

Estudo e análise do processo de desenvolvimento e de ferramentas associados a ebXML

Gustavo Robichez de Carvalho
email:guga@inf.puc-rio.br

Arndt von Staa
email:arndt@inf.puc-rio.br

Puc-RioInf. MCC25/02 Setembro, 2002

Resumo: Em tempos de comércio eletrônico e desenvolvimento de ferramentas que buscam promover a integração de processos e de empresas, os temas processos e ambientes de desenvolvimento para a integração de serviços, aplicações e empresas aparecem como uma área promissora de estudo em engenharia de software.

Entre as iniciativas para o estabelecimento de padrões para a modelagem de processos de negócios está ebXML. O objetivo desta monografia é avaliar a proposta de ebXML e identificar possíveis oportunidades para pesquisa na área de modelagem de processos de negócios. Serão avaliados também alguns ambientes de modelagem com a intenção de identificar alguns requisitos importantes na utilização desta proposta como um processo de desenvolvimento de aplicações e-business.

Palavras-chave: processos de desenvolvimento de software, ferramentas de modelagem de software, comércio eletrônico, XML, ebXML.

Abstract: In the electronic commerce era there are lots of initiatives to promote the integration of processes and enterprises, processes and environments for integration of services, applications and enterprises appear as a promising area of study in software engineering.

Among the business-oriented processes standards initiatives is ebXML. This paper purpose is to evaluate the ebXML proposal and some modeling tools that propose some functionality for the effective use of ebXML proposal as an e-business development process.

Keywords: software development process, software modeling tools, e-commerce, XML, ebXML.

Estudo e análise do processo de desenvolvimento e de ferramentas associados a ebXML

Gustavo Robichez de Carvalho
email:guga@inf.puc-rio.br

Arndt von Staa
email:arndt@inf.puc-rio.br

Resumo: Em tempos de comércio eletrônico e desenvolvimento de ferramentas que buscam promover a integração de processos e de empresas, os temas processos e ambientes de desenvolvimento para a integração de serviços, aplicações e empresas aparecem como uma área promissora de estudo em engenharia de software.

Entre as iniciativas para o estabelecimento de padrões para a modelagem de processos de negócios está ebXML. O objetivo desta monografia é avaliar a proposta de ebXML e identificar possíveis oportunidades para pesquisa na área de modelagem de processos de negócios. Serão avaliados também alguns ambientes de modelagem com a intenção de identificar alguns requisitos importantes na utilização desta proposta como um processo de desenvolvimento de aplicações e-business.

Palavras-chave: processos de desenvolvimento de software, ferramentas de modelagem de software, comércio eletrônico, XML, ebXML.

1. INTRODUÇÃO E MOTIVAÇÃO	3
2. PROCESSOS DE DESENVOLVIMENTO DE SOFTWARE.....	4
2.1 DESVANTAGENS DA NÃO UTILIZAÇÃO DE PROCESSOS DE DESENVOLVIMENTO DE SOFTWARE.....	5
2.2 POR QUE USAR PROCESSOS?.....	5
2.3 PROCESSOS DE DESENVOLVIMENTO E APLICAÇÕES E-BUSINESS	6
3. RUP - RATIONAL UNIFIED PROCESS.....	7
3.1 PRÁTICAS RECOMENDADAS	8
3.1.1 <i>DESENVOLVIMENTO ITERATIVO</i>	8
3.1.2 <i>GERENCIAMENTO DE REQUISITOS</i>	8
3.1.3 <i>ARQUITETURA BASEADA EM COMPONENTES</i>	8
3.1.4 <i>MODELAGEM VISUAL DE SOFTWARE</i>	8
3.2 DIMENSÕES DO PROCESSO.....	9
3.2.1 <i>ESTRUTURA DINÂMICA DO PROCESSO</i>	9
3.2.1.1 Iterações.....	9
3.2.1.2 Fases	9
3.2.2 <i>ESTRUTURA ESTÁTICA DO PROCESSO</i>	10
3.2.2.1 Atividades	11
3.2.2.2 Artefatos	11
3.2.2.3 Disciplinas.....	11
3.3 ALGUNS COMENTÁRIOS SOBRE O RUP.....	14
4. EBXML – ELETRONIC BUSINESS XML	15
4.1 PROCESSO DE DESENVOLVIMENTO UTILIZADO POR EBXML.....	15
4.2 PROCESSOS DE NEGÓCIO.....	17
4.3 LINGUAGEM DE MODELAGEM, PROCESSO DE DESENVOLVIMENTO E MÉTODO.....	19
4.4 CONCLUSÕES	20
5. AMBIENTES DE MODELAGEM DE PROCESSOS DE NEGÓCIOS.....	21
5.1 EIXOS DE EVOLUÇÃO NO DESENVOLVIMENTO DE SOFTWARE	21
5.1.1 <i>EVOLUÇÃO DO NÍVEL DE ABSTRAÇÃO</i>	22
5.1.2 <i>EVOLUÇÃO NO NÍVEL DE FORMALIDADE</i>	22
5.1.3 <i>EVOLUÇÃO NA SATISFAÇÃO DE CRITÉRIOS DE QUALIDADE</i>	23
5.1.3.1 <i>Metáfora : “Errare humanum est”</i>	23
5.1.3.2 <i>Metáfora : “Perseverare in errore est diabolicum”</i>	23
5.1.4 <i>EVOLUÇÃO DOS REQUISITOS</i>	24
5.2 MODELAGEM DE PROCESSOS DE NEGÓCIOS E AMBIENTES DE ENGENHARIA DE SOFTWARE	24
5.3 AVALIAÇÃO DE AMBIENTES DE DESENVOLVIMENTO DE SOFTWARE	25
5.3.1 <i>CAPACIDADE DE DINAMICAMENTE CONSTRUIR E MODIFICAR MODELOS</i>	27
5.3.2 <i>SUPORTE AO APRENDIZADO NA GERAÇÃO DE MODELOS</i>	27
5.3.3 <i>FLEXIBILIDADE NA GESTÃO DE INFORMAÇÕES</i>	28
5.3.4 <i>PROMOÇÃO DE COMUNICAÇÃO DE INFORMAÇÕES</i>	28
5.3.5 <i>FACILITAÇÃO DE VISUALIZAÇÃO</i>	29
5.3.6 <i>AVALIAÇÃO GERAL</i>	29
6. CONCLUSÃO.....	30
7. APÊNDICE A: PROCESSOS COLABORATIVOS UTILIZANDO EBXML.....	31
7.1.1 <i>ILUSTRAÇÃO DE CASO DE USO</i>	34
8. REFERÊNCIAS	37

1. Introdução e Motivação

Em tempos de comércio eletrônico e desenvolvimento de ferramentas que buscam promover a integração de processos e de empresas, os temas processos e ambientes de desenvolvimento para a integração de serviços, aplicações e empresas aparecem como uma área promissora de estudo em engenharia de software.

Um maior grau de automação de processos de negócios entre empresas, a partir do entendimento de regras e restrições intrínsecas a cada instância, pode vir a tornar viável o que muitos chamam de mercado global de oportunidades. Porém estas regras precisam ser estabelecidas e organizadas em um padrão aberto, para que a partir de especificações públicas os participantes possam consultar e publicar informações relevantes para seus negócios.

A concretização de transações para a compra de produtos, informações ou serviços atualmente poderia ser efetuada eletronicamente a partir deste padrão, para isto artefatos precisariam ser gerados para representar a especificação de acordos, a definição dos processos de negócios e a descrição de empresas envolvidas. A partir destas representações seria possível que o fluxo de informações nestas transações e a coordenação de processos fossem feitos de maneira eficiente.

Diversas iniciativas estão sendo desenvolvidas atualmente, porém é de importância vital para o desenvolvimento do comércio eletrônico que seja amadurecido um padrão capaz de organizar, facilitar e promover o comércio entre empresas na Internet.

Ambientes de desenvolvimento são compostos por um conjunto suficiente de ferramentas para modelagem e desenvolvimento, e este precisa estar disponível para que as iniciativas para o desenvolvimento e promoção de padrões tenham abrangência esperada e atinjam a produtividade necessária para sua plena utilização.

Entre as iniciativas para o estabelecimento de padrões e amadurecimento da modelagem de processos de negócios está ebXML[3-9]. ebXML é uma iniciativa de dois grupos, UN/CEFACT[20] e OASIS[12], interessados na promoção do comércio entre empresas e no estabelecimento de padrões utilizando XML[24] respectivamente. Dentre as propostas contidas em ebXML está a utilização do método UMM [21] que dentre suas diretivas propõe o uso de uma instância customizada do Rational Unified Process[16].

O objetivo desta monografia é avaliar a proposta de ebXML e identificar possíveis oportunidades para pesquisa na área de modelagem de processos de negócios. Serão avaliados também alguns ambientes de modelagem com a intenção de identificar a existência de alguns requisitos importantes na utilização desta proposta como um processo de desenvolvimento de aplicações *e-business*.

Com relação a organização da monografia, o segundo capítulo aborda propriedades de processos de desenvolvimento. O terceiro capítulo é um resumo sobre o processo de desenvolvimento RUP, utilizado como origem de alguns conceitos de ebXML. O quarto capítulo resume o processo ebXML, em conjunto com sua análise e alguns tópicos levantados ao longo do estudo. No quinto capítulo são avaliadas ferramentas de modelagem de processos de negócio. No sexto capítulo estão resumidas as

conclusões do estudo e ao final da monografia existe um apêndice onde são descritas algumas motivações para o estudo e desenvolvimento de aplicações de comércio eletrônico.

2. Processos de desenvolvimento de software

Um processo de software define um conjunto coerente de políticas, estruturas organizacionais, tecnologias, procedimentos e artefatos necessários para conceber, desenvolver, implementar e manter um produto de software.

Um processo é um conjunto de passos parcialmente ordenados usado para atingir uma meta previamente estabelecida.

Um processo se bem estabelecido visa aprimorar o grau de competência, técnica e gerencial, que a organização possui para produzir software de boa qualidade, dentro de prazos e custos razoáveis e previsíveis.

O ciclo de desenvolvimento de software define os diferentes estágios no tempo de vida de um produto de software, um processo bem estabelecido deve levar em consideração este ciclo e organizar atividades que auxiliem o desenvolvimento da solução. Na figura 1 está exemplificado o ciclo de desenvolvimento do processo proposto pelo Rational Unified Process[16].



Figura 1. Exemplo de ciclo de vida e disciplinas de um processo de desenvolvimento de software - RUP.

Processos podem ser definidos para atividades como desenvolvimento, manutenção, aquisição e contratação de software.

Mas antes de serem discutidas as vantagens da utilização de um processo de desenvolvimento, é interessante que sejam levantadas algumas desvantagens de organizações ao não utilizarem processos.

2.1 Desvantagens da não utilização de processos de desenvolvimento de software.

Sem um processo de desenvolvimento estabelecido as ferramentas que dão suporte às atividades organizacionais nem sempre ajudam a resolver os problemas. Os procedimentos e padrões, quando existem, são definidos e seguidos de forma burocrática, servindo somente como uma atividade a mais a ser desempenhada pela equipe.

Prejuízos da não utilização de processos ou da utilização de processos ruins são comuns e tendem a ser identificados em diversos projetos. Como exemplo disto podemos citar a ocorrência de equívocos que se repetem de projeto em projeto. Geralmente este tipo de problema ocorre devido à falta de um planejamento eficiente, que tem como consequência a utilização de um número inadequado, geralmente insuficiente ou algumas vezes excessivo, de recursos necessários¹ para se atender a demanda de clientes. Devido a esta mesma falta de planejamento, o trabalho é excessivo e “estressante” sendo comum que as estimativas não sejam cumpridas e que ocorram “corridas” desesperadas contra os prazos de entrega.

Algumas vezes como consequência desta falta de planejamento os produtos gerados são de baixa qualidade podendo apresentar muitos defeitos ou dificuldades para o uso da aplicação. Esta característica pode gerar um certo grau de insatisfação do cliente e isto acarreta em um maior número de reclamações por causa de defeitos reais ou dúvidas quanto ao uso correto da aplicação.

2.2 Por que usar processos?

Além de todas as desvantagens citadas acima, um processo deve promover algumas características que serão enumeradas ao longo deste texto.

Um processo de desenvolvimento pode prever e dar suporte à evolução do software, sendo esta característica muito comum em um ciclo de vida da aplicação. O processo deve ainda difundir conceitos, padrões e tecnologias associados ao ramo de atuação da organização e de qualidade de software. Uma forma de atingir este objetivo é prevendo no processo que estas informações sejam transmitidas através da promoção de palestras, treinamentos, cursos, seminários e publicações.

Um processo de desenvolvimento para ser catalisador de produtividade deve ser compreensível e claro, sendo aceito pelos participantes envolvidos. Além disto é preciso promover o uso de ferramentas harmoniosas² que dêem suporte efetivo a suas atividades, gerando um provável aumento de produtividade das partes envolvidas.

Com relação à compreensibilidade e clareza, um processo deve estar definido explicitamente e os participantes envolvidos devem compreendê-lo com o máximo de facilidade possível, sendo que todas as atividades culminam em resultados claros de forma que o progresso do processo é visível externamente.

¹ Entende-se como recursos necessários para o desenvolvimento de um projeto, as pessoas capazes de desempenhar tarefas para gerar a solução e as ferramentas ou equipamentos utilizados para o auxílio a estas tarefas.

² Ferramentas harmoniosas formam um conjunto em que uma complementa a outra.

Um processo é necessário a uma organização, sendo cada vez mais comum que ambientes de desenvolvimento de software estejam sendo desenvolvidos e lançados no mercado, sem que ao mesmo tempo eles estejam seguindo algum padrão ou normas organizacionais já estabelecidas através de um processo ou ainda sem a menor possibilidade de personalização de atividades para que estas adaptações reflitam práticas e normas destas organizações. Com a utilização destas ferramentas sem normas ou sem a possibilidade de adaptação quase nada funcionará e a produtividade geralmente será baixa, podendo até ocorrer ilhas de excelência que na prática não conseguem se propagar pela organização.

Na prática um processo deve servir como orientação às diversas atividades, recursos e elementos de uma organização. Isto se dá através da definição ou especificação de artefatos que devem ser desenvolvidos, através do planejamento e atribuição de tarefas a elementos individuais ou a equipes.

Quando se fala em processo de desenvolvimento de software se faz referência a três palavras chaves: *compreensão, disciplina e competência*[25]; pois estes conceitos são fatores cruciais para o sucesso no desenvolvimento de um projeto. Porém é preciso ter cuidado na fase de avaliação, configuração e adaptação de um processo pois o ato de “documentar” não basta para “compreender”, a “formalidade” não é equivalente a “disciplina” e o “processo” não é equivalente a “competência”.

As aplicações de software tendem a aumentar bastante a sua sofisticação e complexidade colaborando para a utilização de processos de desenvolvimento de software. Com isto, conviver nesta perspectiva sem regras e políticas estabelecidas em processos de desenvolvimento tenderá a diminuir a produtividade da equipe, podendo acarretar em um grave aumento no risco de desenvolvimento de aplicações.

Na realidade o estabelecimento de um processo de desenvolvimento em uma organização é uma tarefa árdua, sendo importante para isto, a capacidade e o conhecimento dos profissionais envolvidos nesta atividade. É de suma importância que se conheça muito bem não só a tecnologia, o método e o processo a serem implantados, como também a organização e o domínio de aplicações que está sendo tratado.

2.3 Processos de desenvolvimento e aplicações e-business

A realidade de projetos de desenvolvimento de aplicações e-business não é muito diferente de outros domínios de aplicação em engenharia de software, os diversos problemas levantados anteriormente precisam ser corretamente avaliados e um tratamento adequado deve ser dado para evitar perdas de produtividade.

Um fator relevante específico de processos de desenvolvimento de aplicações e-business é a velocidade com que as transformações ocorrem, o que acaba por gerar maiores complicações no que se refere a evolução e manutenção de software. Para conseguir atingir níveis adequados de produtividade nada melhor que um processo de desenvolvimento que contenha um conjunto de diretrizes que organizem o projeto e seu conjunto de disciplinas para o planejamento das atividades.

Outro fato importante relativo à modelagem e ao desenvolvimento de aplicações *e-business* é o aumento na quantidade de participantes envolvidos nas interações,

podendo isto gerar uma maior demanda de esforço no processo de coordenação de atividades e no estabelecimento de padrões para a comunicação de representações, estas últimas terão como finalidade a especificação dos processos de negócios ou informações envolvidas nas transações.

ebXML fornece justamente direções e recomendações para a organização de atividades, para a especificação e disponibilização de infra-estrutura básica baseada em componentes e para o estabelecimento de formas e padrões para a comunicação de especificações de parceiros de negócios.

Para compreender o processo proposto em ebXML na próxima seção será abordado o RUP[16], um framework para processos de desenvolvimento de software proposto pela Rational[15] e utilizado em ebXML como núcleo do processo de desenvolvimento para aplicações e-business proposto pela UN/CEFACT e OASIS.

3. RUP - Rational Unified Process

Nesta seção será abordado o RUP[16], um framework para configuração e adaptação de processos de desenvolvimento de software proposto pela Rational[15] e utilizado em ebXML como núcleo do processo de desenvolvimento para aplicações e-business desenvolvido pela UN/CEFACT e OASIS proposto em UMM[21].

O RUP é um processo de desenvolvimento de software proposto pela Rational. É uma abordagem disciplinar, que ao atribuir tarefas e responsabilidades em uma organização, visa aumentar a produtividade de uma equipe, buscando aprimorar o processo de desenvolvimento, facilitando a administração da complexidade do projeto e a manutenção da integridade do sistema.

O RUP é um *Framework*, isto é, para cada organização em que ele venha a ser utilizado é preciso que sejam feitos estudos detalhados das políticas, práticas estabelecidas e dos objetivos da organização para que a partir destas informações seja possível adaptar a proposta a realidade, necessidades e aplicações de cada empresa gerando ao final um processo “personalizado”. Na prática isto ocorre pois não existe uma solução aplicável a todas organizações e nenhum processo de desenvolvimento de software é adequado para a solução de todos os problemas em qualquer domínio de aplicação.

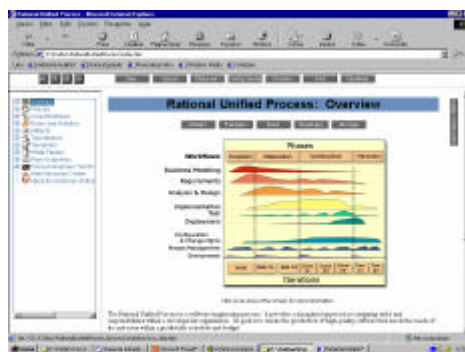


Figura 2.1 : Visão Geral de Disciplinas, Fases e Iterações do RUP



Figura 2.2: Ciclo de vida de uma solução

O propósito defendido pela Rational em se utilizar o RUP é assegurar um excelente nível de qualidade na produção de artefatos de software, aumentando a produtividade de uma equipe dentro de uma organização. A gerência de cronogramas e orçamentos de um projeto é feita em conjunto com a disponibilização de uma base de conhecimento, onde cada membro da equipe terá acesso a guias, modelos e mentores de ferramentas.

No RUP existe o conceito de atividades, onde estas criam e mantêm artefatos. Estes artefatos podem ser modelos que utilizam e promovem o uso de UML - Unified Modeling Language [19]. Em linhas gerais, o RUP é dirigido a *Use Cases*, isto é, este diagrama de UML é utilizado como linguagem para troca de conhecimento entre equipes e entre disciplinas em um projeto de desenvolvimento.

3.1 Práticas Recomendadas

Nesta subseção serão descritas algumas práticas recomendadas pelo RUP para aplicação em processos de desenvolvimento de software. As práticas “Verificação contínua da qualidade” e o “Controle de mudanças de artefatos” não serão descritas por não serem abordadas nesta monografia.

3.1.1 Desenvolvimento Iterativo

Geralmente o desenvolvimento de sistemas complexos gera dificuldades que podem ser identificadas em diferentes áreas e fases. Ao se utilizar um processo iterativo estas dificuldades são resolvidas através de um entendimento incremental do problema, isto ocorre porque através de refinamentos sucessivos a solução tende a atender às necessidades levantadas, sendo que a solução final só é desenvolvida após múltiplas iterações.

3.1.2 Gerenciamento de requisitos

A prática de gerenciamento de requisitos no RUP se dá a partir da descrição de como fazer a elicitação, organização e documentação de restrições e funcionalidades. Geralmente são utilizados *use cases* e *cenários* como principais artefatos no processo de desenvolvimento.

3.1.3 Arquitetura baseada em componentes

Um ponto importante no processo proposto pelo RUP é gerar uma arquitetura baseada em componentes. Estes componentes não são módulos triviais, mas sim subsistemas que desempenham uma função bem definida. Componentes ao serem utilizados em uma arquitetura promovem um maior grau de flexibilidade, pois cada parte do subsistema possui uma semântica própria, podendo a arquitetura ser organizada para que mudanças em componentes sejam previstas e devidamente acomodadas através do planejamento e organização de atividades.

3.1.4 Modelagem Visual de Software

É proposto o uso intensivo de modelagem visual, isto é, são feitas representações gráficas visando capturar a estrutura e o comportamento da arquitetura e de seus componentes. As representações gráficas são utilizadas para promover a compreensão

do problema e de sua respectiva solução, visando diminuir ambigüidades na comunicação.

Como linguagem de comunicação padrão em um projeto é proposto e incentivado o uso de UML, porém é permitida a criação de alguns modelos de documentos para complementar a comunicação de informações.

3.2 Dimensões do processo

O RUP é descrito em duas dimensões ou eixos que podem ser vistos na figura 3. O eixo horizontal representa o tempo e demonstra aspectos dinâmicos do processo, podendo ser representado pelos *Ciclos*, *Fases*, *Iterações* e *Pontos de Controle* existentes na instância do processo. O eixo vertical representa aspectos estáticos do processo, sendo composto por *Atividades*, *Artefatos*, *Papéis* e *Disciplinas*.

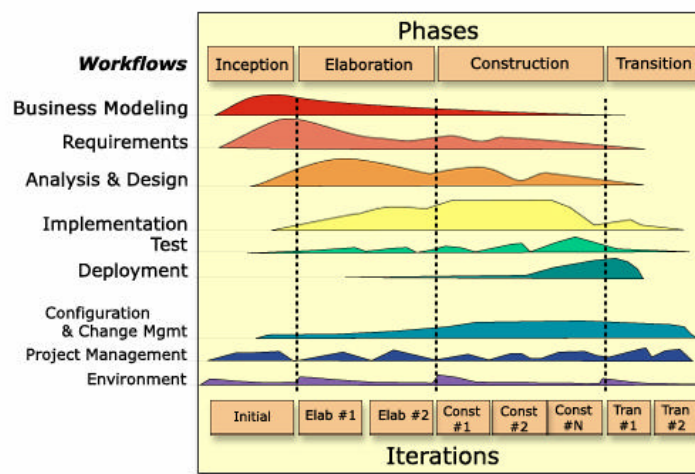


Figura 3: Dimensões do RUP

3.2.1 Estrutura dinâmica do processo

3.2.1.1 Iterações

Cada fase no RUP é dividida em diversas iterações sendo que a construção da solução ocorre de forma incremental. Nesta abordagem iterativa os riscos são identificados o mais cedo possível sendo assim as mudanças são mais fáceis de serem gerenciadas. Além disso o aprendizado ocorre ao longo da iteração visando sempre a melhoria na qualidade.

3.2.1.2 Fases

Sob o ponto de vista gerencial o ciclo de vida no RUP é dividido em quatro fases sequenciais. As fases de *concepção*, *elaboração*, *construção* e *transição*. Todas as fases são concluídas com pontos de controle. Na figura abaixo pode ser vista uma ilustração da demanda por recursos de cada fase e possíveis pontos de controle em cada fase.

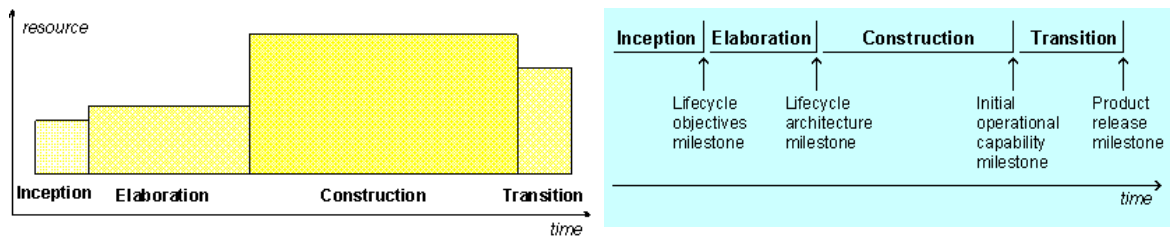


Figura 4 : Fases do processo de desenvolvimento RUP

3.2.1.2.1 Fase de Concepção

Durante esta fase é determinado o escopo do projeto em conjunto com a identificação de atores e a definição da natureza da interação. Para isto é feito um plano de negócio contendo critérios de sucesso, tópicos para avaliação de riscos, estimativas de recursos e cronogramas com os principais pontos de controle para o desenvolvimento do projeto.

3.2.1.2.2 Fase de Elaboração

Esta fase tem como propósito analisar o domínio do problema. Neste ponto se obtém uma noção do todo sem se ater a detalhes. É nesta fase que são estabelecidos os fundamentos arquiteturais, buscando sempre eliminar os principais fatores de risco.

Geralmente esta fase é a mais crítica do processo de desenvolvimento, pois são feitas diversas decisões arquiteturais, como o escopo do projeto, suas funcionalidades principais, seus requisitos não funcionais e seus requisitos de performance.

3.2.1.2.3 Fase de Construção

Esta fase tem como propósito o desenvolvimento de componentes e funcionalidades, sua integração com os demais componentes, a realização de testes, o gerenciamento de recursos e a otimização do controle operacional através do monitoramento e controle de custos, tempo de projeto e qualidade. O trabalho a ser executado nesta fase é dependente da qualidade da arquitetura definida na fase de elaboração.

3.2.1.2.4 Fase de Transição

A fase de transição ocorre quando a *baseline* está estável para ser entregue ao usuário final, isto ocorre quando existe um nível de qualidade aceitável, a documentação já está disponível e já foram corrigidos os problemas críticos do sistema.

3.2.2 Estrutura estática do processo

Um plano de desenvolvimento de software descreve **quem** está fazendo **o quê**, ou **quem** está desempenhando que **papéis** para a geração de determinado **artefato**, ou por fim de que forma e quais **atividades** são utilizadas para atingir um **objetivo** disciplinando o uso de **recursos** e **atividades** ao longo de um projeto.

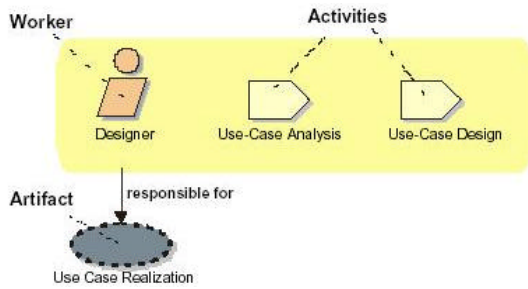


Figura 5.1 : Estrutura estática do processo RUP

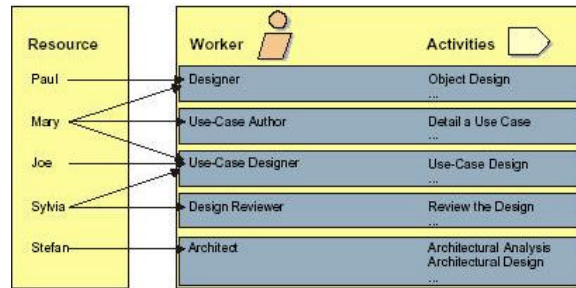


Figura 5.2: Exemplo de associação de recursos a papéis

O papel é uma definição abstrata para um conjunto de atividades realizadas e os artefatos gerados por ela. Um papel pode ser realizado por uma pessoa ou uma equipe e corresponde a responsabilidades e comportamentos que cada pessoa tem dentro de uma organização.

3.2.2.1 Atividades

Atividade é uma unidade de trabalho atribuída a um papel. A atividade deve ter um propósito claro visando a criação e atualização de artefatos e deve ser utilizada como elemento para planejamento e controle de progresso. Um exemplo de atividade é *Planejar uma iteração*, onde o papel sendo desempenhado é de *Gerente de Projeto*.

As atividades estão estritamente ligadas a artefatos, estes podem ser vistos como entradas ou saídas de uma atividade, servindo assim como mecanismo de comunicação entre elas.

3.2.2.2 Artefatos

Um artefato é um produto tangível que é produzido, modificado ou utilizado pelo processo. Artefatos podem ser modelos, elementos de modelo, documentos, código fontes e executáveis, podendo ser expressos visualmente ou textualmente. Na figura 7 estão algumas ilustrações de artefatos que podem ser obtidos a partir do processo de desenvolvimento proposto.

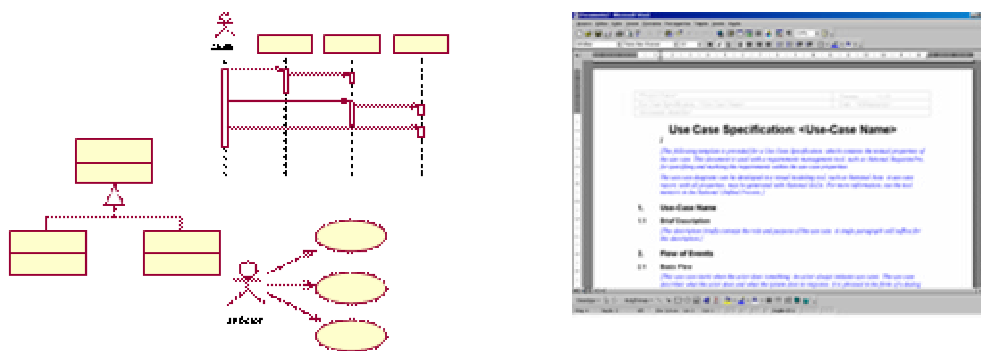


Figura 6: Exemplo de artefatos

3.2.2.3 Disciplinas

Disciplinas são seqüências de atividades que produzem artefatos e esclarecem as interações entre papéis. Abaixo são exemplificados os relacionamentos entre um conjunto de papéis e suas atividades desempenhadas.

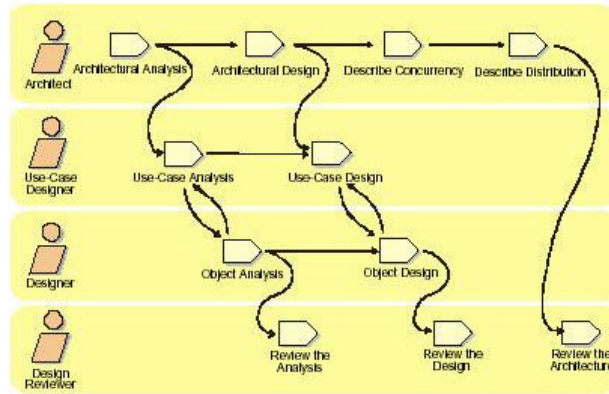


Figura 7 : Atribuição de atividades a papéis

Existem nove disciplinas no RUP, são elas: *Modelagem de negócio, Requisitos, Análise e Design, Implementação, Teste, Deployment, Gerenciamento de Mudanças e Configuração, Gerência de Projetos e Ambiente*. A figura 8 ilustra o relacionamento provável entre as disciplinas e as fases do processo de desenvolvimento. Em seguida serão descritas as disciplinas propostas pelo RUP.

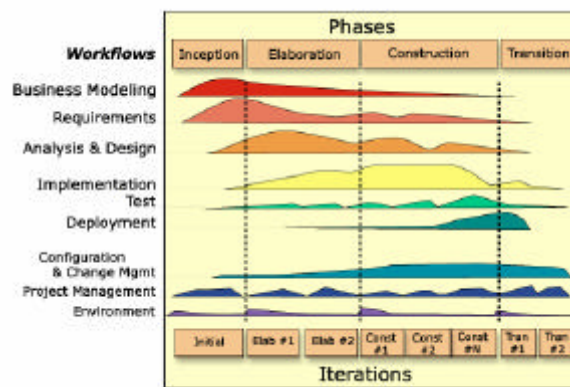


Figura 8 : Disciplinas do RUP

3.2.2.3.1 Modelagem de Negócio

Esta disciplina tem como propósito entender a estrutura e a dinâmica da organização, identificando os problemas atuais e analisando possíveis melhorias. A partir dela é possível derivar os requisitos necessários para atender os objetivos, provendo uma linguagem comum para ambas as comunidades, engenharia de negócio e a engenharia de software. Como principais atividades podem ser citadas o ajuste de expectativas, a captura do vocabulário comum de negócio, a definição de atores e use cases de negócio, a identificação de entidades de negócio e trabalhadores e o detalhamento de use cases de negócio.

3.2.2.3.2 Requisitos

Esta disciplina tem como propósito descrever o quê o sistema deve fazer, promovendo o entendimento dos requisitos aos desenvolvedores, permitindo que os desenvolvedores e clientes cheguem a um acordo. Ao longo desta disciplina é

definida a fronteira do sistema, sendo possível a partir disto, oferecer uma base para a estimativa de custo e tempo de desenvolvimento do projeto.

Nesta disciplina são elicitadas, organizadas e documentadas também as funcionalidades a serem desenvolvidas e restrições levantadas, é gerada uma documentação contendo os desafios, as decisões e a definição do quê o sistema deverá ou não fazer.

As principais atividades desta disciplina são o desenvolvimento do documento de visão, a captura de um vocabulário comum, a identificação de atores e use cases e a criação de um protótipo de interface.

3.2.2.3.3 Análise e Design

Esta disciplina tem como propósito transformar os requisitos levantados em modelos para que os desenvolvedores possam efetuar a implementação do sistema. É nesta etapa que é desenvolvida a arquitetura para o sistema, adaptando o design tornando-o compatível com o ambiente de implementação e projetando questões relativas à performance. Como principais atividades é possível destacar a *Análise Arquitetural*, a *Análise de Use Case*, a *Identificação de Mecanismos de Design*, o *Design do Use Case*, o *Design de Classes* e o *Design da Base de Dados*.

3.2.2.3.4 Implementação

Esta disciplina tem como propósito definir a organização dos artefatos de código que serão implementados, eles são organizados em camadas ou subsistemas de implementação, assim é estruturado o modelo de implementação do projeto, sendo feito o planejamento da integração do sistema, a implementação e integração de componentes.

3.2.2.3.5 Testes

Esta disciplina tem como propósito verificar a interação entre os objetos, a integração entre os diversos componentes, e se todos os requisitos foram implementados corretamente. Ao longo da disciplina é utilizada uma abordagem iterativa, sendo os testes efetuados ao longo do projeto, para que os erros sejam detectados o mais cedo possível e para que haja uma redução no custo de correção de defeitos. As dimensões de qualidade avaliadas são confiabilidade, funcionalidade, performance da aplicação e performance do sistema. Na fase de concepção são gerados a partir desta disciplina os casos de testes a serem utilizados ao longo do desenvolvimento do software.

3.2.2.3.6 Deployment

A disciplina de deployment tem como propósito descrever as atividades associadas a entrega do produto ao usuário. Resumidamente esta fase objetiva produzir versões de produtos, as chamadas releases porém possui como principais atividades, o planejamento da distribuição do produto.

3.2.2.3.7 Gerenciamento de configuração e mudanças

Esta disciplina tem como propósito descrever como controlar os inúmeros artefatos produzidos por diversas pessoas que trabalham em um mesmo projeto, visando evitar

e gerenciar questões como atualizações simultâneas, notificações de alterações e múltiplas versões. Sendo importante para isto a identificação dos itens de configuração, a partir destes itens é definida uma política para o estabelecimento, ordenamento, controle e exame sobre as mudanças efetuadas

A principal atividade desempenhada é o estabelecimento de políticas de gerenciamento de configuração e do processo de controle de mudanças. Além da instalação do ambiente de desenvolvimento, são criadas *baselines*, e o seu uso deve ser efetivamente promovido entre as equipes envolvidas no processo.

3.2.2.3.8 Gerenciamento do projeto

O propósito principal desta disciplina é prover guias para o planejamento e o gerenciamento de riscos, mais especificamente, com relação ao planejamento são feitos esforços relativos à alocação de recursos, execução e monitoramento de atividades.

É nesta disciplina que será exercida a arte de balancear prioridades, objetivos competitivos e conflitantes, os riscos associados a cada atividade do projeto e as necessidades e prioridades de usuários

3.2.2.3.9 Ambiente

Esta disciplina tem como propósito dar enfoque às atividades necessárias para a configuração, escolha e manutenção do processo e ferramentas de suporte ao desenvolvimento. Aqui são enfatizadas as atividades para a configuração do processo no contexto da organização e do projeto.

3.3 Alguns comentários sobre o RUP

O RUP é um processo de desenvolvimento de software complexo, e difícil de entender, implantar e aplicar, muitas vezes não é fácil identificar o que é essencial e o que não é, por onde começar e o que fazer em seguida; o que pode acarretar problemas para os seus usuários.

O sucesso da aplicação de um processo de desenvolvimento como RUP é dependente do conhecimento da equipe com relação a tecnologia, método e práticas impostas e principalmente é necessário que se conheça muito bem a organização onde está sendo implantado o processo para que seja possível evitar ao máximo a imposição de atividades burocráticas tornando assim o processo útil a organização.

A quantidade de artefatos gerados durante o processo pode ser tão grande que muitas vezes é complicado manter o foco no objetivo principal que é a construção de software.

4. ebXML – Eletronic Business XML

Esta seção irá discorrer sobre a proposta de ebXML, dando ênfase ao processo de desenvolvimento proposto e a análise da instância do RUP gerada, analisando porque algumas fases do RUP não foram redefinidas e quais as contribuições geradas pela abordagem.

Abaixo será descrito o processo de desenvolvimento e a proposta apresentada em ebXML. Em seguida será fornecida uma definição para o conceito de processos de negócios que será útil para o entendimento do tema UMM[21], que vem a ser o método utilizado em ebXML e ao final desta seção serão apresentadas algumas conclusões sobre o estudo da proposta. Além disto no *Apêndice A* existe uma pequena descrição sobre a motivação do desenvolvimento de ebXML.

4.1 Processo de desenvolvimento utilizado por ebXML

A visão promovida por *ebXML* é:

Através da utilização de ebXML será possível criar um mercado eletrônico virtual e global onde empreendimentos de qualquer tamanho e em qualquer localização geográfica possam estabelecer e conduzir negócios entre si através de troca de mensagens.

ebXML visa permitir a qualquer um, em qualquer lugar, o estabelecimento de negócios eletronicamente, entretanto, se antecipa que a adoção dos vários componentes ebXML pode ser feita de maneira gradual e incremental ao longo do tempo.

Sob a ótica de engenharia de software, para tornar este processo de desenvolvimento produtivo é preciso que se estabeleçam diretrizes e formas de trabalho visando sobretudo a organização das atividades e o estabelecimento de normas, especificações ou direcionamentos que facilitem este processo de desenvolvimento.

A proposta do consórcio além de gerar um conjunto de recomendações e padrões que devem ser seguidos, especificam a estrutura de repositórios comuns e de processos de negócios para a disponibilização de padrões de representações e componentes, dentre outros, porém estes tópicos não serão o foco da monografia e para os interessados em maiores detalhes são recomendadas algumas referências no final da monografia[3-9].

Resumidamente ebXML propõe um padrão arquitetural de integração de processos de negócios, esta integração é possível através do uso de um conjunto de artefatos para a especificação do comércio eletrônico entre empresas. Para atingir seus objetivos ebXML oferece também um método UMM [21] que organiza a utilização e elaboração de representações. Este método prevê a modelagem tanto de processos de negócios, quanto das informações que serão trocadas entre os participantes do processo de negócio.

O ponto forte da arquitetura proposta por ebXML é promover um framework colaborativo para comércio eletrônico³. Esta proposta permite que empresas

³ Entenda-se framework colaborativo como uma abstração para o processo de desenvolvimento e o processo de execução de software.

trabalhem em conjunto para a especificação de processos de negócios, descoberta de parceiros, estabelecimento de acordos para colaboração e a execução de processos de negócios. Estas atividades significativas formam um ciclo dentro do processo de desenvolvimento e na prática são elas que “implementam” e “executam” ebXML, uma representação deste ciclo está ilustrada na figura 9.



Figura 9. ebXML : Framework colaborativo para o desenvolvimento de aplicações e-business.

O processo de desenvolvimento proposto por ebXML tem início na fase de **Definição do Processo de Negócio** e sua descrição em conjunto com as demais atividades será feita abaixo:

A atividade de **Definição do Processo de Negócio** procura analisar a comunidade de parceiros para estabelecer os processos de negócios a serem utilizados em sua comunidade. Estes processos de negócios são definidos de acordo com algum meta-modelo conhecido. Em *ebXML* por exemplo, estas descrições estarão expressas em *UML* e *XML*[24].

A atividade de **Descoberta de Parceiros de Negócios** prevê o processo de descoberta de negociadores em uma comunidade eletrônica ou mercado virtual. Estes parceiros irão fazer negócios e utilizar os repositórios públicos para efetuar consultas e obter informações sobre possíveis parceiros de negócios.

A atividade de **Estabelecimento de Acordos** é a fase em que acordos, compromissos e regras entre parceiros de negócios são estabelecidos.

A atividade de **Desenvolvimento de Infra-estrutura** é a fase em que cada negociador configura interfaces e aplicativos para a futura execução dos processos de negociação de acordo com os compromissos estabelecidos e com processos de negócios e detalhes de interface do parceiro estabelecidas .

A atividade de **Execução do Processo de Negócio** corresponde a execução das transações propriamente ditas, resumidamente as empresas irão trocar documentos e informações de acordo com o processo de negociação estabelecido e poderão chegar ou não a um acordo.

A atividade de **Gerenciamento do Processo de Negócio** prevê o monitoramento e coordenação da execução dos processos de negociação definidos na fase **Definição do Processo de Negócio** e acordados na fase **Estabelecimento de Compromisso**.

A atividade de **Evolução de Processo de Negócio** prevê a evolução das especificações e software utilizados, é nesta etapa que serão avaliados os processos de negócio existentes, eles serão melhorados através de reengenharia de processos. Esta atividade pode ter como consequência a modificação ou criação de novos processos de negócio para atender às necessidades de mercado. Após a fase de **Evolução de Processos de Negócio** um novo ciclo de desenvolvimento tem início na atividade de **Definição de Processos de Negócio**.

Na próxima seção será feita a conceituação de processos de negócios para que seja possível o entendimento do método proposto e também sua análise.

4.2 Processos de Negócio

Um processo de negócio consiste em um série de atividades estruturadas e pelo conjunto de informações que são trocadas entre os parceiros de negócios para atingir determinado objetivo. É transparente que a complexidade de um processo de negócio, pode variar desde uma atividade que envolva o envio de um correio eletrônico, até todo o conjunto de atividades de uma complicadíssima cadeia de produção de equipamentos eletrônicos.

Processos de Negócios ocorrem quando entidades autônomas colaboram para alcançar um objetivo comum.

Para lidar justamente com a complexidade são estabelecidas informações desde a forma de execução de um conjunto de atividades, por exemplo quantidade de vezes que ela ocorre, atividades precedentes, atividades sucessoras, o seqüenciamento de atividades, a possibilidade de execução em paralelo, condições. A coordenação destas atividades é chamada de coreografia ou orquestração de processos de negócios e cada participante desempenha um papel com tarefas e restrições bem definidas.

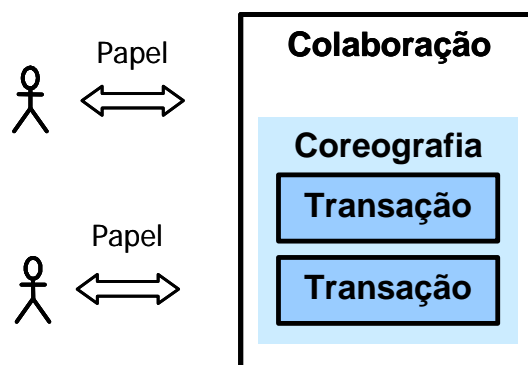


Figura 10. Coreografia entre participantes.

Abaixo pode ser observada uma figura que ilustra conceitualmente uma definição recursiva para processos de negócio. Um processo de negócio pode ser composto de colaborações e estas possuem transações atômicas responsáveis pela realização de atividades, que podem ser por exemplo a troca de documentos entre as partes.

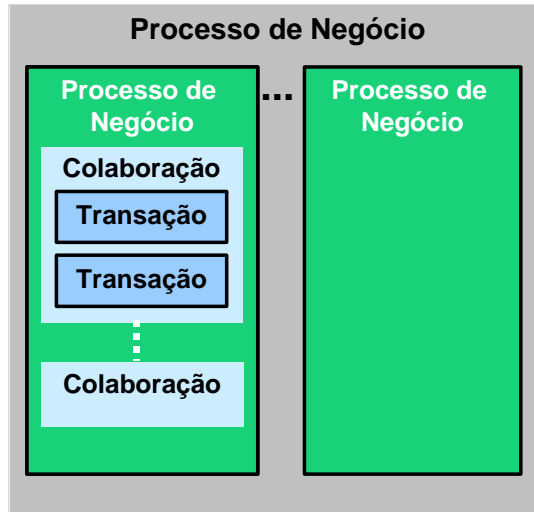


Figura 11. Processos de Negócios :Ilustração conceitual.

Processos de negócio descrevem a interoperabilidade entre aplicações promovendo a colaboração entre parceiros.

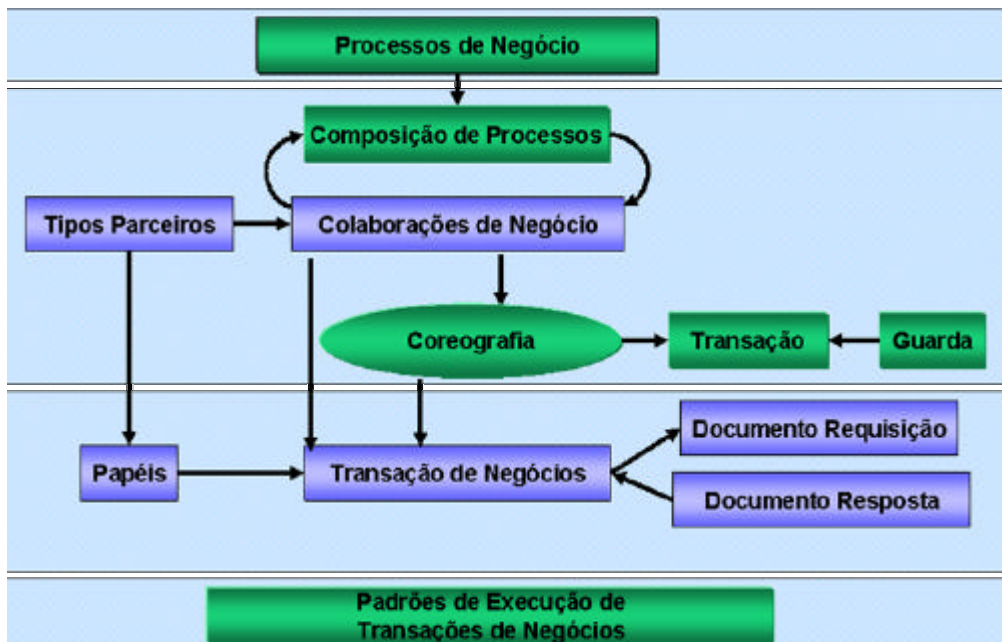


Figura 12. Modelo conceitual da representação necessária para processos de negócios.

Sem entrar em detalhes, um meta-modelo possível para a representação de processos de negócios pode ser observado na figura 12.

4.3 Linguagem de Modelagem, Processo de Desenvolvimento e Método

A modelagem de processos de negócios pode ser uma atividade complexa o suficiente para inviabilizar oportunidades de negócios. Para procurar minimizar esta dificuldade em ebXML foi proposta a utilização do método de desenvolvimento UN/CEFACT Modelling Methodology (UMM).

As principais metas de UMM são a definição de padrões, o suporte a modelagem e desenvolvimento de processos de negócio. UMM é uma método proposto como instância de um conjunto restrito de disciplinas do Rational Unified Process [16], a figura 13 detalha um pouco melhor este relacionamento.

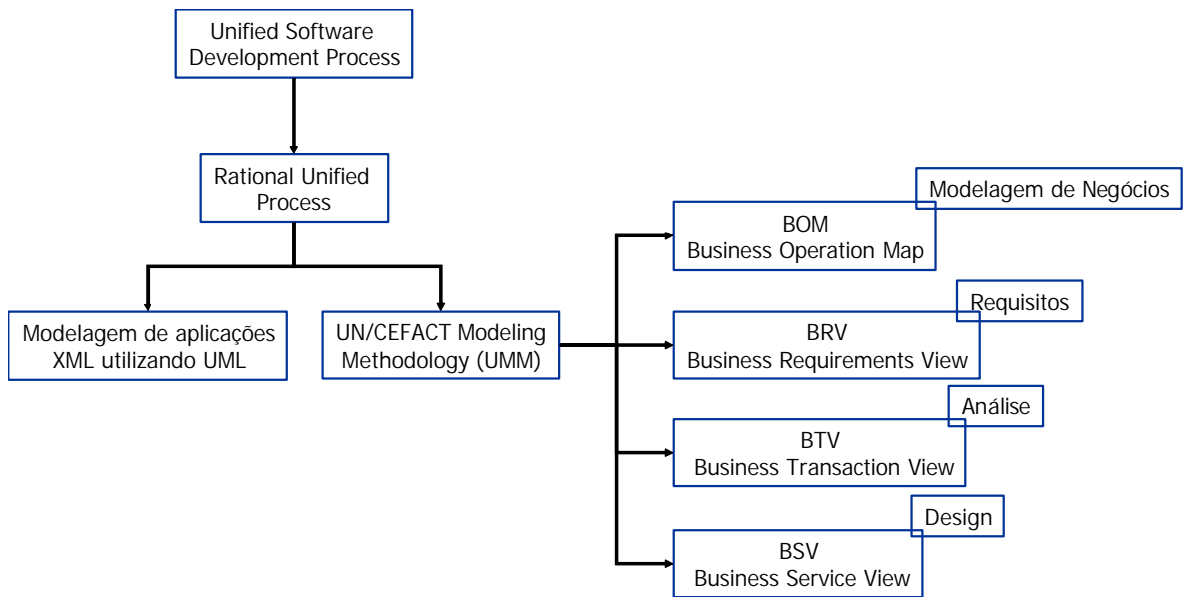


Figura 13. Relacionamento entre RUP e UMM

Como linguagem de modelagem para a definição de processos de negócios é utilizado parte do conjunto de diagramas disponível em UML e para representação de modelos podem ser utilizados UML ou XML.

Como características para o processo de desenvolvimento e seguindo as diretrizes do RUP, UMM prevê um desenvolvimento iterativo, dirigido a use case, é feito uso extensivo de modelos em UML e seu desenvolvimento é baseado em componentes. Abaixo serão descritos os artefatos gerados em ebXML e sua relação com o RUP.

O artefato *Business Operation Map* de UMM é gerado a partir da especialização da disciplina de modelagem de negócios e tem como principal objetivo gerar modelos que descrevam os processos de negócios.

O artefato *Business Requirements View* de UMM é gerado a partir da especialização da disciplina de requisitos e tem como principal objetivo a descrição de colaborações de negócios.

O artefato *Business Transaction View* de UMM é gerado a partir da especialização da disciplina de análise e tem como principal objetivo a descrição de transações de negócios.

O artefato *Business Service View* de UMM é gerado a partir da especialização da disciplina de design e tem como principal objetivo a descrição da dinâmica das colaborações de e-business e no design da troca de informações entre os parceiros de negócios.

O conjunto de disciplinas restantes não são abordadas em UMM pois elas não são disciplinas específicas de processos de desenvolvimento de aplicações e-business, porém elas devem ser abordadas como em qualquer projeto de desenvolvimento tradicional.

UMM mantém o conceito de fases proposto pelo RUP, portanto as fases de concepção, elaboração, construção e transição serão utilizadas para a organizar o controle do processo de desenvolvimento. Em cada fase é utilizado também o conceito de pontos de controle, onde em cada ponto será realizada uma avaliação para determinar se os objetivos descritos inicialmente foram atingidos, em conjunto com uma reavaliação dos riscos do projeto para que se avalie o prosseguimento para uma nova fase.

4.4 Conclusões

ebXML propõe um processo e um método para desenvolvimento de aplicações e-business que tende a tornar mais produtivo o processo de desenvolvimento de software. Porém uma avaliação do processo de desenvolvimento proposto para ser válida precisa ser feita ao longo de um período maior de avaliação e deve ser executado por uma equipe de desenvolvimento com participantes com mais experiência no processo de desenvolvimento de aplicações e-business para a integração de empresas.

ebXML é também uma proposta de arquitetura para a integração e promoção de comércio eletrônico, onde o conjunto de artefatos são capazes de especificar um processo de negócio de maneira que empresas possam informar e modelar formas de interação necessárias em colaborações.

As atividades das quatro disciplinas do RUP especializadas em UMM se concentram nas fases de concepção e elaboração, sendo que a fase de concepção demandará um maior esforço para conciliar o entendimento de todos os parceiros de negócio para a geração da solução, em contrapartida a fase de construção deve ser abreviada ou ter seu esforço reduzido graças a utilização de frameworks que poderiam facilitar o processo de geração de aplicações.

Com relação a clareza da proposta, existem diversas iniciativas que estão promovendo o uso e discutindo dificuldades as dificuldades práticas de ebXML. Podem ser obtidos em [4] um conjunto de workshops e congressos com este fim, além de diversos documentos e especificações que podem ser utilizados como fonte de consulta sobre a proposta. Além disso são mantidos fóruns e listas de discussão pelas organizações promotoras da proposta visando promover o entendimento e amadurecimento de ebXML.

Um dos maiores desafios que não é abordado explicitamente nesta proposta é a coordenação de diversas equipes de desenvolvimento independentes, pois existem diversas organizações envolvidas no processo de desenvolvimento trabalhando em

conjunto para a geração de componentes e especificações que promoverão a interação entre as partes.

Uma outra dificuldade é a necessidade da existência de ferramentas de modelagem e desenvolvimento que dêem suporte efetivo a ebXML. Sem entrar em detalhes neste momento, o tratamento da complexidade do problema será parcialmente avaliado na próxima seção da monografia.

Existe um grupo considerável de instituições comerciais e acadêmicas que estão discutindo a evolução e o amadurecimento da proposta, o que poderá em breve tornar as promessas em avanços. Porém a proposta ainda precisa ser amadurecida e plenamente utilizada para que haja uma validação prática de suas recomendações, sendo que o seu desenvolvimento em um futuro próximo promete ser um grande catalisador e referência para a promoção de empreendimentos na Internet.

5. Ambientes de modelagem de processos de negócios

Ao avaliar ambientes de desenvolvimento de engenharia de software é preciso utilizar algumas convenções e critérios. O critério a ser utilizado durante esta avaliação será a conformidade com parte de um conjunto de requisitos para ambientes de modelagem de processos de negócios descritos em [22]. Na escolha dos requisitos a serem avaliados foram selecionados alguns dos que poderiam contribuir para a evolução do nível de abstração, do nível de formalidade, da satisfação dos critérios de qualidade e por fim da evolução dos requisitos em processos de desenvolvimento de software[1].

O tópico mais importante a ser avaliado é o suporte ao processo de aprendizado oferecido pelos ambientes integrados de engenharia de software. Por se tratar de um processo criativo e em evolução, o desenvolvimento de software atualmente não contém modelos suficientes que garantam a eficiência e a produtividade na geração de resultados. Muitos profissionais adquirem, com a prática e algum estudo, a proficiência responsável pela solução dos problemas existentes. Para que se adquira esta proficiência é preciso que as ferramentas que auxiliam o processo de geração da solução auxiliem também a evolução do conhecimento através do suporte efetivo e programado ao aprendizado da solução no processo de desenvolvimento.

A seção seguinte explicará resumidamente o conceito de eixos de evolução em processos de desenvolvimento de software. Em seguida serão descritos os requisitos de ambientes de desenvolvimento de software que serão avaliados e após isto será identificada a relação entre os requisitos abordados e os eixos de evolução. Por fim os ambientes de desenvolvimento de software serão avaliados.

5.1 Eixos de evolução no desenvolvimento de software

Ao longo de um processo de desenvolvimento incremental é possível que seja feita uma separação abstrata de eixos de evolução de um software, de uma representação ou modelo de solução. Estes eixos podem ser utilizados na análise de ambientes de modelagem com relação ao suporte efetivo ao processo de desenvolvimento de software.

Os eixos a serem analisados são relativos a evolução no nível de abstração, a evolução no nível de formalidade, a evolução da satisfação de critérios de qualidade e por fim, a evolução dos requisitos no desenvolvimento de aplicações [1].

5.1.1 Evolução do nível de abstração

Geralmente um processo de desenvolvimento prossegue de um nível mais abstrato para o menos abstrato, o que pode ser chamado de fase de elaboração. Porém é possível que ocorra o processo inverso, chamado de reflexão, isto é, a partir de um componente menos abstrato, é possível criar ou modificar componentes mais abstratos [Figura 14].

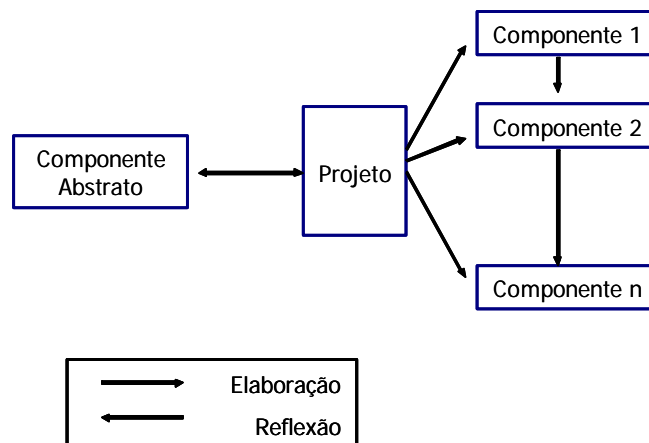


Figura 14: Evolução no Nível de Formalidade

É importante observar que ao longo do processo de criação podem ocorrer transformações entre diferentes representações em um mesmo nível de abstração. A grande vantagem em se utilizar esta abordagem é facilitar o entendimento do problema e da solução para este problema por parte do analista. Ao final deste processo criativo, é possível que haja um conjunto de representações isoladas, porém com um certo grau de interdependência, equivalência ou até mesmo superposição. Para que esta características não se tornem um problema, é preciso que este conjunto de representações geradas neste processo sejam consolidadas e consistidas com outras especificações preexistentes.

O papel de um ambiente de desenvolvimento é a promoção do uso de níveis de abstração através da facilitação do fluxo elaboração ? reflexão e manutenção da consistência entre diferentes representações, procurando sempre minimizar o esforço das pessoas envolvidas no processo para a geração destas representações.

5.1.2 Evolução no nível de formalidade

Entende-se por formalidade, a conformidade com um conjunto de normas, regras e critérios que precisam ser seguidos para que a especificação esteja de acordo com o padrão instituído pela organização onde está sendo desenvolvido o software.

Em um processo criativo de geração de representações é preciso que haja um certo grau de liberdade durante o processo de geração destes artefatos, isto é deve se permitir ao longo deste processo um certo grau de “informalidade” para que as

peças envolvidas possam elaborar suas idéias e amadurecer sua solução sem se ater a detalhes formais para a geração das representações, se concentrando assim na geração da solução para que depois se possa formalizar a solução dada. Um exemplo de um processo de evolução do grau de formalidade pode ser visto na figura 15.

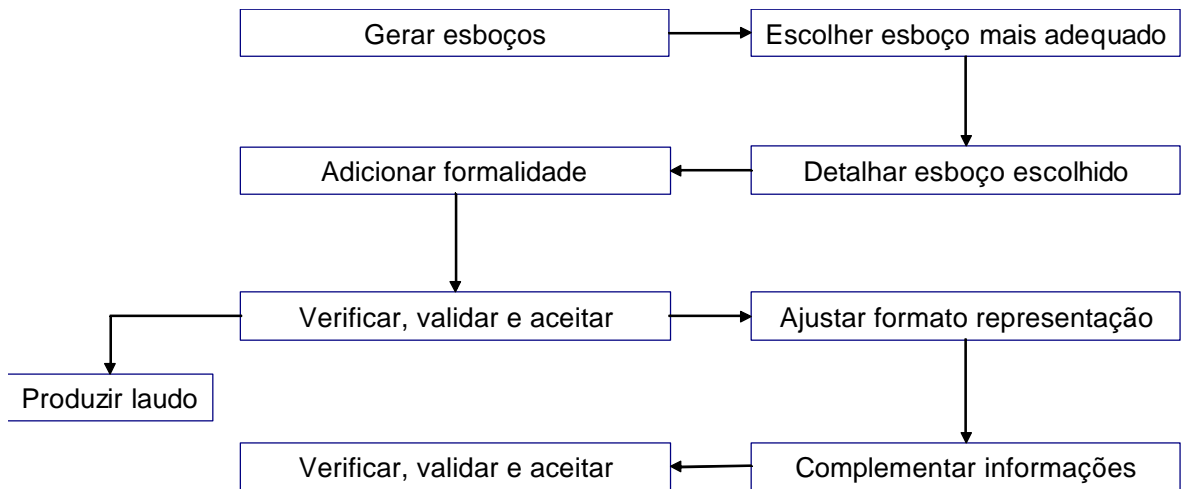


Figura 15: Exemplo de processo de Evolução no Nível de Formalidade

Sendo o processo de desenvolvimento de software um processo de aprendizado é necessário que a ferramenta a ser utilizada promova a simplicidade no uso e na alteração de representações, para que o tempo despendido pelo usuário no processo de geração de representações seja menor do que o tempo de gerá-las sem o uso da ferramenta. Ainda nesta linha de suporte ao aprendizado é necessário que haja o suporte a evolução do conhecimento sobre o problema abordado, tendo a ferramenta uma postura “clemente”, perdoadando erros de uso ou erros de decisão, facilitando a alteração do conjunto de representações, admitindo por exemplo o uso de rascunhos em sucessivos graus de completeza e formalidade.

5.1.3 Evolução na satisfação de critérios de qualidade

5.1.3.1 Metáfora : “*Errare humanum est*”

Seguindo a tese de que o processo de desenvolvimento de software é um processo de aprendizado, é necessário que o usuário não tenha medo de errar, sendo papel da ferramenta oferecer um certo grau de suporte ao aprendizado e que este possa ser sempre associado a bons critérios e métodos de verificação, validação e aceitação.

5.1.3.2 Metáfora : “*Perseverare in errore est diabolicum*”

Em contrapartida, após a identificação de determinado erro em uma representação é preciso que alterações sejam feitas e suas devidas correções sejam aplicadas. Para isto o ambiente deve procurar facilitar este processo, simplificando a manutenção de modelos e representações, pois em geral ao postergar a revisão e a correção de erros é comum que haja um aumento nos custos desta atividade.

5.1.4 Evolução dos Requisitos

Como foi visto anteriormente, é necessário existir um certo grau de suporte ao aprendizado na ferramenta, pois o processo de desenvolvimento de software só termina quando se aprendeu suficientemente sobre o problema e sua respectiva solução[1]. Uma abordagem a ser utilizada é a de aproximações sucessivas, onde cada incremento constitui um pouco de aprendizado, uma avaliação e possíveis correções de rota em virtude do aprendizado.

5.2 Modelagem de Processos de Negócios e Ambientes de Engenharia de Software

Em conjunto com os eixos de evolução já discutidos, um critério a ser utilizado será a conformidade com parte de um conjunto de requisitos identificados para ambientes de modelagem de processos de negócios[22].

Dentre estes requisitos a característica primordial para o processo de construção e modificação de modelos é o auxílio ao tratamento da complexidade do problema. Uma possível abordagem a ser utilizada é a de refinamentos e elaborações iterativas de modelos, que pode ser obtida através do suporte ao desenvolvimento parcial de modelos com o devido tratamento da complexidade que isto possa representar para o processo. Parte desta complexidade se refere ao suporte adequado à integração de modelos, ao gerenciamento de modelos incompletos e inconsistentes e a identificação de modelos incorretos.

Outro requisito importante é o suporte ao aprendizado na geração de modelos e sua avaliação será restrita a existência de funcionalidades como o reconhecimento de padrões de representações, a identificação e simulação de representações alternativas para experimentação e amadurecimento dos modelos, e por fim a avaliação da consistência de *design* parciais.

O processo de gestão de informações é um requisito de suma importância e tem como principal característica a flexibilidade na geração e apresentação de informações. Esta flexibilidade irá se refletir na adaptação de relatórios às realidades e necessidades da equipe de desenvolvimento. Na avaliação serão considerados o processo de obtenção de informações, o gerenciamento de dados em diferentes níveis de formalidade e a representação do processo em diferentes níveis de abstração. Com relação ao suporte ao gerenciamento de modelos parciais é preciso que sejam oferecidas informações suficientes para gerenciar a complexidade e a possível combinação de modelos em um modelo integrado.

Além destes, a promoção da comunicação de informações será um requisito analisado por promover a compreensão de propriedades ou alterações em especificações. Será levado em consideração durante a avaliação a possibilidade de extração de partes de modelos para explicar determinada interação e produção de resumos ou relatórios.

Com relação ao requisito facilitação da visualização de informações, serão analisadas as propriedades e o suporte a gerência de múltiplos modelos interdependentes, onde um ponto importante é capturar a dimensionalidade das interdependências e a possibilidade de integração entre as visões de modelos relevantes.

5.3 Avaliação de ambientes de desenvolvimento de software

Conforme explicado acima serão avaliadas a conformidade com os seguintes requisitos:

- Capacidade de dinamicamente construir e modificar modelos
- Suporte ao Aprendizado na geração de modelos
- Flexibilidade na Gestão de Informações
- Promoção de Comunicação de Informações
- Facilitação de Visualização

Através da análise das propriedades e requisitos foi possível identificar que uma possível concordância com estas propriedades promoverá a evolução no nível de abstração, a evolução no nível de formalidade, a evolução na satisfação de critérios de qualidade e na evolução de requisitos. Abaixo foi criada uma tabela relacionando os requisitos a serem analisados e os eixos de evolução

		CDCMM	SAGM	FGI	PCI	FV
Evolução do Nível de abstração						
	Consistência entre modelos	X				
	Facilita a Reflexão x Elaboração	X	x	x	X	x
Evolução do Nível de formalidade						
	Simplicidade no uso e alteração de modelos	X	x		X	x
	Suporte a evolução do conhecimento sobre determinado problema	X	x	x		
Evolução da Satisfação de critérios de qualidade						
	Habilidade no reconhecimento de padrões		x	x	x	x
Evolução de Requisitos						
	Processo de aproximações sucessivas	X	x	x		
Evolução Geral						
	Suporte ao aprendizado		x	x	x	x
	Facilita a Manutenção de Modelos	X		x	x	x
CDCMM - Capacidade de Dinamicamente Construir e Modificar Modelos		Legenda <input checked="" type="checkbox"/> Relacionável <input type="checkbox"/> Não Relacionável				
SAGM – Suporte ao Aprendizado na Geração de Modelos						
FGI - Flexibilidade na Gestão de Informações						
PCI – Promoção de Comunicação de Informações						
FV - Facilitação de Visualização						

O processo de avaliação de ferramentas será feita seguindo o critério de conformidade com os requisitos explicitados anteriormente e ao final do processo será feita uma análise do conjunto de ferramentas avaliadas.

O período de avaliação das ferramentas foi de trinta dias o que acabou por limitar o estudo completo e detalhado de cada uma delas. Um maior intervalo para avaliação seria utilizado para adquirir um melhor domínio e uma maior compreensão do funcionamento e existência de funcionalidades específicas em todas as ferramentas avaliadas.

Auxiliando ao processo de avaliação foram utilizadas documentações das ferramentas e materiais disponíveis oferecidos pelos fornecedores. Ao final do estudo destes materiais foram feitos alguns testes nas ferramentas visando avaliar a existência dos requisitos acima explicitados.

Para a análise foram escolhidas três ferramentas de desenvolvimento de software que se propõe a modelar processos de negócios. São elas: Mega 5.2 [10], Sybase Power Design [18] e Rational Rose[14]. Abaixo serão oferecidas pequenas descrições sobre as ferramentas e a razão para a escolha destas.

MEGA Software Suite [10] - MEGA Process tem como proposta permitir que organizações tratem adequadamente o desafio de modelar e definir processos de negócios. Outra peculiaridade desta ferramenta é o suporte explícito ao desenvolvimento de aplicações ebXML, sendo que esta ferramenta faz parte de uma solução maior, o ambiente Mega Software Suite, porém nesta monografia somente será avaliada a ferramenta para auxílio ao processo de modelagem, análise e simulação de processos de negócios.

Sybase PowerDesigner[18] é uma ferramenta de suporte ao desenvolvimento de aplicações e-business. Esta ferramenta também suporta explicitamente o desenvolvimento de aplicações envolvendo ebXML e faz parte de um ambiente integrado de desenvolvimento de software chamado Sybase e-business Platform. Esta ferramenta em específico se propõe a modelar e construir aplicações que envolvam processos de negócio.

Rational Rose[14] é uma ferramenta Rational [15] de modelagem difundida no mercado e baseada em diretrizes e conceitos difundidos no mercado como UML[19] e RUP[16]. Esta ferramenta foi escolhida para avaliar o impacto de uma ferramenta que a princípio é recomendada por seu fornecedor como adequada para modelagem em processos de desenvolvimento RUP. Como foi dito anteriormente o método UMM utilizado em ebXML se baseia no RUP e utiliza UML para especificar atividades e artefatos que precisam ser gerados ao longo do processo de desenvolvimento.

Abaixo serão avaliadas as ferramentas com objetivo de identificar a existência ou não de funcionalidades que dêem suporte aos requisitos levantados. Caso não existam estas funcionalidades a ferramenta receberá neste quesito a avaliação **Ausente**. A ferramenta receberá o conceito **Bom** caso possua estas funcionalidades e este suporte seja efetivo e contribua para a modelagem da aplicação, ou receberá o conceito **Regular** caso a funcionalidade ainda precise ser amadurecida.

5.3.1 Capacidade de dinamicamente construir e modificar modelos

	Refinamento e elaboração iterativa de modelos	Suporte ao Desenvolvimento Parcial de Modelos	Identificação modelos que não estejam corretos
Mega	Bom	Bom	Regular
Sybase	Bom	Bom	Regular
Rational	Regular	Regular	Regular
	Suporte adequado a Integração de Modelos	Gerenciamento de modelos incompletos e inconsistentes	
Mega	Bom	Regular	
Sybase	Bom	Regular	
Rational	Regular	Ausente/Não Identificado	

De uma forma geral as ferramentas avaliadas estavam condizentes com este requisito a ser avaliado, o único ponto em destaque no processo de construção e manutenção de modelos foram alguns recursos interessantes oferecidos pelas ferramentas da Sybase e Mega, principalmente no que se refere a existência de informações e estatísticas úteis sobre os modelos que podem ser utilizadas para o acompanhamento do desenvolvimento iterativo.

5.3.2 Suporte ao aprendizado na geração de modelos

	Habilidade em reconhecer padrões	Identificação de possibilidades de design	Proposição designs alternativos
Mega	Ausente/Não Identificado	Ausente/Não Identificado	Ausente/Não Identificado
Sybase	Ausente/Não Identificado	Ausente/Não Identificado	Ausente/Não Identificado
Rational	Ausente/Não Identificado	Ausente/Não Identificado	Ausente/Não Identificado
	Simulação entre modelos alternativos e identificação do modelo correto	Ferramenta para experimentação e amadurecimento de modelos	Avaliação a consistência de design parciais
Mega	Bom	Regular	Regular
Sybase	Ausente/Não Identificado	Ausente/Não Identificado	Ausente/Não Identificado
Rational	Ausente/Não Identificado	Ausente/Não Identificado	Ausente/Não Identificado

Com relação ao suporte ao aprendizado foi possível observar que existem algumas iniciativas interessantes, porém a maioria das ferramentas não oferece suporte efetivo a este requisito, podendo esta ser uma área de importante pesquisa.

A ferramenta que melhor ofereceu suporte ao processo de aprendizado foi a Mega através de processos de simulações de representações. Infelizmente o tempo necessário para a compreensão do uso desta funcionalidade ultrapassa o tempo disponível para a avaliação feita.

5.3.3 Flexibilidade na gestão de informações

	Obtenção de informações	Gerenciamento de dados em diferentes níveis de formalidade	Representação do processo em diferentes níveis de abstração
Mega	Regular	Regular	Regular
Sybase	Bom	Regular	Regular
Rational	Regular	Regular	Regular
	Suporte ao gerenciamento de modelos parciais	Combinação de modelos em um modelo integrado	
Mega	Ausente/Não Identificado	Bom	
Sybase	Ausente/Não Identificado	Regular	
Rational	Ausente/Não Identificado	Regular	

A ferramenta Sybase apresentou um conjunto de funcionalidades que facilitaram o processo de gestão de informações, é importante ressaltar que a facilidade de uso do processo de configuração de relatórios sobre modelos foi de fundamental importância para a avaliação positiva obtida por esta ferramenta.

Não houve suporte efetivo e explícito ao acompanhamento da evolução do nível de formalidade e abstração nas ferramentas avaliadas. Porém não é possível dizer que as ferramentas impedissem estas evoluções, pois em todas elas era possível amadurecer os níveis de detalhe, níveis de formalidade e de abstração sem que necessariamente estas ferramentas estivessem cientes deste processo e desta evolução.

Não foi possível identificar em nenhuma das ferramentas as funcionalidades que dessem suporte efetivo ao gerenciamento parcial de modelos.

Com relação a possibilidade de integração de modelos a ferramenta Mega apresentou auxiliares para guiar o analista ao longo do processo de integração e as demais ferramentas não ofereciam suporte explícito a este processo, ao mesmo tempo que não impediam que isto fosse feito. Esta característica é de fundamental importância para o processo de modelagem de aplicações e-business, pois a complexidade do problema pode ser tanta que torna inviável a geração de uma única representação para a solução.

5.3.4 Promoção de comunicação de informações

	Facilita a compreensão de propriedades ou alterações em especificações	É possível extrair partes de modelos para explicar determinada interação	Produz resumos e relatórios
Mega	Regular	Regular	Bom
Sybase	Regular	Regular	Bom
Rational	Regular	Regular	Regular
	Promove o aprendizado sobre a organização a partir do processo de modelagem de negócios		
Mega	Ausente/Não Identificado		
Sybase	Ausente/Não Identificado		
Rational	Ausente/Não Identificado		

A avaliação do requisito de “Promoção do aprendizado da organização a partir do processo de modelagem” não foi possível pois não existia uma organização real a ser

compreendida e as representações faziam parte de exemplos encontrados na documentação estudada.

A produção de resumos sobre os modelos gerados foi um requisito onde as ferramentas demonstraram certa flexibilidade e completude nas informações descritas nas representações.

A possibilidade de extração de partes de modelos para explicar interações foi feita através da associação dos três requisitos já avaliados e a facilidade na compreensão de propriedades ou alterações em especificações se restringiu na análise destes mesmos requisitos.

5.3.5 Facilitação de visualização

	Existe o suporte ao conceito de múltiplos modelos interdependentes
Mega	Bom
Sybase	Bom
Rational	Regular
	Captura a dimensionalidade das interdependências e a possibilidade de integrar as visões de modelos relevantes.
Mega	Bom
Sybase	Regular
Rational	Regular

Com relação ao requisito de facilitação de visualização, um ponto a favor das ferramentas Sybase e Mega é a existência de representações específicas para a modelagem de processos de negócios o que na prática acaba por facilitar o entendimento das representações geradas.

O conceito de interdependência entre modelos existiu em todas as ferramentas, porém no Rose era preciso estruturar o projeto para que isto fosse possível, em contrapartida nas demais ferramentas existiam gráficos de relacionamentos entre artefato, o que auxiliou no processo de entendimento da estrutura e organização do projeto.

A ferramenta da Mega ofereceu um suporte maior no que diz respeito a possibilidade de integração de representações, disponibilizando até um auxiliar para o processo de integração de modelos.

5.3.6 Avaliação Geral

	CDCMM - Capacidade de Dinamicamente Construir e Modificar Modelos	SAGM - Suporte ao Aprendizado na Geração de Modelos	PCI - Promoção de Comunicação de Informações
Mega	Bom	Regular	Regular
Sybase	Bom	Ausente/Não Identificado	Bom
Rational	Regular	Ausente/Não Identificado	Regular
	FV - Facilitação de Visualização	FGI - Flexibilidade na Gestão de Informações	
Mega	Bom	Regular	
Sybase	Bom	Bom	
Rational	Regular	Regular	

É necessário um tempo maior de avaliação para poder amadurecer o conhecimento de uso da ferramenta e o entendimento do funcionamento e a existência de funcionalidades. O processo de escolha dado estas condições é baseado na concordância/existência de funcionalidades que expressem os requisitos levantados segundo este critério a melhor ferramenta para a modelagem de processos de negócios foi a Mega pois foi possível observar que esta ferramenta teve uma avaliação superior as demais, seguidas pela Sybase e por último a Rational.

Em uma futura avaliação é preciso avaliar as soluções completas propostas pelas três companhias fornecedoras de produtos.

De acordo com a avaliação geral é possível observar que é preciso que os ambientes avaliados melhorem bastante, principalmente ao que se refere a suporte ao aprendizado, onde praticamente não foi encontrado suporte efetivo nas ferramentas de modelagem.

Durante a avaliação foi possível observar que o Rational Rose não é tão flexível no que se refere a customizações e adaptações feitas ao processo RUP, pois geralmente estas adaptações geram peculiaridades que se bem tratadas em ferramentas de desenvolvimento poderiam facilitar o processo de desenvolvimento.

6. Conclusão

A proposta de ebXML marca um avanço e o início de organização do processo de desenvolvimento de aplicações e-business. Através dela é possível compartilhar iniciativas e experiências de diversas equipes de desenvolvimento que através da submissão de propostas e revisões contribuem para a evolução do processo e conjunto de especificações propostas.

É importante observar que o conjunto de especificações de ebXML tende a amadurecer com o tempo e a prática do desenvolvimento deste processo será de grande auxílio para a evolução e utilidade da proposta.

Com relação a instância do RUP gerada, é possível observar que não foram feitas recomendações específicas para as disciplinas de implementação, teste, deployment, gerenciamento de configuração e mudanças, gerenciamento do projeto e ambiente. Este conjunto de disciplinas não é específico a processos de desenvolvimento de aplicações e-business portanto estas disciplinas especificadas no RUP podem ser reutilizadas no desenvolvimento de uma aplicação utilizando ebXML.

Foi possível observar que ainda há muito o que evoluir no que diz respeito a adequação de ambientes de desenvolvimento de software para auxílio ao processo de modelagem de processos de negócios. Ferramentas direcionadas são melhores do que ferramentas genéricas sem possibilidade efetiva de customização ou adaptação.

7. Apêndice A: Processos Colaborativos utilizando ebXML

O comércio eletrônico colaborativo (*c-commerce*) é um modelo de negócios baseado em tecnologias Internet que promove interações dentro de empresas, entre empresas e seus parceiros comerciais, e entre participantes de uma comunidade comercial.

Em *c-commerce*, empresas promovem a colaboração dinâmica e a troca de informações em tempo real. A colaboração dinâmica tem como objetivo especificar, executar e monitorar processos de negócios que ultrapassem as fronteiras organizacionais. Como consequência destas interações é necessário que empresas coordenem o comércio e o fluxo de informações, permitindo que várias empresas, colaborando, resolvam seus problemas de negócios.

A figura 14 ilustra o framework conceitual para aplicações de comércio colaborativo proposto em [11]. Nele são descritos dois grupos de tecnologias que irão ser determinantes no processo de desenvolvimento de aplicações para *c-commerce*, o primeiro corresponde às tecnologias que facilitam a troca de informações entre empresas, no caso em estudo estas tecnologias se referem a Internet, que representa o canal onde os negócios ocorrem, nesta camada existem 3 classes de tecnologia, *Rede*, *Linguagem* e *Interpretação*. Já o segundo grupo corresponde às tecnologias que aprimoram o gerenciamento de processos, sendo estas disponibilizada por uma aplicação desenvolvida por participantes de uma negociação, nesta são definidas 3 classes de tecnologias, *Regras de Negócio*, *Processos* e *Aplicação*.

Regras de negócio são diretivas que definem ou restringem aspectos particulares de um negócio, é a indicação do que é possível ocorrer em um processo e o que não é permitido. São definidos através de regras de negociação os relacionamentos em objetos de negócio, as restrições de integridade de dados, regras de inferências lógicas e seqüências de processamento.

Processos representam instâncias executáveis de regras de negócio, são fluxos de informações, produtos, e serviços que controlam quando e onde os recursos são utilizados.

Aplicação é a ferramenta responsável pela coordenação e gerenciamento de processos e seu objetivo é oferecer informações necessárias para modelos de decisão.

Rede representa os recursos disponíveis, isto é, o meio onde ocorrem as trocas de mensagens. Exemplos de redes são ETN (eletronic trading networks) e VAN (value added networks).

Linguagem é uma convenção entre receptor e emissor que permite que haja um entendimento entre eles, fornecendo significado para um conjunto de dados e permitindo a troca de informações em um formato predefinido.

Tradução é o processo pelo qual uma aplicação obtém dados através de linguagens e a interpreta, transformando-a em informações úteis para serem utilizadas no gerenciamento de processos.

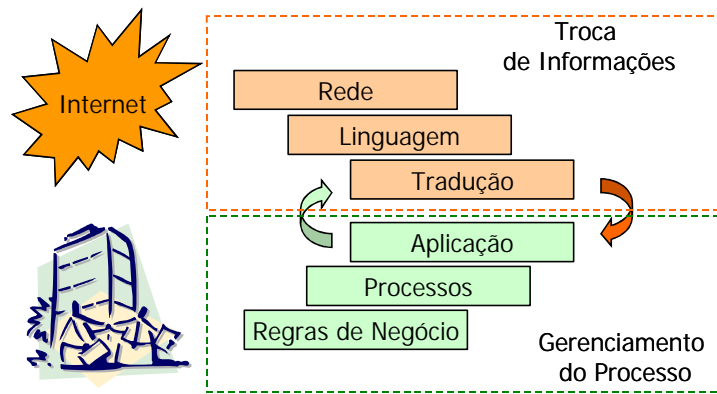


Figura 14. Framework para c-commerce

ebXML [3-9] (*Electronic Business using eXtensible Markup Language*) é um framework colaborativo para o desenvolvimento de aplicações de comércio eletrônico. É uma iniciativa promovida pelo consórcio OASIS [12], para estabelecimento de padrões XML, e pela UN/CEFACT [20], órgão das Nações Unidas para a promoção comercial entre as nações, incluindo em suas responsabilidades a promoção e o desenvolvimento de comércio eletrônico.

A visão promovida por *ebXML* é:

Através da utilização de ebXML será possível criar um mercado eletrônico virtual e global onde empreendimentos de qualquer tamanho e em qualquer localização geográfica possam estabelecer e conduzir negócios entre si através de troca de mensagens.

ebXML visa permitir a qualquer um, em qualquer lugar, o estabelecimento de negócios eletronicamente, entretanto, se antecipa que a adoção dos vários componentes ebXML pode ser feita de maneira gradual e incremental ao longo do tempo.

ebXML pretende atingir seus objetivos oferecendo um conjunto de especificações que permitem que empresas conduzam negócios através da Internet. Utilizando ebXML, empresas têm um método padrão para troca de mensagens, gerenciamento de relacionamentos comerciais, comunicação de dados utilizando uma mesma terminologia e a definição e registro de processos de negócios.

Uma ilustração para o ciclo para o estabelecimento de acordos utilizando ebXML pode ser visto na figura 15.

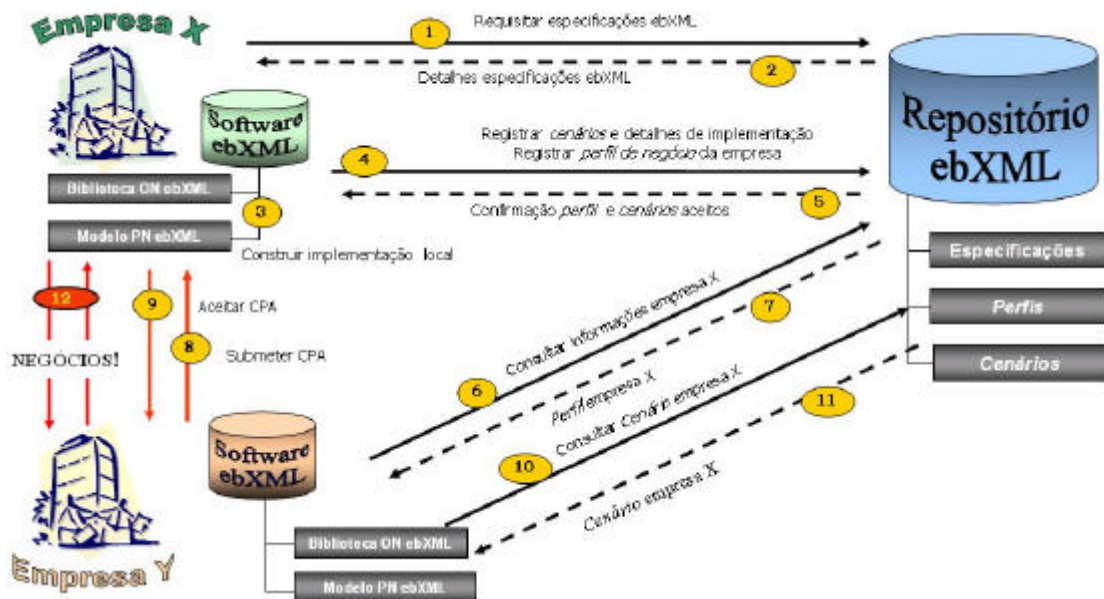


Figura 15 - Sequência de execução de negócios utilizando ebXML.

Antes de empresas conduzirem negócios eletronicamente entre si, é necessário obter os parceiros para negociação. Estes parceiros são outras empresas que oferecem serviços ou produtos que atendem a sua necessidade. Para o estabelecimento de acordos é preciso determinar que processos de negócio e documentos são necessários na obtenção destes produtos e serviços. Após identificá-los é preciso definir como a troca de informações será feita e como serão estabelecidos os acordos, estes contendo termos contratuais e em que condições isto será feito. Uma vez que estas configurações estão estabelecidas é possível trocar informações, produtos e serviços.

Para facilitar isto, ebXML oferece uma infra-estrutura para a interoperabilidade na comunicação de dados, um framework semântico para interoperabilidade comercial, e mecanismos que permitem que empresas busquem outras empresas, estabeleçam relacionamentos e conduzam negócios entre si.

A interoperabilidade na comunicação de dados é garantida por um mecanismo de transporte de mensagens padrão com uma interface bem definida, regras de empacotamento, e um modelo de entrega previsível, assim como uma interface para o tratamento de mensagens.

A interoperabilidade comercial é conceitualmente estabelecida através da especificação de esquemas para a definição de processos de negócios, de componentes chaves no processo e de um modelo de contexto para a definição de documentos de negócios. *ebXML* recomenda uma metodologia e fornece um conjunto de *worksheets* e *guidelines* para a criação destes modelos. Uma biblioteca (catálogo) de negócios e modelos de informações promove a eficiência nos negócios por encorajar o reuso de processos de negócios ou partes de processos de negócios pré-definidos.

Para que o processo possa ocorrer, ebXML fornece um repositório compartilhado onde empresas podem obter informações sobre outras empresas, além de um processo para o estabelecimento de regras e compromissos entre empresas para o

estabelecimento de negócios (*Collaboration Protocol Agreement*, CPA). Este repositório também é utilizado para disponibilizar *profiles* de empresas, especificações de processos de negócios, e mensagens de negócio relevantes.

Abaixo será ilustrado um problema simplificado com a utilização de agentes de software em um processo de compra de pacotes de turismo.

7.1.1 Ilustração de caso de uso

Deseja-se criar um agente de software capaz de negociar uma *commodity*. Este agente possui funções ou regras, instruídas por agentes humanos, que o tornam capaz de avaliar pequenas peculiaridades e diferenças entre produtos similares oferecidos. Por exemplo, o *Agente X* avalia um menor prazo de entrega como fator positivo a ser considerado na escolha do produto.

Seria de muita valia que o agente de software fosse capaz de obter e atender às regras de interação com fornecedores da *commodity*, sem que para isto, para cada novo fornecedor ou aplicação, um designer humano fosse obrigado a configurar e especificar a aplicação obedecendo à **coreografia** que o agente deve seguir. Entende-se por coreografia o conjunto de etapas, papéis, seqüências e ações específicas ao processo de negociação eletrônica que devem ser respeitadas.

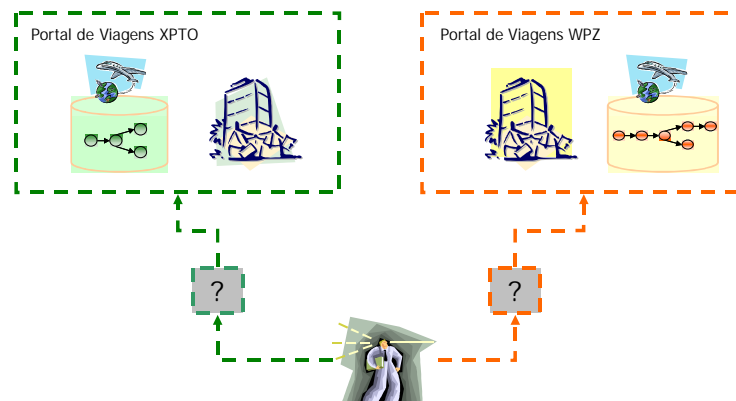


Figura 16: Compra de pacotes de turismo sem interpretação de protocolos

O “entendimento” do protocolo de interação a partir de uma descrição do processo de negociação [Figura 16] poderia ser atribuição do agente de software e um exemplo, onde a *commodity* negociada é um pacote turístico, está ilustrado abaixo [Figura 17].

Em um primeiro momento é feita uma busca para obtenção de fornecedores que atendem as necessidades do agente de software.

Na etapa de busca, hipoteticamente, foram encontrados dois fornecedores para o pacote de turismo desejado, o portal XPTO e o portal WPZ. Para cada portal é estabelecido pelo fornecedor um conjunto de regras, o protocolo de negociação, a ser utilizado na interação para a aquisição do bem.

Após isto, o agente obtém a descrição do processo no qual ele deve interagir para a aquisição do bem. Esta descrição pode ser armazenada no respectivo portal ou em um repositório compartilhado contendo informações sobre processos [Figura 17.1].

O agente de software após “ler” o protocolo, interpreta as regras estabelecidas e suas restrições em uma máquina de execução de tarefas e ações. É nesta etapa de interpretação que será feita o seqüenciamento e execução de tarefas de acordo com o que foi estabelecido no protocolo [Figura 17.2].

A interação com fornecedores é feita através da máquina de execução de protocolos[Figura 17.3]. Após diversas etapas de negociação com diversos fornecedores distintos será possível obter o melhor acordo atendendo as preferências do agente de software [Figura 17.4].

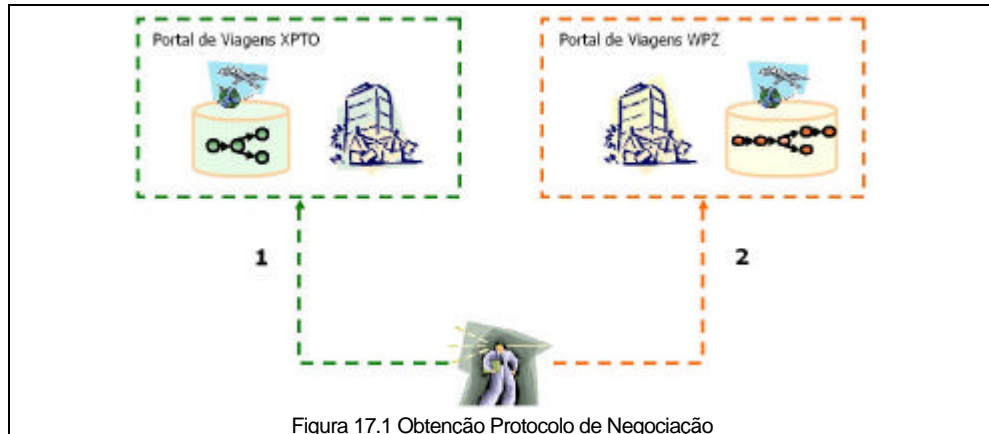


Figura 17.1 Obtenção Protocolo de Negociação

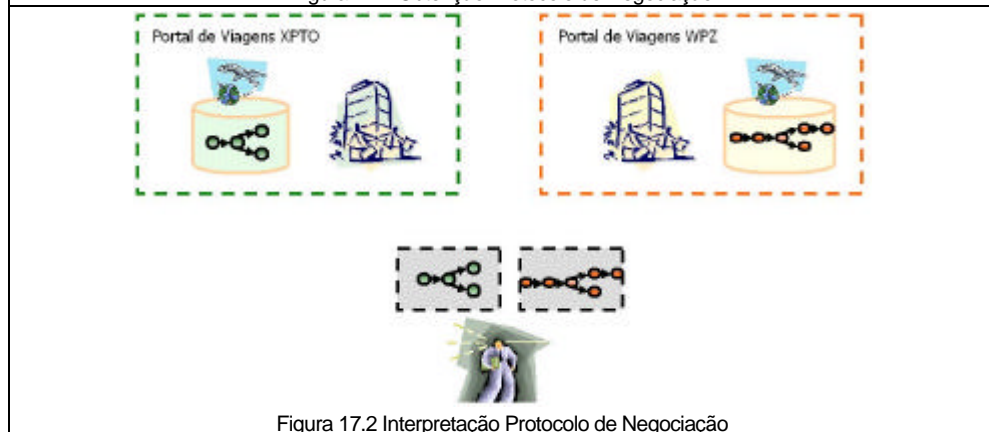


Figura 17.2 Interpretação Protocolo de Negociação

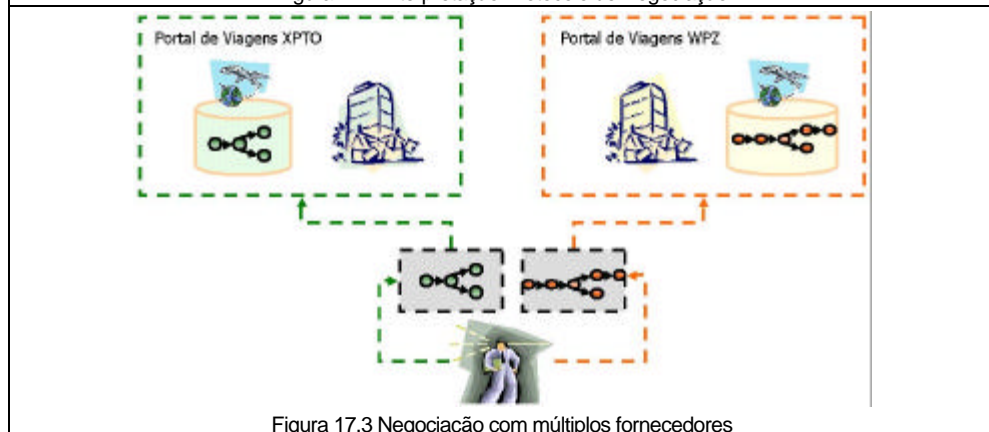


Figura 17.3 Negociação com múltiplos fornecedores

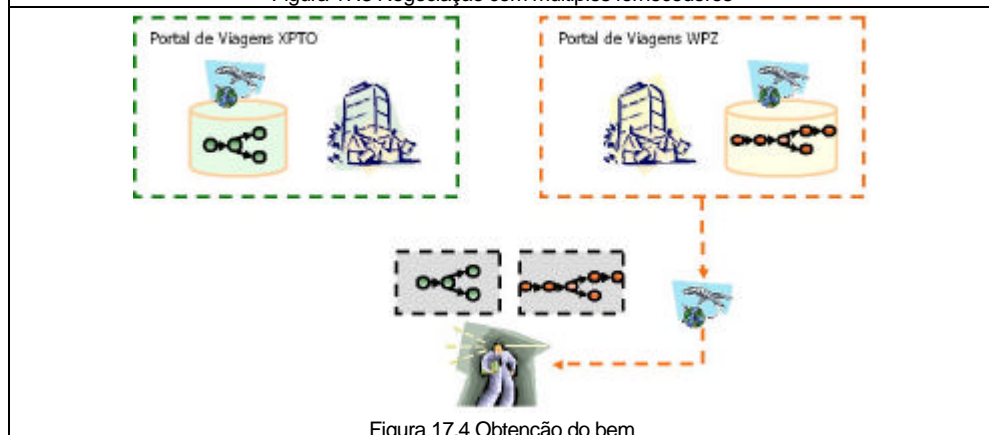


Figura 17.4 Obtenção do bem

Figura 17. Negociação de Commodity – Pacotes de turismo – Processo de Interpretação de Protocolo

8. Referências

- [1] Arndt von Staa. “*Ambientes e processos*”, Apresentação do curso “Processos e Ambientes de Engenharia de Software 2002.1”.
- [2] Arndt von Staa. Apresentações e aulas do curso “Processos e Ambientes de Engenharia de Software 2002.1”.
- [3] Business Process and Business Information Analysis Overview. Version 1.0. Date 11 May 2001. ebXML Business Process Project Team
- [4] ebXML <http://www.ebxml.org/>
- [5] ebXML Concept - Context and Re-Usability of Core Components. Version 1.04. 11 May, 2001. ebXML Core Components Project Team.
- [6] ebXML Registry Information Model. Version 1.0 11 May 2001. ebXML Registry Project Team.
- [7] ebXML Registry Services. Version 1.0 11 May 2001. ebXML Registry Project Team.
- [8] ebXML Technical Architecture Specification. Version 1.0.4 16 February 2001. ebXML Technical Architecture Project Team.
- [9] ebXML Catalog of Common Business Processes. Version 1.0. Date May 11, 2001. ebXML Business Process Project Team
- [10] MEGA Homepage. <http://www.mega.com/us/home/index.asp>
- [11] Michael E. Porter, *Competitive Advantage: Creating and Sustaining Superior Performance*, 1998, Harvard Business School Press.
- [12] OASIS Homepage <http://www.oasis-open.org>
- [13] Object Management Group – Agent Platform Special Interest Group. Agent Technology – Green Paper. Version 1.0, September 2000.
- [14] Rational Rose Homepage <http://www.rational.com/products/rose/>
- [15] Rational Homepage, <http://www.rational.com>
- [16] RUP Homepage, <http://www.rational.com/rup>
- [17] Supply Chain Operations Reference model, The Supply Chain Council (<http://www.supply-chain.org/>)
- [18] Sybase Power Designer <http://www.sybase.com/products/enterprisemodeling/powerdesigner>
- [19] UML Homepage, <http://www.omg.org/uml>
- [20] UN/CEFACT Homepage www.unece.org/cefact
- [21] UN/CEFACT Modeling Methodology. CEFACT/TMWG/N090R9.1. UN/CEFACT Technical Modeling Working Group.
- [22] University of Toronto. “Designing tools to Support Business Process Reengineering”. Enterprise Integration Laboratory, Department of Industrial Engineering.
- [23] Workflow Management Coalition. The Workflow Reference Model. <http://www.wfmc.org/> .
- [24] XML, <http://www.oasis-open.org/cover/xml.html>
- [25] I. Sommerville *Software Engineering 6th Edition* (editors), Addison-Wesley, 2001.