

4

Contract Oriented Web Services Model (COWS)

Contract Oriented Web Services (COWS) define um modelo para um ambiente adequado para diálogos de *E-Business* implementados utilizando *Web Services*, baseados em contratos bem definidos acordados por todos os parceiros envolvidos, que incorporam vários níveis de políticas que são aplicáveis a esses contratos.

COWS separa as descrições de serviço em duas partes: a descrição relativa ao domínio do serviço e a descrição relativa ao contrato do serviço. A descrição do domínio do serviço contém a informação sobre os dados que são o objeto da transação em si, a parte funcional do serviço. A descrição do contrato contém informações sobre os parceiros envolvidos no negócio, o protocolo de comunicação que deve ser estabelecido para que se alcancem os resultados esperados por todos os parceiros envolvidos (coreografia), assim como as restrições que são aplicáveis, tais como, qualidade de serviço (QoS), políticas de privacidade etc.

Um documento especificado em WSDL descreve o formato das mensagens e o tipo de variáveis contidas em mensagens que são trocadas entre *Web Services*. Especificações em OWL-S e WSMO descrevem o domínio do serviço com mais informação semântica do que uma especificação WSDL, mas todas elas descrevem a informação de contrato apenas pela perspectiva do fornecedor do serviço. O solicitante do serviço é obrigado a aceitar o contrato de forma incondicional, sem possibilidades de modificá-lo. Por outro lado, o fornecedor do serviço não tem informação alguma sobre as capacidades do solicitante e de sua real possibilidade de atender a restrições que também podem existir por parte do fornecedor em relação ao solicitante do serviço.

Fornecedores de serviços podem anunciar os serviços oferecidos em agências de casamento de serviços — especializadas em casamento de serviços de parceiros de negócios —, mas os solicitantes de serviços não têm a possibilidade de anunciar suas necessidades utilizando algum tipo análogo de agência.

Conseqüentemente, os processos atuais de *E-Business* são sempre iniciados por um solicitante de serviço, que procura por um fornecedor. Nessa busca, não existe uma perspectiva igualitária quanto à possibilidade de definição de políticas por parte do fornecedor e do solicitante do serviço.

O processo de casamento de serviços de parceiros de negócios precisa ser simétrico. Não se deve restringir o anúncio apenas aos fornecedores e a busca ao solicitante. Além disso, se um diálogo de *E-Business* envolve mais de dois participantes, não existe a especificação de um contrato global que abranja todos os envolvidos. O que existe são apenas contratos bilaterais entre cada um dos pares.

O objetivo do modelo abstrato do COWS é ampliar os esforços atuais para a agregação de informação semântica aos *Web Services*, aproximando o *E-Business* do ambiente de negócios do mundo real, que é regido por contratos que estabelecem relacionamentos entre parceiros, incluindo os aspectos relacionados ao protocolo de coordenação (coreografia) e as restrições que são aplicáveis ao processo (políticas).

Quando algum agente deseja estabelecer um diálogo de *E-Business*, em um ambiente COWS, ele informa a um serviço de casamento de serviços de parceiros de negócios qual é o seu domínio de informação e o papel que ele deseja desempenhar no processo de *E-Business*, que é definido por um conjunto de padrões de contrato. Com base nessas informações, o serviço de casamento de serviços pode procurar parceiros que estabeleçam relacionamentos entre papéis específicos em contratos bem definidos.

A visão do COWS é a de que o contrato deve ser a entidade central de um negócio envolvendo parceiros comerciais, que estabelecem relacionamentos entre papéis específicos, desempenhados por cada um dos parceiros, e que têm um objetivo comum a ser alcançado por todos.

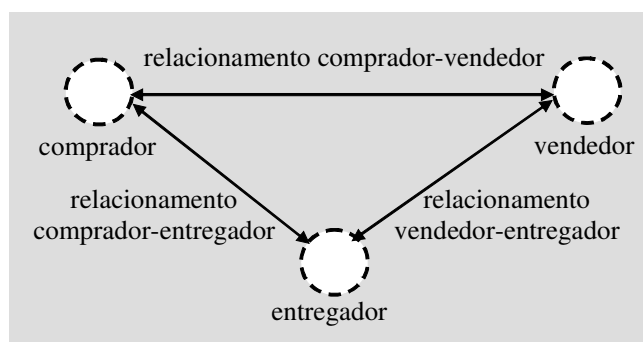


Figura 7 - Exemplo de contrato envolvendo três papéis

Vamos considerar um exemplo de um usuário que deseja comprar um produto. Podemos ter nessa situação um relacionamento entre três papéis: um comprador, um vendedor e um entregador (Figura 7).

Vamos considerar que o processo de negócios relativo à compra e venda do produto pode ser representado pelo seguinte conjunto de PIPS RosettaNet, representado na Figura 8:

- PIP3A2 – o comprador solicita ao vendedor o preço e a disponibilidade do produto.
- PIP3A4 – o comprador envia ao vendedor uma ordem de compra do produto.
- PIP3C3 – o vendedor envia ao comprador uma notificação de efetuação do pedido de compra do produto.
- PIP3B1 – o vendedor envia ao entregador uma solicitação de entrega do produto.
- PIP3B3 – o comprador requisita ao entregador uma notificação do status da entrega do produto.
- PIP4B2 – o comprador envia ao vendedor uma notificação de recebimento da entrega.
- PIP3C5 – o vendedor envia ao comprador uma notificação para o pagamento do produto.

O serviço de casamento de serviços tem que procurar por parceiros — comprador, vendedor e entregador — que atendam à coreografia de interações especificada pelos PIPs, como definido acima. Além disso, uma série de restrições pode ser agregada à definição do processo, que inclui, por exemplo, formas de pagamento, localização geográfica dos parceiros, políticas de segurança, quais as credenciais a ser apresentadas para o estabelecimento de confiança entre os parceiros, entre outros.

A especificação de um contrato e a definição por parte dos parceiros de negócios de qual seria a sua participação no contrato, em conjunto com as restrições requeridas e as restrições aceitas por cada um dos parceiros, pode auxiliar na aproximação mais eficiente de parceiros de negócios.

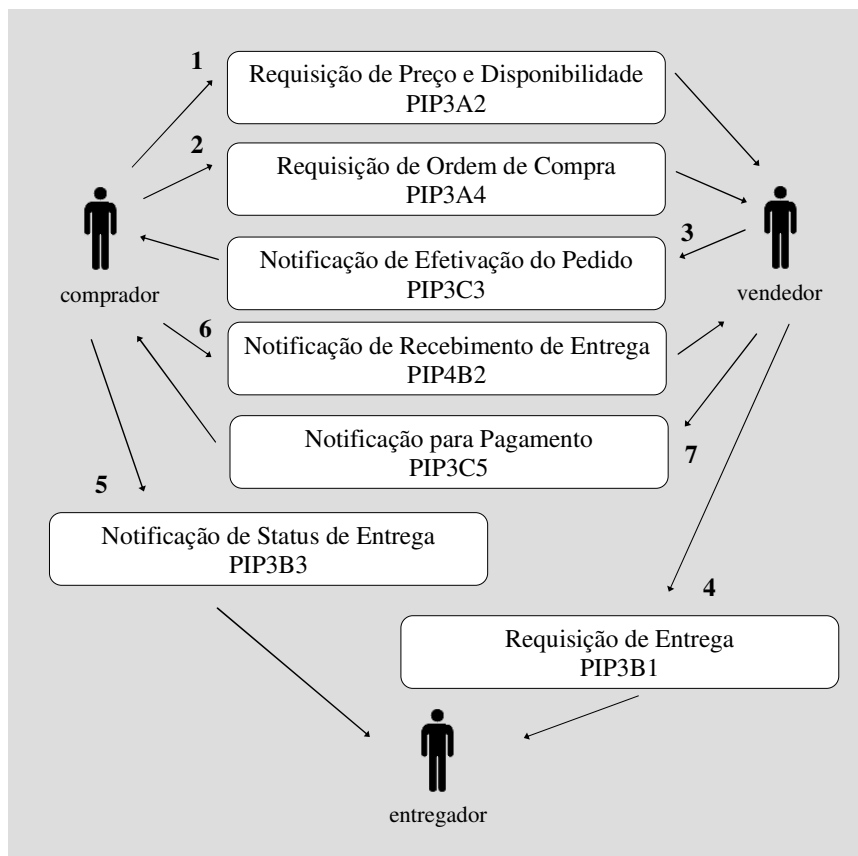


Figura 8 - Exemplo de processo de negócios definido com PIPs RosettaNet

Neste capítulo, é apresentado, inicialmente, um resumo da sintaxe *Notation 3*, visando facilitar a compreensão da formalização utilizada para o modelo conceitual do COWS. Em seguida, cada elemento do modelo conceitual do COWS é descrito, em conjunto com a sua formalização na representação N3.

4.1. Notation 3 – N3

Notation 3 (N3) é uma notação similar à *Resource Description Framework* (RDF), em sua sintaxe XML, porém, de mais fácil legibilidade. Em RDF, uma informação é uma coleção de declarações, cada uma delas com um sujeito, um predicado e um objeto — uma tripla RDF. O predicado expressa a relação entre um sujeito e um objeto.

Em N3, uma tripla RDF pode ser expressa da seguinte forma:

- #paula #amiga #maria.
- #paula # idade 29.
- #maria # idade 27.

As declarações acima poderiam ser lidas como “#paula é #amiga de #maria”, “#paula tem 29 anos de #idade” e “#maria tem 27 anos de #idade”. O ponto final é obrigatório como terminação de uma declaração em N3.

Todos os elementos são identificados por meio de *Uniform Resource Identifiers* (URI), que fornecem uma forma simples e extensível de se identificar recursos, por exemplo:

- <http://www.tecweb.inf.puc-rio.br/pessoas#paula>

Nesse caso, a parte da URI antes do caractere “#” refere-se a um esquema e a parte após o caractere “#” refere-se a um componente de identificação de fragmento da URI, que permite a identificação indireta de um segundo recurso com referência a um primeiro recurso. Caso a informação de esquema seja suprimida, será entendido que a referência é relativa a algo definido no próprio documento.

Existe uma exceção quanto à forma de identificação por URIs: objetos também podem ser expressos como literais — inteiros, *strings* etc.

Existem duas maneiras resumidas para se expressar diversas declarações sobre o mesmo sujeito: o caractere “;” indica outra propriedade acerca do mesmo objeto e o caractere “,” indica outro objeto para o mesmo sujeito e predicado. Por exemplo:

- #paula #amiga #maria, #joana; #idade 29.

A declaração acima pode ser lida como “#paula é #amiga de #maria e de #joana e tem 29 anos de #idade”.

Algumas vezes, não existe a necessidade de se ter um identificador explícito em uma declaração. Por exemplo:

- #paula #amiga [#idade 27].

A declaração acima pode ser lida como “#paula tem uma #amiga de 27 anos de #idade”. Os colchetes declaram que alguma coisa existe e que tem determinadas propriedades, mas não existe a possibilidade de referenciá-la em nenhum documento.

Os identificadores não têm nenhum significado em si — são apenas identificadores. Para se definir alguém que tenha o nome “Paula”, associado ao identificador #paula, seria necessária a seguinte declaração:

- #paula #nome “Paula”.

O mesmo se aplica aos predicados: #amiga não tem o significado semântico de amizade, ou seja, de que #paula é amiga de #maria.

4.2. Modelo Conceitual

O modelo abstrato do COWS emprega alguns dos conceitos de WS-CDL. Um diálogo de *E-Business* é visto como uma troca de informações que depende de relacionamentos entre parceiros desempenhando papéis específicos. O ambiente operacional do COWS tem um registro onde um possível parceiro de negócios anuncia o papel que ele deseja desempenhar, utilizando um conjunto de contratos, além de informações de domínio relacionadas ao objetivo a ser alcançado — da forma como já é publicada atualmente —, e de políticas que expressam restrições impostas por cada um dos parceiros.

Em adição ao modelo tradicional de casamento de objetivos, o serviço de busca introduz mais um nível de restrição: a procura de parceiros que têm possibilidade de estabelecer relacionamentos entre papéis que são desempenhados por esses parceiros e são definidos nos contratos referenciados. A descrição semântica das informações de domínio pode ser feita da forma usual, como WSDL, OWL-S, WSMO etc.

A Figura 9 apresenta um diagrama UML com a representação do modelo conceitual do COWS.

Um determinado contrato é composto por diversos papéis, que podem, por sua vez, estar vinculados a outros papéis, definidos em outros contratos. Cada contrato é regido por uma coreografia que estabelece o protocolo de interações entre os diversos agentes que participam de um contrato.

Um agente participa de um ou mais contratos, desempenhando um ou mais papéis em cada um dos contratos, de acordo com diretrizes estabelecidas em foros específicos. Esses foros têm associados a si, políticas que devem ser atendidas pelos contratos sob sua jurisdição. Cada papel, desempenhado por um agente, estabelece relacionamentos com os agentes que desempenham os outros papéis definidos no contrato. Esses relacionamentos podem exigir seqüências de trocas de credenciais que venham a estabelecer a confiança entre os parceiros. Cada papel tem associado a si a descrição funcional do serviço por ele implementado,

que pode estar especificada de acordo com diversos meta-modelos como, por exemplo, OWL-S e WSMO.

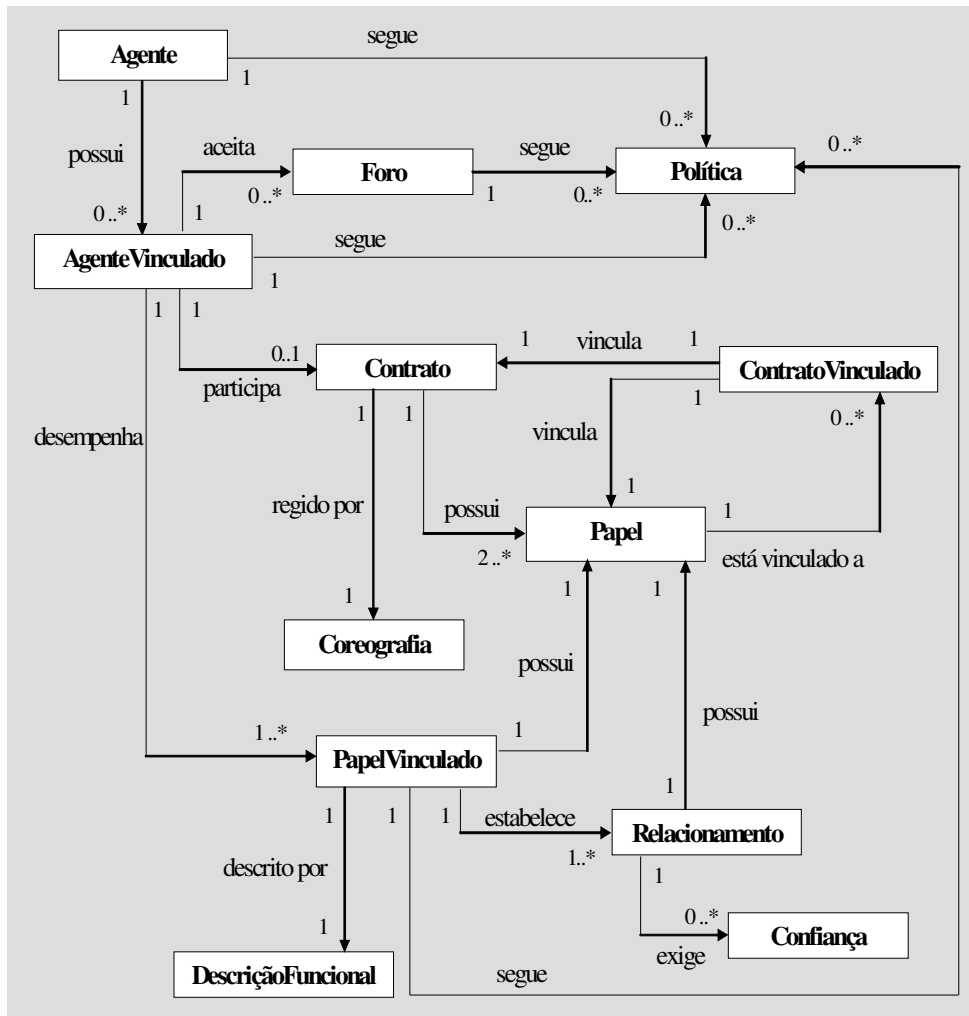


Figura 9 - Modelo Conceitual do COWS

Diversas políticas, compostas por um conjunto de condições, podem ser especificadas, restringindo a possibilidade de agentes, que não satisfaçam às condições estabelecidas, em participar dos contratos. Essas políticas podem ser definidas em diversos níveis: no nível do agente, no nível do contrato em que um agente participa, no nível do papel desempenhado por um agente em um contrato, além de políticas definidas no âmbito dos foros associados aos contratos.

Nas seções seguintes, são apresentadas, em detalhe, as especificações de cada uma das classes do modelo, utilizando-se a notação N3.

4.2.1. Contrato

Um contrato define um conjunto de parceiros que desempenham papéis e são regidos por uma coreografia que deve conter todas as informações

relacionadas às interações necessárias entre os parceiros. Um contrato precisa ter pelo menos dois papéis para que se estabeleça pelo menos um relacionamento entre dois parceiros.

No caso de uma agência para casamento de serviços de parceiros de negócios, um agente, ao se publicar como candidato para a participação em processos de negócios, deverá indicar, além das especificações funcionais associadas ao domínio do negócio propriamente dito, quais os tipos de contrato que são aceitos por esse agente e quais os respectivos papéis que ele se propõe a desempenhar em cada um dos tipos de contrato.

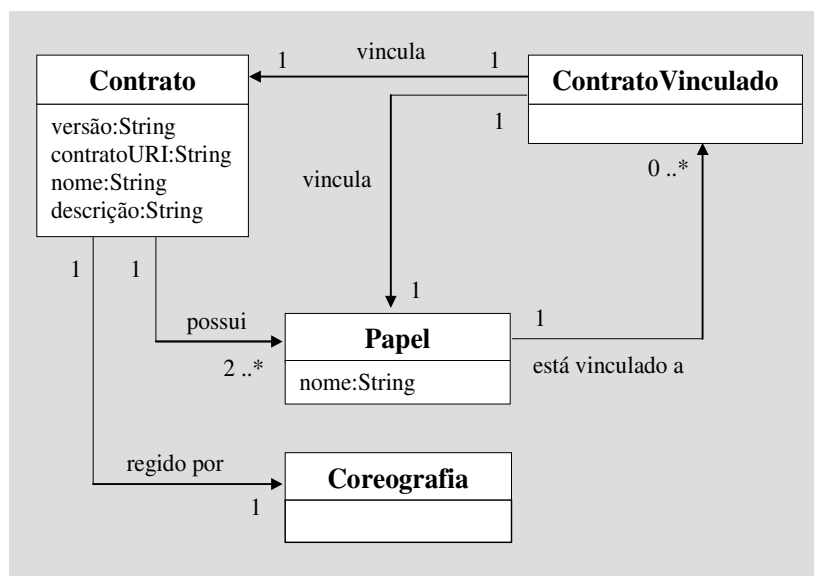


Figura 10 - Contrato

Um papel definido em um contrato pode estar vinculado a um papel definido em um outro contrato, permitindo assim que uma cadeia de contratos possa ser estabelecida, para que se alcance um determinado objetivo.

Por exemplo, um tipo de contrato de compra e venda (C1) pode conter dois papéis, comprador e vendedor, sendo que o papel de comprador pode estar vinculado ao papel de vendedor em um outro tipo de contrato de compra e venda (C2), que, por sua vez, estabelece o relacionamento entre os papéis de comprador e vendedor. Assim sendo, um relacionamento indireto pode ser estabelecido entre dois agentes, por intermédio de um terceiro agente, que desempenha dois papéis vinculados em tipos de contratos distintos.

A coreografia associada a um contrato deve representar todos os comportamentos externos observáveis dos parceiros envolvidos na realização do negócio, de forma a que todos tenham uma visão global do processo. Dessa

forma, pode-se garantir que todos os parceiros envolvidos no processo têm pleno conhecimento dos requisitos operacionais necessários para a execução efetiva do processo.

Não faz parte do escopo deste trabalho a definição da coreografia associada a um contrato. Uma das opções para a descrição da coreografia pode ser a utilização de *Web Services Choreography Description Language* (WS-CDL). O fundamental é que essa descrição seja compreendida e aceita pelos signatários do contrato, que assumem o compromisso de que sua forma operacional atenda à especificação da coreografia.

As Figuras 11a e 11b apresentam a representação, em N3, do diagrama da Figura 10.

```

@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns>.
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
@prefix xsd: <http://www.w3.org/2001/XMLSchema#>.
@prefix owl: <http://www.w3.org/2002/07/owl#>.
@prefix : <http://www.tecweb.inf.puc-rio.br/ont/2005/05/cows#>.

# Class:                Contract
# DatatypeProperty:    version?, contractURI, name, description?
# ObjectProperty:      hasRole+, hasChoreography
:Contract a owl:Class;
  rdfs:subClassOf [a owl:Restriction; owl:onProperty :version; owl:maxCardinality "1"];
  rdfs:subClassOf [a owl:Restriction; owl:onProperty :contractURI; owl:cardinality "1"];
  rdfs:subClassOf [a owl:Restriction; owl:onProperty :name; owl:cardinality "1"];
  rdfs:subClassOf [a owl:Restriction; owl:onProperty :hasRole; owl:minCardinality "2"];
  rdfs:subClassOf [a owl:Restriction; owl:onProperty :description; owl:maxCardinality "1"];
  rdfs:subClassOf [a owl:Restriction; owl:onProperty :hasChoreography; owl:cardinality "1"].
:version a owl:DatatypeProperty;
  rdfs:domain :Contract;
  rdfs:range xsd:string.
:contractURI a owl:DatatypeProperty;
  rdfs:domain [owl:unionOf (:Contract :ContractBinding)];
  rdfs:range xsd:anyURI.
:name a owl:DatatypeProperty;
  rdfs:domain :Contract;
  rdfs:range xsd:string.
:description a owl:DatatypeProperty;
  rdfs:domain :Contract;
  rdfs:range xsd:string.
:hasRole a owl:ObjectProperty;
  rdfs:domain :Contract;
  rdfs:range :Role.
:hasChoreography a owl:ObjectProperty;
  rdfs:domain :Contract;
  rdfs:range :Choreography.

# Class:                Role
# DatatypeProperty:    roleName
# ObjectProperty:      hasContractBinding*
:Role a owl:Class;
  rdfs:subClassOf [a owl:Restriction; owl:onProperty :roleName; owl:cardinality "1"].
:roleName a owl:DatatypeProperty;
  rdfs:domain :Role;
  rdfs:range xsd:string.
:hasContractBinding a owl:ObjectProperty;
  rdfs:domain :Role;
  rdfs:range :ContractBinding.

```

Figura 11a - Especificação de contrato em N3

```

# Class:          ContractBinding
# DatatypeProperty:
# ObjectProperty:  hasRole, hasContract
:ContractBinding a owl:Class;
  rdfs:subClassOf [a owl:Restriction; owl:onProperty :hasRole; owl:cardinality "1"];
  rdfs:subClassOf [a owl:Restriction; owl:onProperty :hasContract; owl:cardinality "1"];
:hasRole a owl:ObjectProperty;
  rdfs:domain :ContractBinding;
  rdfs:range :Role.
:hasContract a owl:ObjectProperty;
  rdfs:domain :ContractBinding;
  rdfs:range :Contract.

# Class:          Choreography
# DatatypeProperty: definition
# ObjectProperty:
:Role a owl:Class;
  rdfs:subClassOf [a owl:Restriction; owl:onProperty :definition; owl:cardinality "1"];
:definition a owl:DatatypeProperty;
  rdfs:domain :Choreography;
  rdfs:range xsd:string.
    
```

Figura 11b - Especificação de contrato em N3

4.2.2. Agente

Um agente participa de um ou mais contratos, desempenhando um ou mais papéis em cada um dos contratos, de acordo com diretrizes estabelecidas em foros específicos. Um agente segue políticas, especificadas como um conjunto de condições que restringem a possibilidade de outros agentes, que não satisfaçam às condições estabelecidas, em participar dos contratos.

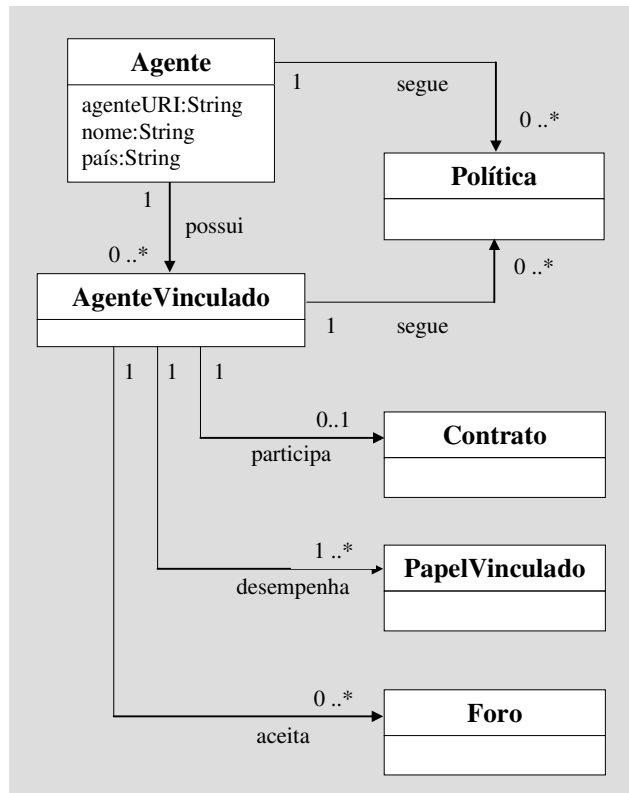


Figura 12 - Agente

Ao se candidatar a ser um dos parceiros de um determinado processo de negócios, um agente deve indicar em quais contratos ele aceita desempenhar um papel específico. Em determinados contratos, um agente pode desempenhar mais de um papel. Por exemplo, podemos considerar um contrato de compra e venda (C1) que defina três papéis — comprador, vendedor e entregador. Esses três papéis poderiam ser desempenhados por três agentes diferentes, ou, então, um determinado agente poderia desempenhar o papel de comprador e um segundo agente os papéis de vendedor e entregador.

```
# Class: Agent
# DatatypeProperty: agentURI, name, country
# ObjectProperty: hasAgentBinding*, hasPolicy*
Agent a owl:Class;
  rdfs:subClassOf [a owl:Restriction; owl:onProperty :agentURI; owl:cardinality "1"];
  rdfs:subClassOf [a owl:Restriction; owl:onProperty :name; owl:cardinality "1"];
  rdfs:subClassOf [a owl:Restriction; owl:onProperty :country; owl:cardinality "1"].
:agentURI a owl:DatatypeProperty;
  rdfs:domain :Agent;
  rdfs:range xsd:anyURI.
:name a owl:DatatypeProperty;
  rdfs:domain :Agent;
  rdfs:range xsd:string.
:country a owl:DatatypeProperty;
  rdfs:domain :Agent;
  rdfs:range xsd:string.
:hasAgentBinding a owl:ObjectProperty;
  rdfs:domain :Agent;
  rdfs:range :AgentBinding.
:hasPolicy a owl:ObjectProperty;
  rdfs:domain :Agent;
  rdfs:range :Policy.

# Class: AgentBinding
# DatatypeProperty:
# ObjectProperty: hasContract, hasForum*, hasRoleBinding+, hasPolicy*
AgentBinding a owl:Class;
  rdfs:subClassOf [a owl:Restriction; owl:onProperty :hasContract; owl:minCardinality "0"];
  rdfs:subClassOf [a owl:Restriction; owl:onProperty :hasContract; owl:maxCardinality "1"];
  rdfs:subClassOf [a owl:Restriction; owl:onProperty :hasRoleBinding; owl:minCardinality "1"].
:hasContract a owl:ObjectProperty;
  rdfs:domain :AgentBinding;
  rdfs:range :Contract.
:hasForum a owl:ObjectProperty;
  rdfs:domain :AgentBinding;
  rdfs:range :Forum.
:hasRoleBinding a owl:ObjectProperty;
  rdfs:domain :AgentBinding;
  rdfs:range :RoleBinding.
:hasPolicy a owl:ObjectProperty;
  rdfs:domain :AgentBinding;
  rdfs:range :Policy.
```

Figura 13 - Especificação de agente em N3

Para cada contrato, que o agente indica como opção para sua participação, devem ser indicados quais os foros que são aceitos para esse contrato. Por exemplo, um agente A1 pode indicar que aceita desempenhar um papel de comprador no contrato C1 sob a jurisdição do foro F1. Os agentes que estarão, inicialmente, habilitados a participar do contrato C1 com o agente A1 terão que se

candidatar para desempenhar o papel de vendedor ou o papel de entregador, ambos aceitando também o foro F1. Um agente que se candidate a participar do contrato C1 sob um foro diferente de F1 estará impedido de participar desse negócio.

A Figura 13 apresenta a representação, em N3, do diagrama da Figura 12.

4.2.3.Papel

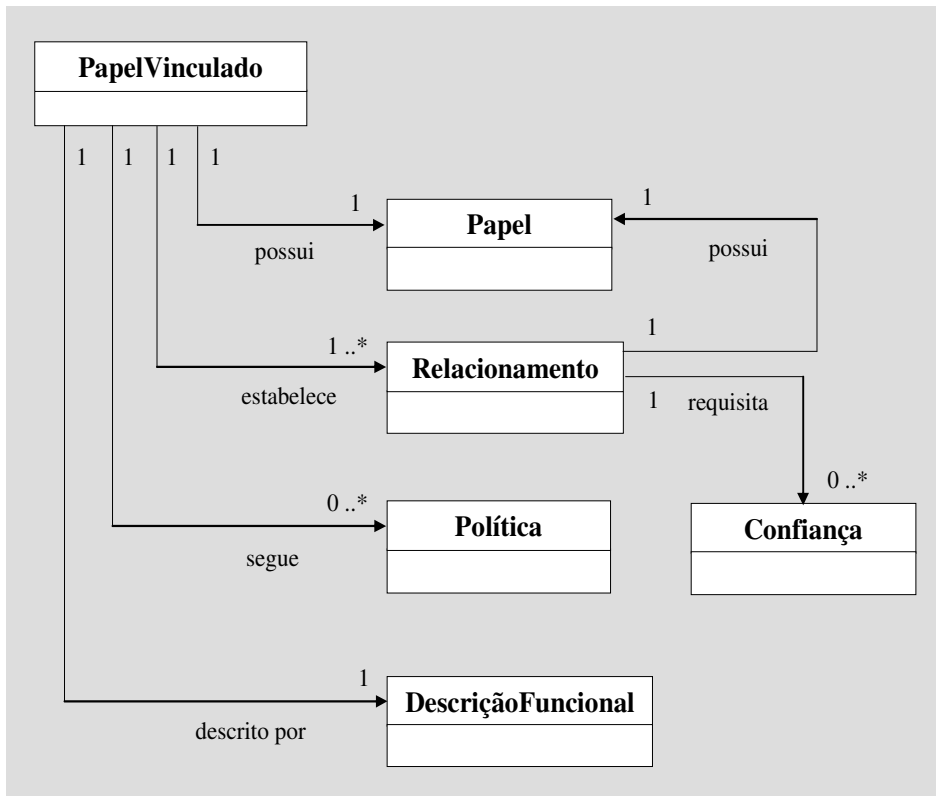


Figura 14 - Papel

Diálogos de negócios ocorrem entre parceiros que desempenham diferentes papéis e que são essencialmente compostos por interações entre parceiros. O modelo básico da arquitetura de *Web Services* considera que existem apenas dois papéis que são desenvolvidos pelos agentes responsáveis pela implementação do serviço: um agente solicitante e um agente fornecedor. O modelo estipula que o *Web Service* é uma funcionalidade que será fornecida por um agente fornecedor a um agente solicitante de acordo com um contrato publicado pelo fornecedor, descrito em WSDL, que deve ser aceito de forma incondicional pelo solicitante. WSDL define um contrato apenas como a parte operacional da interface do serviço. Nesse caso, a semântica do *Web Service* é um modelo do comportamento do serviço, representando o significado e o propósito da interação.

Os papéis de fornecedor e solicitante, no modelo de arquitetura de *Web Services*, podem ser vistos como papéis sintáticos. Eles não dependem da semântica do serviço que está sendo oferecido.

Os papéis sintáticos são de baixo nível e estão associados à execução da tarefa em si — o protocolo para se obter o serviço propriamente dito. Os papéis semânticos estão associados ao negócio a ser estabelecido. No caso de um negócio de compra e venda do produto, os papéis semânticos envolvidos no negócio poderiam ser, por exemplo, comprador, vendedor e entregador.

```
# Class:                RoleBinding
# DatatypeProperty:
# ObjectProperty:      hasRole, hasRelationship+, hasPolicy*, hasFunctionalDescription
:RoleBinding a owl:Class;
  rdfs:subClassOf [a owl:Restriction; owl:onProperty :hasRole; owl:cardinality "1"];
  rdfs:subClassOf [a owl:Restriction; owl:onProperty :hasRelationship; owl:minCardinality "1"];
  rdfs:subClassOf [a owl:Restriction; owl:onProperty :hasFunctionalDescription; owl:cardinality "1"].
:hasRole a owl:ObjectProperty;
  rdfs:domain :RoleBinding;
  rdfs:range :Role.
:hasRelationship a owl:ObjectProperty;
  rdfs:domain :RoleBinding;
  rdfs:range :Relationship.
:hasPolicy a owl:ObjectProperty;
  rdfs:domain :RoleBinding;
  rdfs:range :Policy.
:hasFunctionalDescription a owl:ObjectProperty;
  rdfs:domain :RoleBinding;
  rdfs:range :FunctionalDescription.

# Class:                Relationship
# DatatypeProperty:
# ObjectProperty:      hasRole, hasTrust*
:Relationship a owl:Class;
  rdfs:subClassOf [a owl:Restriction; owl:onProperty :hasRole; owl:cardinality "1"].
:hasRole a owl:ObjectProperty;
  rdfs:domain :Relationship;
  rdfs:range :Role.
:hasTrust a owl:ObjectProperty;
  rdfs:domain :Relationship;
  rdfs:range :Trust.

# Class:                FunctionalDescription
# DatatypeProperty:
# ObjectProperty:      metaModelType
:FunctionalDescription a owl:Class;
  rdfs:subClassOf [a owl:Restriction; owl:onProperty :metaModelType; owl:cardinality "1"].
:metaModelType a owl:DatatypeProperty;
  rdfs:domain :FunctionalDescription;
  rdfs:range xsd:anyURI.
```

Figura 15 - Especificação de papel em N3

O papel que é considerado no modelo conceitual do COWS é o papel semântico desempenhado por cada um dos parceiros no processo de negócios.

Um agente participa de um ou mais contratos, desempenhando um ou mais papéis em cada um dos contratos. Um agente, desempenhando um determinado

papel definido em um contrato específico, estabelece relacionamentos com os outros agentes, que desempenham os demais papéis definidos naquele contrato.

Esses relacionamentos podem exigir seqüências de trocas de credenciais que venham a estabelecer a confiança entre os parceiros. Cada papel tem associado a si a descrição funcional do serviço por ele implementado, que pode estar especificada de acordo com diversos meta-modelos como, por exemplo, OWL-S e WSMO.

A Figura 15 apresenta a representação, em N3, do diagrama da Figura 14.

4.2.4. Política

Diversas políticas — compostas por um conjunto de condições — podem ser especificadas de forma a restringir a possibilidade de agentes, que não satisfaçam às condições estabelecidas, em participar dos contratos. Essas políticas podem ser definidas em diversos níveis: no nível do agente, no nível do contrato em que um agente participa, no nível do papel desempenhado por um agente em um contrato, além de políticas definidas no âmbito dos foros associados aos contratos.

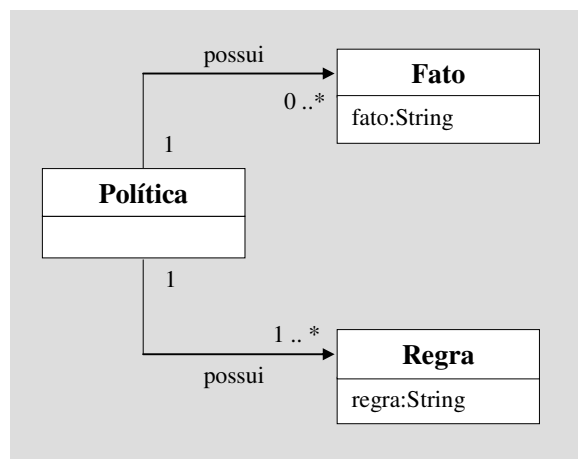


Figura 16 - Política

Como foi visto no capítulo anterior, existem diversas iniciativas que definem formas de se especificar políticas entre diferentes agentes. No presente modelo do COWS, uma política é considerada como um conjunto de fatos e de regras, que definem condições a ser atendidas pelos parceiros de um processo de negócios. Uma política definida por um agente ou um foro estabelece um conjunto de fatos e regras que podem ser aplicadas sobre os dados associados aos outros agentes.

```

# Class: Policy
# DatatypeProperty:
# ObjectProperty: hasFact*, hasRule+
:Policy a owl:Class;
  rdfs:subClassOf [a owl:Restriction; owl:onProperty :hasRule; owl:minCardinality "1"];
:hasFact a owl:ObjectProperty;
  rdfs:domain :Policy;
  rdfs:range :Fact.
:hasRule a owl: ObjectProperty;
  rdfs:domain :Policy;
  rdfs:range :Rule.

# Class: Fact
# DatatypeProperty: factText
# ObjectProperty:
:Fact a owl:Class;
  rdfs:subClassOf [a owl:Restriction; owl:onProperty : factText ; owl:cardinality "1"];
:factText a owl:DatatypeProperty;
  rdfs:domain :Fact;
  rdfs:range xsd:string.

# Class: Rule
# DatatypeProperty: ruleText
# ObjectProperty:
:Rule a owl:Class;
  rdfs:subClassOf [a owl:Restriction; owl:onProperty : ruleText; owl:cardinality "1"];
:ruleText a owl:DatatypeProperty;
  rdfs:domain :Rule;
  rdfs:range xsd:string.
    
```

Figura 17 - Especificação de política em N3

4.2.5. Confiança

Se um contrato utilizado em um processo de negócios tem políticas de controle de acesso, que solicitam a apresentação de credenciais, é desejável que um agente possa indicar quais são as credenciais e os certificados digitais que ele possui, e quais as credenciais e os certificados digitais que esse agente solicita dos outros agentes que participam do negócio. Dessa forma, o processo de busca de parceiros pode filtrar os agentes que não têm possibilidade de satisfazer as políticas de controle de acesso.

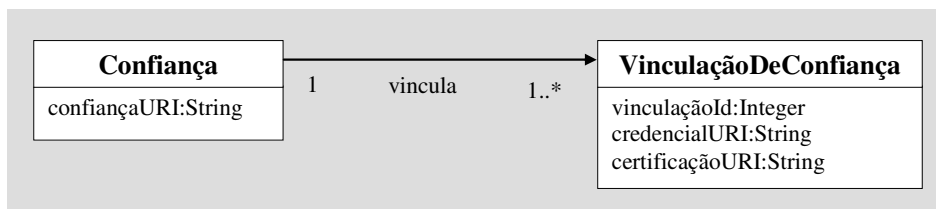


Figura 18 - Confiança

A modelo conceitual do COWS define uma classe especial que permite a um agente especificar quais são as vinculações de credenciais/certificações para

um protocolo de confiança específico (Figura 18). A Figura 19 apresenta a representação, em N3, do diagrama da Figura 18.

```
# Class:          Trust
# DatatypeProperty: trustURI
# ObjectProperty: hasTrustBinding+
:Trust a owl:Class;
  rdfs:subClassOf [a owl:Restriction; owl:onProperty :trustURI; owl:cardinality "1"];
  rdfs:subClassOf [a owl:Restriction; owl:onProperty :hasTrustBinding; owl:minCardinality "1"].
:trustURI a owl:DatatypeProperty;
  rdfs:domain :Trust;
  rdfs:range  xsd:anyURI.
:hasTrustBinding a owl:ObjectProperty;
  rdfs:domain :Trust;
  rdfs:range  :TrustBinding.

# Class:          TrustBinding
# DatatypeProperty: bindId, credentialURI, certificationURI+
# ObjectProperty:
:TrustBinding a owl:Class;
  rdfs:subClassOf [a owl:Restriction; owl:onProperty :bindId; owl:cardinality "1"];
  rdfs:subClassOf [a owl:Restriction; owl:onProperty :credentialURI; owl:cardinality "1"];
  rdfs:subClassOf [a owl:Restriction; owl:onProperty :certificationURI; owl:minCardinality "1"].
:bindId a owl:DatatypeProperty;
  rdfs:domain :TrustBinding;
  rdfs:range  xsd:integer.
:credentialURI a owl:DatatypeProperty;
  rdfs:domain :TrustBinding;
  rdfs:range  xsd:anyURI.
:certificationURI a owl:DatatypeProperty;
  rdfs:domain :TrustBinding;
  rdfs:range  xsd:anyURI.
```

Figura 19 - Especificação de confiança em N3

As credenciais/certificações de um agente e a sua ordem de apresentação são utilizadas como forma de verificar a possibilidade de aproximação de parceiros. Se o protocolo de confiança não pode ser atendido, não existe motivo para que se aproximem parceiros que durante a fase de execução irão descartar a realização do negócio, por falta de confiança.

4.2.6. Foro

No mundo real, negócios são estabelecidos entre parceiros, pessoas ou organizações, de acordo com um ordenamento jurídico. Esse ordenamento jurídico governa contratos que são acordados entre os diversos parceiros e serve como sustentação legal para possíveis ações relativas ao não-cumprimento de obrigações estipuladas nos contratos. Esses contratos têm validade em determinados foros, que por sua vez têm determinadas leis e regras definidas, que são consideradas *default* nos contratos sob sua jurisdição. Algumas dessas regras podem ser redefinidas pelos parceiros, outras são imposições do foro e não podem ser alteradas e têm validade independente de sua especificação no contrato.

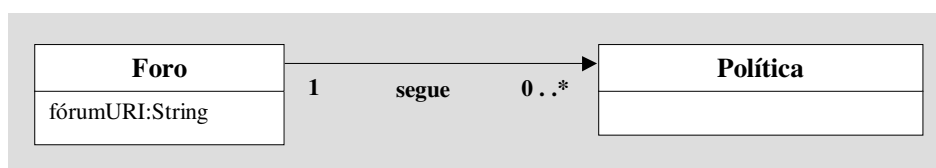


Figura 20 - Foro

Um agente participa de um ou mais contratos, desempenhando um ou mais papéis em cada um dos contratos, de acordo com diretrizes estabelecidas em foros específicos. Esses foros têm associados a si, políticas que devem ser atendidas pelos agentes que participem de contratos sob sua jurisdição (Figura 20). A Figura 21 apresenta a representação, em N3, do diagrama da Figura 20.

```

# Class:          Forum
# DatatypeProperty:  forumURI
# ObjectProperty:   hasPolicy*
:Forum a owl:Class;
  rdfs:subClassOf [a owl:Restriction; owl:onProperty :forumURI; owl:cardinality "1"].
:forumURI a owl:DatatypeProperty;
  rdfs:domain :Forum;
  rdfs:range  xsd:anyURI.
:hasPolicy a owl:ObjectProperty;
  rdfs:domain :Forum;
  rdfs:range  :Policy.
  
```

Figura 21 - Especificação de foro em N3

4.3. O Ambiente Operacional

O ambiente operacional do COWS supõe a existência de agentes que se anunciam como candidatos a participar de contratos que definem processos de negócios, a ser executados sob a regência de coreografias, e aceitam as condições impostas por foros e pelos demais parceiros, todos com um objetivo comum de negócios.

4.3.1. Casamento de Parceiros

Para o anúncio dos agentes é necessária a existência de uma agência de casamento de parceiros de negócios, onde os diversos agentes publicam seus anúncios, suas ofertas de contratos (Figura 22). Os contratos, que definem os papéis que podem ser desempenhados, e a coreografia, que irá reger as interações entre os parceiros durante a fase de execução, podem ser especificados por órgãos independentes de padronização ou por organizações que queiram definir modelos

para processos de negócios de seu interesse. Da mesma forma, foros poderiam ser definidos, de forma a conter políticas de uso comum entre diversas organizações.

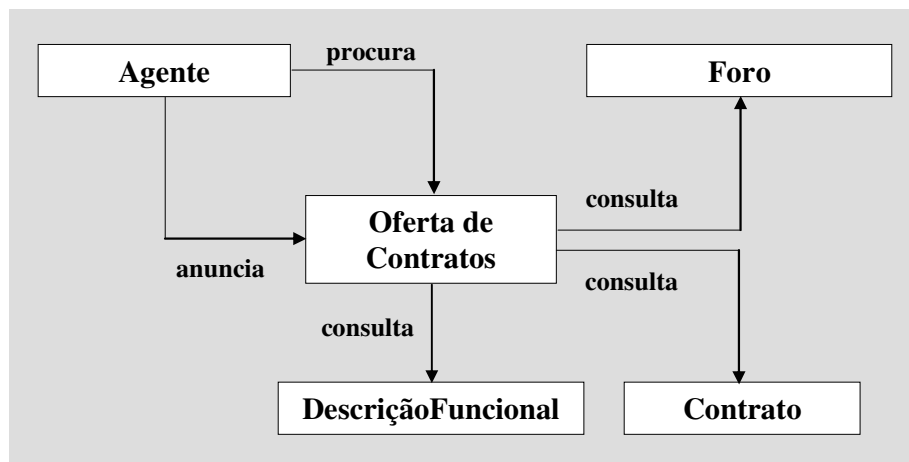


Figura 22 - Ambiente operacional COWS

Essa agência tem como função aproximar parceiros que, dentro das especificações de contrato, foros e políticas, possam participar de processos de negócios, com um objetivo comum. Esse registro deve permitir, além dos anúncios, que agentes possam efetuar buscas por outros parceiros.

Para que um conjunto de agentes seja considerado uma solução para um contrato relacionado a um processo de negócios, eles devem satisfazer as condições quanto aos papéis desempenhados no contrato, às políticas definidas e atender aos protocolos de confiança especificados nos relacionamentos entre parceiros. Se os agentes não satisfazem as possíveis credenciais/certificações e a sua ordem de apresentação, o *matchmaker* do COWS os descarta como parceiros. O *matchmaker* do COWS não verifica nem atesta a validade das informações das credenciais. A apresentação e a verificação das credenciais ocorre durante a fase de execução. Se algum dos agentes não é sincero quanto a essas informações, esse fato será capturado durante a fase de execução.

Uma vez que o *matchmaker* do COWS tenha selecionado um conjunto de soluções envolvendo parceiros em um processo de negócios, esse conjunto é submetido ao *matchmaker* associado à parte funcional do processo. Esse *matchmaker* é responsável pelo casamento dos parceiros a partir das descrições funcionais, que podem estar especificadas de acordo com os diversos meta-modelos existentes, por exemplo, WSDL, OWL-S, WSMO.

O modelo operacional do COWS introduz um nível semântico global de aproximação dos parceiros, que é realizado antes do casamento funcional, como é

realizado nos modelos atuais. Esse casamento semântico introduz uma limitação quanto ao espaço de soluções possíveis, e tem como um de seus objetivos definir, de forma mais precisa, a natureza dos negócios estabelecidos entre os parceiros.

4.3.2. Fase de Execução

Uma vez que o registro, a partir de uma busca iniciada por um agente, encontre uma solução de parceiros para a realização de um negócio, será iniciada a fase de execução do processo de negócios.

A fase de execução do processo de negócios deve ser realizada de forma descentralizada e distribuída, ou seja, não deve existir um controle do processo executado individualmente por nenhum dos parceiros. No contrato deve estar incluída a forma operacional do relacionamento entre os parceiros, ou seja, a sequência de passos para se executar o processo — a coreografia associada dos *Web Services*, no caso das partes automatizadas, e os pontos de sincronização, no caso das partes humanizadas. Essa sequência deve conter as situações de falhas e exceções do processo e eventuais procedimentos de compensação — como desfazer certas operações em caso de falhas.

A coreografia deve, portanto, prever todas as situações que poderão ser executadas pelos parceiros. Não faz parte do escopo deste trabalho a especificação das coreografias.

Da mesma forma que no mundo real não é possível garantir que os parceiros em processos de negócios cumpram com as suas obrigações, não se pode garantir que os parceiros em processos de *E-Business* cumpram com suas obrigações, mesmo que se tenham especificações detalhadas das coreografias. No ambiente do mundo real dos negócios, essas situações — o não-cumprimento de obrigações — são resolvidas em tribunais, onde os litigantes apresentam suas acusações e defesas. Para dar respaldo às acusações/defesas, são apresentadas provas, muitas das vezes na forma de documentos, que possam confirmar as declarações de cada uma das partes.

Uma assinatura não é uma parte substancial de uma transação comercial, mas, de alguma forma, a representa. Podemos listar algumas propriedades decorrentes das assinaturas:

- Evidência: uma assinatura autentica um documento por meio da identificação do assinante com o documento assinado. Quando o assinante faz uma marca de uma forma distinta, esse documento passa a ser atribuído ao assinante.
- Cerimonial: o ato da assinatura faz com que o assinante tenha a percepção do significado legal do ato de assinar e previne, de alguma forma, comprometimentos irresponsáveis.
- Aprovação: em certos contextos definidos na prática ou por lei, a assinatura exprime a aprovação ou a autorização do documento, ou a intenção do assinante de que o documento tem efeito legal.

A formalização das transações dá mais segurança aos parceiros quanto à validade e à execução dos negócios. Na prática, a formalização freqüentemente envolve documentar a transação e assinar ou autenticar os documentos.

Apesar da natureza básica das transações não ter sido modificada, a lei ainda está no início de sua adaptação aos avanços tecnológicos. As comunidades legais e do mundo dos negócios ainda precisam desenvolver regras e práticas que utilizem novas tecnologias para se adaptar ao *E-Business*.

Para se alcançar os propósitos básicos das assinaturas, que foram apresentados acima, uma assinatura necessita ter os seguintes atributos:

- Autenticação do assinante: uma assinatura precisa indicar quem assinou o documento, sendo de extrema dificuldade que outra pessoa possa produzir tal assinatura sem autorização.
- Autenticação do documento: uma assinatura precisa identificar o que está sendo assinado, tornando impraticável a falsificação ou alteração da assinatura ou do objeto da assinatura, sem que esse fato seja detectado.

Existem, atualmente, diversas iniciativas que buscam definir padrões para assinaturas e autenticações digitais, entre elas, a especificação XML Signature da W3C (XML-SIGNATURE, 2002), que fornece um mecanismo flexível de assinatura digital, utilizando a sintaxe XML, e busca garantir a integridade, a autenticação de mensagens e a autenticação de assinaturas em ambientes digitais.

As coreografias, associadas à fase de execução dos processos de negócios, devem incluir a troca de documentos que sejam assinados e autenticados

eletronicamente, de forma a que possíveis litígios, a ser resolvidos em tribunais, possam ser respaldados pela apresentação de comprovações das operações.