

2 Fundamentação – Conceitos Básicos

Neste capítulo serão apresentados conceitos básicos necessários para o entendimento do Processo Unificado KUP. São abordadas as definições de Processos, Frameworks de Processos e Gestão de Conhecimento. Conceitos como Ontologias, Engenharia de Ontologias, a Web Semântica e sua evolução já foram tratados no capítulo de Introdução.

2.1. Processos e Frameworks de Processos

Como já foi citado e será visto mais adiante, o que se propõe neste trabalho é a unificação de diversas características e abordagens oriundas de metodologias distintas em um framework de processos abrangente, que permita a instanciação de processos para o projeto e desenvolvimento de ontologias e bases de conhecimento. Entende-se como um processo um conjunto de atividades e resultados associados a essas atividades com o objetivo de garantir a geração de um produto final (Sommerville, 1998), seja este produto um software, uma ontologia ou uma ontologia associada a sua base de conhecimento.

Segundo (Booch, 1999), um processo define “quem” está fazendo “o quê”, “quando” e “como” para alcançar certo objetivo. Em engenharia de software, este objetivo é construir um produto de software ou aprimorar um software existente. Em engenharia de ontologias o resultado primordial deve ser uma ontologia explícita, um artefato computacional, podendo até se estender a aplicações que façam uso das informações semânticas descritas pela ontologia em uma base de conhecimento. Um processo efetivo deve prover orientações para um desenvolvimento eficiente, capturando e apresentando as “melhores práticas” que o estado-da-arte permite. Essas providências irão permitir a redução do risco associado ao projeto, aumentando a previsibilidade e a antecipação de problemas (Kruchten, 2000). Um processo unificado combina resultados e contribuições de diversos autores, o que permite uma consolidação de experiências e evita a repetição de erros.

2.1.1. Terminologia

2.1.1.1. Metodologia

Uma metodologia é uma série integrada e compreensiva de técnicas ou métodos criando uma teoria sobre como uma classe de trabalho intelectual deve ser realizada (IEEE, 1995).

2.1.1.2. Método

Um método é um processo ou procedimento seqüencial usado na construção de um produto ou na execução de um serviço (IEEE, 1990).

2.1.1.3. Técnica

Uma técnica, por sua vez, é um procedimento técnico ou gerencial usado na obtenção de um objetivo definido (IEEE, 1995).

2.1.1.4. Processo

Como já foi visto, um processo é um conjunto de atividades e resultados associados a essas atividades com o objetivo de garantir a geração de um produto final (Sommerville, 1998), seja este produto um software, uma ontologia ou uma ontologia associada a sua base de conhecimento.

2.1.1.5. Framework de Processos

Todo processo proposto para permitir a geração de um produto específico possui vantagens e desvantagens. Sua aplicação e execução não devem apenas seguir a teoria ou um roteiro pré-definido mas também ser adaptado às necessidades individuais de cada time, projeto, empresa ou cliente, para que possa atingir de forma eficaz e eficiente seus objetivos (Gnatz et al., 2001).

No entanto, para que essa adaptação vire realidade, é necessário identificar e explicitar todos os componentes de um processo, bem como as relações existentes entre eles. Com o intuito de resolver esse problema – identificando um conjunto básico de elementos e definições comuns a todos os modelos de processos – que foi concebida a idéia de modelos de frameworks de processos.

Esse tipo de framework deve permitir a integração de vários modelos de processos existentes, servindo como uma base comum para a definição de um processo de desenvolvimento concreto, isto é, uma instância do framework. Este framework deve, portanto, incorporar características e benefícios de diferentes origens sendo flexível e facilmente adaptável a conjuntos de requisitos e situações distintas.

Adotar um novo processo para uma equipe ou organização envolve a análise das melhores práticas implantadas na indústria e do que já se mostrou adequado à própria organização. Nesses casos, a utilização de frameworks de processos parece ser uma boa estratégia, uma vez que tanto elementos de processos quanto práticas consagradas podem ser integradas a um processo já existente, resultando em um processo mais adequado para o projeto ou organização em questão (Kruchten, 2000).

Em (Booch, 1999), por exemplo, o processo unificado para o desenvolvimento de software (RUP) é citado apenas como sendo uma orientação ao projetista, sendo raramente aplicado como está descrito. Isto indica, portanto, que o processo unificado aqui proposto não deve ser algo estanque, imutável e rígido mas que deve ser adaptado a cada necessidade, escolhendo-se quais fases são aplicáveis a cada projeto. Mesmo que um processo de desenvolvimento tenha sido definido através da abordagem de framework de processos, ainda assim é necessário adaptá-lo a diferentes projetos e situações, como mostra a Figura 7.

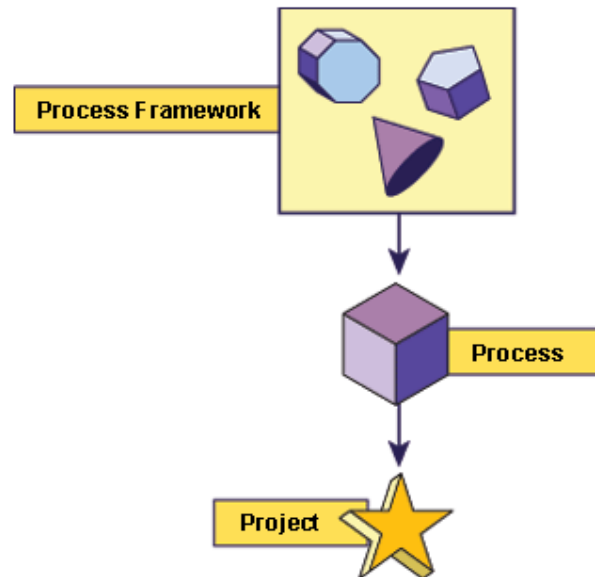


Figura 7 - A instanciação de um Framework de Processos em um Processo Concreto

2.2.Gestão de Conhecimento

Atualmente, a Gestão de Conhecimento está se tornando cada vez mais um fator crítico de sucesso para diferentes tipos e tamanhos de organizações. É senso comum dizer que o conhecimento é o mais importante ativo intangível de uma corporação. No entanto, a grande maioria das companhias vem implementando, até hoje, apenas iniciativas incidentais – ou mesmo inconscientes – com o objetivo de gerenciar seus ativos de conhecimento. Mais recentemente, essas mesmas companhias começaram a perceber que, para serem mais produtivas, competitivas e lucrativas, era fundamental aprimorar a forma como essas iniciativas vinham sendo conduzidas.

A Gestão de Conhecimento pode ser definida como o esforço em se aprimorar a efetividade corporativa e humana através do estabelecimento de conexões (Terra et al., 2002) entre pessoas e o conhecimento disponível internamente ou externamente. Isso engloba não apenas prover aos funcionários um acesso universal às informações e ao conhecimento da empresa, mas também motivar e dar suporte ao compartilhamento de conhecimento entre eles. Esse esforço pode ser construído através de uma combinação entre uma nova cultura e processos organizacionais com soluções de tecnologia da informação, para que seja possível adquirir, gerenciar e acessar o conhecimento necessário para a execução das tarefas diárias dentro e fora

da empresa. Contudo, apesar de todo esse interesse ao redor deste tema, Gestão de Conhecimento não é uma idéia nova. No entanto, a rápida evolução da Web e o desenvolvimento de novas tecnologias de comunicação como a Web Semântica vêm tornando possível a criação de um ambiente no qual pessoas e companhias podem disseminar, adquirir, armazenar e recuperar conhecimento relacionado a suas atividades de uma forma muito mais dinâmica, universal e eficiente.

Uma organização pode ser entendida como uma rede de pessoas que compartilham metas e conceitos em prol do atingimento de objetivos comuns. Grupos de trabalho dentro de uma empresa devem compartilhar conhecimento e experiências em determinados domínios, constantemente. Pessoas que executam algum tipo de trabalho intelectual podem dispendir desnecessariamente tempo e recursos para resolver problemas que já foram experimentados por outras pessoas, justamente porque não existe uma cultura corporativa de colaboração e compartilhamento de conhecimento.

Uma solução tecnológica de Gestão de Conhecimento deve integrar conhecimento formal, semi-formal e informal para facilitar o acesso ao conhecimento da organização, seu compartilhamento e reuso, aprimorando a solução individual ou coletiva de problemas (Dieng et al., 1999). Nesse contexto, o conhecimento deve ser modelado e estruturado, para que então seja possível estabelecer todas as relações existentes entre suas manifestações. Portais de Conhecimento (Staab et al., 2001) são exemplos de ferramentas efetivas que permitem disponibilizar este conhecimento entre diversos níveis de uma companhia.

2.3. Agentes de Software e Sistemas Multi-Agentes

A Engenharia de Software de Sistemas Multi-Agentes foi proposta como uma complementação a Engenharia de Software Orientada a Objetos, com o objetivo de controlar a complexidade associada ao desenvolvimento de sistemas distribuídos em larga escala. Nos últimos anos, com os avanços das tecnologias da Web (Heflin e Hendler, 2000), observa-se uma forte tendência na utilização de novas arquiteturas de software. Estes estão deixando de ser construídos basicamente como arquiteturas monolíticas, baseadas em componentes passivos, para serem desenvolvidos em arquiteturas distribuídas, compostas pela organização de componentes autônomos que operam e se movem de forma coordenada através de diferentes ambientes em

prol de seus objetivos ((Minsky et al., 2000), (Zambonelli et al., 2002)). Mesmo sendo muito bem sucedida no desenvolvimento de sistemas de software com qualidade, a engenharia de software orientada a objetos não atende a diversos requisitos deparados durante essa transição entre arquiteturas, uma vez que as abstrações, linguagens de modelagem e metodologias não estão preparadas para dar suporte a esse novo paradigma (Silva et al., 2003).

Essas limitações motivaram a pesquisa de novas abordagens para a engenharia de software de sistemas multi-agentes (Jennings, 2000), englobando atividades que vão desde a modelagem conceitual até a modelagem computacional de sistemas em larga-escala (Silva et al., 2003). Enquanto a abstração de objetos continua sendo útil para a modelagem de recursos e de componentes passivos, a abstração de agentes é utilizada para representar componentes autônomos no sistema de software.

A noção de sistemas multi-agentes (Silva et al., 2003) e diversas teorias relacionadas descrevendo suas propriedades torna muito mais natural o suporte a características como autonomia, coordenação, mobilidade, organização, abertura e inteligência. Neste contexto, a Engenharia de Software procura entender como as lições aprendidas com a aplicação das teorias sobre agência na área de Inteligência Artificial podem ser usadas na superação das limitações encontradas na Engenharia de Software Orientada a Objetos e no controle da complexidade em sistemas de software em larga escala.

Um Sistema Multi-Agente (MAS) compreende classes e instâncias de agentes, objetos e organizações (Silva et al., 2003). Uma organização agrupa agentes de um MAS. Agentes, organizações e objetos encontram-se imersos ou habitam um ambiente (Jennings, 1999) que provê recursos e disponibiliza serviços. Recursos são entidades não-autônomas que podem ser usadas pelos agentes e organizações, como programas externos ou bases de dados. Em (Silva et al., 2003) objetos são usados como uma abstração para modelar esses recursos. Os agentes – as entidades autônomas – manipulam os objetos que representam os recursos do ambiente. Serviços são aparatos disponibilizados e utilizados pelas entidades do sistema (Kinny et al., 1997). Agentes, objetos e organizações podem disponibilizar funções – serviços – a serem usados por outras entidades. Uma organização descreve um conjunto de papéis que limita o comportamento de seus agentes, objetos e sub-organizações (Silva et al., 2003). Todo agente em um MAS deve estar relacionado com pelo menos um papel na organização. Agentes devem interagir entre si e cooperar para atingir

objetivos pessoais ou comuns. Essas interações são baseadas em relacionamentos entre os papéis definidos em uma organização.

Um agente, por sua vez, é um elemento autônomo, adaptativo e interativo que apresenta um estado mental (Silva et al., 2003). Agentes de software são objetos complexos com uma atitude (Bradshaw, 1997), isto é, são elementos que estendem objetos com um estado estruturado e propriedades de agência. Além da autonomia, adaptatividade e interatividade – as propriedades que caracterizam um agente, existem propriedades adicionais como racionalidade, capacidade de aprendizado e mobilidade, que por si só não garantem esta caracterização. O estado mental de um agente é expressado através de componentes como crenças, objetivos, planos e ações (Jennings, 2000), podendo mudar dinamicamente durante sua existência. As crenças de um agente incluem premissas e conhecimento sobre o ambiente, sobre o próprio agente e sobre outros agentes. Isto inclui também o que o agente sabe, o que ele observa, sua memória e percepções sobre tudo o que acontece no MAS. Os objetivos de um agente compreendem os estados futuros que o agente deve atingir ou satisfazer. Como agentes são entidades orientadas a objetivos, a existência de pelo menos um objetivo é fundamental. Estes objetivos devem ser alcançados através da execução de um plano. Um plano é definido como uma seqüência de ações, que devem ser executadas por um agente. Um agente atualiza seu estado mental, modifica seus papéis, percebe e gera eventos e troca mensagens enquanto executa ações. Estas possuem um conjunto de pré e pós-condições, que serão verificadas antes e depois de sua execução (Silva et al., 2003).

O comportamento de um agente é expresso através de seus planos e ações sendo baseado nas citadas propriedades de agência – interatividade, autonomia e adaptatividade. Agentes são interativos pois possuem a habilidade de interagir com outros elementos enquanto executam seus papéis na organização, a partir de seus relacionamentos. Diversos tipos de relacionamentos possíveis entre elementos de um MAS são descritos em (Silva et al., 2003). A autonomia de um agente refere-se a sua capacidade pró-ativa, pois este não depende de estímulos externos para iniciar uma tarefa. Agentes são adaptativos pois podem adaptar seu estado e seu comportamento em resposta a mensagens recebidas do ambiente ou de outros agentes.

Por fim, podem ser distingüidas duas principais classes de sistemas com múltiplos agentes (Juchem e Bastos, 2001), (Zambonelli et al., 2000):

- **Sistemas de Resolução Distribuída de Problemas:** Neste tipo de sistema os agentes são explicitamente projetados para, de maneira cooperativa, alcançarem um objetivo comum. Todos são conhecidos *a priori* e possuem confiança mútua em relação a suas interações.
- **Sistemas Abertos:** Neste caso, os agentes podem ingressar e sair do sistema dinamicamente, o que implica que estes não são conhecidos inicialmente e não foram projetados para resolver um problema de forma específica.