dessa forma, múltiplos esquemas de classificação podem ser sobrescritos dentro

² No processo de alienação, o momento em que a característica de ser uma “coisa” se torna típica da realidade objetiva, ou seja, quando se consegue entender de forma clara o que o conceito apresentado.

de uma entidade UDDI particular. Esta capacidade permite que organização estenda o conjunto de sistemas que os registros UDDI suportam. Um registro pode não estar atrelado a um único sistema, podendo empregar muitos sistemas de classificação diferentes simultaneamente.

- Segundo, UDDI permite que tais sistemas de classificação sejam usados em cada entidade dentro do modelo de informação. É importante que taxonomias sejam utilizadas para publicar dados em um registro UDDI. Se códigos padrões são usados ou uma nova taxonomia é criada e distribuída, é intrínseco que dados UDDI (*businessEntity*, *businessService*, *bindingTemplate* e *tModel*) estejam associados com metadados.
- Terceiro, o conjunto API de pesquisa UDDI (UDDI Inquiry API) provê a habilidade de emitir buscas precisas baseado em diferentes esquemas de classificação. Um intervalo de consultas que realiza diferentes junções através das entidades UDDI pode ser gerado, tais dados podem ser descobertos e acessados. Além disso, registrando informações tais como códigos de indústria, códigos de produto, códigos geográficos e códigos de identificação do negócio permitem que outros serviços de busca usem estas informações de classificação como um ponto de partida para prover indexação e classificação com valor.

Sistemas de classificação e identificação, tomados juntos, são chamados de “conjuntos de valor” (*value sets*) em um UDDI. Conjuntos de valor podem ser checados e não checados e ambos usados para categorização e identificação. Um conjunto de valor checado significa que o UDDI não permite o uso de qualquer valor que não esteja dentro dos valores pré-definidos. Eles são usados quando o autor da taxonomia consegue enumerar todos os valores distintos possíveis para a taxonomia. Conjunto de valores não checados não prescrevem quaisquer valores fixos e qualquer nome e par de valor pode ser usado para categorizar entidades do UDDI. Eles são usados, por exemplo, para registrar volume, peso, preço etc. A diferença entre eles é que, toda vez que um conjunto de valor checado é usado, o seu uso é inspecionado para verificar se ele está conforme com os requisitos do conjunto de valor. Conjuntos de valores não checados não têm o uso inspecionado.

2.3 Segurança no UDDI

O modelo de segurança para um registro UDDI pode ser caracterizado por uma coleção de políticas de registro e de nó e a implementação destas políticas por um nó UDDI. As principais áreas de políticas e mecanismos de segurança na especificação UDDI estão relacionadas com gestão de dados, identificação de usuários, autenticação de usuário, autorização de usuário, confidencialidade de mensagens e integridade de dados.

A fim de autorizar ou restringir acesso aos dados em um registro UDDI, uma implementação de UDDI pode estar integrada com um ou mais sistemas de identificação. UDDI também suporta assinaturas digitais XML nos dados UDDI para permitir que pesquisadores (*inquirers*) verifiquem a integridade dos dados em relação ao publicador dos mesmos.

Os modelos de segurança para um registro e nó podem ser estendidos além dos mecanismos descritos na especificação e representados pelo modelo de web services de UDDI e através de documentação de políticas de nó e registro.

Segundo [ORACLE, 2008a], a especificação UDDI não define um mecanismo de controle de acesso. A especificação permite a modificação de uma entidade específica apenas pelo seu proprietário (o criador da entidade). Isto não escala em um ambiente de empresa aonde o direito para modificar ou apagar uma entidade de UDDI deve estar atribuída a muitas identidades, ou melhor ainda, a alguns perfis.

2.4 Usando WSDL em um registro UDDI

Descrições de serviço em WSDL podem ser estruturadas de múltiplas formas. Entretanto, se a informação reutilizável está separada da informação que é específica para uma dada instância de serviço, o uso de WSDL e UDDI torna-se relativamente simples [Colgrave e Januszewski, 2002].

Colgrave e Januszewski [2002] ressaltam que é importante notar que o elemento *import* em um WSDL permite a separação de elementos da descrição de um serviço em duas partes, chamadas de “definição da interface do serviço” e “definição da implementação” do serviço. Tipicamente, informações comuns a certa categoria de serviços de negócio, tais como formatos de mensagens, *portTypes* (interfaces abstratas), e protocolos de *binding*, estão incluídas na porção reutilizável, enquanto informações que pertencem a um *endpoint* de serviço particular (por exemplo, definição de porta) está incluída na parte de definição da implementação do serviço. No contexto do UDDI, a parte a ser considerada é a parte reutilizável do WSDL do serviço.

O processo para um WSDL suportar a criação de entidades de *businessService*, pode ser descrito nas seguintes partes:

1. Criar o WSDL da definição da interface do serviço. A definição da interface do serviço inclui interfaces e protocolos de *bindings* do serviço, e irá estar disponível de forma pública. As definições de interface do WSDL do serviço são então registradas como *tModels* do UDDI; o campo *overviewDoc* de cada *tModel* novo apontará para o documento WSDL correspondente. Estes *tModels* são chamados de “*wsdlSpec tModels*”. Cada *tModel* deve ser classificado utilizando taxonomia de tipos do *uddi-org:types*. Um exemplo é apresentado na Figura 4.

```
<tModel authorizedName="..." operator="..." tModelKey="...">
  <name>StockQuote Service</name>
  <description xml:lang="en">
    WSDL description of a standard stock quote service interface
  </description>
  <overviewDoc>
    <description xml:lang="en">WSDL source document.
    </description>
    <overviewURL>
      http://stockquote-definitions/stq.wsdl
    </overviewURL>
  </overviewDoc>
  <categoryBag>
    <keyedReference
      tModelKey="uuid:C1ACF26D-9672-4404-9D70-39B756E62AB4"
      keyName="uddi-org:types"
      keyValue="wsdlSpec"/>
  </categoryBag>
</tModel>
```

Figura 4 – Exemplo de *tModel*

Quando um *tModel* refere a documento WSDL como apresentado na Figura 4, ele refere-se ao conteúdo inteiro do documento, incluindo todos os seus *bindings*. É sempre conveniente que um *tModel* referencie um único *binding*. Isto pode ser realizado colocando na *overviewURL* colocando um XPointer para o *binding* como um fragmento identificador. A Figura 5 apresenta um exemplo de *tModel* referenciando especificamente *StockQuoteSoapBinding*.

```
<overviewURL>
http://stockquote-
definitions/stq.wsdl#xmlns (wsdl=http://schemas.xmlsoap.org/wsdl/)
xpointer (//wsdl:binding[@name='StockQuoteSoapBinding'])
</overviewURL>
```

Figura 5 – Exemplo de *overviewURL* com Xpointer para *binding*

2. Manualmente ou usando o ferramental do UDDI, programadores recuperarão a descrição do *tModel* e seguindo o link do *overviewDoc* obterão o documento de definição do WSDL correspondente. Ferramenta do UDDI pode ajudar a gerar uma implementação que suporta os padrões de *bindings* e interfaces.
3. Finalmente, os novos serviços devem ser instalados e registrados no repositório UDDI, manualmente ou usando ferramental do UDDI. Uma estrutura de dados *businessService* do UDDI é criada, e então registrada.

As informações contidas no novo *businessService* referencia os padrões implementados e provê detalhes de instalação adicionais:

- Um *bindingTemplate* é criado para cada *endpoint* de acesso do serviço. O endereço de rede do ponto de acesso é codificado no elemento *accessPoint*.
- Um *tModelInstanceInfo* é criado no *bindingTemplate* para cada *tModel* que é relevante para o ponto de acesso do serviço descrito, em particular, para cada *tModel wsdlSpec* que define interfaces e *bindings* suportados pelo serviço.

A Figura 6 apresenta um exemplo de WSDL, enquanto que a Figura 7 apresenta o seu *tModel* correspondente e a Figura 8 o *bindingTemplate* correspondente referenciando o *tModel* da Figura 7.

```
<?xml version="1.0"?>
<definitions name="StockQuote"
  targetNamespace="http://example.com/stockquote.wsdl"
  xmlns:tns="http://example.com/stockquote.wsdl"
  xmlns:xsd="http://example.com/stockquote.xsd"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns="http://schemas.xmlsoap.org/wsdl/">
  <types>
    <schema targetNamespace="http://example.com/stockquote.xsd"
      xmlns="http://www.w3.org/2001/XMLSchema" >
      <element name="TradePriceRequest">
        <complexType>
          <all>
            <element name="tickerSymbol" type="string"/>
          </all>
        </complexType>
      </element>
    <element name="TradePrice">
```



```

        <complexType>
            <all>
                <element name="price" type="float"/>
            </all>
        </complexType>
    </element>
</schema>
</types>

<message name="GetLastTradePriceInput">
    <part name="body" element="xsd1:TradePriceRequest"/>
</message>

<message name="GetLastTradePriceOutput">
    <part name="body" element="xsd1:TradePrice"/>
</message>

<portType name="StockQuotePortType">
    <operation name="GetLastTradePrice">
        <input message="tns:GetLastTradePriceInput"/>
        <output message="tns:GetLastTradePriceOutput"/>
    </operation>
</portType>

<binding name="StockQuoteSoapBinding"
    type="tns:StockQuotePortType">
    <soap:binding style="document"
        transport="http://schemas.xmlsoap.org/soap/http"/>
    <operation name="GetLastTradePrice">
        <soap:operation
            soapAction="http://example.com/GetLastTradePrice"/>
        <input>
            <soap:body use="literal"
                namespace="http://example.com/stockquote.xsd"
                encodingStyle=
                    "http://schemas.xmlsoap.org/soap/encoding"/>
        </input>
        <output>
            <soap:body use="literal"
                namespace="http://example.com/stockquote.xsd"
                encodingStyle=
                    "http://schemas.xmlsoap.org/soap/encoding"/>
        </output>
    </operation>
</binding>
</definitions>

```

Figura 6 – Exemplo de WSDL

```

<tModel authorizedName="..." operator="..." tModelKey="...">
    <name>StockQuote Service</name>
    <description xml:lang="en">
        WSDL description of a standard stock quote service interface
    </description>
    <overviewDoc>
        <description xml:lang="en">
            WSDL source document.
        </description>
        <overviewURL>
            http://example.com/stockquote.wsdl
        </overviewURL>
    </overviewDoc>
    <categoryBag>
        <keyedReference

```

```

        tModelKey=" uuid:C1ACF26D-9672-4404-9D70-39B756E62AB4"
        keyName="uddi-org:types"
        keyValue="wsdlSpec"/>
    </categoryBag>
</tModel>

```

Figura 7 – Exemplo de *tModel* para o WSDL da Figura 6

```

<businessService businessKey="..." serviceKey="...">
  <name>StockQuoteService</name>
  <description> (...) </description>
  <bindingTemplates>
    <bindingTemplate>
      (...)
      <accessPoint urlType="http">
        http://example.com/stockquote
      </accessPoint>
      <tModelInstanceDetails>
        <tModelInstanceInfo tModelKey="...">
          </tModelInstanceInfo>
        </tModelInstanceDetails>
      </bindingTemplate>
    </bindingTemplates>
  </businessService>

```

Figura 8 – *BindingTemplate* para o WSDL da Figura 6 referenciando o *tModel* da Figura

7

2.5 UDDI Schemas

UDDI usa a linguagem XML Schema para formalmente descrever sua estrutura de dados. A definição completa inclui 9 arquivos de esquema listados a seguir, cujos detalhes são apresentados em Bellwood [2004].

- UDDI API Schema
- UDDI Custody Schema
- UDDI Subscription Schema
- UDDI Subscription Listener Schema
- UDDI Replication Schema
- UDDI Value Set Validation Schema
- UDDI Value Set Caching
- UDDI Policy
- UDDI Policy Instance Parameters

3 Oracle Service Registry

Oracle Service Registry 10.3 corresponde a uma consolidação dos seguintes produtos anteriores BEA ALSR (BEA Aqualogic Service Registry), do fornecedor BEA antes de ter sido adquirida pela Oracle, e o Oracle OSR 10G (Oracle Service Registry) [ORACLE, 2008a].

Oracle Service Registry (OSR) está de acordo com a especificação UDDI V3 e corresponde a uma ferramenta composta de funcionalidades baseadas em padrões para publicação e descoberta de web services e recursos relacionados como, por exemplo, XML Schemas ou transformações XSLT. A ferramenta pode ser instalada na maioria dos ambientes Java e trabalha com os SGBDs (Sistemas Gerenciadores de Bancos de Dados) mais conhecidos.

O registro armazena descrições e metadados relacionados aos serviços, tais como, políticas, classificações e dependências. O registro não armazena os ativos correspondentes, tais como, WSDLs, esquemas, e outros, mas ao invés disso ele aponta para estes ativos. O principal propósito do registro é tornar disponível estes metadados dos serviços para serem consumidos por aplicações para descoberta e uso - este é o motivo para o qual serviços são "publicados" no registro para serem descobertos e utilizados por toda a empresa.

O OSR trata funcionalidades para aplicações da empresa, tais como:

- Segurança: permite definir controle de acesso granular para cada componente registrado. O responsável pela publicação do componente pode definir permissões de acesso para operações de localização (*find*), recuperação (*get*), modificação e remoção para cada componente publicado;
- Mecanismos para controle de qualidade e precisão de dados: para garantir que componentes registrados estão precisos e atualizados em seus respectivos ambientes. OSR define responsabilidades para cada componente registrado. Ele oferece mecanismos para promoção e aprovação de componentes entre ambientes de desenvolvimento, controle de qualidade e produção;
- Subscrição e notificação: para automaticamente notificar usuários a respeito das mudanças nos componentes que eles dependem;
- Replicação seletiva entre múltiplos registros: permite propagação automatizada entre diferentes registros (por exemplo, entre registros internos e externos);
- Gestão avançada de taxonomia para garantir taxonomias bem definidas;
- Gestão de desempenho: para controle, *logging* e auditoria dos processos de publicação e descoberta;
- Desempenho e escalabilidade: pela implementação de algoritmos para pilhas de serviços web e bancos de dados e suportando balanceamento de carga e clusterização.

O OSR não implementa algumas partes da especificação, como os requisitos de replicação, descrição de políticas e canonização XML para assinaturas digitais [CANONICALIZATION, 2002]. O mapeamento entre WSDL e UDDI é feito seguindo o padrão especificado pela OASIS [WSDL-UDDI, 2004], o qual contempla o novo vocabulário WSDL versão 2.0 [WSDL, 2008a].

O OSR faz parte da família Oracle Fusion Middleware e corresponde a um repositório que permite configurar, tratar escalabilidade e segurança das informações dos serviços publicadas no repositório. As principais características do OSR são:

- Permite que provedores de serviço publiquem e divulguem suas ofertas de serviços;

- Permite que consumidores localizem e recuperem serviços que estejam de acordo com suas necessidades;
- Serve como um ponto único de controle para governança e garantia de políticas em uma abordagem SOA. Políticas referem-se a restrições sobre entidades e restrições de como um cliente pode usar um serviço para facilitar o estabelecimento de um contrato entre um provedor de serviço e um usuário do serviço.

Os principais benefícios do OSR são:

- Provê agilidade do negócio através de infraestrutura dinâmica.
- Oferece controle do ambiente SOA através de políticas de publicação e atribuição em tempo de execução.
- Facilita descoberta e reuso de serviço em tempo de execução. Com o OSR, aplicações têm garantia de que requisições que têm obrigações contratuais (tempo de resposta, disponibilidade etc.) são roteadas para a localização do serviço apropriada que está de acordo com o acordo contratual.
- É uma fonte de autorização para informação de referências a *endpoints* – “DNS” para serviços.
- Minimiza necessidade de hardware adicional para escalabilidade ao permitir que organizações adicionem localizações adicionais para serviços, registrando-os no OSR para localização por consumidores no momento da requisição.
- Permite resiliência à falha. Tendo um registro UDDI para identificação da localização apropriada do serviço permite que aplicações não tenham mais que manter conexões ponto-a-ponto com localizações de serviços específicos, tornando mudanças e falhas mais transparentes para as aplicações.

O OSR é integrado diretamente com outros produtos do Oracle Fusion Middleware, tais como Oracle BPEL Process Manager, Oracle Enterprise Service Bus, Oracle Service Bus, Oracle Web Services Manager e Oracle JDeveloper. O OSR é uma solução independente de plataforma que pode ser facilmente instalada com diferentes configurações. O OSR também integra com diretórios LDAP³ (Lightweight Directory Access Protocol), incluindo Oracle Internet Directory e Microsoft ActiveDirectory.

3.1 Instalação

O OSR pode ser instalado em diferentes plataformas. Em relação a sistemas gerenciadores de bancos de dados, ele se conecta com as seguintes versões de SGBDs: Oracle 10g Release 1, Oracle 9i Release 2, Microsoft SQL Server 2005 e DB2 8.0.

O OSR pode ser instalado nos seguintes servidores de aplicação: Oracle Application Server (10.1.2 e 10.1.3) e Oracle WebLogic Server (9.2 e 10.3.0).

O OSR permite escolher cinco tipos de instalação:

³ LDAP é um protocolo de aplicação para consulta e modificação de serviços de diretórios executando sobre TCP/IP. Um diretório corresponde a um conjunto de objetos com atributos organizados de uma maneira lógica e hierárquica. Um exemplo de diretório é uma lista telefônica.

- *Standalone Registry*: Essa é a instalação padrão, na qual o servidor do OSR é instalado no Oracle Application Server e conecta a um banco de dados de registro externo.
- *Approval Process Registry*: Neste caso, a instalação do OSR é dividida em dois servidores, *Publication registry* e *Discovery registry*. O registro de publicação é o servidor preliminar para publicação, teste e aprovação de dados. Depois que os dados estão aprovados, eles são promovidos para o registro de descoberta. O registro de descoberta é configurado para pesquisas.
- *External Accounts Integration*: o servidor OSR pode ser opcionalmente configurado para usar contas externas em um servidor LDAP ou outro repositório de contas. É possível configurar integração de contas externas durante a instalação do banco de dados.
- *Registry Cluster*: Um cluster UDDI é um grupo de registros UDDI instalados em múltiplos servidores provavelmente com um cluster de bancos de dados no *back-end*. Balanceamento de carga é utilizado para distribuir solicitações entre servidores OSR a fim de obter a melhor distribuição.

Uma instalação de múltiplos registros deve ser escolhida se a organização deseja impor um maior controle sobre os conteúdos dos registros disponíveis para os consumidores. Este processo de controle de qualidade é obtido pela separação dos registros de publicação e descoberta, e usando um processo de aprovação para promoção dos serviços de um estágio para produção. O processo de aprovação pode ser configurado para usar aprovação manual ou automatizado para promover informações. Algumas observações importantes para este ambiente são:

- Cada registro requer um único *tablespace* e esquema dentro de um banco de dados para servir os metadados armazenados. Entretanto, ambos os *tablespaces* e esquemas podem ser criados dentro da mesma instância do banco de dados.
- Os registros de publicação e descoberta podem ser instalados em *hosts* separados do OracleAS⁴ (Oracle Internet Application Server) [Oracle, 2009a] ou no mesmo host.
- Se ambos os registros forem instalados na mesma instância do OAS, então idealmente eles devem ser instalados em instâncias separadas do OC4J⁵ [Oracle, 2009b], devido a requisitos de memória.

Além disso, uma ou mais instâncias intermediárias de registros podem ser instaladas. Uma instância intermediária de registro fica entre um ou mais registros de publicação e o registro de descoberta do nível mais alto.

Após a instalação os seguintes diretórios são criados:

- *bin*: contém os scripts de linha de comando para executar o OSR;
- *conf*: contém os arquivos de configuração do OSR;

⁴ Oracle Internet Application Server provê uma solução integrada para um infraestrutura de *middleware* incluindo Oracle Containers para J2EE, Oracle Web Cache, Oracle HTTP Server, Oracle Forms, Oracle Reports, Oracle Portal e Oracle Discoverer. Também incluído nesta infraestrutura estão segurança integrada, gestão e tecnologias de integração. Oracle Internet Application Server é o component da família de produtos Oracle Fusion Middleware.

⁵ Oracle Containers for J2EE (OC4J) é o componente principal de tempo de execução de J2EE do Oracle Application Server.

- demos: contém demonstrações de funcionalidades do OSR;
- dist: contém pacotes de clientes do OSR;
- doc: contém a documentação;
- etc: contém scripts e dados adicionais;
- log: contém logs de instalação, configuração, e saída do servidor;
- work: este diretório é a área de trabalho usada pelas ferramentas de linha de comando.

Maiores detalhes sobre instalação do OSR em ambiente *Windows* podem ser encontrados no (

Apêndice 1).

3.2 Fluxo de aprovação

As mudanças realizadas sobre os serviços podem ser controladas por um fluxo de aprovação definido. Além disso, modificações podem ser notificadas a usuários e processos sobre os elementos que foram alterados através da funcionalidade de subscrição e notificação do OSR. No OSR, a notificação pode ser síncrona ou assíncrona. Na notificação síncrona, a parte interessada explicitamente pergunta sobre mudanças que ocorreram desde a última notificação, e o OSR responde. Notificações assíncronas executam periodicamente em um intervalo configurável e a parte interessada é notificada sempre que a entidade que atende a consulta é criada, modificada ou removida.

3.3 Subscrição e notificação

O mecanismo de subscrição permite que usuários monitorem entradas novas, alteradas ou apagadas referentes a *businessEntities*, *businessServices*, *bindingTemplates*, *tModels* ou *publisherAssertions*. O conjunto de entidades que um usuário está interessado é expressa por um filtro de subscrição (*SubscriptionFilter*), o qual pode ser qualquer uma das seguintes consultas de API de UDDI v3:

- `find_business`, `find_relatedBusinesses`, `find_services`, `find_bindings`, `find_tmodel`
- `get_businessDetail`, `get_serviceDetail`, `get_bindingDetail`, `get_tModelDetail`, `get_assertionStatusReport`

Além disso, no Business Service Control, usuários podem criar subscrições para recursos (WSDL, XML, XSD e XSLT) sem conhecimento detalhado de como os recursos são mapeados para estruturas de dados UDDI.

Subscrições podem ter os seguintes argumentos: chave de subscrição (*SubscriptionKey*), filtro de subscrição (*SubscriptionFilter*), tempo de expiração (*Expires after*), intervalo de notificação, número máximo de entidades, chave de ligação (*bindingKey* - opcional - aponta para o *bindingTemplate* que inclui o *endpoint* do serviço que trata a notificação) e resumo (*brief* - por exemplo, se *brief* estiver habilitado apenas identificadores de entidades são enviados na notificação de *businessEntity*).

3.4 Classificação de serviços usando taxonomias

Serviços podem ser classificados no OSR, tanto ao nível técnico como do negócio, de acordo com taxonomias definidas. Além disso, o OSR inclui um conjunto de serviços de validação de taxonomia como definido pelas especificações UDDI. Ele ainda permite a definição de taxonomias da empresa. Usuários podem baixar (*download*) e publicar (*upload*) qualquer taxonomia como um arquivo XML. OSR oferece ferramentas para criar, modificar e navegar em taxonomias através de interface web e níveis de API SOAP.

O negócio também pode especificar sua própria classificação usando taxonomias que mapeiam as características específicas do negócio para facilitar a descoberta de serviços. No Apêndice 10 é apresentado um roteiro para associar um serviço de negócio a um provedor de negócio, enquanto que no Apêndice 12 é apresentado como utilizar taxonomias para pesquisar serviços. Classificações podem ser estendidas para

especificar dados adicionais, por exemplo, criação de campos de dados adicionais para serem associados aos WSDLs ou interfaces dos serviços durante a publicação. O serviço de validação interna faz a checagem para garantir que uma categorização específica está válida dentro de uma taxonomia.

3.5 Segurança

OSR permite definir que múltiplos usuários ou perfis possam ter acesso para modificar ou apagar uma entidade específica UDDI. O Apêndice 4 apresenta um roteiro para gerenciamento de permissões de usuários e grupo de usuários.

A segurança no OSR é feita através de uma extensão do ACL (Access Control List) para o modelo de segurança de UDDI. Cada entidade UDDI pode estar associada com a ACL que define quem pode encontrá-la (*find* - listá-la em algum resultado de consulta UDDI), recuperar (*get* - recuperar todos os detalhes do objeto do UDDI), modificá-la ou apagá-la. O ACL pode referenciar ambos a conta de um usuário específico ou grupo de usuários. A versão 3 da especificação UDDI provê suporte para assinaturas digitais. No OSR, o publicador da estrutura do UDDI pode digitalmente assinar esta estrutura. A assinatura digital pode ser validada para verificar se as informações não podem ser modificadas por nenhuma razão e confirmar a identidade do publicador.

O conteúdo do registro UDDI pode ser replicado no OSR usando o modelo de master e escravo (*master-slave*). O registro UDDI pode replicar dados de acordo com múltiplas definições de replicação que são definidas usando consultas padrão do UDDI. A relação *master-slave* é específica a uma definição de replicação. Logo, um registro pode ser master para uma definição de replicação específica e escravo para outra. As definições de segurança (ACL, usuários e grupos) não estão sujeitas à replicação, mas podem ser ajustadas permissões sobre os dados replicados.

O OSR possui uma ferramenta web de controle de serviços de negócio que permite que usuários não familiares com UDDI possam olhar modelos publicados (WSDL, XML Schema, XSLT, BPEL, etc.) ao invés de terem que analisar tModels e entidades do negócio, reduzindo custo de treinamento e tempo de implementação de serviços de negócio.

3.6 Processo de aprovação

O processo de aprovação provê funcionalidades para garantir consistência e qualidade de dados armazenados no OSR. Dois tipos de registros estão envolvidos no processo:

- Registro de publicação: usado para teste e verificação de dados;
- Registro de descoberta: contém apenas dados que foram aprovados e promovidos do registro de publicação.

Dois tipos de usuários estão envolvidos no processo de aprovação:

- Um *requisitante* que é o usuário do registro de publicação que pode requisitar aprovação de dados para promoção para o registro de descoberta. Além disso, o requisitante pode pesquisar, obter e cancelar requisições, além de sincronizar dados do registro de publicação com os dados do registro de descoberta;

- Um *avaliador* (*approver*) que é o usuário que pode aprovar ou rejeitar requisições por promoções de dados.

Administradores podem especificar:

- Usuários ou grupos de usuários que podem ser avaliadores;
- Usuários ou grupos de usuários cujas requisições eles podem validar.

Cada usuário pode requisitar por aprovação, mas para ter dados considerados para promoção, um usuário deve ter um validador atribuído pelo administrador. Ou seja, o usuário pode solicitar que um determinado dado seja validado, mas pode não ter permissão para solicitar que ele seja promovido.

O ciclo de aprovação é apresentado na Figura 9. Um requisitante pode criar uma requisição. Uma vez a requisição criada, o requisitante pode adicionar estruturas UDDI ou recursos (WSDL, XML, XSD e XSLT) na requisição. Observe que o requisitante não precisa saber como recursos são mapeados para estruturas de dados do UDDI. Quando um requisitante adiciona um recurso à requisição, todas as estruturas UDDI (*bindings*, *tModels*) que o recurso representa são automaticamente adicionadas à requisição. Uma vez que o requisitante especifica todas as entidades para serem promovidas, a requisição é submetida para aprovação. Neste momento, todos os dados referenciados pela requisição ficam bloqueados para escrita até que a requisição é cancelada pelo requisitante ou aprovada ou rejeitada pelo avaliador.

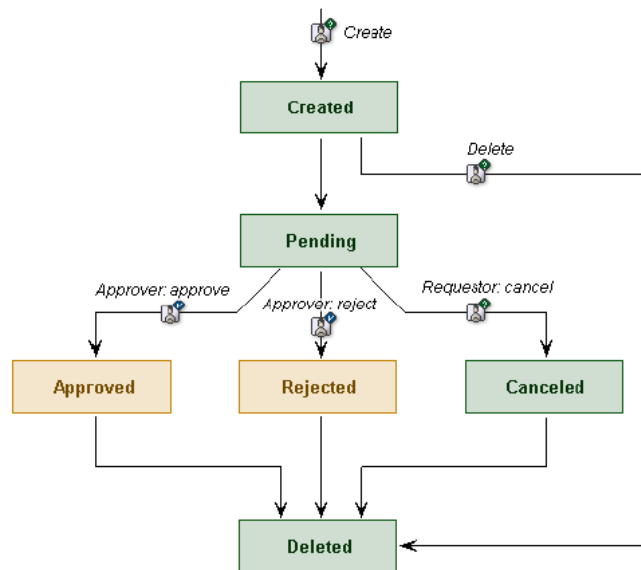


Figura 9 – Ciclo de aprovação

O avaliador avalia a requisição, e então aprova ou rejeita a requisição. Se o avaliador aprova a requisição, ela é imediatamente promovida para o registro de descoberta. Se o requisitante não está satisfeito com o tempo da aprovação, o usuário pode lembrar o avaliador de revisar suas requisições. O requisitante também pode cancelar requisições.

O OSR realiza checagem de contexto para garantir a consistência das requisições sendo realizadas no registro de publicação para serem publicadas no registro de descoberta como, por exemplo, verificar se está sendo requisitada a remoção de um dado que não existe no registro de descoberta.

Um requisitante pode receber o status de auto-avaliador (*AutoApprover*). Neste caso, todas as requisições deste usuário são automaticamente promovidas, sem ocorrer o fluxo de aprovação.

Opcionalmente uma validação de conteúdo (*content checking*) pode ser definida pelo aprovador, por exemplo: cada *businessService* deve incluir um *templateBinding*; cada *businessService* deve estar categorizado em determinadas categorias. Para realizar estas validações, o avaliador deve escrever uma implementação na API de checagem (*API Checker*) para garantir estas checagens.

No processo de aprovação, e-mails de notificações são enviados para requisitante e avaliadores quando ocorrem ações sobre dados que eles estão envolvidos. O formato do email é definido por transformações XSL.

3.7 Consoles do OSR

O OSR provê duas interfaces com o usuário para gestão de entidades do OSR.

- O Business Service Control permite que desenvolvedores, arquitetos e usuários do negócio possam navegar sobre informações de esquemas, nomes locais de interfaces ou namespaces etc. Além disso, ele permite realizar publicações usando *wizards* de publicação.

O Business Service Control pode ser encontrado na URL <http://<Business Service Control hostname>:<port>/<context>/uddi/bsc/web>. A porta padrão é 8888 ou 80 dependendo da configuração do servidor de aplicação.

- Usando o Registry Control, usuários podem navegar e publicar conteúdo no registro, criar subscrições e realizar mudanças de propriedade. Esta é a principal ferramenta para administradores realizarem gestão do registro.

O Registry Control pode ser encontrado na URL <http://<hostname>:<port>/<context>/uddi/web>. A porta padrão é 8888 ou 80 dependendo da configuração do servidor de aplicação.

Existem dados pré-instalados no OSR para uso do Business Service Control e do Registry Control.

3.7.1 Business Service Control

O Business Service Control tem os seguintes itens de menu:

- *Home*: página inicial
- *Catalog*: permite listar, pesquisar e publicar entidades
- *Tools*: permite visualizar e gerenciar subscrições e requisições de aprovação
- *Report*: permite visualizar o conjunto pré-definido de relatórios
- *Configure*: permite configurar o Business Service Control.

A Figura 10 e a Figura 11 apresentam exemplos de telas do Business Service Control: aba *Catalog* do Business Service Control e página de listagem de WSDL de serviços. As letras destacam funcionalidades importantes da ferramenta, estas funcionalidades são descritas em [Oracle, 2008a].

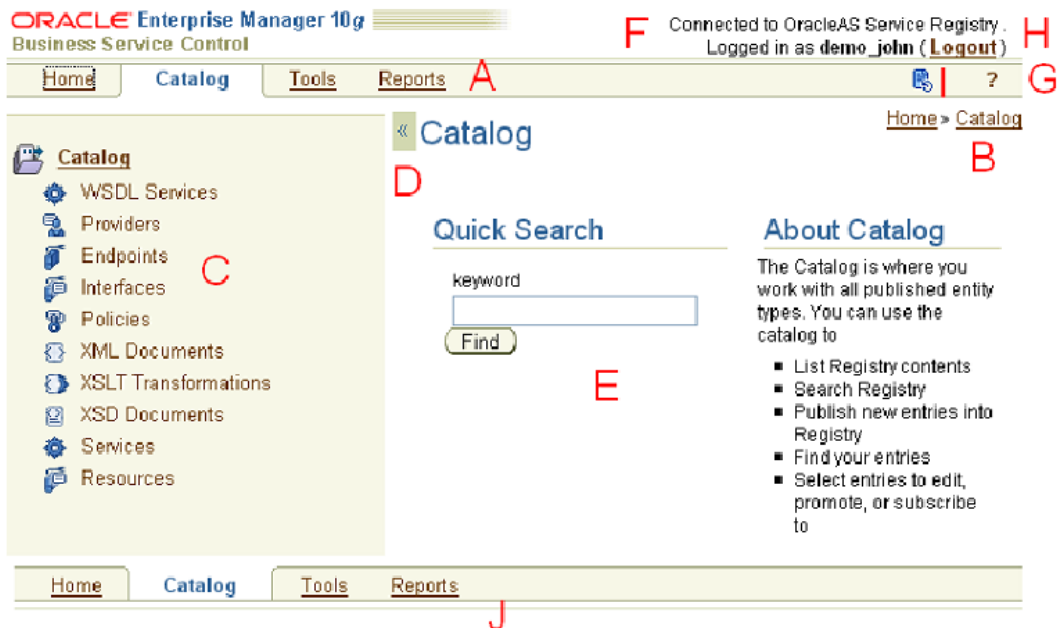


Figura 10 – Página da funcionalidade *Catalog* do Business Service Control

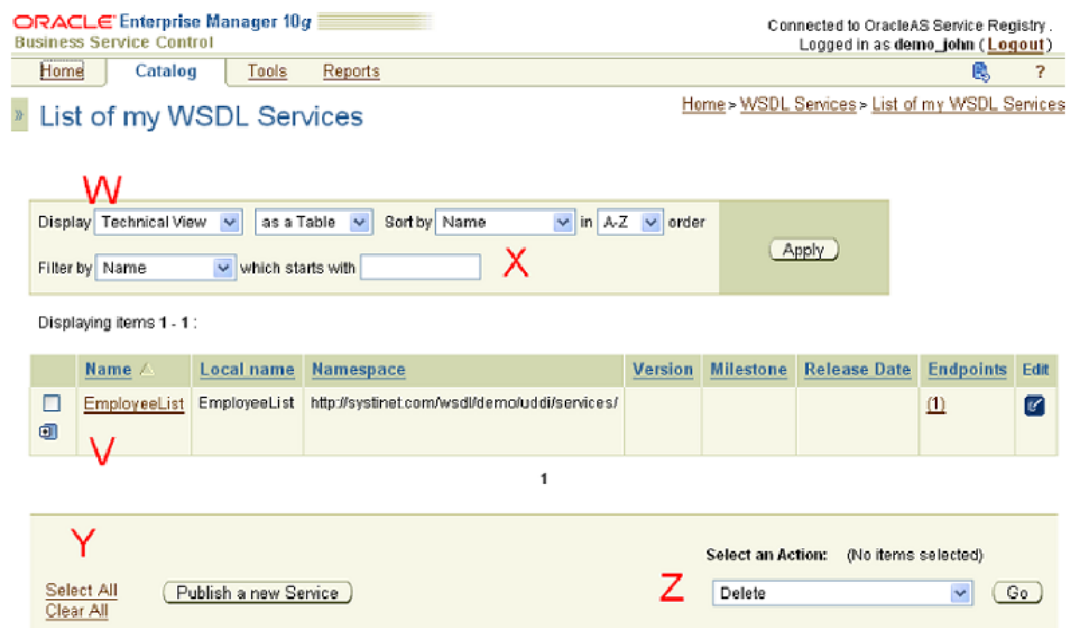


Figura 11 – Página de listagem de WSDL de serviços do Business Service Control

3.7.1.1 Conta de usuário

Antes de publicar dados no OSR, é necessário ter uma conta no mesmo. Ao cadastrar uma conta, cujos passos são apresentados no Apêndice 2, dentre as informações que devem ser preenchidas destacam-se:

- Intervalo de notificação padrão: especifica o intervalo que e-mails de notificação serão enviados.
- Número máximo de entradas a serem enviadas em um único email.

- Exibir atualizações realizadas no último período, por exemplo, 3 dias.

3.7.1.2 Perfis previamente definidos

O OSR possui uma lista de perfis (*profiles*) previamente definidos. Eles diferem nos itens das abas que estarão disponíveis para cada um. Cada *profile* também contém uma definição dos formatos padrões para visualizações de resultados. O administrador do registro pode atualizar estes *profiles*:

- *Business Expert*: entende os problemas que precisam ser resolvidos e relacionamentos e implicações em relação a outros sistemas dentro da empresa. Ele propõe componentes funcionais reutilizáveis (serviços de negócio futuro) e como eles resolvem problemas específicos. Este usuário associa requisitos funcionais (descrições) e não funcionais (capacidades e restrições em alto nível) aos componentes. Ele também sugere reuso de serviços existentes.

- *Developer*: Implementa serviços de negócio de acordo com descrições e restrições/capacidades associadas aos serviços. Este usuário reutiliza infraestrutura de baixo nível para serviços para realizar a implementação.

Depois do desenvolvimento, serviços de negócio têm sua qualidade avaliada (QA - Quality Assurance) e são testados.

- *SOA Architect*: refatora as entradas do *Business Expert*. Este usuário realiza o seguinte:
 - Traduz instalações definidas pelo *Business Expert* em um conjunto de serviços de negócio reutilizável.
 - Transforma capacidades e restrições em alto nível para capacidades e restrições baseadas em padrão que podem ser garantidos e implementados por outros papéis (desenvolvedores, administradores e gerentes de operação).
 - Define capacidades e restrições (tais como restrições de acordo) que garantem estar de acordo a padrões e implementações comuns e práticas de desenvolvimento de serviços da empresa.
 - Garante que serviços estão de acordo com padrões (SOAP, WSDL, UDDI, WS-S, WS-RM etc.).
 - Sugere reuso de serviços existentes.
- *Operator*: disponibiliza e gerencia no ambiente de produção os serviços de negócio implementados pelo *Developer*.
 - Publica *endpoint* de serviços e outros dados de tempos de execução sobre o serviço instalado.
 - Garante que serviços de negócio estão sendo gerenciados e seguros apropriadamente pela marcação do serviço com categorias apropriadas que inicia segurança e o processo de registro no WSM (Web Service Manager⁶).

⁶ Oracle Web Services Manager é uma solução para gestão de política e segurança da infraestrutura de serviços. Ele provê visibilidade e controle de políticas através de interface de administração centralizada oferecida pelo Oracle Enterprise Manager.

- *SOA Administrator*: realiza as mesmas funções do *operator*, mas com maiores privilégios.
- *Anonymous User Profile*: este perfil se aplica a usuários não autenticados. O perfil é uma configuração para usuários que não estão logados no Business Service Control.

3.7.1.3 Pesquisa por entidades

O OSR permite realizar pesquisas por provedores, serviços, *endpoints*, interfaces e artefatos publicados. A Figura 12 apresenta a tela de pesquisa do OSR.

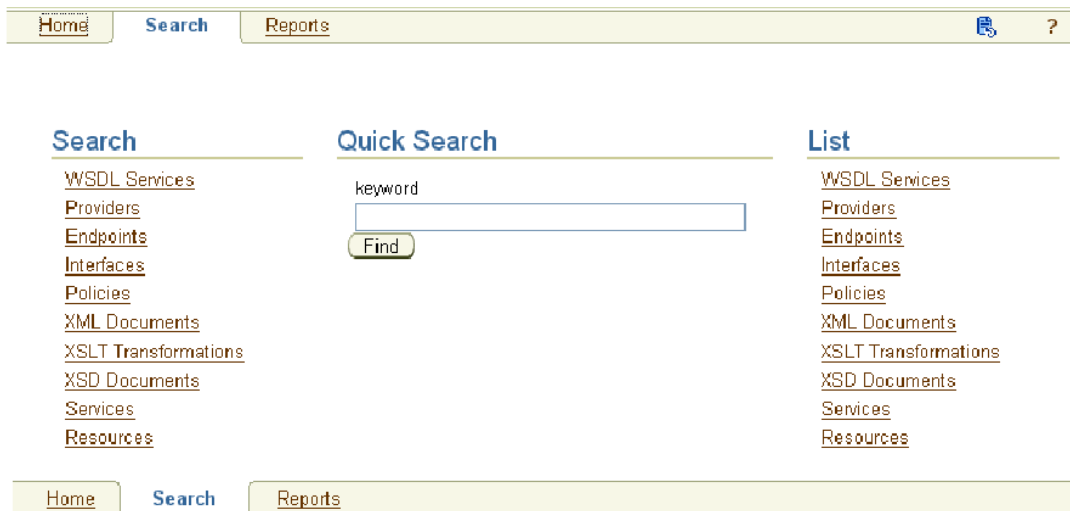


Figura 12 – Tela de pesquisa

Pesquisas podem utilizar *wildcards* (por exemplo, “%”) e são customizadas para cada tipo de pesquisa. A Figura 13 apresenta a tela de pesquisa por *endpoints*. Observe que se pode pesquisar por propriedades operacionais, tais como, disponível, indisponível, degradado, em manutenção, em teste etc.

Search Terms **Results**

Common properties

Usage

WS-I Compliance Basic Profile 1.0 Basic Profile 1.1

Operation properties

Availability Available Degraded Unavailable

Status In Maintenance In Test Not Available Operational

Keywords =

Technical properties

Local name

Namespace

Binding properties

Protocol soap http

Transport http

Figura 13 – Tela de pesquisa por endpoints

Resultados podem ser customizados, como apresentado na Figura 14. Observe que o formato de visualização pode ser definido. Por exemplo, uma das opções é visualização técnica (*Technical View*).

Search Terms **Results**

Display as a Sort by in order

Filter by which starts with

Displaying items 1 - 1 :

	AccessPoint ▲	UseType	Namespace
	http://company.com/accou.../account	http://schemas.xmlsoap.org/soap/http	

Figura 14 – Resultado de busca por endpoint

3.7.1.4 Publicação

Pode-se publicar os seguintes dados no OSR:

- Provedores: nome do provedor e descrição, classificação taxonômica do provedor e pessoas de contato. O Apêndice 5 apresenta um roteiro para publicação de provedores. O último passo para publicação do provedor é definir se será realizado um fluxo de aprovação.

- Serviços: *wizard* para publicação do serviço, suas interfaces e seus *endpoints*.
- Interfaces
- Recursos: arquivos WSDL, XML, XML Schema e XSLT.

O último passo para publicação é definir se será realizado um fluxo de aprovação.

Um serviço pode ser publicado a partir de um WSDL ou sem WSDL. A URL do WSDL do serviço que se deseja publicar pode ser encontrada utilizando a aplicação de busca de serviço Web Services Inspection Language⁷ (WSIL) disponibilizado pelo Oracle Application Server, ou diretamente a partir do arquivo WSDL armazenado em algum diretório. Os passos para publicação de serviço a partir do seu WSDL são apresentados nas figuras: Figura 15, Figura 16, Figura 17, Figura 18 e Figura 19.

Figura 15 – Publicação de WSDL

Figura 16 – Preenchimento de credenciais para autenticação

⁷ <http://www.ibm.com/developerworks/library/specification/ws-wsilspec/>

Step 2 of 4

Service properties

Service creation method:

new service
 rewrite service ---- no service ----

Service properties

Name	EmployeeList	
Description		
Usage		
Keyword		= <input type="text"/> <input type="button" value="Add another"/>
Certification	<input type="radio"/> Certified <input type="radio"/> Not Applicable <input type="radio"/> Pending <input checked="" type="radio"/> N/A	
Release date		
Version		
Milestone		

Step 2 of 4

Figura 17 – Publicação de propriedades do serviço

Step 3 of 4

Interface properties

Interface creation method

new interface
 rewrite interface EmployeeList_portType
 reuse interface EmployeeList_portType

Interface properties:

Local name	EmployeeList_portType	
Description		
Usage		
Keyword		= <input type="text"/> <input type="button" value="Add another"/>
Status	<input type="radio"/> Beta <input type="radio"/> Deprecated <input type="radio"/> Obsolete <input type="radio"/> Stable <input checked="" type="radio"/> N/A	
Compliance	N/A	

Step 3 of 4

Figura 18 – Publicação das propriedades da interface do serviço

Step 4 of 4

Endpoint properties

AccessPoint

Usage

Keyword =

Compliance

Availability
 Available
 Degraded
 Unavailable
 N/A

Status
 In Maintenance
 In Test
 Not Available
 Operational
 N/A

Step 4 of 4

Figura 19 – Publicação das propriedades do endpoint do serviço

Maiores detalhes sobre publicação de um serviço utilizando WSDL no OSR podem ser encontrados no Apêndice 6.

3.7.1.5 Relatórios

O Business Service Control contém um conjunto pré-definido de relatórios, como apresentado na Figura 20. Os relatórios do primeiro nível da árvore listam serviços, *endpoints* ou interfaces de acordo com as taxonomias: `systinet-com:taxonomy:usage`; `systinet-com:taxonomy:endpoint:status`; `systinet-com:taxonomy:interface:status`; `uddi-org:xml:namespace taxonomy`; `uddi-org:xml:localName`; `systinet-com:taxonomy:service:certification`; `systinet-com:taxonomy:endpoint:availability`; `ws-i-org:conformsTo:2002_12`; `systinet-com:versioning:milestone`; `systinet-com:versioning:releaseDate`; `systinet-com:versioning:version`.

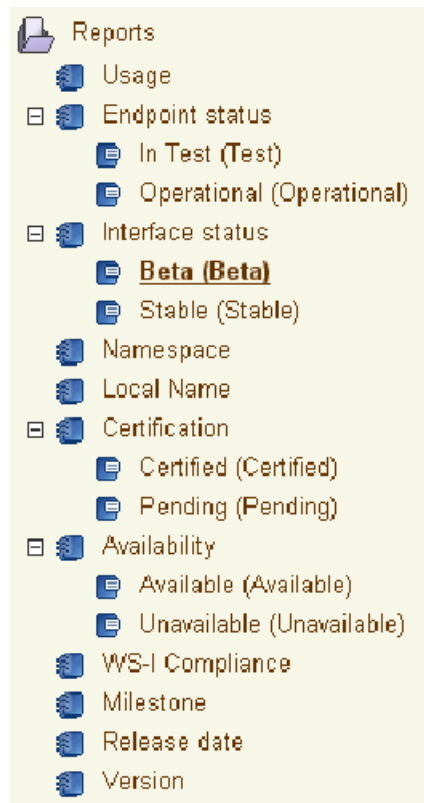


Figura 20 – Relatórios pré-definidos no Business Service Control

3.7.1.6 Subscrições e notificações

Subscrições podem ser criadas no OSR para alertar usuários interessados em acompanhar mudanças em estruturas existentes no OSR. O Business Service Control permite criar e gerenciar subscrições para monitoramento de entidades novas, alteradas ou apagadas. As seguintes entidades podem ser monitoradas: provedores, serviços, interfaces e *endpoints*, bem como recursos (WSDL, XML, XSD e XSLT). Uma subscrição pode ser definida para um conjunto de entidades que se esteja interessado ou a uma consulta de busca específica. Usuários podem receber notificações sobre estruturas modificadas via email ou eles podem visualizar as modificações das entidades utilizando ferramentas do OSR, em *My Subscription Results*. No caso de escolhido receber notificações por email, então se pode configurar o intervalo de notificação.

A Figura 21 apresenta um exemplo de criação de subscrição para o provedor *HR Services*, enquanto que a Figura 22 apresenta os ajustes de notificação para acompanhamento das mudanças no provedor escolhido. Os passos para criar uma subscrição a partir de uma busca são os mesmos, exceto pelo fato de ter sido feita uma busca antes de se chegar à listagem das entidades que atendem a pesquisa, ou seja, antes de se ter a tela apresentada na Figura 21. A Figura 23 mostra a tela de criação de subscrição para uma consulta de serviços, ou seja, tendo sido realiza antes uma busca por serviços. É importante notar que não é possível modificar a consulta para a qual foi feita uma subscrição. Logo, se for necessário alterá-la, então é necessário cancelá-la e recriá-la.

Display as a Sort by in order
 Filter by which starts with

Displaying items 1 - 3 :

	Provider name ▲	Contact name	Contact phone	Contact email	Edit
<input type="checkbox"/>	Headquarter Headquarter department	Joe Black	1-234-567-890		<input type="checkbox"/>
<input checked="" type="checkbox"/>	HR Services Provides HR related services				<input type="checkbox"/>
<input type="checkbox"/>	IT IT department	John Demo	1-123-456-7890		<input type="checkbox"/>

1

Select an Action: (1 item(s) selected)

[Reports](#)

Figura 21 – Criação de subscrição para provedor *HR Services*

Notification setup

Subscription Filter

Subscribed to:

Name	Type
HR Services	Provider

Select notification parameters for this subscription:

Email Address: No notifications

Notification Interval:

Notifications Expire After: No Expiration
 (MM/DD/YYYY) / /

Maximum Updates to Send: No Maximum
 Entries

Suppress Empty Notifications

Send Raw XML

Step 1 of 1

Figura 22 – Ajustes de parâmetros de notificação da subscrição

Search Terms		Results	
Display	Default View	as a Table	
Sort by	Name	in A-Z order	
Filter by	Name	which starts with	
<input type="button" value="Apply"/>			
Displaying items 1 - 5 of 17 : (Single page)			
Name	Provider name	Description	Edit
<input type="checkbox"/> AccountService	Account Services	The account service provides the account related operations	
<input type="checkbox"/> AddCustomerService	Customer Management System	This service allows a customer to be added to the enterprise customer system.	
<input type="checkbox"/> BillPaymentService	Account Services	The bill payment service provides the ability to establish bill payment service, cancel bill payment service and get information about bill payment for a customer.	
<input type="checkbox"/> CheckOrderService	Account Services	This service supports new check orders, check reorders, check order inquiry	
<input type="checkbox"/> CustomerNotificationService	Customer Management System	This service provides notification messages for various customer changes	
1 2 3 4			Next →
Select Page Select All Clear Page Clear All		<input type="button" value="Publish a new Service"/>	Select an Action: (No items selected) <input type="button" value="Go"/>
		<input type="button" value="Delete"/>	
		<input type="button" value="Promote/Demote"/>	
		<input type="button" value="Subscribe to Selected Services"/>	
		<input type="button" value="Subscribe using this Search"/>	
Home	Catalog	Tools	Reports Configure

Figura 23 – Tela de criação de subscrição a partir de busca por serviços

Para revisar e ajustar subscrições deve-se utilizar a funcionalidade *Manage My Subscriptions* (Figura 24).

Manage My Subscriptions

Type	Notifications to	Notifications Expire	
<input type="checkbox"/> Get Provider	demo_john@mycompany.com	Nov 26, 2006 1:17:29 PM	
<input type="checkbox"/> Query Service	demo_john@mycompany.com	Nov 23, 2006 11:59:59 PM	
Select an Action: (No items selected)			
Select All <input type="button" value="Back"/> Clear All		<input type="button" value="Delete"/> <input type="button" value="Go"/>	

Figura 24 – Tela de gestão de subscrições

3.7.1.7 Configuração

Na aba *Configure* é possível configurar [Oracle, 2008b]:

- *Tabs Display*: permite definir as abas que serão exibidas para os usuários que têm um perfil específico.

- *Search Result View Display*: permite configurar os tipos de visualizações de resultado para cada perfil de usuário.
- *Browsable Taxonomies*: permite definir quais taxonomias são navegáveis.
- *User Interface Configuration*: permite configurar a interface do Business Service Control. Por exemplo, customizar URL do registro, diretório onde ficarão as páginas estáticas, diretório para *Upload* etc.
- *Customizable Taxonomies*: permite definir quais taxonomias podem ser customizadas pelos usuários quando este estiver criando, editando ou pesquisando entidades.

3.7.2 Registry Control

O Registry Control é utilizado por administradores de sistema e usuários avançados para configuração de sistema, e gerência de usuários e segurança, e manipulação direta do modelo de dados do UDDI.

As principais funcionalidades do Registry Control são:

- *Navegação (Browse)*: permite navegar em entidades do UDDI utilizando taxonomias.
- *Pesquisa*: permite realizar pesquisa no registro. Pode-se realizar uma busca nas entidades do UDDI e encontrar *businessEntity*, serviços, *bindings*, *tModels*, etc.
- *Publicar*: permite publicar estruturas do UDDI (*businessEntity*, *businessServices*, *bindingTemplates* e *tModels*).
- *Profile*: para gestão de propriedades de contas de usuários, grupos de contas e taxonomias.
- *Manage*: permite realizar tarefas de gestão.

A interface principal do Registry Control é apresentada na Figura 25. As letras destacam as funcionalidades e são descritas em detalhes em [Oracle, 2008a].

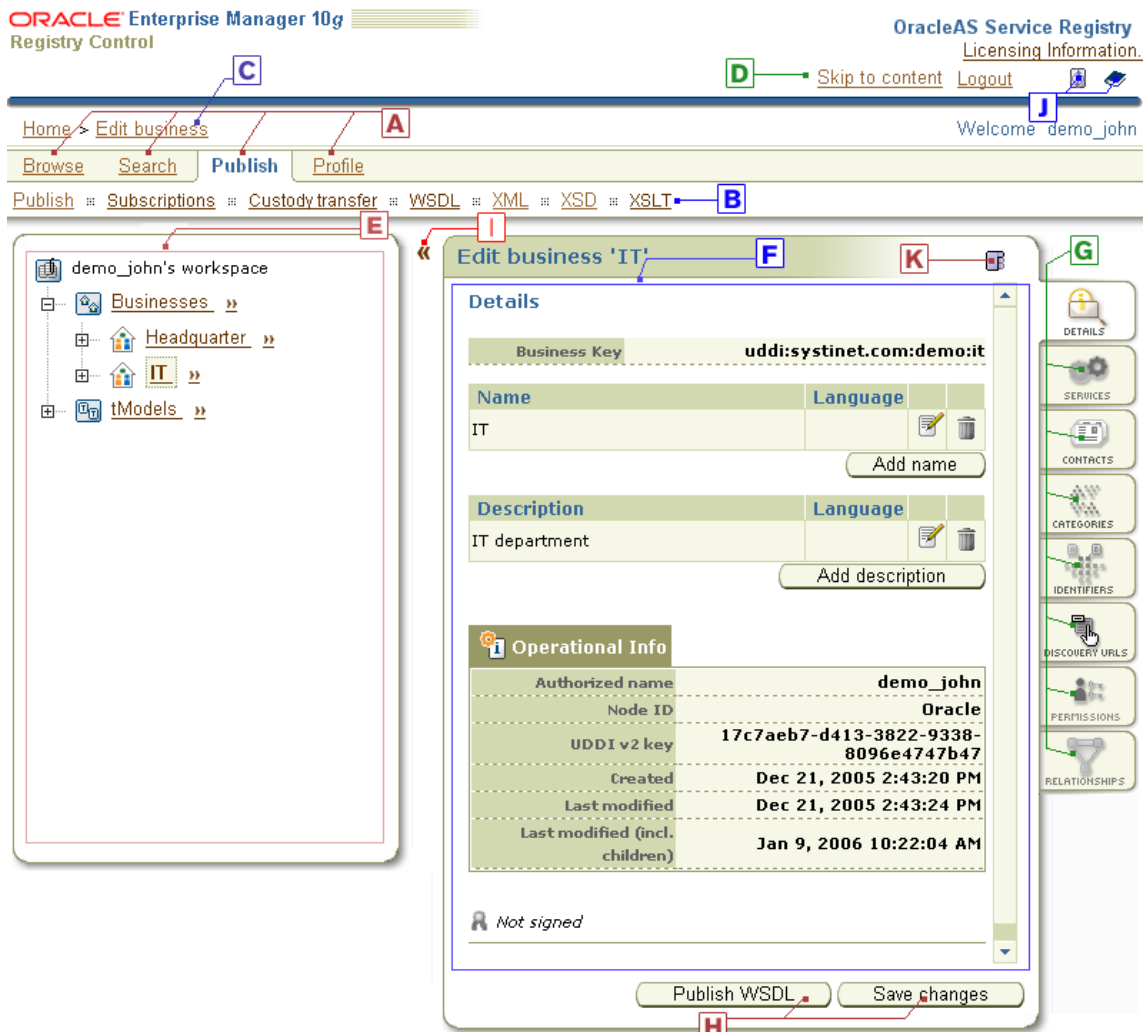


Figura 25 – Interface do Registry Control

3.7.2.1 Perfil de usuário

Para o usuário logado alterar dados de sua conta, ele deve clicar em *Profile* (Figura 26), em seguida, clicar em *My account* e então em *Edit account* (Figura 27). Também é possível editar grupos (Figura 28) e a associação de grupos com usuários (Figura 29).



Figura 26 – Detalhes do menu *Profile*

View account

Login name:	john
Email:	john@company.com
Full name:	John Johnson
Default language:	English
Description:	
Business Name:	
Phone:	1-858-4648442
Alternate phone:	1-858-4648444
Address:	4585 Pacific Beach Ave
City:	San Diego
State province:	CA
Country:	USA
Zip:	92107
User profile:	Developer Profile
Last logged in:	Mon Jan 09 14:53:47 ICT 2006
External:	No
Blocked:	No
Assertions limit:	10
Bindings limit:	2
Businesses limit:	1
Services limit:	4




Back Delete Account Edit Account

Figura 27 – Tela de visualização de conta de usuário

Groups

Filter: Filter:

Displaying results 1 - 2 of 2

Group name	Description	Visibility		
john_group_a	John Group A	public		
john_group_b	John Group B	private		

1

Add Group

Figura 28 – Tela de visualização de grupos de usuários

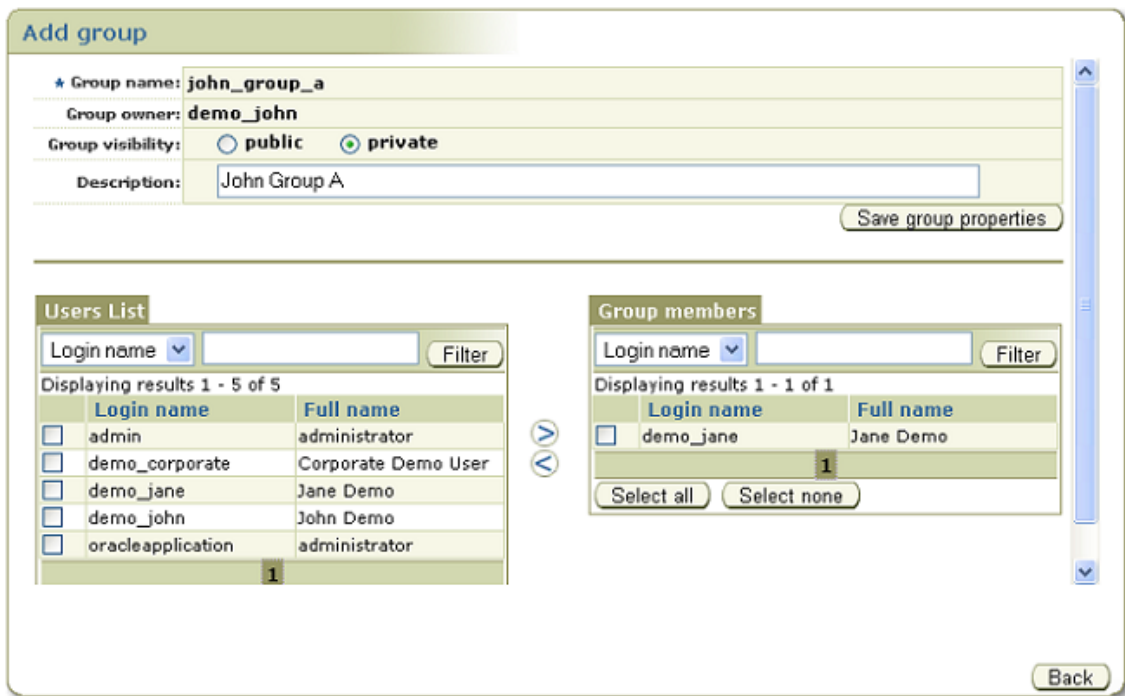


Figura 29 – Edição de relacionamento de grupos e usuários

O usuário pode definir quais são suas taxonomias favoritas. Taxonomias favoritas ajudam na pesquisa e categorização de entidades UDDI. A Figura 30 apresenta a tela de gerência de taxonomias favoritas. À esquerda são listadas todas as taxonomias existentes, enquanto que à direita estão as taxonomias favoritas escolhidas utilizando os botões “>” e “<”.



Figura 30 – Tela de gerência de taxonomias favoritas

Maiores detalhes sobre o gerenciamento de perfís de usuários e grupos no OSR podem ser obtidos em Apêndice 2 e Apêndice 3, respectivamente.

3.7.2.2 Navegação

No item de menu *Browse* pode-se realizar pesquisas por taxonomias (Figura 31). Um exemplo de pesquisa por filtro é apresentado na Figura 32.

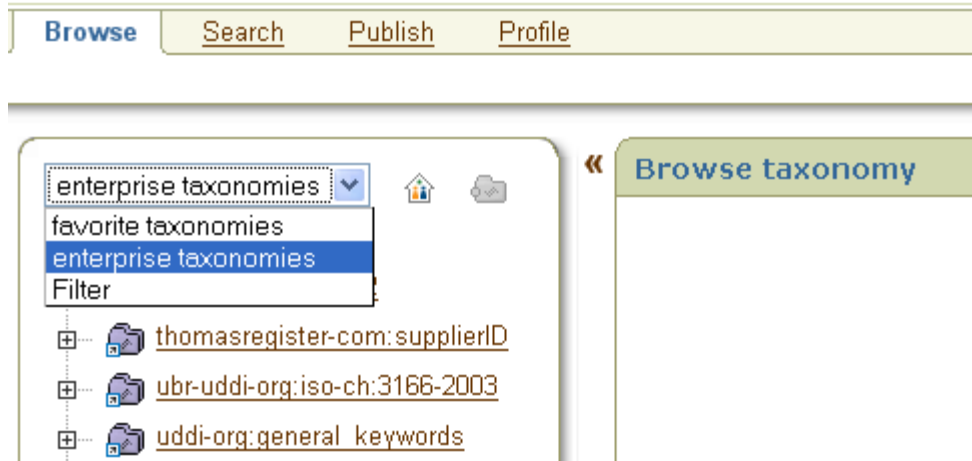


Figura 31 – Item de menu *Browse*



Figura 32 – Exemplo de pesquisa na taxonomia *demo:location:floor*

O número de taxonomias listadas pode ser reduzido através da definição de filtro (Figura 33).

[Browse](#) | [Search](#) | [Publish](#) | [Profile](#)

Name:

Type:
 categorization
 identifier

Compatibility:
 businessEntity
 businessService
 bindingTemplate
 tModel

Validation:
 all
 checked
 unchecked

Show system taxonomies:

[Browse](#) | [Filter](#) | [Query](#)

Figura 33 – Tela de definição de filtro

Também é possível combinar critérios de pesquisa em uma consulta. Para isto, utilize o botão “Add to query” apresentado na Figura 32. Dessa forma, é possível expandir uma nova taxonomia e especificar um novo critério. Um resultado de consulta é apresentado na Figura 34.

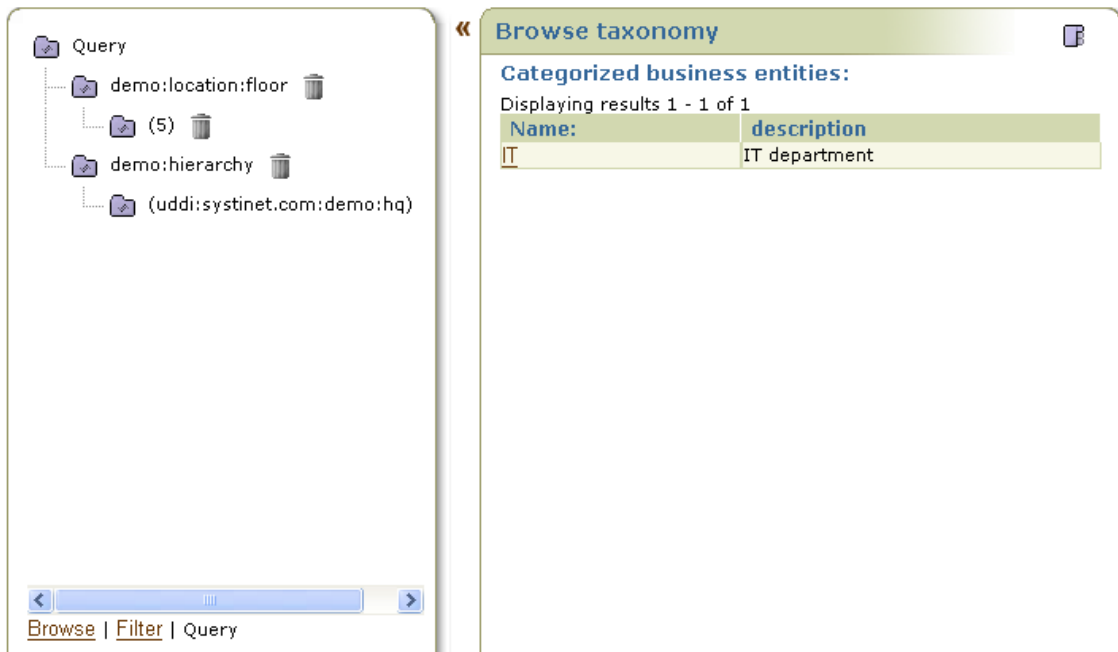


Figura 34 – Resultado de busca considerando dois critérios

3.7.2.3 Pesquisa

Utilizando o Registry Control, é possível pesquisar por:

- Estruturas de dados do UDDI: *BusinessEntity*, *BusinessService*, *BindingTemplate* e *tModel*.
- Acesso direto (*Direct get*): pode-se recuperar dados do OSR quando se sabe a chave da entidade UDDI que se deseja recuperar.
- Recursos: WSDL, XML, XSD e XSLT.

3.7.2.3.1 Pesquisa por *BusinessEntity*

BusinessEntity podem ser pesquisadas por nome, categorias, identificadores, URL de descoberta e *tModel*. Na Figura 35, estas opções foram destacadas à direita. Para todos os tipos de pesquisa, qualificadores podem ser especificados (Figura 35). Após a pesquisa, os detalhes do *BusinessEntity* retornado podem ser visualizados (Figura 36).

Find business

By qualifiers

Key combination	<input checked="" type="radio"/> default	<input type="radio"/> andAllKeys	<input type="radio"/> orAllKeys	<input type="radio"/> orLikeKeys
Sort by name	<input checked="" type="radio"/> default	<input type="radio"/> ascending	<input type="radio"/> descending	
Sort by date	<input checked="" type="radio"/> default	<input type="radio"/> ascending	<input type="radio"/> descending	
Category bag	<input checked="" type="radio"/> default	<input type="radio"/> combine	<input type="radio"/> service subset	<input type="radio"/> binding subset
Approximate match	<input type="radio"/> no	<input checked="" type="radio"/> yes		
Case insensitive match	<input type="radio"/> no	<input checked="" type="radio"/> yes		
Binary sort	<input checked="" type="radio"/> no	<input type="radio"/> yes		
Case insensitive sort	<input checked="" type="radio"/> no	<input type="radio"/> yes		
Signature present	<input checked="" type="radio"/> no	<input type="radio"/> yes		
Suppress projected services	<input checked="" type="radio"/> no	<input type="radio"/> yes		
Show only my entities	<input checked="" type="radio"/> no	<input type="radio"/> yes		

NAME

CATEGORIES

IDENTIFIERS

DISCOVERY URLS

T-MODELS

FIND QUALIFIERS

Figura 35 – Tela de definição de qualificadores de pesquisa

Home » IT »

- Contacts »
- EmployeeList »
- Support »

View business 'IT'

Details

Business Key uddi:system.com:demo:it	
name:	language
IT	
description	language
IT department	

Operational Info

Authorized name	demo_john
Node ID	Oracle
UDDI v2 key	17c7aeb7-d413-3822-9338-8096e4747b47
Created	Apr 13, 2006 5:43:23 PM
Last modified	Apr 13, 2006 5:43:30 PM
Last modified (incl. children)	Apr 13, 2006 5:43:30 PM

Not signed

Back View as XML Delete Edit

DETAILS

SERVICES

CONTACTS

CATEGORIES

IDENTIFIERS

DISCOVERY URLS

PERMISSIONS

RELATIONSHIPS

Figura 36 – Tela de detalhes do *BusinessEntity*

Na busca por categoria (Figura 37), mais de uma categoria pode ser adicionada, utilizando o botão *Add category*. Pode-se pesquisar também por um intervalo de valores (Figura 38).

Find business

By categories

Quantifier	tModel key	Key name
	Taxonomy list	
	demo:location:floor	
Quantifier:	exists	
Key name:		
★ Key value:	1	
Operator:		

Figura 37 – Busca por *BusinessEntity* por categoria

Find business

By categories

Quantifier	tModel key	Key name	Operator	KeyValue
exists	demo:location:floor		>	1

Figura 38 – Busca por *BusinessEntity* por intervalo de categoria

Na busca por identificadores (Figura 39), pode-se adicionar mais de um identificador na pesquisa.

Find business

By identifiers

tModel key	Key name
Taxonomy list	
demo:departmentID	
Key name:	
★ Key value:	002

Figura 39 - Busca por *BusinessEntity* por identificadores

Na busca por URL de descoberta, deve-se clicar na opção “*Discovered URLs*” (Figura 35), digitar a URL de pesquisa, na tela que se abre, e clicar no botão *Find*.

3.7.2.3.2 Pesquisa por *BusinessService*, *BindingTemplate* e *tModel*

A pesquisa por *BusinessService*, *BindingTemplate* e *tModel* é semelhante a pesquisa por *BusinessEntity*, descrita na seção anterior.

3.7.2.3.3 Pesquisa por acesso direto

A pesquisa por acesso direto (Figura 40) permite especificar chaves na versão 2 ou 3 de UDDI.

The screenshot shows a web interface titled "Direct get". It features a header with the text "Enter key value in UDDI version 3 format:". Below this, there are four rows of input fields, each with a "View" button: "Business key" with "View business", "Service key" with "View service", "Binding key" with "View binding", and "tModel key" with "View tModel". At the bottom, there is a section for "WSDL location" with four input fields: "WSDL location:", "tModel key:", "Service key:", and "Binding key:", followed by a "Get WSDL" button. On the right side, there are two search icons: "V3 FIND BY U3" and "V2 FIND BY U2".

Figura 40 – Tela de pesquisa por acesso direto

Também é possível obter o formato XML de *BusinessEntity*, *BusinessServices*, *BindingTemplates* e *tModels* para uso em processamento automatizado pela digitação da chave da estrutura em uma URI. A forma da URI é:

```
http://<hostname>:<port>/<context>/uddi/web/directGetXml?<structureKey>=<key>
```

Exemplos na versão 3 do UDDI são apresentados a seguir:

- <http://localhost:8888/registry/uddi/web/directGetXml?businessKey=uddi:systinet.com:uddinodebusinessKey>
- <http://localhost:8888/registry/uddi/web/directGetXml?serviceKey=...>
- <http://localhost:8888/registry/uddi/web/directGetXml?bindingKey=...>
- <http://localhost:8888/registry/uddi/web/directGetXml?tModelKey=...>

3.7.2.3.4 Pesquisa por WSDL, XML, XSD e XSLT

Telas de pesquisa por WSDL, XML, XSD e XSLT são apresentadas nas Figura 41, Figura 42, Figura 43 e Figura 44, respectivamente.

Find WSDL Document

You can either find all WSDL documents published in Oracle Registry or find a particular one by providing its URI location and then examine how its components are published in Oracle Registry.

Use below button to find all business services representing WSDL documents published in Oracle Registry

Find all published WSDLs

Enter location of WSDL document to see how it is published in Oracle Registry

★ WSDL location (URI) Examine this WSDL

Cancel

Search Result

Figura 41 – Tela de pesquisa por WSDL

Find XML resource

Enter one of following fields (use % as wildcard). You can also leave all fields blank to search for all XML resources published in the Oracle Registry.

Location (URI)

Enter XML schema that the searched XML shall be compatible with.

XML schema

Find Cancel

Search Result

Figura 42 – Tela de pesquisa por XML

Find XSD resource

Enter one of following fields (use % as wildcard). You can also leave all fields blank to search for all XSD resources published in the Oracle Registry.

Location (URI)

Enter namespace used in the searched XSD.

Namespace

Enter toplevel type defined in the searched XSD.

Defined type

Enter toplevel element defined in the searched XSD.

Defined element

Figura 43 – Tela de pesquisa por XSD

Find XSLT resource

Enter one of following fields (use % as wildcard). You can also leave all fields blank to search for all XSLT resources published in the Oracle Registry.

Find XSL transformation

Location

Enter XML schema that the searched XSLT shall be able to process.

	Type	Value		
Input XML schema	<input type="radio"/> Key	<input type="text"/>	*	<input type="button" value="✓"/>
	<input checked="" type="radio"/> Namespace	<input type="text" value="Select XML schema"/>		<input type="button" value="✗"/>

Enter XML schema that the documents generated by the searched XSLT shall be compatible with.

Output XML schema

[Search](#) [Result](#)

Figura 44 – Tela de pesquisa por XSLT

3.8 Controle de acesso

Uma das formas de especificar controle de acesso no OSR é utilizar permissão de acesso a dados – ACL (Access Control List). Uma ACL é composta de:

- Parte: usuário ou grupo de usuários
- Estrutura principal: uma das principais estruturas de dados do UDDI: *businessEntity*, *businessService*, *bindingTemplate* ou *tModel*.

- Ação: uma operação "find", "get", "save", ou "delete" sobre a entidade, e uma ação especial "create", que indica que podem ser realizadas alterações em sub-entidades. Por exemplo, o usuário com a permissão *create* em um *businessService* pode salvar novos *bindingTemplates* abaixo do *businessService*, mas não pode atualizar o *businessService* como um todo. Esta operação só faz sentido para *businessEntity* e *businessService* pois *bindingTemplates* e *tModels* não têm sub-entidades.

ACL sobrepõem o controle de acesso padrão do UDDI, pois, segundo o padrão, apenas o proprietário da estrutura principal do UDDI pode atualizar ou removê-la. Cada usuário pode procurar ou obter a estrutura.

Quando uma permissão não é explicitamente definida para as operações *find/get*, qualquer usuário pode executá-las. Quando nenhuma permissão está definida explicitamente para as ações *save/delete* em uma entidade, apenas o proprietário da entidade pode salvá-la ou apagá-la. Este é o funcionamento do controle de acesso padrão do UDDI. Quando uma permissão explícita é ajustada para uma ação, um controle de acesso totalmente diferente é utilizado, seguindo as seguintes regras.

- Proprietário sempre tem controle total.
- Uma permissão negativa para usuário sobrescreve uma permissão positiva para usuário se forem definidas para o mesmo objeto. O mesmo ocorre para o caso de grupo.
- Permissão para usuário tem mais peso do que permissão para grupo.
- O proprietário de uma entidade pode sempre executar *get_XXX* sobre uma sub-entidade direta.
- Permissões positivas para *delete* e *save* são herdadas de entidades pais e sobrescrevem permissões negativas em sub-entidades.
- Para realizar atualização com a operação *save_XXX*, é necessário ter ambas as permissões *save* e *get*.

Maiores detalhes sobre segurança no OSR são encontrados em [Oracle, 2008a] e no Apêndice 7 é apresentado um roteiro para controlar o acesso a um serviço de negócio no Oracle Service Registry.

3.9 Taxonomia

Uma taxonomia, ou conjunto de valores na terminologia da especificação UDDI, é um *tModel* que pode ser usado em *categoryBags*, *identifierBags*, ou *Publisher Assertions*. Este *tModel* deve estar em um formato específico para que o OSR possa reconhecê-lo como uma taxonomia. O *tModel* deve ser categorizado com o tipo de taxonomia e, opcionalmente, com informação a respeito se e como validar os valores em *keyedReferences*. Por exemplo, números de taxa US ou ISBN só fazem sentido se sabe-se que eles são número de taxas dos Estados Unidos ou ISBNs.

Categorizações é um dos conceitos mais importantes em um UDDI. Qualquer entidade UDDI pode estar associada com uma categoria que é definida em alguma taxonomia. Uma categoria é essencialmente um par composto por nome e valor e uma taxonomia é um conjunto organizado hierarquicamente de categorias. Categorias podem ser usadas pra pesquisar por entidades do UDDI, como por exemplo, a localização de serviços de negócio (Apêndice 12). Existem categorias criadas

previamente no UDDI. Além disso, existem taxonomias criadas por empresas que estão disponíveis para serem utilizadas, tais como, taxonomia para mecanismos de autenticação segura, taxonomia para tipo de componente distribuído etc. O registro UDDI permite também a definição de taxonomias customizadas construídas especificamente para organizações ou aplicações [Oracle, 2008b].

A especificação UDDI distingue quatro tipos de taxonomias:

- **Categorizações:** podem ser usadas em todas as quatro estruturas do UDDI. Elas são usadas para marcar as estruturas com informações adicionais, tais como identidade, localização, e o que mais a taxonomia descrever.
- **Grupos de categorização:** agrupam várias categorizações em uma categorização lógica. Por exemplo, uma localização geográfica incluindo duas categorizações: longitude e latitude.
- **Identificadores:** usados em *businessEntity* e *tModels* referenciam informações públicas.
- **Relacionamentos:** usados apenas em *Publisher Assertions*, definem o relacionamento entre duas *businessEntities*.

O Apêndice 8 apresenta um Roteiro para criar taxonomias no Oracle Service Registry.

3.10 Mapeamento de recursos

Esta seção descreve o mapeamento dos seguintes recursos para um UDDI: WSDL, XML, XSD e XSLT.

3.10.1 Mapeamento de WSDL

O mapeamento de elementos de um WSDL (**Tabela 1**) para um UDDI ocorre como esquematizado na Figura 46 e detalhado a seguir.

Tabela 1 – Elementos de um WSDL

Elemento	Descrição
<types>	Definições de tipos para o conteúdo das mensagens.
<message>	Definição abstrata do tipo de dados sendo transmitido. Uma mensagem consiste de partes, as quais descrevem o conteúdo lógico e abstrato de uma mensagem.
<portType>	Coleção abstrata de operações suportadas pelo serviço.
<operation>	Descrição abstrata de uma ação suportada pelo serviço. Cada operação refere-se a uma mensagem de entrada e mensagem de saída.
<binding>	Define o protocolo concreto e especificações de formato de dados para as operações e mensagens definidas por um determinado <i>portType</i> .
<port>	Especifica um endereço para um <i>binding</i> , portanto definindo um <i>endpoint</i> de comunicação único.
<service>	Coleção de portas ou <i>endpoints</i> .

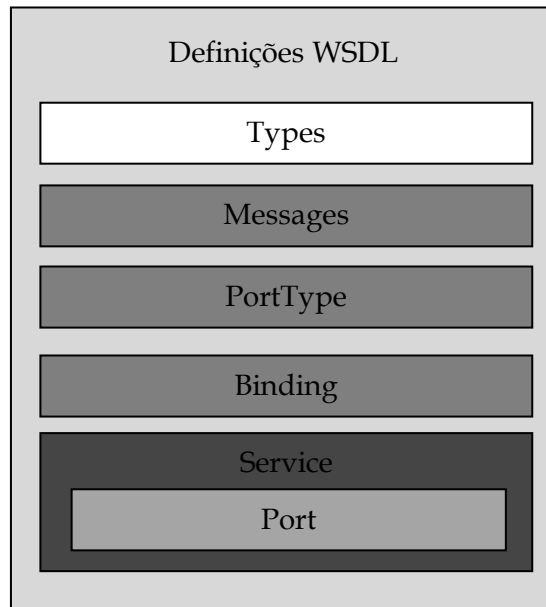


Figura 45 – Elementos de um WSDL

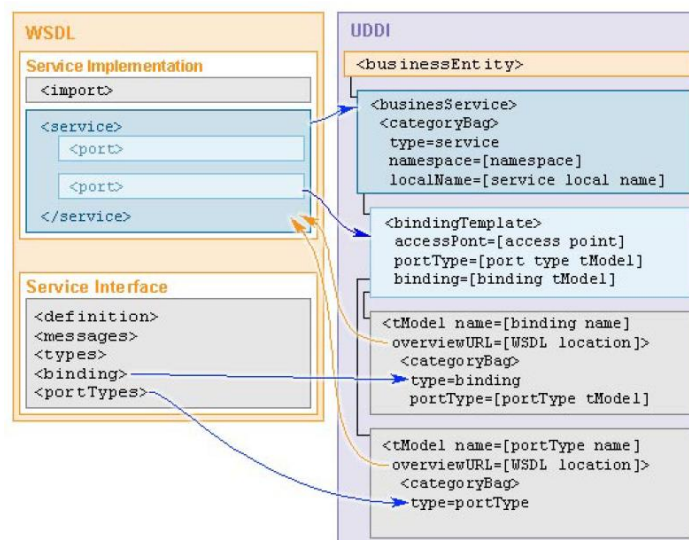


Figura 46 – Esquema de mapeamento de WSDL para o UDDI

Cada *portType* do WSDL é mapeamento em um *tModel* como apresentado na Tabela 2.

Tabela 2 – Mapeamento de *portType* para o UDDI

WSDL	UDDI
<i>portType</i>	<i>tModel</i> (categorizado como <i>portType</i>)
Namespace do <i>portType</i>	<i>keyedReference</i> no <i>categoryBag</i>
Nome local do <i>portType</i>	Nome do <i>tModel</i>
Localização do WSDL	<i>overviewURL</i>

Cada *binding* do WSDL é mapeamento em um *tModel* como apresentado na Tabela 3.

Tabela 3 – Mapeamento de *binding* para o UDDI

WSDL	UDDI
<i>Binding</i>	<i>tModel</i> (categorizado como <i>binding</i> e <i>wsdlSpec</i>)

<i>Namespace do binding</i>	<i>keyedReference no categoryBag</i>
Local name do <i>binding</i>	Nome do <i>tModel</i>
Localização do WSDL	<i>overviewURL</i>
<i>portType binding</i>	<i>keyedReference no categoryBag</i>
<i>Protocol</i>	<i>keyedReference no categoryBag</i>
<i>Transport</i>	<i>keyedReference no categoryBag</i>

A seção *service* é representada como *businessService* do UDDI, como apresentado na **Tabela 4**. O *businessService* também contém um *bindingTemplate* contendo:

- O tipo de acesso é categorizado como um ponto de acesso do serviço.
- O *portType* é categorizado como o *tModel* do *portType*.
- O *binding* é categorizado como o *tModel* da informação de *binding*.
- O nome local é categorizado como o nome local do *port*.

Tabela 4 – Mapeamento da seção *service* para o UDDI

WSDL	UDDI
<i>Service</i>	<i>businessService</i> (categorizado como <i>service</i>)
<i>Namespace do service</i>	<i>keyedReference no categoryBag</i>
Nome local do <i>service</i>	<i>keyedReference no categoryBag</i> ; opcionalmente nome usado do <i>service</i>

No Oracle Service Registry, existem funcionalidades para publicar um WSDL, definir uma nova forma de mapeamento de WSDL, pesquisar por um WSDL pela localização do WSDL, desregistrar ou republicar um WSDL.

O mapeamento captura informação suficiente dos documentos WSDL para permitir consultas precisas por *web services* sem ter que acessar os documentos WSDL fontes e permitir que os documentos WSDL apropriados sejam recuperados uma vez que uma combinação de busca tenha sido alcançada.

3.10.2 Mapeamento de arquivo XML

Um arquivo XML é mapeado para um *tModel* seguindo o formato apresentado na Figura 47. A localização do arquivo XML é adicionada ao elemento *overviewURL* do *tModel*. *Namespaces* são mapeados para *keyedReferences* no *categoryBag* do *tModel*. Cada *namespace* é mapeado para um *tModel*.

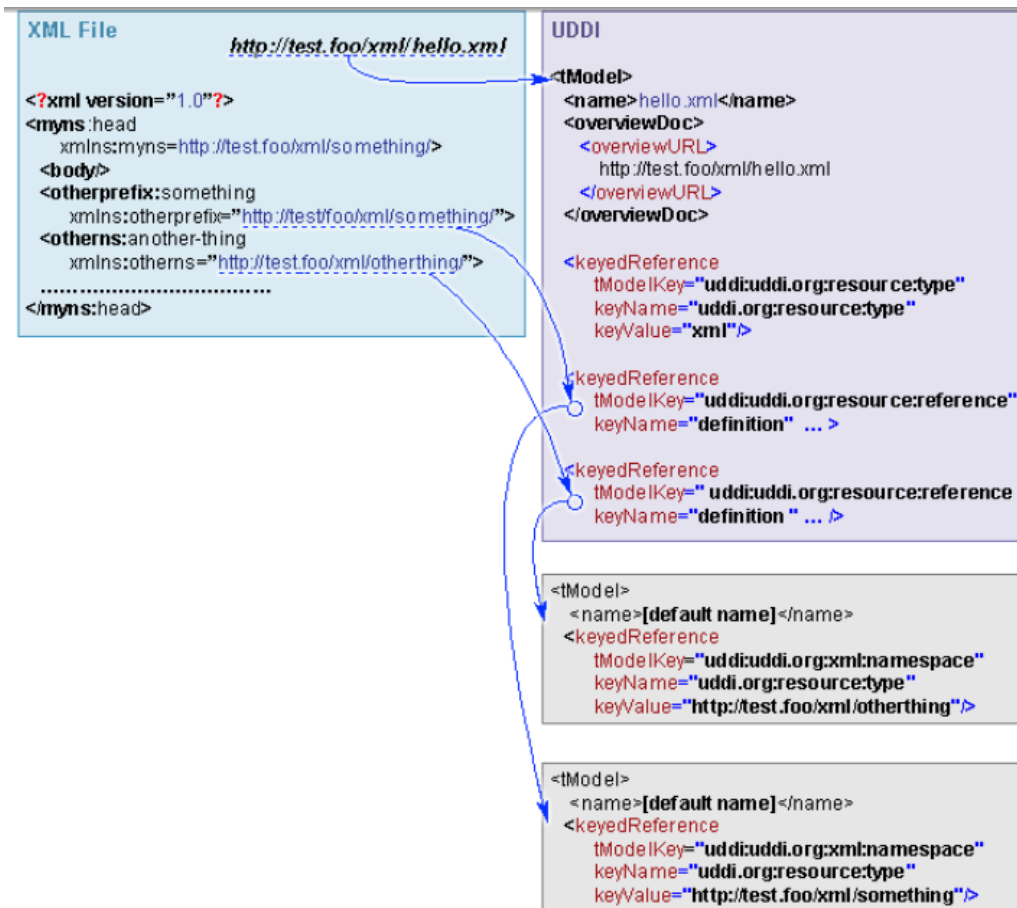


Figura 47 – Mapeamento de XML para UDDI

O OSR permite publicar um documento XML e também é possível definir a forma de mapeamento de documento XML. Além disso é possível desregistrar ou republicar um arquivo XML. O OSR permite pesquisar por um documento XML das seguintes maneiras:

- Pesquisar por um arquivo XML contendo dados de um determinado tipo (XSD).
- Pesquisar por um arquivo XML de uma pasta ou servidor específico, usando um critério de busca, prefixo URI e caracteres especiais.
- Pesquisar por um arquivo XML que é entrada ou saída de um XSLT específico.
- Pesquisar por um processador de um arquivo XML de entrada específico.

3.10.3 Mapeamento de arquivo XSD

Um arquivo XSD é mapeado para um *tModel* como apresentado na Figura 48. A URI de localização do arquivo XSD é colocado no elemento *overviewURL* do *tModel* e o *target namespace* é mapeado para um *keyedReference* no *categoryBag* do *tModel*. Os *xsd:types*, *xsd:elements* e *xsd:imports* são mapeados para *keyedReferences* no *tModel*. Para cada tipo, elemento ou import, um novo *tModel* é criado.



Figura 48 – Mapeamento de XSD para o UDDI

O OSR permite publicar, desregistrar e republicar arquivos XML Schema. Além disso, ele permite especificar como artefatos do XML Schema devem ser mapeados para estruturas existentes no UDDI. Em relação à pesquisa, é permitido:

- Pesquisar por um XML Schema que importa artefatos declarados em um arquivo XSD específico.
- Pesquisar por XML Schema localizado em um servidor ou pasta específica.
- Pesquisar por todos os XSLT que podem processar documentos usando um XSD específico.
- Pesquisar por todos os XSLT que produzem documentos que usam um XSD específico.

3.10.4 Mapeamento de arquivo XSLT

O mapeamento de XSLT para UDDI é apresentado na Figura 49. A URI de localização do arquivo XSLT é adicionada ao elemento `overviewURL` do `tModel`. `Namespaces` são mapeados em `keyedReferences` no `categoryBag` do `tModel`. Elementos `xsl:import` são também mapeados para `keyedReferences` no `categoryBag` do `tModel`. Para cada `import` e `namespace`, um novo `tModel` é criado.



Figura 49 – Mapeamento de XSLT para o UDDI

O OSR permite publicar um XSLT, bem como especificar como artefatos de um arquivo XSLT serão mapeados para estruturas existentes no UDDI.

3.11 Desenvolvimento do lado do cliente e do servidor

Do lado do cliente, o desenvolvimento pode utilizar:

- APIs do UDDI para pesquisar e publicar informações no registro.
- APIs avançadas para validação, taxonomias, categorias, aprovação, utilitários de administração, replicação, estatísticas, pesquisa, subscrição e publicação de recursos.
- APIs de segurança para tratar contas, grupos e permissões.

Além disso, é possível preparar um cliente para distribuição para acessar o OSR via interface SOAP e criar um cliente que se autentica utilizando HTTP básico. Detalhes a respeito destas funcionalidades são apresentados em [Oracle, 2008a].

O manual [Oracle, 2008a] detalha atividades que podem ser realizadas do lado do servidor, tais como:

- Acesso de APIs de *backend* via *servlet* instalado no servidor de aplicação.
- Módulos de customização do OSR
- Uso de *interceptors* para monitorar ou modificar requisições e respostas do OSR.
- Escrita de serviços de validação customizados

- Escrita de serviços de notificação de subscrição.
- Framework JSP.

3.12 Integração do OSR com OSB

O OSR e o OSB (Oracle Service Bus [Azevedo, 2009]) têm forte integração pelo fato de que serviços registrados no OSR podem ser automaticamente pesquisados e consumidos a partir do OSB *console*. OSB automaticamente cria serviços de negócio e artefatos dependentes quando consome serviços do OSR. Por exemplo, pela definição de fluxos dentro do OSB utilizando os serviços registrados no OSR. De maneira análoga, serviços de *proxy* podem ser exportados do OSB para o OSR para registro automatizado de serviços de *proxy* no OSR. Isto facilita a localização dos serviços de *proxy* pelos usuários e taxonomias específicas do OSB podem ser utilizadas no registro.

Não se deve confundir o registro do OSB com o OSR. O registro do OSB é um mecanismo de pesquisa local apenas. Ou seja, ele permite pesquisar e usar serviços internamente. Ele não é um registro UDDI nem tem a intenção se de expor ativos no OSB para uso fora do OSB.

Já o OSR é um registro no nível da empresa que permite qualquer consumidor pesquisar e navegar por serviços armazenados no registro. O OSR também provê governança com *workflow* e aprovações para gerenciar o ciclo de vida de serviços.

O roteamento de serviços utilizando OSB e OSR é ilustrado na Figura 50, e inclui os seguintes passos:

1. Consumidor do serviço envia requisição juntamente com metadados do contrato.
2. OSB invoca OSR para localizar *endpoint* que esteja de acordo com a definição do contrato.
3. OSR envia *endpoint* apropriado para o OSB.
4. OSB roteia requisição para o *endpoint* retornado.
5. Provedor processa a requisição e envia a resposta para o OSB.
6. OSB roteia resposta para o consumidor do serviço.

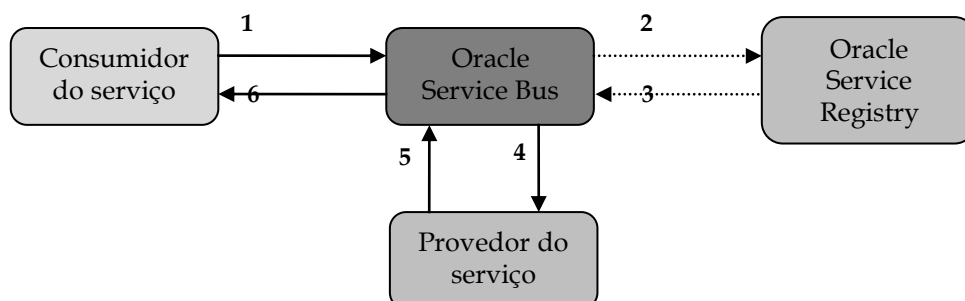


Figura 50 – Roteamento dinâmico considerando OSB e OSR

4 Conclusões

As especificações UDDI formam a fundamentação técnica necessária para publicação e descoberta de implementações de web services [Bellwood *et al.*, 2004]. Ele atua em conjunto com o Enterprise Service Bus, sendo também uma das ferramentas mais importantes para implantação de SOA em uma organização. Dessa forma, o correto entendimento da especificação UDDI e de ferramenta de apoio é intrínseco para o sucesso de SOA.

Este relatório apresentou as principais características da especificação UDDI versão 3.0 e apresentou um estudo do OSR (Oracle Service Registry), ferramenta da Oracle que implementa a especificação.

5 Referências

- AZEVEDO, L.G., SOUSA, H.P., SANTORO, F., BAIÃO, F., Estudos de ESB e análise do OSB. Relatório técnico Petrobras/GDIEP, 2009. (a ser publicado)
- BELLWOOD *et al.*, T., UDDI Version 3.0.2. UDDI Spec Technical Committee Draft, 2004. Disponível em <<http://www.oasis-open.org/committees/uddi-spec/doc/spec/v3/uddi-v3.0.2-20041019.htm>>. Acessado em 27 Nov. 2009.
- CANONICALIZATION, Exclusive XML Canonicalization Version 1.0, 2002. Disponível em <<http://www.w3.org/TR/2002/REC-xml-exc-c14n-20020718/>>. Acessado em 25 Nov. 2009.
- COLGRAVE, J., JANUSZEWSKI, K., Using WSDL in a UDDI Registry, Version 1.08. OASIS UDDI Spec TC, 2002.
- ORACLE. Oracle Fusion Middleware Service Registry 10g Release 3 (10.3) E13665-01. Oracle Corporation, 2008a. Disponível em <<http://download.oracle.com/otndocs/tech/soa/OSR103ProductDocumentation.pdf>>. Acessado em 25 Nov. 2009.
- ORACLE. BEA Aqualogic Registry Repository 3: Create an It Ecosystem - Student Guide. Oracle Corporation, vol. II, 2008b.
- OASIS, OASIS UDDI Specifications TC - Committee Best Practices, Advancing Open Source Standards for Information Society (OASIS). Disponível em <<http://www.oasis-open.org/committees/uddi-spec/doc/bps.htm>>. Acesso em 27 Nov. 2009b.
- OASIS, OASIS UDDI Specifications TC - Committee Technical Notes, Advancing Open Source Standards for Information Society (OASIS). Disponível em <<http://www.oasis-open.org/committees/uddi-spec/doc/tns.htm>>. Acesso em 27 Nov. 2009b.
- ORACLE. Oracle Internet Application Server. Oracle Corporation. Disponível em <<http://www.oracle.com/technology/products/ias/index.html>>. Acessado em 25 Nov. 2009a.
- ORACLE. Oracle Containers for J2EE (OC4J). Oracle Corporation. Disponível em <<http://www.oracle.com/technology/tech/java/oc4j/1013/index.html>>. Acessado em 25 Nov. 2009b.

- UDDI, Universal Description, Discovery & Integration Specification version 3, 2004. Disponível em <<http://www.oasis-open.org/committees/uddi-spec/doc/tcspecs.htm#uddiv3>>. Acessado em 25 Nov. 2009.
- WSDL, Especificação da Web Service Description Language, 2008. Disponível em <<http://www.w3.org/TR/wsdl>>. Acessado em 25 Nov. 2008.
- WSDL-UDDI, Using WSDL in a UDDI Registry, version 2.0.2, Technical Note, 2004. Disponível em <<http://www.oasis-open.org/committees/uddi-spec/doc/tn/uddi-spec-tc-tn-wsdl-v2.htm>>. Acessado em 25 Nov. 2004.

Apêndice 1 – Roteiro de Instalação do Oracle Service Registry versão 10.3

Esta seção descreve, de forma resumida, os passos para instalação do Oracle Service Registry, no cenário *Standalone Registry*. Informações detalhadas sobre a instalação do produto, inclusive em outros cenários possíveis, podem ser obtidas em [Oracle, 2008a].

Os passos a seguir foram instalados no seguinte ambiente operacional:

- ✓ Windows XP Professional versão 2002 SP3
- ✓ Java Development Kit versão 1.6.0_13-b03
- ✓ Oracle WebLogic Server 10.3
- ✓ Banco de Dados Oracle versão 10g Release 1 (10.1)

1. Para iniciar o processo de instalação, digite o seguinte comando no *prompt* do Windows:

```
java -jar oracle-service-registry-10.3.jar
```

2. A primeira tela do processo de instalação apresenta informações sobre os requisitos de sistema recomendados para a instalação, inclusive possibilita visualizar a documentação do produto (Figura 51).

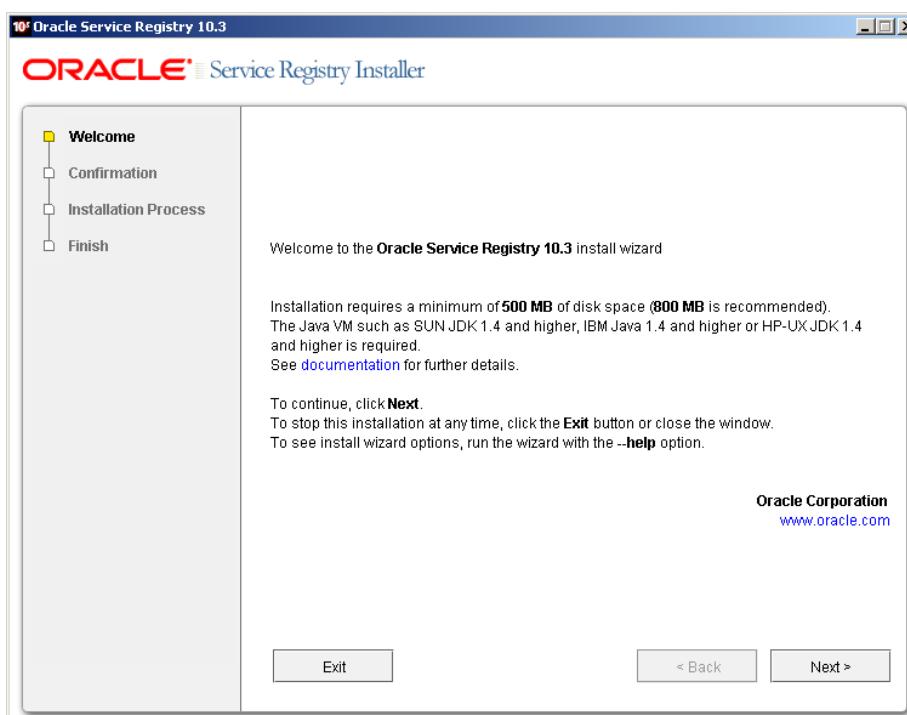


Figura 51 – Tela de boas vindas da instalação do OSR

3. Selecione o cenário de instalação desejado, neste caso *Standalone Registry* (Figura 52).

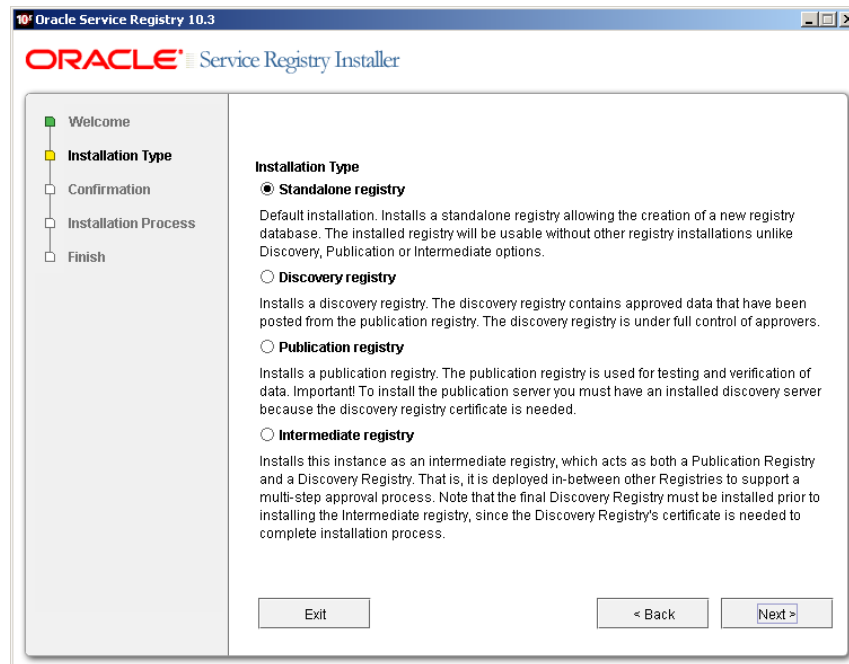


Figura 52 – Tela de seleção do tipo de instalação

4. Informe o caminho do diretório de instalação do OSR (o default para Windows é `c:\oracle\registry_10_3`)(Figura 53).

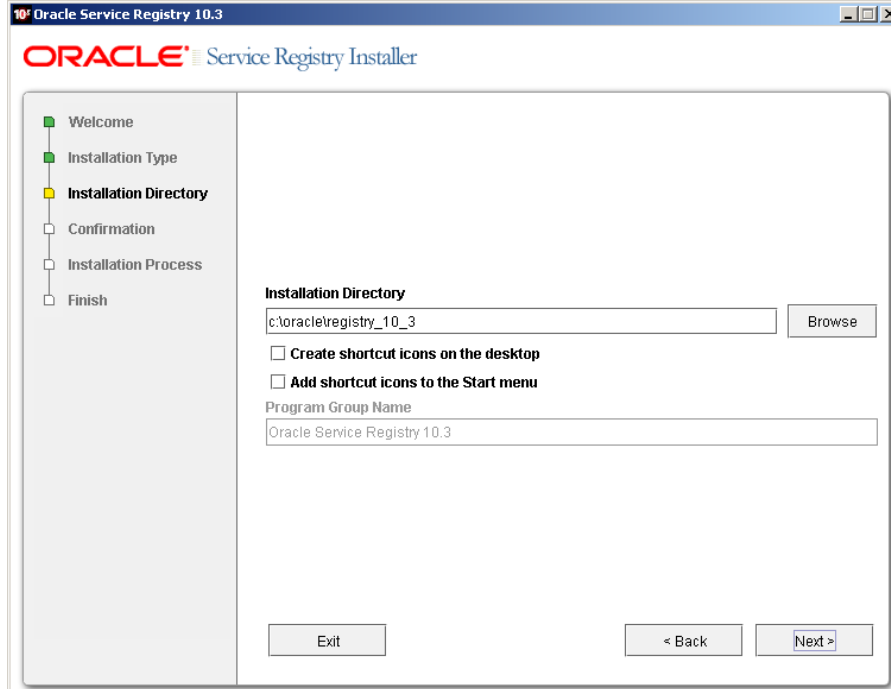


Figura 53 – Tela de seleção do diretório de instalação

5. Informe as configurações SMTP. Elas são importantes para o envio de notificações, como por exemplo, mudanças nos serviços assinados (*subscriptions*)(Figura 54).

The screenshot shows the 'Oracle Service Registry 10.3' window with the 'Service Registry Setup' title. On the left, a navigation pane shows the progress: Oracle Home, Setup, Registry (highlighted), Confirmation, Installation Process, and Finish. The main area is titled 'SMTP Configuration' and contains the instruction 'Enter the registry configuration properties.' Below this are several input fields: 'SMTP Hostname' (placeholder: [SMTP server hostname]), 'SMTP Port' (value: 25), 'SMTP Account Name', 'SMTP Password', 'Confirm Password', 'SMTP Default Sender E-mail', and 'SMTP Default Sender Name'. At the bottom, there are three buttons: 'Exit', '< Back', and 'Next >'.

Figura 54 – Tela de configurações SMTP

6. Defina a conta de administração do OSR (Figura 55)

The screenshot shows the 'Oracle Service Registry 10.3' window with the 'Service Registry Installer' title. On the left, a navigation pane shows the progress: Welcome, Installation Type, Installation Directory, Registry, Administrator (highlighted), Confirmation, Installation Process, and Finish. The main area is titled 'Administrator Account Configuration' and contains the instruction 'Specify Oracle Service Registry administrator account.' Below this are four input fields: 'Administrator Username' (value: admin), 'Administrator Password' (masked with dots), 'Confirm Password' (masked with dots), and 'Administrator Email' (value: flavio.faria@uniriotec.br). At the bottom, there are three buttons: 'Exit', '< Back', and 'Next >'.

Figura 55 – Tela de criação da conta de administração

7. Configuração do banco de dados

- a. Utilize a opção *Create New Tablespace/database* para criar o banco de dados. É necessário existir um *schema* criado e possuir acesso de administrador no banco de dados (Figura 56).

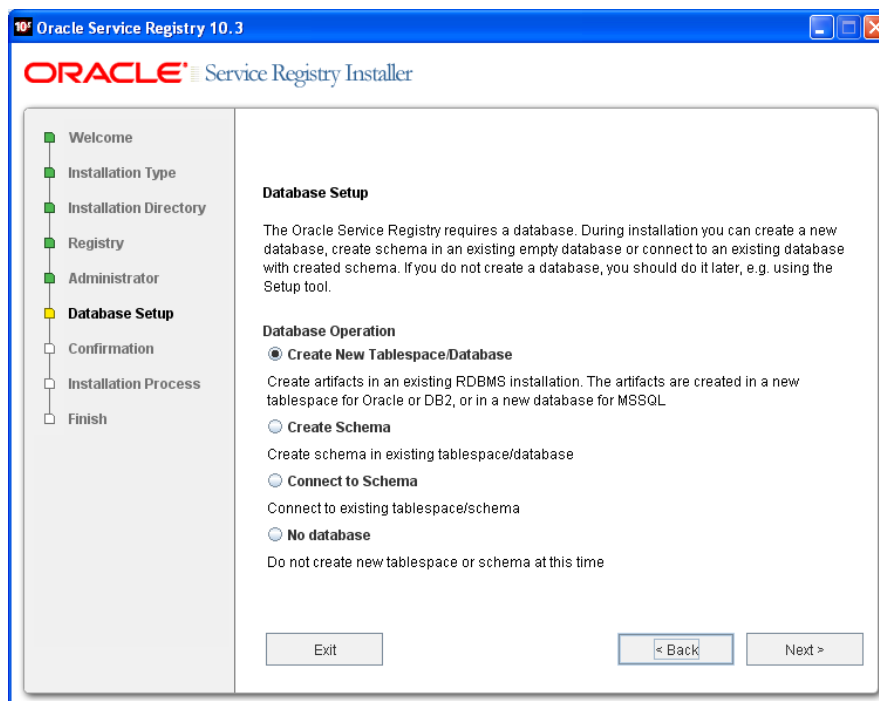


Figura 56 – Tela de seleção do método de criação do banco de dados

- b. Selecione o banco de dados utilizado. Marque a opção *Install demo data to the database* para que sejam instalados os exemplos do OSR (Figura 57).

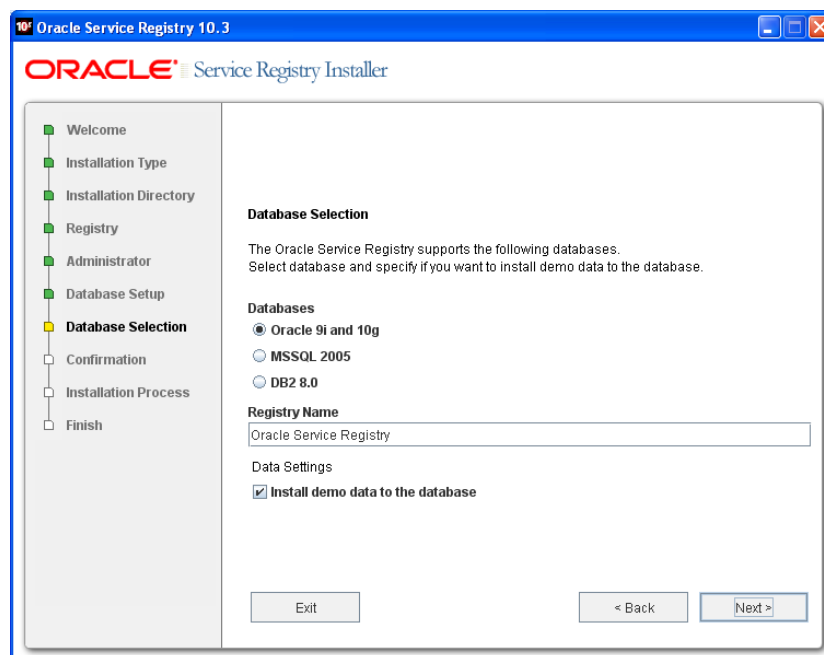


Figura 57 – Tela de seleção do banco de dados

- c. Forneça as informações do servidor de banco de dados que será utilizado pela instalação do OSR (Figura 58).

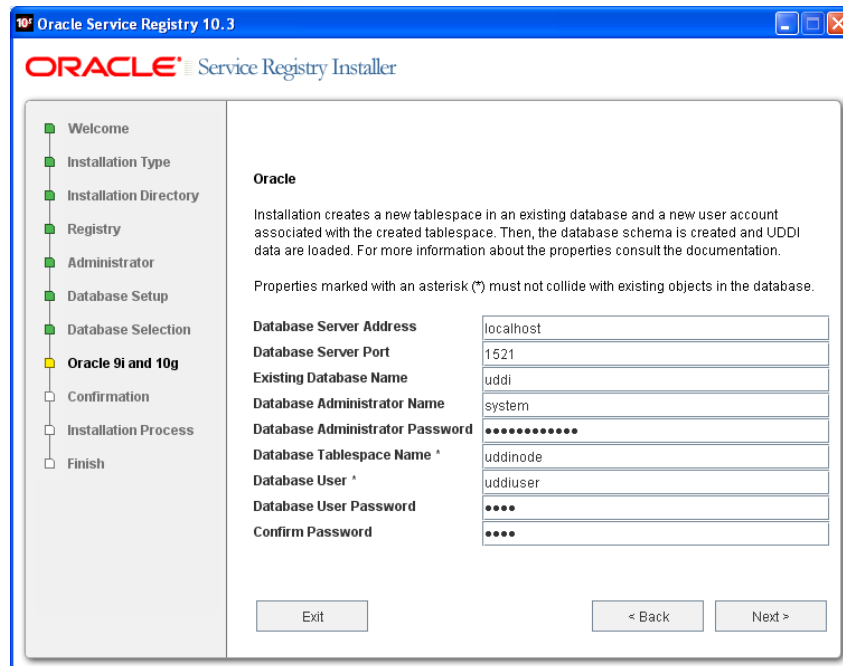


Figura 58 – Tela de configurações do banco de dados

8. Informe o diretório do driver JDBC que será utilizado pelo OSR para conectar o banco de dados. O driver pode ser encontrado no diretório de instalação do Oracle 10g (Figura 59).

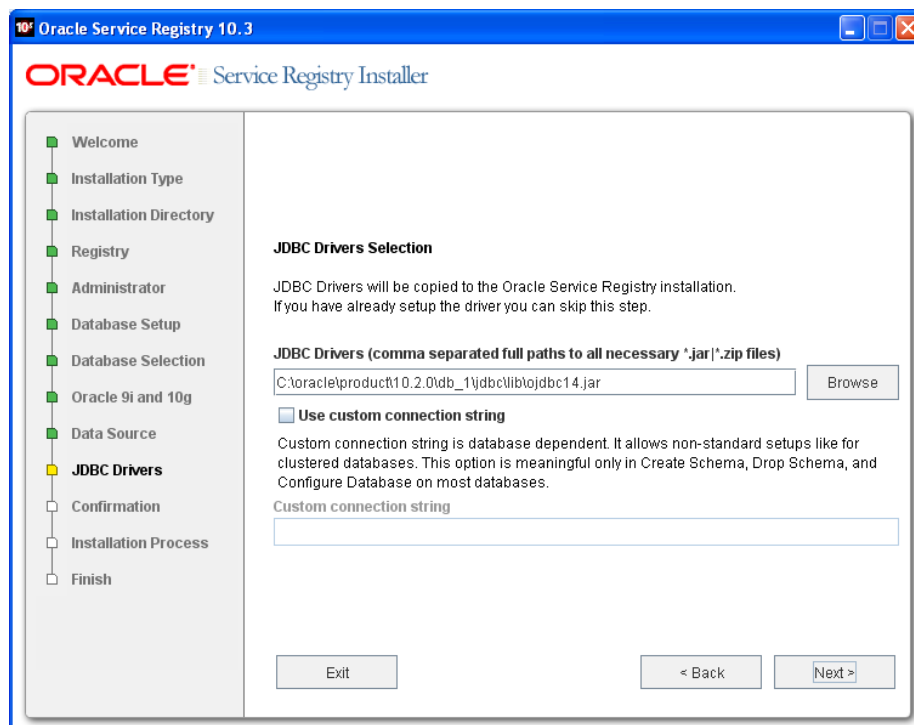


Figura 59 – Tela de seleção do driver JDBC

9. Selecione o provedor *Database* para autenticação a ser utilizado pelo OSR (Figura 60).

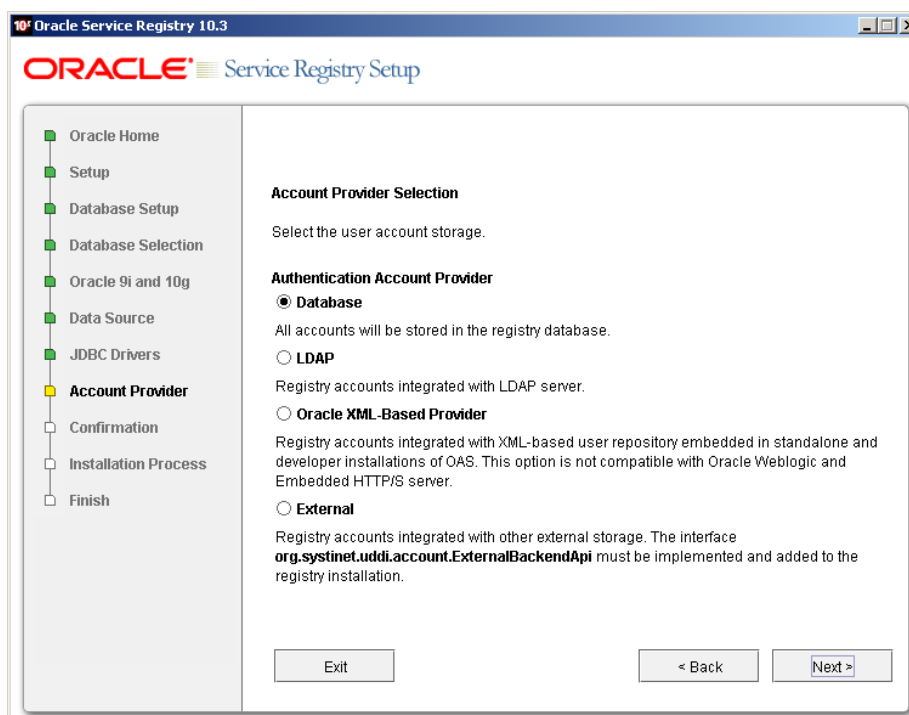


Figura 60 – Tela de seleção do provedor de autenticação

10. Selecione o servidor de aplicação:
a. Marque a opção *Oracle WebLogic 9.2 or 10.3* (Figura 61).

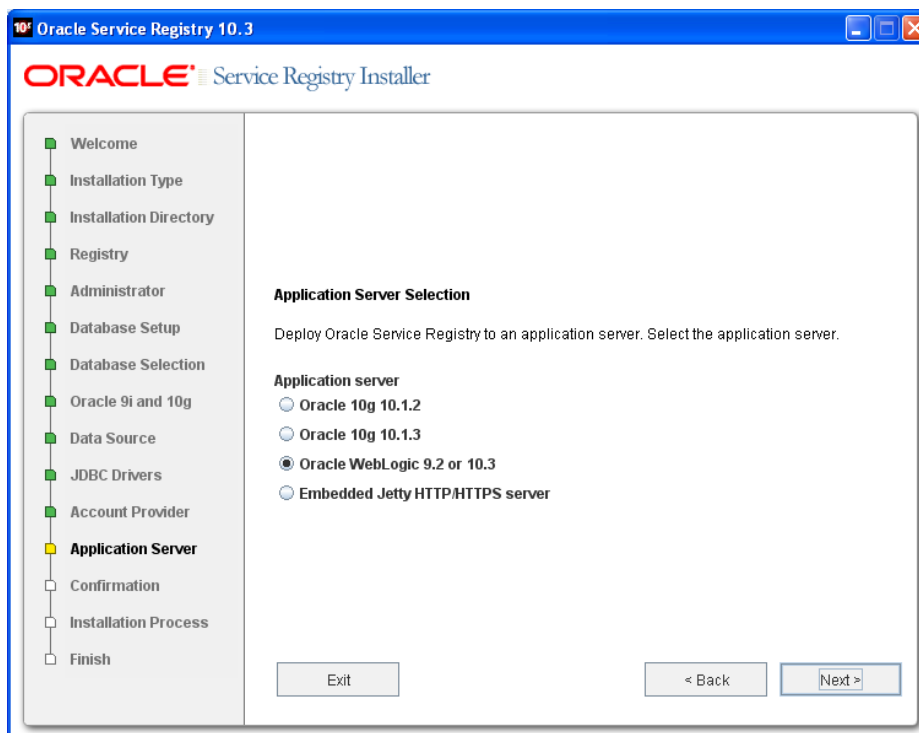


Figura 61 – Tela de seleção do servidor de aplicação

- b. Forneça as informações para deploy no servidor de aplicação e marque a opção *Deploy at the end of installation* (Figura 62).

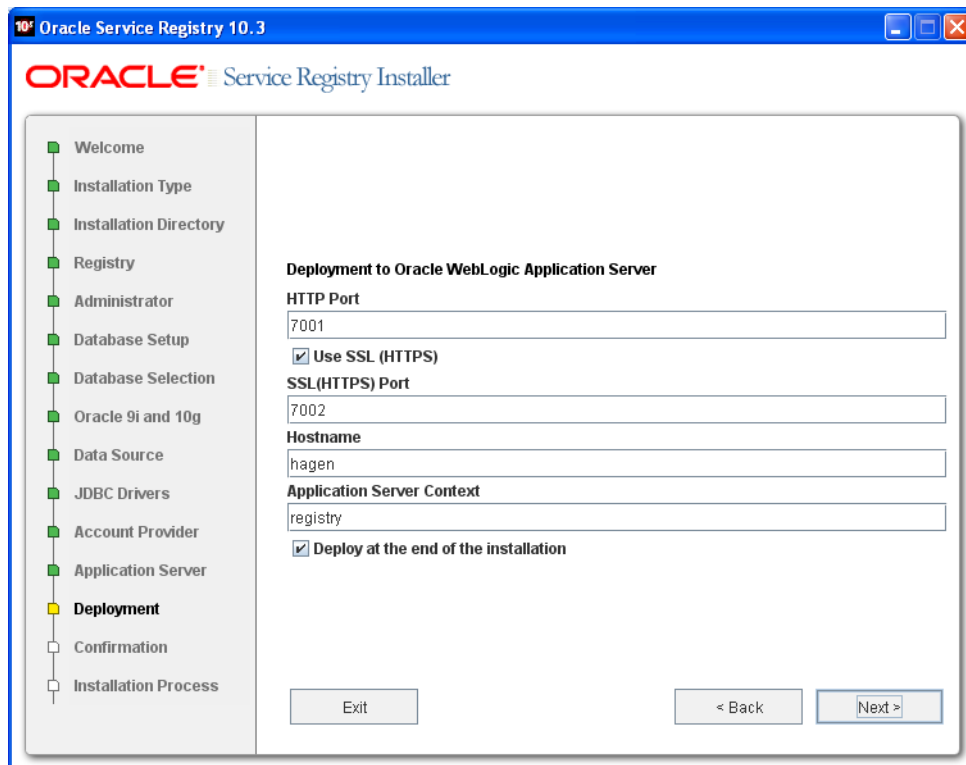


Figura 62 – Tela de configuração de deploy no servidor de aplicação

11. Neste passo é necessário informar o diretório de instalação do Oracle Weblogic Server, JAVA_HOME, host, porta e o usuário e senha do administrador do Oracle Weblogic Server (Figura 63).

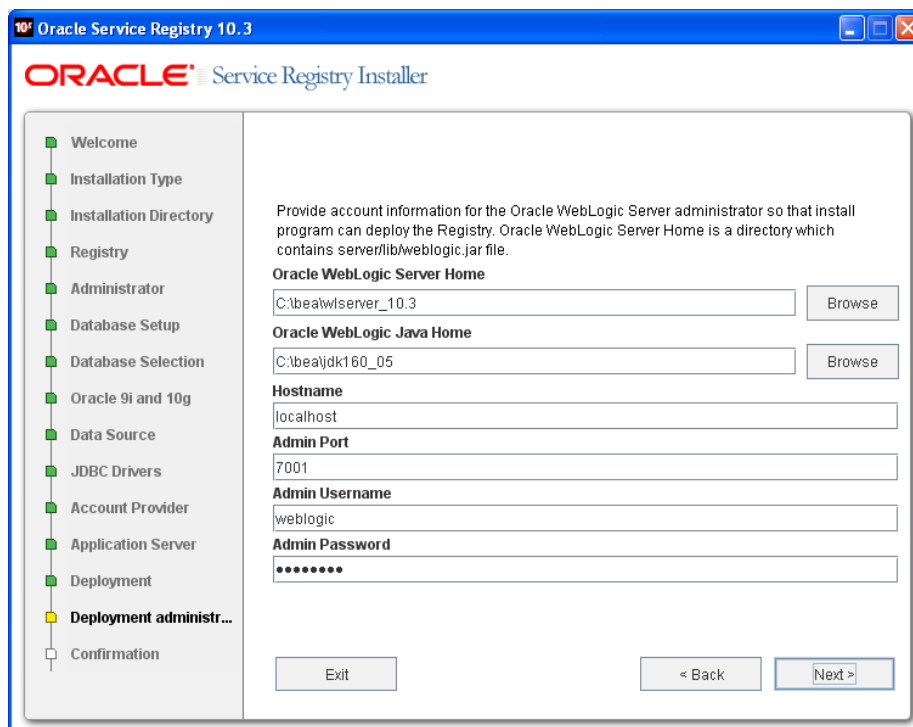


Figura 63 – Tela de configurações do Oracle Weblogic Server

12. A tela a seguir possui as informações que foram definidas durante o processo de instalação do OSR (Figura 64).

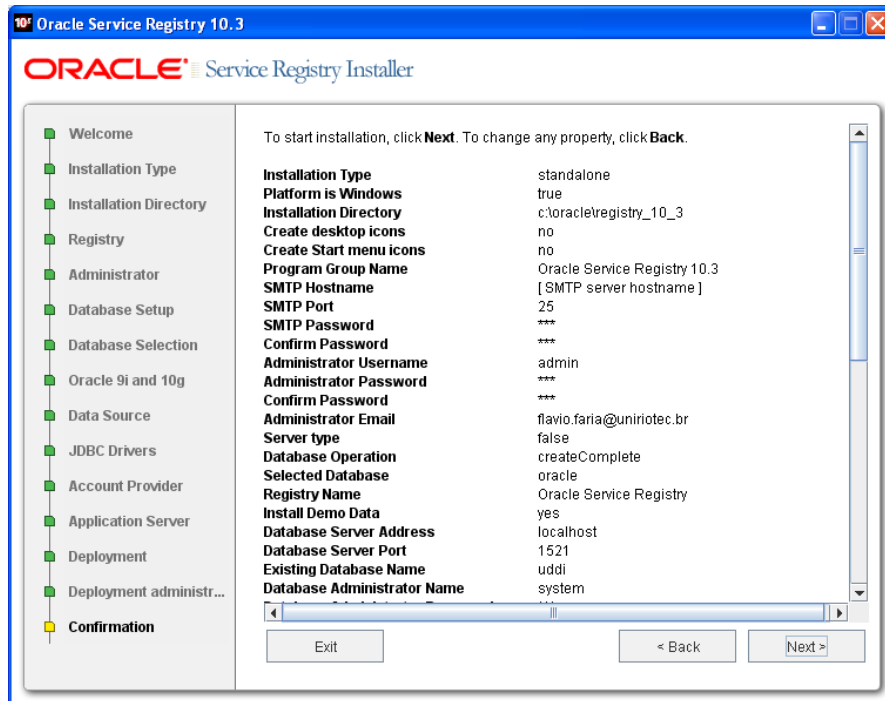


Figura 64 – Tela de confirmação da instalação

13. Esta tela indica que o processo de instalação foi finalizado com sucesso e apresenta algumas informações importantes (Figura 65).

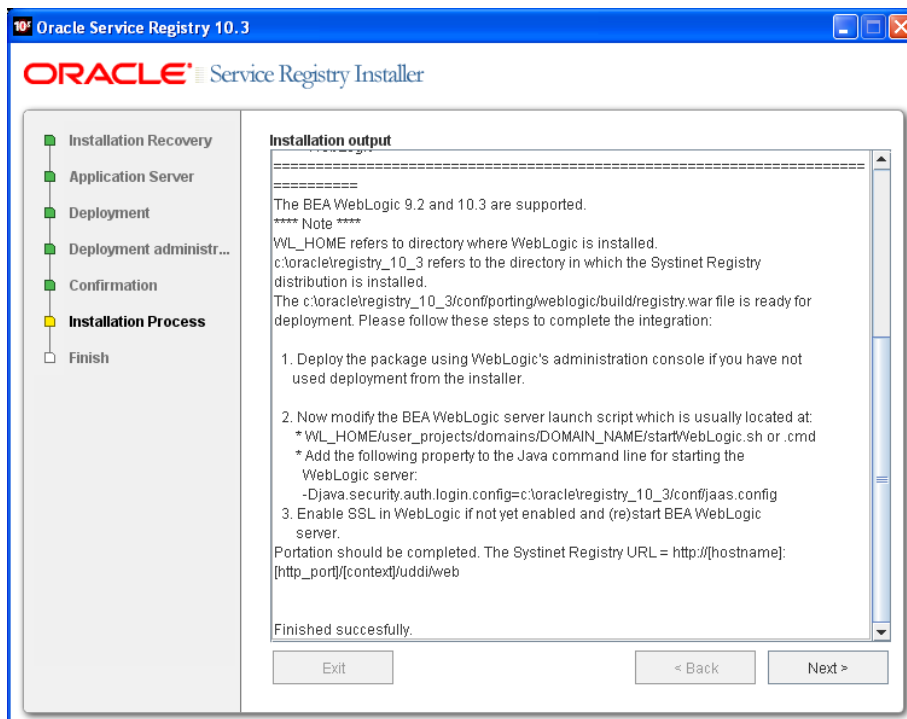


Figura 65 – Tela de instalação realizada com sucesso

É importante ressaltar que após o processo de instalação é necessário configurar o arquivo de inicialização do Oracle Weblogic Server (startWebLogic.cmd) que permite a autenticação dos usuários no OSR (existem outras formas de autenticação maiores detalhes em [Oracle, 2008a]):

```
set JAVA_OPTIONS= -Djava.security.auth.login.config=  
<OSR_HOME>/conf/jaas.config %JAVA_OPTIONS%
```

Apêndice 2 – Roteiro para gerenciamento de contas de usuários no Oracle Service Registry versão 10.3

Esta seção aborda o gerenciamento de usuários no OSR utilizando o *Registry Control* (<http://<home>:7001/registry/uddi/web>).

Para acessar o gerenciamento de usuários, acesse o *Registry Control* e faça *login* como administrador. Clique na aba *Manage* e selecione a opção *Registry Management* e clique no botão *Account management* (Figura 66).

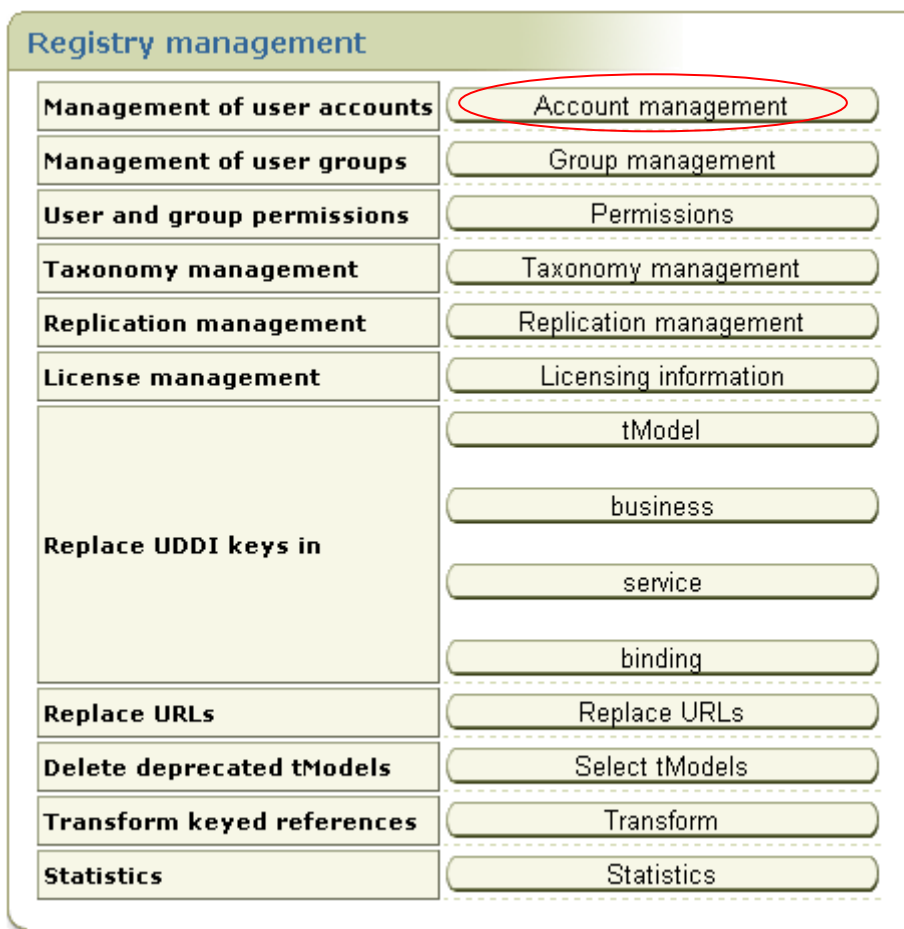


Figura 66 – Tela Registry Management (Account management)

Somente usuários com privilégios de administrador terão acesso a aba *Management*.

1. Para criar uma conta de usuário, clique na opção *Create Account* (Figura 67).

Create account

Required fields are marked *

* Login name:	john
* Email:	john@company.com
* Password:	*****
* Retype password:	*****
* Full name:	John Johnson
Default language:	English
Description:	
Business Name:	
Phone:	
Alternate phone:	
Address:	
City:	
State/Province:	
Country:	
Zip/Postal Code:	
* User profile:	Developer Profile
Blocked:	<input type="checkbox"/>
Assertions limit:	10
Bindings limit:	2
Businesses limit:	1
Services limit:	4
Subscriptions limit:	5
TModels limit:	100

Create account Cancel

Figura 67 – Tela para criação de conta de usuário

Para um melhor entendimento do campo *User profile*, consulte a descrição dos principais perfis pré-definidos pelo OSR na seção 3.7.1.2 .

Apêndice 3 – Roteiro para gerenciamento de grupo de usuários no Oracle Service Registry versão 10.3

Esta seção aborda o gerenciamento de grupo de usuários no OSR utilizando o *Registry Control* (<http://<home>:7001/registry/uddi/web>).

Para acessar o gerenciamento de grupo de usuários, acesse o *Registry Control* e faça *login* como administrador. Clique na aba *Manage* e selecione a opção *Registry Management* e clique no botão *Group management* (Figura 68).

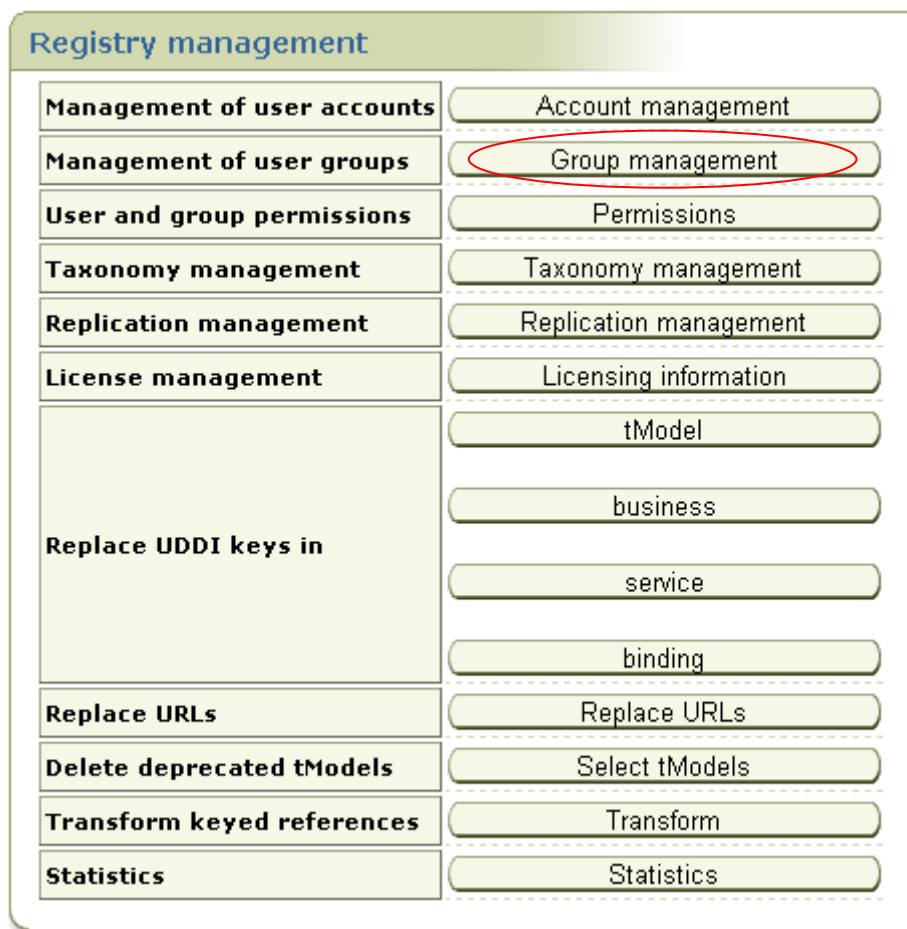


Figura 68 - Tela Registry Management (Group management)

1. Para criar um grupo de usuários, clique na opção *Add Group*.
2. Informe o nome do grupo, o proprietário, a visibilidade do grupo e uma descrição (opcional) (Figura 69).

Add group

* Group name:

* Group owner:

Group visibility: public private

Description:

This group does not exist yet. Save its properties to be able to add group members.

Figura 69 – Tela para criação de grupo de usuários

A visibilidade de um grupo (*Group visibility*) pode ser definida como pública ou privada. A diferença entre um grupo público e um grupo privado é que os membros de um grupo público podem ser vistos por qualquer usuário de registro, enquanto que os membros de um grupo privado podem ser vistos apenas pelo proprietário (*owner*) do grupo.

3. Após clicar no botão *Save group properties*, será apresentada uma lista de usuários (*User list*) e a lista de membros do grupo (*Group Members*). Defina os membros do grupo e clique em *Save group properties*. (Figura 70)

Add group

* Group name:

* Group owner:

Group visibility: public private

Description:

Users List

Login name

Displaying results 1 - 3 of 3

<input type="checkbox"/>	Login name	Full name
<input type="checkbox"/>	demo_corporate	Corporate Demo User
<input type="checkbox"/>	demo_jane	Jane Demo
<input type="checkbox"/>	demo_john	John Demo

Group members

Login name

Displaying results 1 - 2 of 2

<input type="checkbox"/>	Login name	Full name
<input type="checkbox"/>	demo_jane	Jane Demo
<input type="checkbox"/>	demo_john	John Demo

Figura 70 – Tela de edição de membros do grupo

Apêndice 4 – Roteiro para gerenciamento de permissões de usuários e grupo de usuários no Oracle Service Registry versão 10.3

Esta seção aborda o gerenciamento de permissões de usuários e grupo de usuários no OSR utilizando o Registry Control (<http://<home>:7001/registry/uddi/web>). O OSR utiliza a mesma *interface* para o gerenciamento de permissões de usuários e grupos. Nesta seção, serão abordadas apenas as permissões de usuários. Para um melhor entendimento sobre os princípios de permissão no OSR, apresentaremos um resumo sobre alguns conceitos.

O OSR divide a configuração de permissão em controle de acesso às APIs e seus métodos e controle de acesso a dados. Permissão no OSR prove um mecanismo que permite ao administrador gerenciar acesso a diferentes partes do registro para diferentes usuários. A permissão é definida como sendo o direito (privilegio) de executar uma ação (método) em uma determinada interface. O OSR utiliza o recurso Access Control que permite controlar o acesso às estruturas do UDDI.

1. Para acessar o gerenciamento de permissões de usuários, acesse o *Registry Control* e faça *login* como administrador. Clique na aba *Manage* e selecione a opção *Registry Management* e clique no botão *Permissions* (Figura 71). Observe que para gerenciar grupos basta escolher a opção *Group* no *radiobutton* nas opções de filtro.

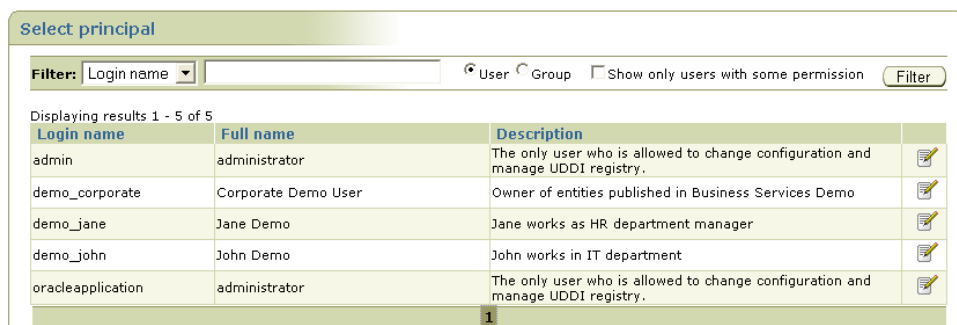


Figura 71 – Tela de seleção de usuário e grupo

2. Selecione o usuário ou grupo desejado e clique no ícone *Edit Permissions* (). Na tela que será exibida (*Permissions*) clique em *Add Permissions* e será exibida a tela referente à Figura 72.

Add permissions

Principal: admin
Principal type: user

Permission type: org.systinet.uddi.security.permission.ApiUserPermission

Permission name: org.systinet.uddi.account.AccountApi

Actions:

- delete_userAccount
- enable_userAccount
- find_userAccount
- get_userAccount
- save_userAccount
- *

Save Changes Cancel

Figura 72 – Tela para adicionar permissão em usuário ou grupo

3. Selecione o tipo de permissão a partir da lista (*drop down*) *Permission type*.
 - a. Selecione o nome da permissão desejada da lista (*drop down*) *Permission Name*
 - b. Marque os *checkbox* das ações associadas as permissões do nome selecionado em *Permission Name*, para dar permissão (*grant*) para executar essas ações. Para permitir todas as ações listadas, marque o *checkbox* referente a opção asterísco (*)

Apêndice 5 – Roteiro para publicação de um provedor de serviço no Oracle Service Registry versão 10.3

Esta seção aborda a publicação de um provedor no OSR utilizando o *Business Service Control* (BSC) (<http://<home>:7001/registry/uddi/bsc/web>). Para publicar um provedor é necessário fazer *login* no BSC.

1. Clique na aba (*menu tab*) *Catalog* (Figura 73).

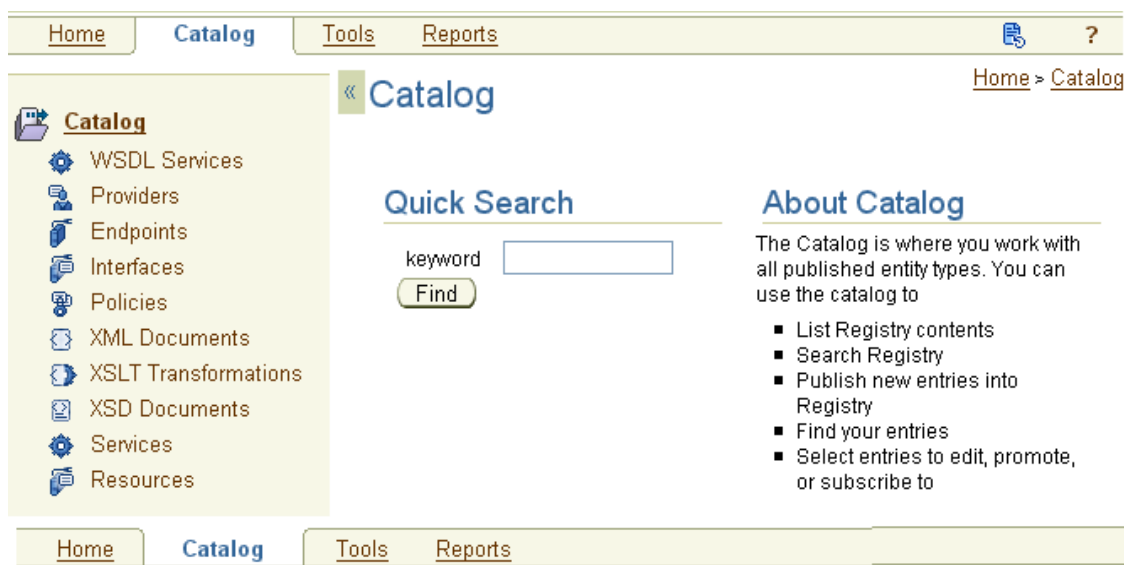


Figura 73 – Tela *treeview Catalog*

2. Clique na opção *Providers* e, em seguida, *Publish a new provider* (Figura 74).

[Home](#) > [Providers](#) > [Publish new provider](#)

Publish new provider

Step 1 of 2

Name and description

★ Name	<input type="text" value="HR Services"/>
Description	<input type="text" value="Provides services for human resource management"/>
Homepage	<input type="text" value="http://www.mycompany.com/hr"/>

Keyword =

Step 1 of 2

Figura 74 – Tela para publicação de um provedor (passo 1)

- Informe o nome do provedor, uma descrição (recomendado) e clique em *next* (Figura 75)

Publish new provider

[Home](#) > [Providers](#) > [Publish new provider](#)

Step 2 of 2

Contacts

Person name:	UseType:	Description:	Email:	Phone:	Actions
Jane Smith			janeS@company.com	1-123-123-1237	<input type="button" value="edit"/> <input type="button" value="delete"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="button" value="Add"/>

Step 2 of 2

Figura 75 – Tela para publicação de um provedor (passo 2)

- Informe o nome da pessoa de contato no provedor (recomendado) e clique em *finish*.

Apêndice 6 – Roteiro para publicação de um serviço de negócio no Oracle Service Registry versão 10.3

Esta seção aborda a publicação de um serviço de negócio no OSR utilizando o *Business Service Control* (BSC) (<http://<home>:7001/registry/uddi/bsc/web>). O serviço será criado a partir de um arquivo WSDL. Para publicar um serviço de negócio é necessário fazer *login* no BSC.

1. Clique na aba (*menu tab*) *Catalog* (Figura 73).
2. Clique na opção *WSDL Services* e, em seguida, *Publish a new WSDL service* (Figura 76).

Publish new WSDL service

[Home](#) > [WSDL Services](#) > [Publish new WSDL service](#)

Cancel Step 1 Next

WSDL Location:

★ Provider: HR Services

★ WSDL Location:

Cancel Step 1 Next

Figura 76 – Tela para publicação de um serviço de negócio (passo 1)

3. Selecione o provedor de serviço, informe a localização do arquivo WSDL e clique em next.
 - a. Caso seja necessário autenticação para acessar o arquivo WSDL será necessário informar as credenciais (usuário e senha) (Figura 77).

Publish new WSDL service

[Home](#) > [WSDL Services](#) > [Publish new WSDL service](#)

Cancel Back Step 2 of 3 Next

Enter Credentials For Secured WSDL Access

Enter Credentials For Secured WSDL Access.

★ Username:

★ Password:

Cancel Back Step 2 of 3 Next

Figura 77 – Tela para acesso a WSDL seguro

4. Informe as propriedades do serviço.

Step 2 of 4

Service properties

Service creation method:

new service
 rewrite service ---- no service ----

Service properties

Name	EmployeeList	
Description		
Usage		
Keyword		<input type="button" value="Add another"/>
Certification	<input type="radio"/> Certified <input type="radio"/> Not Applicable <input type="radio"/> Pending <input checked="" type="radio"/> N/A	
Release date		
Version		
Milestone		

Step 2 of 4

Figura 78 – Tela de propriedades do serviço

- Defina as propriedades da *interface* do serviço (Figura 79).

[Publish new WSDL service](#)
[Home](#) > [WSDL Services](#) > [Publish new WSDL service](#)

Step 3 of 4

Interface properties

Interface creation method

new interface
 rewrite interface EmployeeList_portType
 reuse interface EmployeeList_portType

Interface properties:

Local name	EmployeeList_portType	
Description		
Usage		
Keyword		<input type="button" value="Add another"/>
Status	<input type="radio"/> Beta <input type="radio"/> Deprecated <input type="radio"/> Obsolete <input type="radio"/> Stable <input checked="" type="radio"/> N/A	
Compliance	N/A	

Step 3 of 4

Figura 79 – Tela de propriedades da interface do serviço

6. Especifique o *endpoint* do serviço (Figura 80).

Publish new WSDL service

[Home](#) > [WSDL Services](#) > [Publish new WSDL service](#)

Step 4 of 4

Endpoint properties

AccessPoint	<input type="text" value="urn:unknown-location-uri"/>
Usage	<input type="text"/>
Keyword	<input type="text"/> = <input type="text"/> <input type="button" value="Add another"/>
Compliance	<input type="text" value="N/A"/> <input type="button" value="v"/>
Availability	<input type="radio"/> Available <input type="radio"/> Degraded <input type="radio"/> Unavailable <input checked="" type="radio"/> N/A
Status	<input type="radio"/> In Maintenance <input type="radio"/> In Test <input type="radio"/> Not Available <input type="radio"/> Operational <input checked="" type="radio"/> N/A

Step 4 of 4

Figura 80 – Tela para especificação do *endpoint* do serviço

Apêndice 7 – Roteiro para controlar o acesso a um serviço de negócio no Oracle Service Registry versão 10.3

Esta seção apresenta como controlar o acesso a um serviço de negócio no OSR. Para alterar as permissões de um serviço de negócio é necessário fazer *login* no OSR e o usuário deve ter privilégios administrativos ou ser o *owner* do serviço.

1. A partir de uma listagem de serviço, selecione um serviço, clique no botão *Edit* (neste caso foi utilizado o *AccountService*) e clique em *permissions* (Figura 81).



The screenshot shows the 'Edit service 'AccountService'' interface. The 'Permissions' section is active, displaying a table of users and groups. The table has columns for 'Login name', 'Full name', and 'Description'. The 'PERMISSIONS' button in the right-hand navigation menu is highlighted with a red circle.

Login name	Full name	Description	
<input type="checkbox"/>	admin	administrator	The only user who is allowed to change configuration and manage UDDI registry .
<input type="checkbox"/>	demo_corporate	Corporate Demo User	Owner of entities published in Business Services Demo
<input type="checkbox"/>	demo_jane	Jane Demo	Jane works as HR department manager
<input type="checkbox"/>	demo_john	John Demo	John works in IT department
<input type="checkbox"/>	flavio.architect	Flávio de Paula Faria	
<input type="checkbox"/>	flavio.business	Flávio de Paula Faria	
<input type="checkbox"/>	flavio.faria	Flávio de Paula Faria	
<input type="checkbox"/>	oracle	administrator	The only user who is allowed to change configuration and manage UDDI registry .
<input type="checkbox"/>	teste	conta de teste permissão	
<input type="checkbox"/>	teste1	conta de teste permissão	

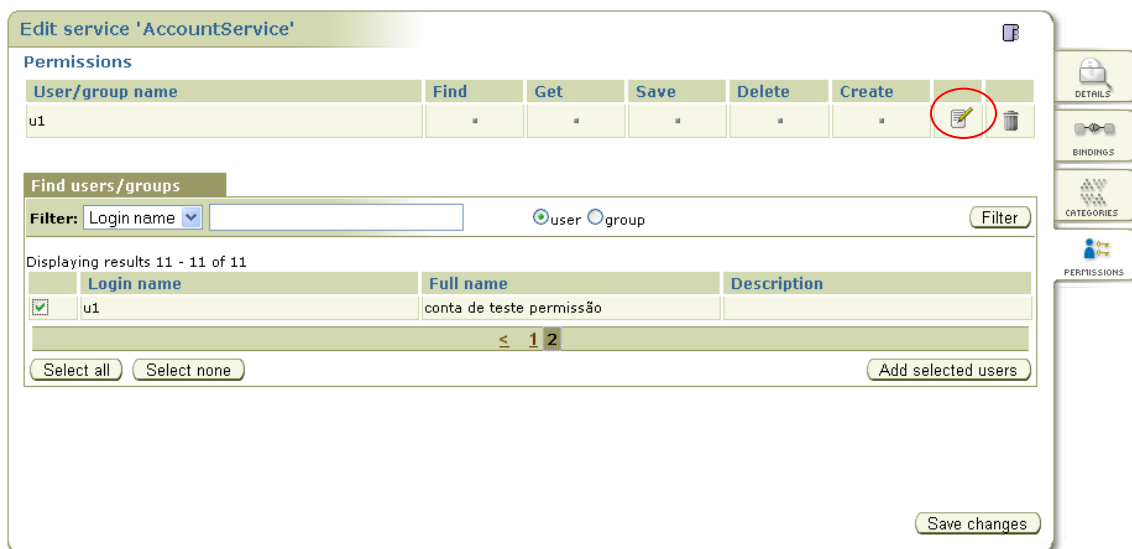
Figura 81 – Tela de edição de serviço

2. Selecione o usuário (por exemplo *U1*) ou grupo que será definida a permissão e clique em *Add selected user* (Figura 82). O usuário irá aparecerá na aba de cima.



Figura 82 – Tela de seleção do usuário para permissão no serviço

3. Edite a permissão do usuário selecionado (U1).
 - a. Clique no ícone *edit permission for user u1*



- b. Selecione as permissões desejadas e clique *update permission of user U1e* salve as alterações clicando em *save changes* (Figura 83).

Edit service 'AccountService'

Permissions

User/group name	Find	Get	Save	Delete	Create		
u1	denied	denied	denied	denied	denied	<input type="checkbox"/>	<input type="checkbox"/>

Find users/groups

Filter: Login name user group

Displaying results 11 - 11 of 11

	Login name	Full name	Description
<input type="checkbox"/>	u1	conta de teste permissão	

< 1 2

Figura 83 – Tela de definição de permissão de usuário no serviço

Dessa forma o usuário U1 tem permissão negada para qualquer tipo de operação que pode ser realizada no serviço.

Apêndice 8 – Roteiro para criar taxonomias no Oracle Service Registry versão 10.3

Esta seção apresenta como criar taxonomias utilizando o OSR.

Para acessar o gerenciamento de taxonomia, acesse o *Registry Control* (<http://<home>:7001/registry/uddi/web>) e faça *login* como administrador.

1. Clique na aba *Manage* e selecione a opção *Taxonomy Management*.

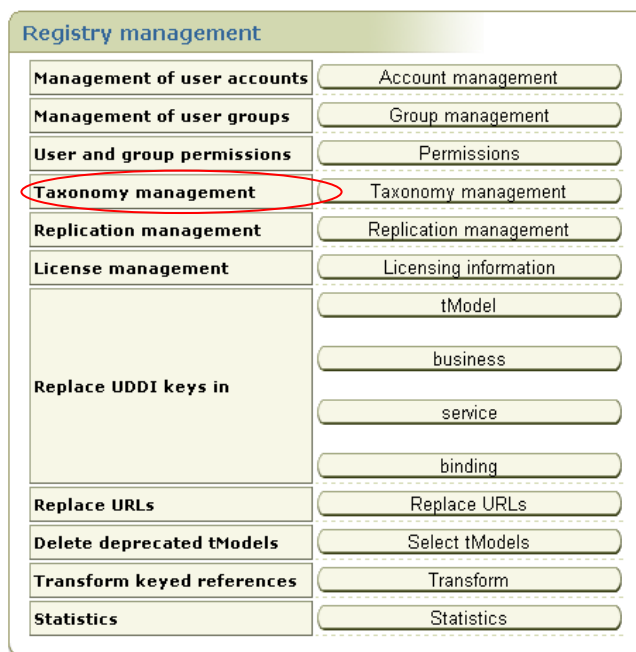


Figura 84 – Tela *Registry Management (Taxonomy management)*

2. Clique no botão *Add Taxonomy* (Figura 85).

Find taxonomy

Show:

Name:

Type: categorization identifier relationship
 categorizationGroup

Compatibility: businessEntity businessService bindingTemplate
 tModel

Validation: all Checked unchecked

Displaying results 1 - 10 of 74

Name	Description	Checked	Edit	Delete	Edit Structure	Download
Shippers	Shippers taxonomy	<input checked="" type="checkbox"/>				
Test		<input checked="" type="checkbox"/>				
demo:departmentID	Identifier of the department	<input type="checkbox"/>				
demo:hierarchy	Business hierarchy taxonomy	<input type="checkbox"/>				
demo:location:floor	Specifies the floor on which the department is located	<input type="checkbox"/>				
dnb-com:D-U-N-S	Thomas Registry Suppliers	<input type="checkbox"/>				
ebxml-org:specifications	ebXML Specifications Taxonomy	<input type="checkbox"/>				
http://schemas.xmlsoap.org/ws/2003/03/localpolicyreference	Category system used for UDDI entities to point to a WS-Policy Policy Expression tModel that describes their characteristics. See WS-PolicyAttachment specification for further details.	<input checked="" type="checkbox"/>				
http://schemas.xmlsoap.org/ws/2003/03/policytypes	WS-Policy Types category system used for UDDI tModels to characterize them as WS-Policy - based Policy Expressions.	<input checked="" type="checkbox"/>				
http://schemas.xmlsoap.org/ws/2003/03/remotepolicyreference	Category system used for UDDI entities to point to an external WS-PolicyAttachment Policy Expression that describes their characteristics. See WS-PolicyAttachment specification for further details.	<input type="checkbox"/>				

1 2 3 4 5 6 7 8 >

Figura 85 – Tela de seleção de taxonomia

3. Preencha as informações e clique em *save taxonomy*.

Figura 86 – Tela de inclusão de taxonomia

4. Para adicionar categoria(s) a taxonomia criada clique no ícone *Edit structure* na tela do resultado da pesquisa de taxonomias (Figura 87).

Name	Description	Checked	Edit	Delete	Edit Structure	Download
Shipper	Teste Taxonomia	✓	[Edit Icon]	[Delete Icon]	[Edit Structure Icon]	[Download Icon]
UNIRIO		✓	[Edit Icon]	[Delete Icon]	[Edit Structure Icon]	[Download Icon]

Figura 87 – Tela do resultado da consulta de taxonomias favoritas

- a. Clique no botão *add category* para adicionar uma categoria (Figura 88).

Figura 88 – Tela de edição de taxonomia

- b. Informe o nome e um valor para a nova categoria e clique em *save category*.

Figura 89 – Tela de inclusão de uma nova categoria

- a. Para criar uma sub-categoria de uma já existente, clique no ícone *add child category to*<nome da categoria parent> e informe o nome e valor para a nova sub-categoria

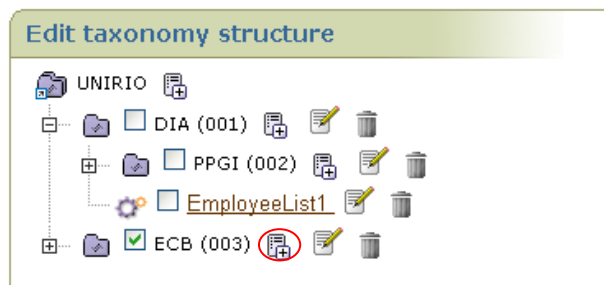


Figura 90 - Tela de inclusão de sub-categoria

Apêndice 9 – Roteiro para associar uma categoria a um serviço de negócio no Oracle Service Registry versão 10.3

Esta seção apresenta como associar uma categoria a um serviço de negócio utilizando o OSR. O objetivo dessa associação é facilitar a localização de um serviço através de uma categoria (criada na taxonomia). Após a associação de um serviço a uma categoria, é possível localizar serviços a partir de uma consulta utilizando a taxonomia (Apêndice 12).

1. Edite o serviço que será associado à categoria e acesse a *tab categories* e clique no botão *add category*.

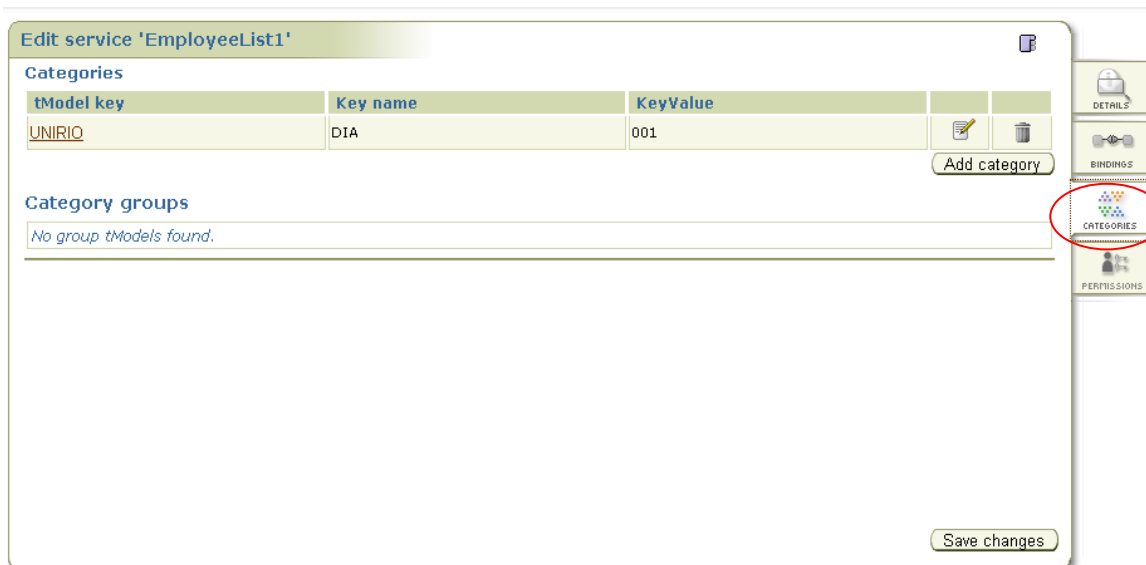


Figura 91 – Tela de edição de serviço

2. Selecione a categoria da lista de taxonomias e clique em um dos ícones *add category* ou *add category with parents*.

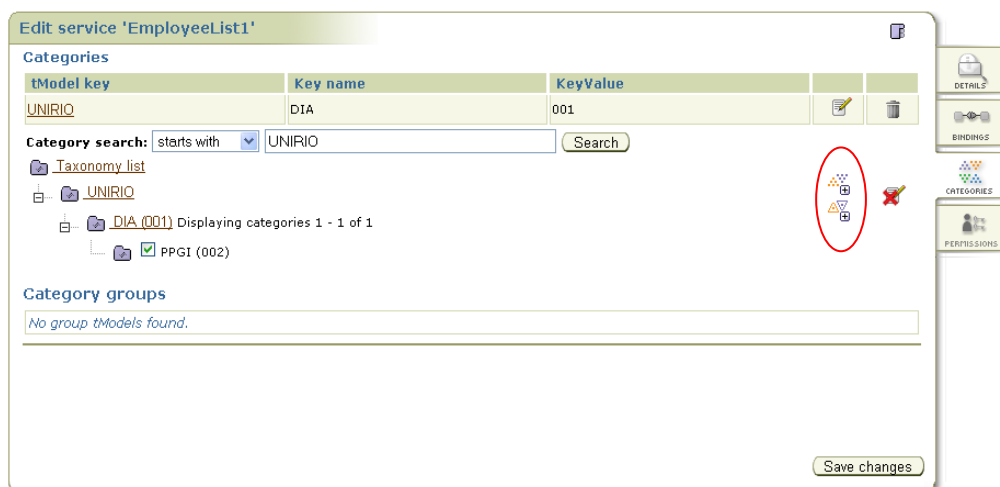


Figura 92 - Tela de adição de categoria na edição de serviço

Apêndice 10 – Roteiro para associar um serviço de negócio a um provedor de negócio no Oracle Service Registry versão 10.3

Esta seção apresenta como associar um serviço de negócio a um provedor de negócio (também chamado de *business* ou *business entity*) utilizando o OSR. O objetivo dessa associação é facilitar a localização de um serviço a partir de um provedor de negócio.

1. Edite o provedor (*business*) e acesse a aba (*tab*) *services* (Figura 93).

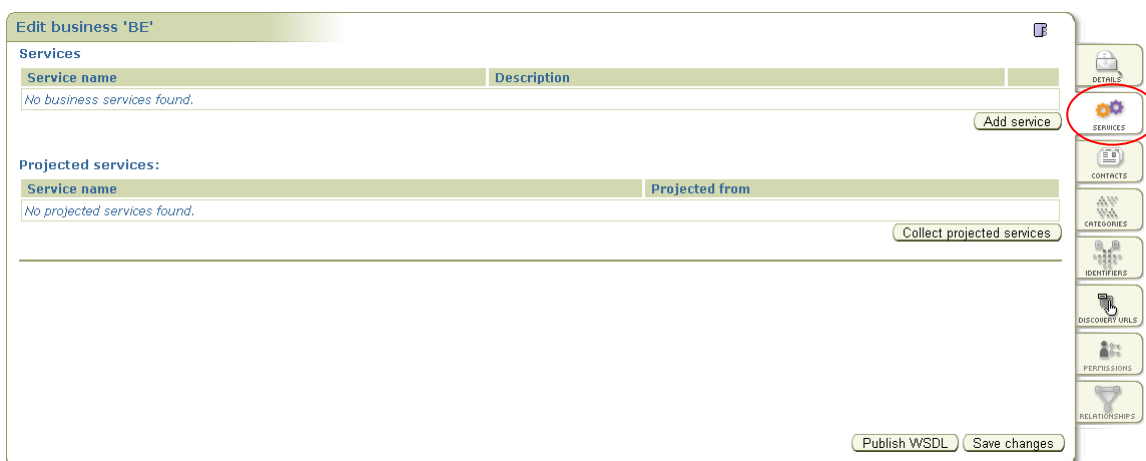


Figura 93 – Tela de edição de um provedor (*business*)

2. Clique no botão *add service* para adicionar o serviço desejado (Figura 94).

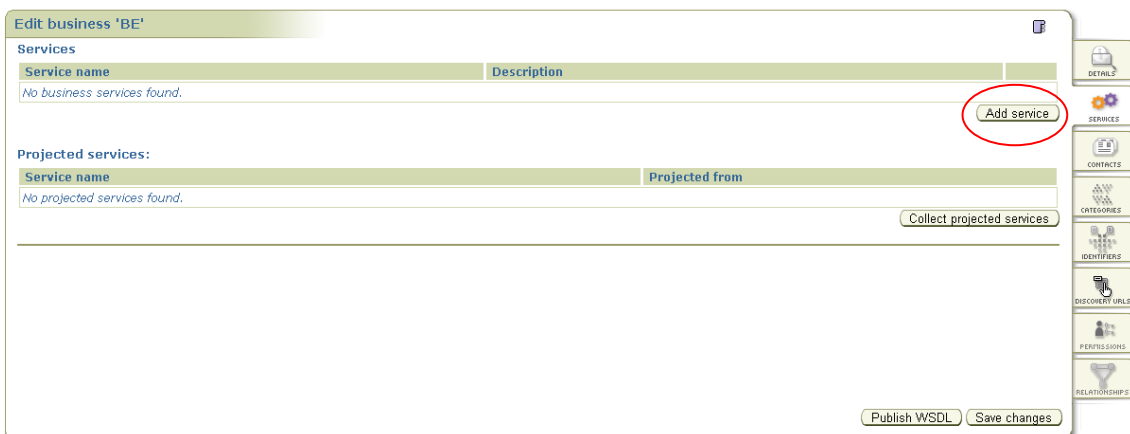


Figura 94 - Tela de adição de serviço ao provedor

3. Informe o nome do serviço que será associado ao provedor (*business*) e clique no botão *add service*. O *Service key* é o identificador único do serviço no OSR.

Add service			
Name:	EmployeeList	Language:	English
Description:		Language:	English
Service key:			

Figura 95 – Tela de inclusão de serviço no provedor (*business*)

Apêndice 11 – Roteiro para criar um *binding template* no Oracle Service Registry versão 10.3

Esta seção apresenta como criar um *binding template* possibilitando a localização de serviços de negócio utilizando o OSR.

O *binding template* é uma instância de um *web service* onde podemos obter, além de outras informações, detalhes técnicos de como invocar um serviço. Recomenda-se a utilização do *binding template* para os casos em que essas informações são compartilhadas por mais de um provedor de serviço (*business entity*).

1. Selecione o menu *Publish* e clique na aba (*tab*) *Publish*.
2. Selecione o serviço ao qual será adicionado o *binding* (Figura 96) e, em seguida, clique no link *Add Binding*.

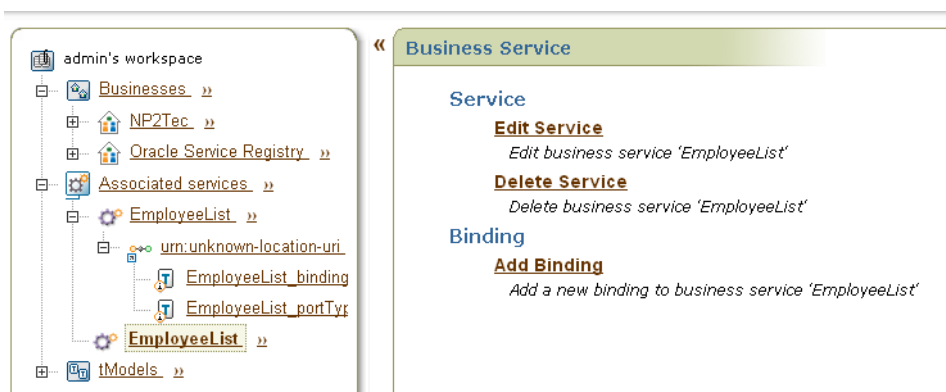


Figura 96 – Tela de seleção de serviço para inclusão do *binding*

3. Informe o *access point* que indica onde poderá ser encontrado o *end point* do serviço (*web service*). São exemplos de *access point* uma URL, um endereço de e-mail ou até mesmo um número de telefone. Clique no botão *Add Binding* para salvar.

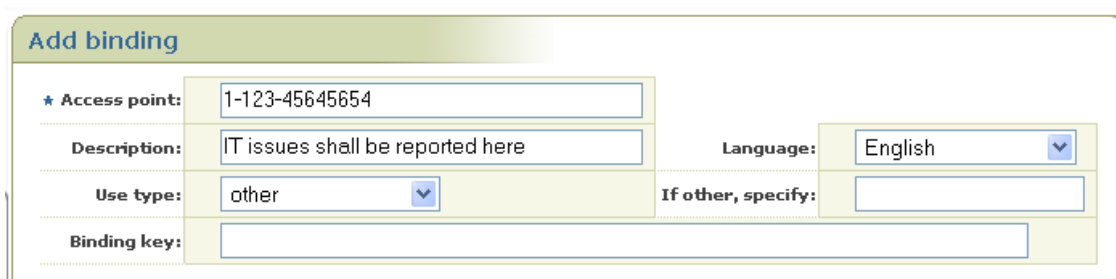
The image shows a screenshot of the 'Add binding' form. The form has a title bar 'Add binding' and several input fields. The 'Access point' field contains the value '1-123-45645654'. The 'Description' field contains 'IT issues shall be reported here'. The 'Language' dropdown menu is set to 'English'. The 'Use type' dropdown menu is set to 'other'. The 'If other, specify' field is empty. The 'Binding key' field is also empty.

Figura 97 – Tela de inclusão do *binding*

Apêndice 12 – Utilizando taxonomia para pesquisa de serviços no Oracle Service Registry versão 10.3

Esta seção apresenta como utilizar taxonomia para a localização de serviços de negócio utilizando o OSR.

1. A partir da tela de pesquisa de serviços, acesse a aba *Categories* e clique no botão *Add Category* (Figura 98).

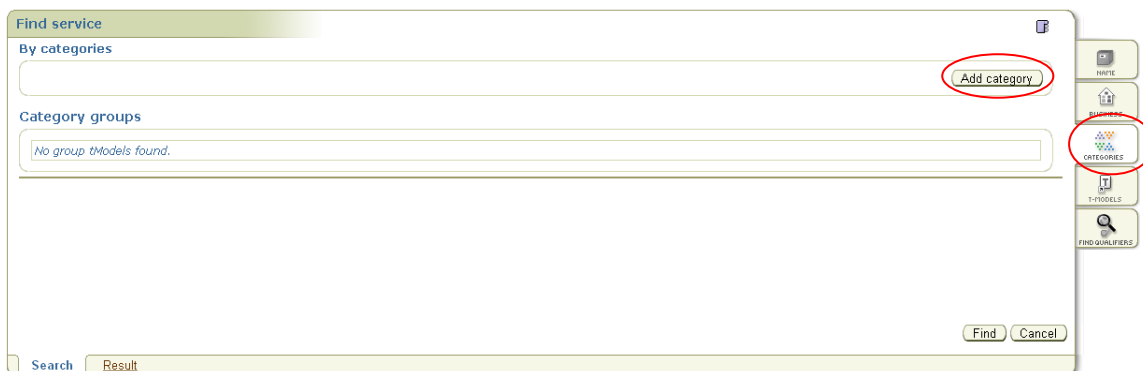


Figura 98 – Tela de pesquisa de serviços

2. Selecione a taxonomia desejada da lista de taxonomias (*Taxonomy list*). Utilize os ícones *add categories* ou *add with parent* para selecionar a(s) categoria(s) as opções de filtro (Figura 99).

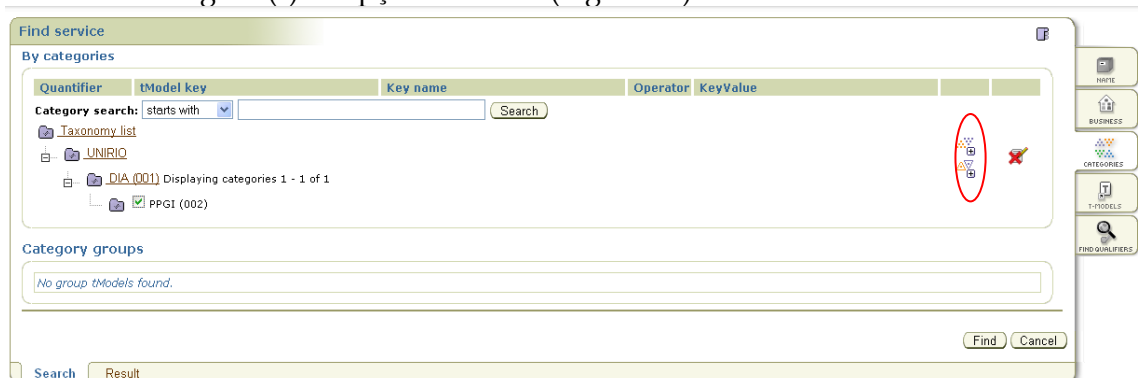


Figura 99 – Tela de pesquisa de serviços por categoria

3. Após a adição das categorias desejadas clique no botão *find* (Figura 100).

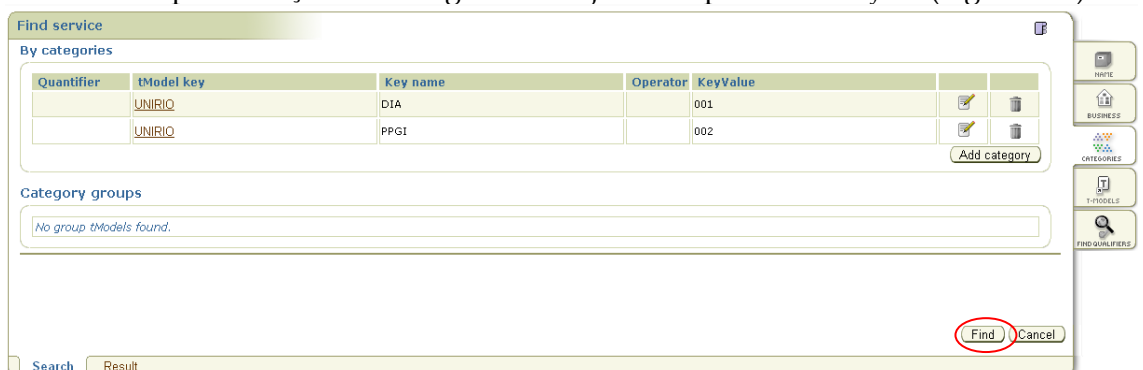


Figura 100 – Tela de pesquisa de serviços com as categorias selecionadas