

5 Processo de Unificação e o KUP

5.1. O KUP

O KUP – Knowledge Unified Process – é um Processo Unificado para Engenharia de Ontologias que deverão ser utilizadas como base para o desenvolvimento de diversos tipos de aplicações para a Web Semântica, como aplicações hipermídia com anotação, web services semânticos ou agentes de software inteligentes, apenas para citar alguns exemplos.

A unificação de diversos elementos de processos em um processo unificado não pode ocorrer de forma *ad-hoc*, uma vez que o objetivo principal deste esforço é integrar, em um processo compreensivo e detalhado, as melhores práticas (ou mesmo práticas consagradas que demonstraram sua eficiência em diversas ocasiões) aplicadas na indústria.

Um conjunto de critérios explícitos precisa ser definido, para que as diversas metodologias existentes possam ser então analisadas e avaliadas. Esses critérios foram definidos com base em três fontes principais, descritas a seguir:

Em 1996 a IEEE definiu um padrão para a construção de processos de ciclo de vida de desenvolvimento de software (IEEE, 1996). Com o intuito de adaptar este padrão a metodologias e processos de desenvolvimento de ontologias, *Fernández-López et alli* (2003) propuseram então um framework de avaliação de metodologias, utilizando as bases da proposta da IEEE. Por fim, o trabalho apresentado aqui adapta e integra as melhores práticas em desenvolvimento de software apresentadas por (Rational, 1998) e a experiência adquirida em projetos anteriores para finalizar o conjunto de critérios para avaliação de metodologias e processos de engenharia de ontologias.

5.2. O Framework de Avaliação de Metodologias e Processos para Engenharia de Ontologias

Os critérios utilizados nesta primeira seção foram baseados em um framework de avaliação de metodologias para a construção de ontologias proposto e descrito em (Fernandez-López et al., 2003).

5.2.1. Critérios para Avaliação

Os critérios apresentados por este framework podem ser divididos em 4 grupos principais (Fernandez-López, 2002):

- Análise da Estratégia de Construção Proposta
- Análise do Processo de Desenvolvimento da Ontologia Proposta
- Uso atual da Metodologia ou Processo
- Suporte de ferramentas para a Metodologia ou Processo

A análise da Estratégia de Construção Proposta se divide em outros quatro tópicos, a seguir:

1. O Ciclo de Vida proposto para a ontologia, quando são analisadas questões como:
 - a. Quais são os estágios pelos quais a ontologia passa em seu ciclo de vida?
 - b. Quais são as atividades que devem ser desenvolvidas em cada estágio do ciclo de vida da ontologia?
 - c. Como os estágios estão relacionados entre si?
 - d. Como se dá a transição entre os artefatos de cada estágio?
2. A dependência da Estratégia de Construção com relação às aplicações relacionadas, de acordo com 3 grupos distintos:
 - a. Dependente de Aplicação:

Diz-se que a ontologia é dependente de aplicação quando sua construção parte de uma aplicação já existente, como uma base de conhecimento ou um software.
 - b. Semi-dependente da aplicação:

Neste caso, são identificados durante a especificação possíveis cenários de uso da ontologia.
 - c. Independente de aplicação:

Quando a ontologia tem como objetivo representar apenas a conceitualização de um domínio de conhecimento.

3. O suporte oferecido ao uso de ontologias base (core ontologies), quando os envolvidos na construção da ontologia podem usar conceitos e relações de outras ontologias como base para a definição da nova ontologia proposta.
4. E, finalmente, a estratégia para identificação dos conceitos, que pode ser caracterizada como:
 - a. Bottom-up: do mais concreto ao mais abstrato
 - b. Top-down: do mais abstrato ao mais concreto
 - c. Middle-out: do mais relevante aos mais abstratos e concretos

Com relação ao Processo de Desenvolvimento da Ontologia, têm-se as análises complementares do Processo de Gerência do Projeto, dos Processos de Desenvolvimento da Ontologia propriamente ditos e dos Processos Complementares.

1. Para a Gerência de Projeto avalia-se se a metodologia ou processo identifica a importância da definição de elementos como membros da equipe, cronograma, monitoramento e controle e gerência de qualidade, entre outros.
2. Para os Processos de Desenvolvimento da Ontologia propriamente ditos os critérios de avaliação são bem mais complexos. Eles se subdividem novamente em Processos de Pré-Desenvolvimento, Processos de Desenvolvimento e Processos de Pós-Desenvolvimento.

a. Processo de Pré-desenvolvimento:

Este processo envolve a forma como a metodologia aborda o estudo do ambiente de uso da ontologia e, em seguida, como este estudo é trabalhado em um estudo da viabilidade para projeto de construção da ontologia.

b. Processo de Desenvolvimento:

A forma como a metodologia aborda o desenvolvimento da ontologia é crucial para sua avaliação. O framework de avaliação proposto por Fernández-López aponta as três disciplinas principais desse processo, que são:

- Análise de Requisitos

As atividades desta disciplina devem permitir o levantamento e especificação dos requisitos que deverão orientar o projeto da ontologia.

- Projeto

Definidos os requisitos, a Disciplina de Projeto terá como resultado o projeto conceitual ou formal da ontologia, para posterior implementação.

- Implementação

Nesta disciplina deverá ser gerado um artefato que representará a ontologia em uma linguagem identificada como a mais adequada para sua implementação. Esta escolha deve ser baseada em critérios de análise das demandas do projeto, os quais são abordados nesse trabalho.

c. Processos de Pós-Desenvolvimento:

Após a implementação da ontologia seguem-se os processos de Pós-desenvolvimento, que devem envolver a instalação, operação, suporte, manutenção e arquivamento da ontologia.

d. Os Processos Complementares devem ocorrer de forma paralela aos outros processos de desenvolvimento, e englobam:

- Aquisição de Conhecimento
- Verificação e Validação da Ontologia
- Gerência de Configurações da Ontologia
- Desenvolvimento de Documentação
- Treinamento

A avaliação do uso atual da metodologia ou processo deve responder as

seguintes questões:

- A abordagem já foi ou está sendo usada em projetos reais?
- Qual a aceitação da metodologia em outros grupos que não o grupo que a originou?
- Que ontologias foram desenvolvidas segundo essa metodologia?
- Quais os domínios nos quais ela já foi testada?
- Quais os sistemas ou ferramentas usadas?

Por fim, o framework de avaliação propõe a análise do suporte tecnológico dado às disciplinas do processo, identificando tanto a existência de ferramentas para a execução das atividades descritas quanto a abrangência dessas ferramentas. Entre as categorias de ferramentas, API, utilitários e ambientes disponíveis podemos citar (DAML):

- Analisadores de Ontologia
- Armazenamento de Ontologias
- Bases de Conhecimento
- Bibliotecas de Ontologias
- Controle de Versão
- Crawlers para Ontologias
- Editores de Ontologias
- Ferramentas de Anotação
- Ferramentas de Marcação de Ontologias
- Ferramentas para Acesso Remoto
- Ferramentas para Apresentação
- Ferramentas para Autoria de Ontologias
- Ferramentas para Construção de Thesaurus
- Ferramentas para Exportação de Ontologias
- Ferramentas para Geração de Relatórios
- Ferramentas para Importação de Ontologias
- Ferramentas para Persistência
- Ferramentas para Querying em Ontologias
- Ferramentas para Segurança
- Ferramentas para Transformação de Ontologias entre Linguagens
- Máquinas de Busca
- Máquinas de Inferência
- Navegadores para Ontologias
- Parsers para Ontologias
- Tradução de Ontologias
- Validadores de Ontologias
- Visualizadores de Grafos de Ontologias
- Visualizadores de Ontologias

5.2.2. Melhores Práticas em Processos de Desenvolvimento

Durante muito tempo, os processos tradicionais de desenvolvimento de software foram atacados por apresentarem diversos problemas críticos (Rational,

1998). A abordagem em cascata, por exemplo, defende a completa captura de requisitos durante a fase inicial do processo, seguida então pelo projeto do sistema, codificação e integração e teste. Esta organização parece bem natural a princípio, uma vez que o objetivo do processo é produzir um artefato de software após a execução seqüencial de um conjunto de passos bem especificados. Nenhum passo é executado sem que o passo anterior seja completado e os artefatos esperados sejam gerados. A codificação e o teste são feitos antes em componentes individuais, fazendo com que as principais inconsistências sejam identificadas apenas na fase de integração, quando esses diversos componentes serão combinados em um sistema único. Essas características podem levar a dilatações nos prazos e nos orçamentos, a necessidades de mudanças no projeto altamente custosas ou mesmo ao abandono de requisitos críticos ao sistema. O principal problema apontado é que o risco se mantém alto durante todo o processo, até que seja tarde demais para mudar alguma decisão que tenha sido tomada anteriormente.

Para resolver esse tipo de problema foram identificadas seis práticas principais:

- Desenvolvimento iterativo

A idéia de dividir o processo em iterações tem por objetivo minimizar os riscos durante o processo, e não apenas em suas fases avançadas. Cada iteração pode ser entendida como um processo em cascata de escopo reduzido, com a realização de todas as disciplinas desde a definição ou reavaliação dos requisitos e resultando em um artefato de software executável. Dessa forma, as falhas são descobertas rapidamente e as decisões podem ser tomadas sem grandes impactos. A integração destes resultados executa o papel das atividades de verificação. Uma outra vantagem desta abordagem é a geração de versões simplificadas do software, permitindo a entrega de versões e distribuições auto-contidas mesmo na ocorrência de imprevistos. Da mesma forma, algumas metodologias para o desenvolvimento de ontologias trabalham com a idéia de evolução de protótipos – como as metodologias Methontology e On-To-Knowledge - permitindo a geração de artefatos intermediários que resolvam sub-conjuntos de problemas no domínio.

- Gerência de Requisitos

Para que o artefato desenvolvido atinja os objetivos de seus projetistas e patrocinadores é importante uma adequada gerência dos requisitos envolvidos. A gerência de requisitos deve implementar uma abordagem sistemática de elicitação, documentação, gerência e acompanhamento das mudanças dos requisitos do projeto. Como diversos requisitos podem aparecer apenas durante a disciplina de projeto, ao contrário do que é visto no modelo de desenvolvimento em cascata, é importante manter o rastreamento entre requisitos e projeto propriamente dito. Quando o artefato a ser desenvolvido é uma ontologia, é possível que diversos novos requisitos surjam ao longo do seu ciclo de vida, exigindo mudanças freqüentes em sua estrutura.

- Uso de Componentes Arquiteturais

Para que o desenvolvimento possa ocorrer de forma iterativa a arquitetura do sistema deve ser pensada para permitir facilmente seu reuso. Módulos, pacotes e até subsistemas podem ser encarados como componentes arquiteturais. Por terem funcionalidades bem definidas e limites claros, estes elementos facilitam a integração com outros elementos do projeto (*design*) que fazem parte da arquitetura. Além disso, podem ser identificados, isolados, projetados, desenvolvidos e testados individualmente, para então ser integrados a outros componentes da arquitetura. No caso de ontologias, este tipo de abordagem pode ser usada para a divisão do domínio de conhecimento em sub-domínios bem definidos, facilitando o uso de ontologias já existentes que permitam resolver partes do problema. Essa abordagem foi utilizada no segundo caso de teste realizado, com sucesso.

- Modelagem Visual

Modelagens Visuais facilitam a comunicação entre os membros do projeto, uma vez que apresentam uma visão concisa do sistema sem que seja necessário analisar o código do mesmo. Modelos que descrevem os relacionamentos taxonômicos e não-taxonômicos entre os conceitos de uma ontologia podem facilitar bastante seu desenvolvimento, assim como aqueles que descrevam o uso da ontologia e das bases de conhecimento por sistemas e aplicações.

- Verificação contínua de qualidade

Durante um processo de desenvolvimento iterativo os procedimentos de teste são realizados a cada iteração. Isso garante uma verificação contínua de qualidade, dado que a cada vez que um novo requisito é identificado e incluído no sistema, novos testes são realizados. Em se tratando do desenvolvimento de uma ontologia, essa verificação contínua de qualidade continua sendo importante, pois este tipo de artefato apresenta um alto grau de evolução e dinamismo, com a identificação freqüente de novos requisitos oriundos das aplicações que dele farão uso.

- Gerência de Mudanças

Mudanças nos requisitos são um acontecimento inevitável em qualquer ciclo de desenvolvimento de software, não obstante o modelo usado seja o iterativo ou em cascata. Por causa disso, essas mudanças devem ser previstas pelo processo de desenvolvimento, principalmente quando este processo envolve a participação de diversas pessoas. Devem ser gerenciados quando e como as mudanças são feitas nos artefatos do projeto, por quem elas podem ser feitas e o que caracteriza uma nova versão daquele artefato. Esta mesma idéia deve ser aplicada a um processo de desenvolvimento de uma ontologia, dado que tão natural quanto a existência de um novo requisito em um software é a identificação de um novo requisito para uma ontologia.

Apesar destas melhores práticas terem sido propostas para resolver principalmente problemas presentes em desenvolvimento de software, diversas lições podem ser aproveitadas para a avaliação de metodologias de engenharia de ontologias e, posteriormente, para a unificação do processo proposto.

5.2.3. Metodologias Avaliadas

Diversas abordagens foram avaliadas utilizando-se o framework descrito. Grande parte dessas avaliações foi obtida em um relatório apresentado em (Fernandez-López, 2002). Como foi visto, essas abordagens foram avaliadas também

de acordo com as melhores práticas apresentadas anteriormente, gerando-se ao fim uma tabela comparativa entre:

- Método de Uschold e Gruninger
- Método Cyc
- Método de Uschold e King (baseado na experiência de desenvolvimento da Enterprise Ontology)
- Metodologia de Grüninger e Fox (baseado na experiência de desenvolvimento do projeto TOVE)
- Método KACTUS (direcionado ao reuso de conhecimento através do desenvolvimento de ontologias)
- Methontology
- Métodos Sensus (desenvolvida para a geração de ferramentas de processamento de linguagem natural)
- Metodologia On-to-knowledge (focada na geração de aplicações de knowledge management)
- Método OOHDM (Schwabe e Rossi, 1998) (projeto e desenvolvimento de aplicações hipermídia)

5.3. Componentes do KUP

5.3.1. O Meta-Modelo SPEM

O SPEM (SPEM, 2002) – Software Process Engineering Metamodel - é um meta-modelo proposto pela OMG (OMG) para permitir tanto a descrição de processos de desenvolvimento de software quanto de famílias de processos de desenvolvimento de software relacionados. O SPEM usa uma abordagem orientada a objetos para que esta modelagem seja realizada, usando a notação de UML. Em SPEM (2002) é definido um conjunto mínimo de elementos de modelagem de processos necessários para descrever qualquer processo de desenvolvimento de software.

No modelo conceitual do SPEM reside a idéia de que um processo de desenvolvimento é uma colaboração entre entidades abstratas chamadas *process roles* – aqui chamados de papéis – através de *activities* – atividades – em *work products* – denominados neste documento como artefatos. Os diversos papéis interagem através da troca de artefatos e da inicialização ou execução de atividades. O principal objetivo do processo é culminar em um estado bem definido para os artefatos produzidos.

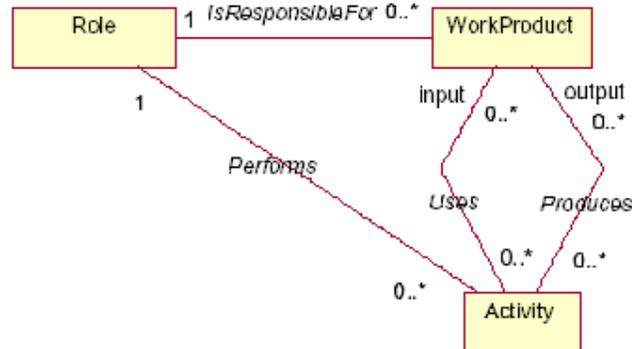


Figura 13 - Modelo Conceitual do SPEM : Papéis, Artefatos e Atividades

5.3.1.1. Papéis

Um papel é uma abstração que permite classificar executores de um conjunto específico de atividades no contexto de um processo. Essa classificação também estabelece responsabilidades com relação a artefatos que serão manipulados nessas atividades, sendo gerados ou consumidos a partir de sua execução.

5.3.1.2. Operações e Fases

Uma Operação – originalmente denominada *Work Definition* no SPEM – descreve um escopo de trabalho realizado ao longo de um processo. Atividades, Fases, Iterações e Ciclos de Vida são tipos possíveis de Operações que podem integrar um processo. Uma operação pode ser decomposta em outras operações, contendo entradas e saídas bem definidas. Atividade é o mais importante sub-tipo de Operação. Ela descreve o trabalho realizado por um Papel – *Process Role*. Uma atividade pode ser dividida em elementos atômicos denominados Passos. Um Processo, por sua vez, engloba um conjunto de operações de forma completa e auto-contida.

Uma Fase é uma especialização de uma Operação para a qual foram especificados tanto os critérios de entrada e seus resultados esperados quanto seus objetivos – *Milestones* – ou seja, os critérios de saída. Fases devem ser executadas de

forma seqüencial, com a geração dos “*Milestones*” ao longo do tempo. Um Ciclo de Vida por sua vez é definido como uma seqüência de fases que resulta em um objetivo específico, ao passo que uma Iteração é uma composição de operações que resultam na geração de “*Milestones*” intermediários (SPEM, 2002).

5.3.1.3. Disciplinas

Uma disciplina agrupa atividades que integram um processo específico de acordo com determinado tema. Essa abordagem facilita a categorização de Guias e de Artefatos sob um mesmo tema, facilitando o acesso a esse tipo de elemento.

5.3.1.4. Artefatos

Um artefato – ou *Work Product* – é tudo aquilo que é produzido, consumido ou modificado por um processo. Existem diversos tipos de artefatos possíveis, como documentos, modelos, código fonte e até mesmo programas inteiros.

5.3.1.5. Guias

Para cada elemento do modelo do SPEM podem ser associados Guias – *guidances* – para que os responsáveis pela execução do processo possam acessar maiores informações sobre tais elementos. A seguir são apresentados alguns tipos possíveis de Guias em uma lista que pode ser estendida de acordo com a necessidade do processo:

5.3.1.5.1. Guidelines - Orientações

Uma Orientação é um conjunto de regras e recomendações sobre como um artefato precisa ser organizado, ou mesmo sobre como ele deve parecer.

5.3.1.5.2. Techniques - Técnicas

Uma técnica, procedimento ou diretiva é, segundo a especificação SPEM, um algoritmo preciso e detalhado usado para a criação de um artefato.

5.3.1.5.3. Estimates – Estimativas

Uma estimativa descreve o esforço esperado associado a um elemento de processo em particular.

5.3.1.5.4. Examples – Exemplos

Um exemplo descreve a execução de uma atividade ou um artefato através de um exemplo concreto, facilitando o entendimento.

5.3.1.5.5. Tool mentors – “Guru” de Ferramenta

Um “Guru” de Ferramenta mostra como uma determinada atividade pode ser realizada com o auxílio de uma ferramenta. Esta ferramenta pode ser um CASE, um editor de ontologias, entre diversos outros tipos.

5.3.1.5.6. Lista de Verificação - Checklist

Uma Checklist contém uma lista de passos que devem ser completados na execução de uma atividade, ou mesmo uma lista de características que devem constar em um artefato.

5.3.1.5.7. Templates

Um Template é uma estrutura de um documento que deverá ser completada para a confecção de um determinado artefato.

5.3.2. Notação para Elementos Estruturais do KUP inspirada no SPEM

Os seguintes conceitos definidos pelo SPEM serão utilizados para a estruturação do KUP, acompanhados dos ícones propostos para sua notação:

Elemento	Ícone proposto pelo SPEM (ou inspirado em alguma superclasse correspondente do elemento no SPEM)
----------	---

Processos	
Papéis	
Atividades	
Artefatos	
Disciplinas	 (de Packages)
Fases	
Iterações	 (de WorkDefinition)
Guias (Templates, Exemplos, Orientações, etc)	

Tabela 2 - Notação utilizada na definição do KUP

5.3.3. Meta-Estrutura do KUP

A Figura 14 apresenta o meta-modelo do KUP utilizando-se a notação proposta anteriormente. O KUP é um processo dividido em Fases, as quais são por sua vez executadas através de Iterações nas Disciplinas de Engenharia de Ontologias. Papéis específicos relacionados a essas disciplinas são responsáveis pela execução de Atividades, consumo e geração de Artefatos, de acordo com Guias oferecidos para o Processo.

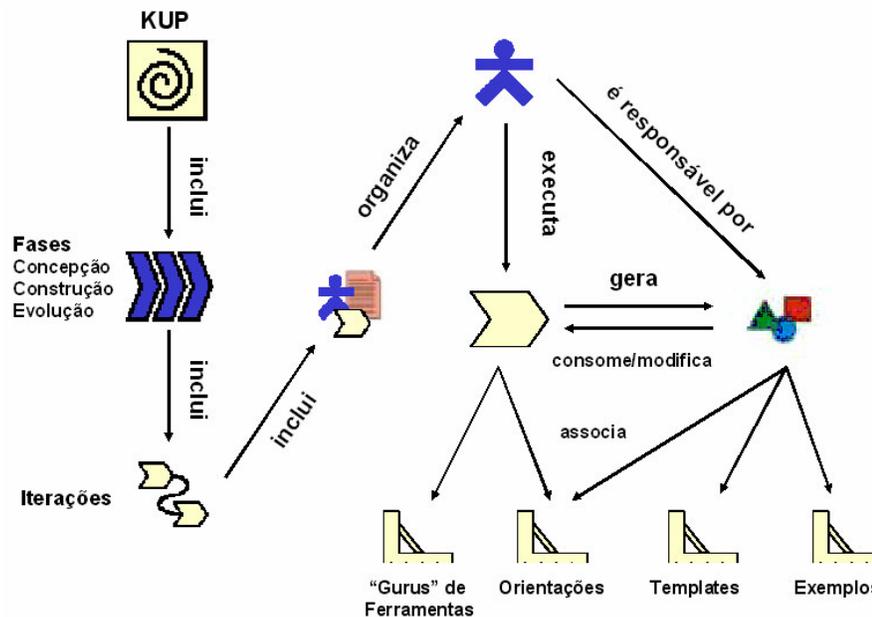


Figura 14 - Meta-Modelo do KUP

5.3.4. Organização do KUP segundo suas Fases, Disciplinas, Iterações e Atividades

5.3.4.1. Fases

Seguindo as melhores práticas apresentadas neste documento e procurando desenvolver um processo que atendesse aos critérios propostos pelo framework de avaliação de Fernández-López o Processo Unificado foi proposto contendo três fases principais:

5.3.4.1.1. Fase de Concepção

Nesta fase será analisada a viabilidade da construção da ontologia, a estratégia de desenvolvimento, seu escopo inicial e os requisitos preliminares identificados, sendo dado também um pontapé inicial no projeto e na implementação da ontologia.

5.3.4.1.2.Fase de Construção

Na fase de construção o foco será dado no projeto, na implementação e na implantação da ontologia, sem deixar de lado os novos requisitos que podem surgir e pequenas mudanças de estratégia que sejam identificadas como necessárias.

5.3.4.1.3.Fase de Evolução

As iterações da fase de evolução têm como principal objetivo integrar novos requisitos a ontologia, através da inclusão de novos conceitos, relações e axiomas, por exemplo, permitindo que novos tipos de consultas ou tarefas sejam resolvidas com o apoio da mesma.

O Diagrama de Fases do KUP – desenvolvido fortemente com base em um popular diagrama do RUP – ilustra a distribuição de esforço entre as fases e disciplinas do KUP.

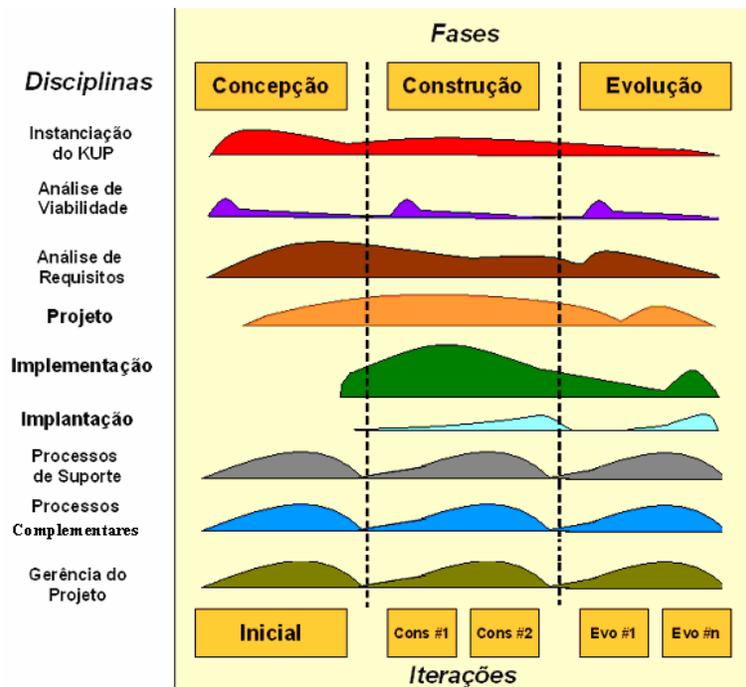


Figura 15 - Diagrama de Fases do KUP

5.3.5. Disciplinas, Atividades e Artefatos

Seguindo a mesma abordagem, as seguintes disciplinas organizam as atividades e artefatos do KUP em torno de temas relacionados:

5.3.5.1. Instanciação do Framework de Processos KUP

Definidos os elementos de processo que devem compor o KUP cabe ao projetista escolher - de acordo com os recursos disponíveis e características do projeto - quais fases, disciplinas, papéis, atividades e artefatos farão parte do processo concreto utilizado no projeto em desenvolvimento. Devem ser escolhidos os elementos que ao mesmo tempo garantam a expressividade necessária para a execução do processo bem como a facilidade desta execução. Como se sabe, um processo difícil de ser usado certamente não será usado. O processo além de ser de fácil execução, deve ser também eficiente e compreensivo, desde que não exagerado. Indo além, na abordagem do KUP o responsável pela instanciação do processo tem a possibilidade de não apenas escolher as atividades e artefatos que serão adotados mas também entre diferentes estratégias de execução propostas.

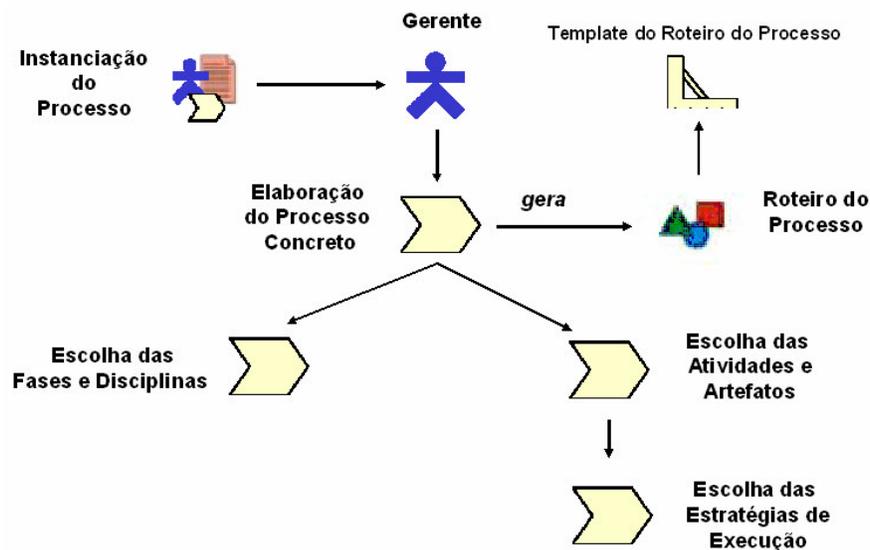


Figura 16 - Instanciação do Framework de Processos em um Processo Concreto

A Instanciação do Framework de Processos KUP em um processo concreto envolve a seguinte seqüência de passos:

1. Elaboração do Processo Concreto
 - a. Identificação do responsável geral pelo projeto da ontologia
 - b. Escolha das fases e disciplinas do processo unificado que serão executadas no processo concreto
 - c. Escolha das atividades e artefatos que compõem o processo concreto
 - d. Identificação dos responsáveis por cada disciplina e atividade
 - e. Identificação dos participantes em cada disciplina e atividade e seus respectivos papéis (vide referência de papéis)
 - f. Identificação dos participantes responsáveis por cada artefato
 - g. Identificação dos recursos que serão utilizadas em cada atividade – incluindo recursos de software
 - h. Estimativa de prazos para execução de cada atividade (se possível, com identificação de datas de início e fim para monitoramento e controle e critérios para gerência de qualidade)
 - i. Elaboração do Fluxo (Roteiro) para a Execução do Processo Concreto

Esta primeira atividade deve gerar um cronograma para o projeto com informações sobre as disciplinas, atividades e artefatos que compõem o processo concreto, o responsável por cada um desses elementos, as pessoas e recursos envolvidos, seus papéis, o tempo estimado para conclusão de cada elemento e critérios de monitoramento e controle e de gerência de qualidade. Deve ser gerado também um fluxo para a execução do processo concreto, identificando as disciplinas, atividades e artefatos envolvidos com as respectivas dependências entre tais elementos.

Para facilitar a definição dos papéis dos participantes do projeto, o KUP apresenta alguns papéis de referência, ou seja, exemplos de papéis e possíveis atividades às quais ele pode estar envolvido (Jenz, 2003).

Papel de Referência	Disciplinas e Atividades
Gerente de Projeto	Todas as disciplinas e atividades;
Analista de Requisitos	Disciplina de Análise de Requisitos e atividades de Validação e Refinamento
Engenheiro de Conhecimento	Disciplinas de Análise de Requisitos, Projeto e Implementação da ontologia.
Especialista de Domínio	Disciplinas de Análise de Requisitos e

	Projeto.
Arquiteto de Aplicação	Disciplina de Implantação

Tabela 3 - Papéis de Referência e Disciplinas Relacionadas

5.3.5.2. Análise de Viabilidade

Diversos fatores não necessariamente tecnológicos podem fadar o projeto da ontologia ao sucesso ou ao fracasso (Sure et al., 2002c). A análise de viabilidade do projeto deve permitir a identificação prévia de possíveis oportunidades que podem acelerar o processo de desenvolvimento ou de possíveis problemas que podem aumentar o seu risco. A análise de viabilidade deve focar em aspectos técnicos, econômicos e de projeto, de forma a minimizar os riscos nas fases seguintes.

De acordo com o framework de avaliação proposto por (Fernandéz-López et al., 2003), adaptando (IEEE, 1996), a Análise de Viabilidade do KUP mapeia as etapas de “Estudo do Ambiente” e “Estudos de Viabilidade”. Estas etapas estão relacionadas aos sub-processos de pré-desenvolvimento relacionados ao processo de desenvolvimento da ontologia propriamente dita.

A Metodologia On-To-Knowledge (OTK) foi escolhida como a base para a Análise de Viabilidade do KUP uma vez que apresenta um alto grau de maturidade, tendo sido utilizada em uma série de estudos de caso (Sure et al., 2002c). Outras abordagens, quando levam em consideração estas etapas o fazem de forma muito superficial e pouco descritiva. No entanto, algumas alterações e aprimoramentos foram feitos apenas no KUP, uma vez que a OTK não apresentava todas as características apontadas pelo framework de avaliação.

Segundo Sure et alli, (2002c), a análise de viabilidade deve ser conduzida para permitir a futura identificação:

- dos participantes do projeto, tanto usuários – humanos e agentes de software - quanto patrocinadores e executores;
- dos casos de uso que descrevam cenários de uso da ontologia ou de aplicações resultantes por potenciais usuários;
- dos casos de uso que descrevam cenários de uso de suporte da ontologia ou aplicações resultantes.

A metodologia CommonKADS (Schreiber et al., 1999) apresenta três modelos para a realização da análise de viabilidade: o modelo da organização, de tarefas e de agentes. Para a construção desses três modelos os seguintes passos devem ser executados (Sure et al., 2002c):

- Realização de um estudo sobre o problema e seu escopo, dividido em duas partes:
 - Identificação de problemas e oportunidades e das soluções em potencial
 - Decisão a respeito da viabilidade econômica, técnica e de projeto para suavização dos riscos nas etapas seguintes
- Realização de um estudo sobre os impactos e melhorias que o desenvolvimento da ontologia e aplicações relacionadas poderá trazer aos seus usuários e patrocinadores, também dividido em duas partes:
 - Estimar - através de uma análise da utilização da ontologia e suas aplicações, dos atores envolvidos e do uso do conhecimento mapeado – que melhorias que podem ser alcançadas após implantação dos resultados do projeto.
 - Tomar providências no sentido de se aplicar métricas e gestão de mudanças para permitir a adoção dos resultados do projeto.

No contexto do KUP, estas abordagens são adaptadas para que seja possível responder às seguintes questões:

1. Com relação ao Ambiente e Motivação para o Projeto:
 - Identificação de Oportunidades:
 1. Quais são as necessidades observadas e os problemas ainda não resolvidos?
 2. Quem está interessado em resolver estes problemas?
 - Identificação de Problemas:
 1. Quais são os problemas que podem afetar o projeto e desenvolvimento da ontologia?
 2. Como esses problemas podem ser contornados durante o projeto?
 - Identificação de Potenciais Soluções:
 1. Que soluções podem ser apresentadas para resolver as necessidades e problemas ainda não resolvidos?
 2. Que ontologias poderiam ajudar nessa solução?
 3. Que aplicações poderiam ajudar nessa solução?
2. Com relação à Análise dos Recursos Disponíveis:
 1. Qual a disponibilidade de recursos técnicos?
 - Ferramentas, Fontes de Informação, Especialistas no domínio, etc.
 2. Qual a disponibilidade de recursos econômicos?

- Capital
- 3. Qual a disponibilidade de recursos de projeto?
 - Tempo, Equipe, etc.

Após a Análise de Viabilidades o projetista deve ser capaz de decidir se o projeto é viável ou não ou mesmo tomar iniciativas que o tornem viável, bem como direcionar as próximas etapas no sentido de minimizar os riscos avaliados. A Figura 17 expressa a seqüência de atividades e a geração do Documento de Análise de Viabilidade gerado ao fim desta disciplina. O Template para este documento encontra-se incluso no Anexo I.

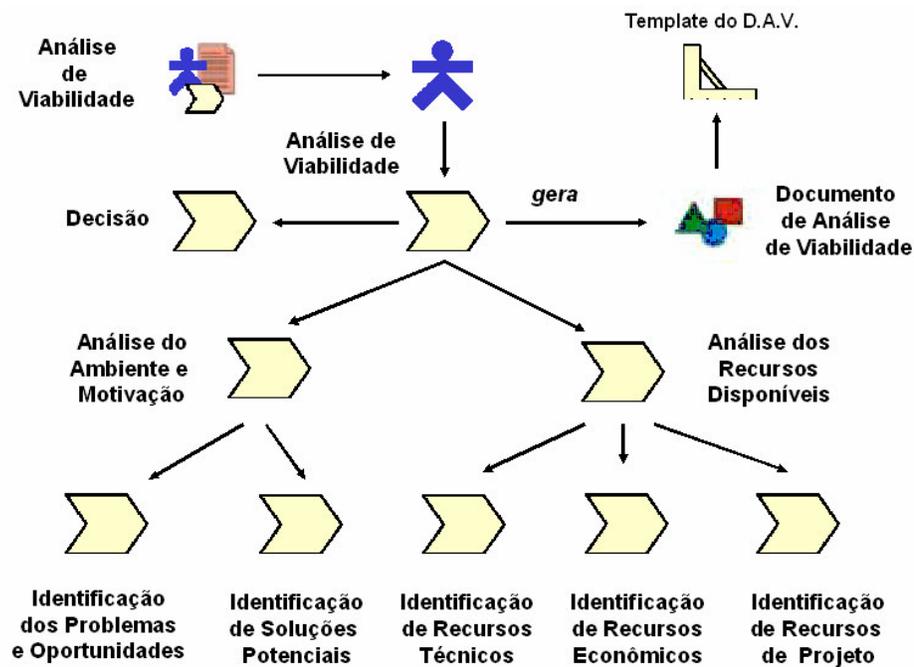


Figura 17 - Modelo da Disciplina de Análise de Viabilidade

5.3.5.3. Análise de Requisitos

A Disciplina de Análise de Requisitos compreende atividades iterativas que permitem desenvolver a especificação de requisitos da ontologia. Esta disciplina mapeia - de acordo com o framework de avaliação - as atividades relacionadas ao Processo de Requisitos pertencente aos sub-processos de Desenvolvimento relacionados ao Processo de Desenvolvimento da Ontologia propriamente dito.

Para a definição da Análise de Requisitos do KUP foram extraídos conceitos e abordagens de diversas metodologias, processos e métodos distintos, a seguir.

- Método de Uschold & Gruninger
- Método de Uschold & King
- Metodologia de Gruninger & Fox
- Método KACTUS
- Metodologia Methontology
- Método Sensus
- Metodologia On-To-Knowledge
- Método OOHDM
- Desenvolvimento de Ontologias baseado em Léxicos

Na disciplina de Análise de Requisitos devem ser iterativamente levantados os requisitos da ontologia ou das aplicações que serão desenvolvidas a partir dela. Primeiramente, é importante identificar o motivo pelo qual a ontologia está sendo construída e quais os usos que serão feitos dela. Uschold e Gruninger (1996) e Fernandez-Lopez *et alli* (1997) propuseram uma atividade de Identificação do Propósito e Escopo da ontologia, quando devem ser identificados seus Espaços de Uso e seus potenciais usuários. Cabe avaliar, também, os benefícios que essa ontologia trará aos usuários identificados, bem como as possíveis aplicações que podem ser desenvolvidas a partir da mesma, o que irá permitir uma definição mais completa de seus requisitos.

Existem diversas descrições sobre ontologias e seus usos na literatura (Uschold e Gruninger, 1996). A grande maioria delas tem como objetivo possibilitar o reuso, com propósito implícito em uma ou mais das diversas definições de ontologia - como a definição de um vocabulário, a conceitualização ou teoria sobre um domínio, entre outras. Outra forma de descrever o uso da ontologia é através da natureza dos softwares que utilizarão a ontologia, seja ela direcionada a apenas um tipo em especial de software ou flexível o suficiente para atender a uma gama variada de sistemas. Uma ontologia pode ser usada apenas para estruturar uma base de conhecimento, para a anotação de páginas na web, como um meio para integrar aplicações, entre diversas outras utilidades. Para organizar a descrição do uso de uma ontologia, Uschold e Gruninger (1996) propuseram o conceito de Espaços de Uso seguindo três categorias:

- Comunicação
- Interoperabilidade
- Engenharia de Sistemas: Especificação, Confiabilidade e Reusabilidade

Como já foi visto, o uso de ontologias permite um entendimento compartilhado sobre um determinado domínio, podendo ser de grande valia para diversos tipos de organizações. Este entendimento compartilhado é um fator crucial para a comunicação entre pessoas com diferentes necessidades e vivendo contextos particulares. Os autores identificam, nesta primeira categoria, diferentes aspectos relacionados ao uso de ontologias para facilitar a comunicação entre pessoas de uma organização. São eles:

- Geração de Modelos Normativos
 - Em qualquer sistema de software em larga escala que precise ser integrado, diferentes pessoas necessitam ter um entendimento compartilhado sobre o sistema e seus objetivos. Através de uma ontologia é possível criar um Modelo Normativo para o mesmo, criando para ele uma semântica compartilhada. Essa semântica pode permitir, no futuro, transformações entre diferentes contextos.
- Criação de uma Rede de Relacionamentos
 - Uma ontologia pode ser criada com o objetivo de se construir uma rede de relacionamento entre itens de conhecimento, permitindo identificar que elementos estão semanticamente relacionados entre si, explorar os relacionamentos na resolução de tarefas e navegar através dessas relações. As ontologias são utilizadas para a definição explícita de todas as conexões lógicas entre os elementos que compõem o sistema, permitindo, novamente, um entendimento compartilhado de sua semântica.
- Garantia de Consistência e Inexistência de Ambigüidades
 - Uma das principais atribuições de uma ontologia é prover definições não ambíguas para termos que são usados em algum sistema. Esta tarefa envolve a identificação das diferentes perspectivas dos usuários, ferramentas ou outras ontologias envolvidas para a identificação de sinônimos e termos relevantes.
- Integração de Diferentes Perspectivas
 - No caso de um sistema com diversos agentes – humanos ou de software – comunicando-se entre si, a integração através de um entendimento compartilhado é crucial para seu correto funcionamento. O processo de desenvolvimento de uma ontologia também pode contribuir para que pessoas com diferentes perspectivas organizacionais cheguem a um consenso e compartilhem visões diferenciadas.

Além do seu uso na Comunicação, as ontologias também são freqüentemente aplicadas para atender a requisitos de Interoperabilidade. Isto significa a existência de diversos usuários, sistemas e agentes que precisam trocar dados entre si. A seguir

algumas aplicações diferenciadas de ontologias para promover interoperabilidade são descritas:

- Uso de Ontologia como Inter-língua
 - Ontologias podem ser usadas para dar suporte a tradução entre diferentes linguagens e representações. Uma abordagem comum é a confecção de tradutores para cada par de representações. Obviamente, isto iria exigir n^2 tradutores para n representações. O uso de uma Inter-língua diminui a quantidade de tradutores necessários para $2n$, uma vez que qualquer ontologia só precisará ser traduzida para a ontologia comum e desta para a outra desejada.

- Possibilitar a Troca de Informações
 - A troca de informações entre sistemas de diversos tipos é o principal requisito de interoperabilidade. No entanto, existem diversos tipos e dimensões distintas de interoperabilidade, descritos a seguir:
 1. Quanto a fronteira da interoperabilidade:
 - a. Interoperabilidade Interna
 - Neste caso, todos os sistemas que necessitam trocar informações são de responsabilidade de uma única organização.
 - b. Interoperabilidade Externa
 - Trata-se da interoperabilidade entre sistemas internos de uma organização e dados externos, provenientes de outras fontes.
 2. Quanto ao escopo da interoperabilidade:
 - a. Integração de Ontologias de Diferentes Domínios
 - Este tipo de integração reflete a necessidade de se integrar ontologias de diferentes domínios com o intuito de se dar suporte a determinado conjunto de tarefas.
 - b. Integração de Ontologias de Diferentes Ferramentas
 - Neste caso são integradas ontologias de um mesmo domínio mas provenientes de diferentes ferramentas ou sistemas.

A terceira categoria envolve a aplicação de ontologias no suporte ao projeto e desenvolvimento de sistemas de software. Esta categoria apresenta três linhas principais, que são apresentadas abaixo:

- Especificação
 - Um entendimento compartilhado do problema e das tarefas a serem resolvidas certamente facilita o trabalho de especificação de um sistema de software, principalmente em projetos que envolvam diversos participantes. Essa abordagem é utilizada por várias metodologias e processos de desenvolvimento, com um número considerável de variantes. Essas variantes estão relacionadas a diversos fatores, como, por exemplo, o grau de formalidade na utilização destas tecnologias e quanto trabalho pode ser automatizado no projeto do sistema. Ontologias podem facilitar o processo

de identificação dos requisitos do sistema bem como o entendimento das relações entre seus componentes. A ontologia também pode ser utilizada para a geração de especificações declarativas sobre o sistema a ser desenvolvido, especialmente em metodologias que envolvam especificação formal.

- **Confiabilidade**
 - De acordo com seu grau de formalidade, as ontologias podem ser usadas tanto para auxiliar na checagem do projeto do sistema de acordo com a sua especificação até na geração de checagem automática dos sistemas frente à especificação declarativa. As ontologias podem também ser usadas para explicitar várias decisões sobre componentes do sistema.

- **Reusabilidade**
 - Ontologias podem ser usadas para dar suporte a reusabilidade, fornecendo informações sobre componentes arquiteturais que possam ser importados ou exportados entre diferentes sistemas.

Resumidamente, nesta primeira etapa da análise de requisitos os participantes do projeto devem ser capazes de responder às seguintes questões:

1. Identificação do propósito e escopo
 - a. Por que a ontologia será construída?
 - Quais são os usos pretendidos?
 - Elaboração da descrição
 - Indicação da associação aos espaços de uso de Uschold e Grüninger (1996)
 - b. Quais são os potenciais usuários?
 - c. Quais os benefícios que a ontologia trará para a execução das tarefas dos potenciais usuários?
 - d. Quais as aplicações potenciais deverão fazer uso da ontologia?

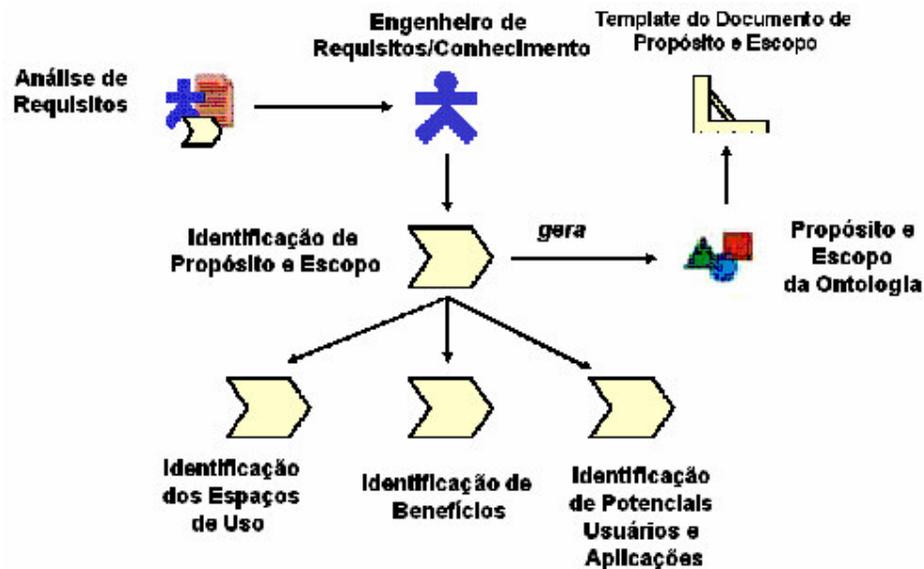


Figura 18 - Análise de Requisitos - Identificação do Propósito e Escopo

Em seguida a identificação do propósito e escopo da ontologia podem iniciarse as atividades de captura dos Cenários Motivacionais. Cenários Motivacionais tem como objetivo identificar situações nas quais as ontologias ou aplicações existentes atualmente não são capazes de resolver as necessidades descritas na fase de Análise de Viabilidade. Este tipo de abordagem foi proposta por Grüninger e Fox e é extremamente útil para as atividades seguintes de elicitação de requisitos. No contexto do KUP seu formato segue a descrição de cenários através de narrativas em linguagem natural. Obviamente, seguindo a idéia de iterações, diversas questões não levantadas durante a identificação de propósito e escopo - ou mesmo na Análise de Viabilidade – poderão ser percebidas durante a captura desses cenários.

Devem ser respondidas as seguintes perguntas:

1. Captura dos Cenários Motivacionais (Grüninger e Fox, 1995)
 - a. Quais são os cenários não atendidos pelas ontologias disponíveis?
 - Como esses Cenários podem ser descritos em linguagem natural?
 - b. Por que as ontologias atuais não resolvem os problemas descritos na disciplina de Análise de Viabilidade?

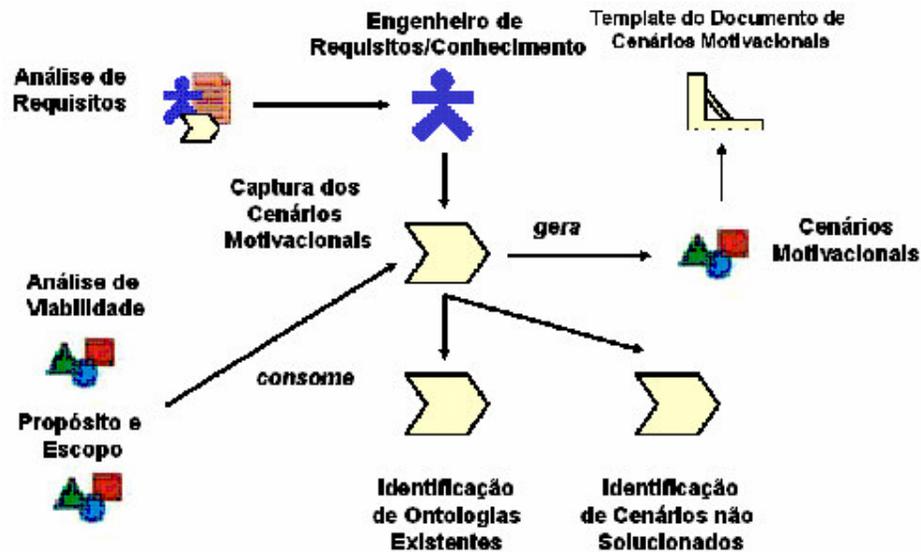


Figura 19 - Análise de Requisitos : Captura dos Cenários Motivacionais

Após a Captura dos Cenários Motivacionais passa-se às atividades de Elicitação de Requisitos, as mais importantes da disciplina de Análise de Requisitos. A primeira atividade reside na Identificação das Fontes de Informação para a construção da ontologia e de sua base de conhecimento. A metodologia On-To-Knowledge apresenta uma lista de tipos de fontes de informação possíveis, atualizada no contexto do KUP.

1. Especialistas no domínio
2. Ontologias existentes
3. Dicionários
4. *Thesauri*
5. Fontes internas
 - a. Bancos de Dados
 - b. Esquemas de Bancos de Dados
 - c. Modelos
 - d. Modelos de Sistemas de Software
 - e. Templates
 - f. Relatórios
 - g. Gráficos e diagramas
 - h. Listas de Índices
 - i. Legislações
 - j. Descrições de produtos, projetos, pessoas, etc
 - k. White-papers
 - l. Páginas Web
 - m. Documentos em geral
6. Especificações

7. Padrões

A existência de fontes de informação é crucial tanto para o desenvolvimento da ontologia quanto para as aplicações que serão desenvolvidas a partir dela. A escassez dessas fontes pode impactar na viabilidade do projeto, e deve ser avaliada também nas atividades de identificação de recursos técnicos disponíveis quanto sua facilidade de obtenção, sem necessidade de enumeração.

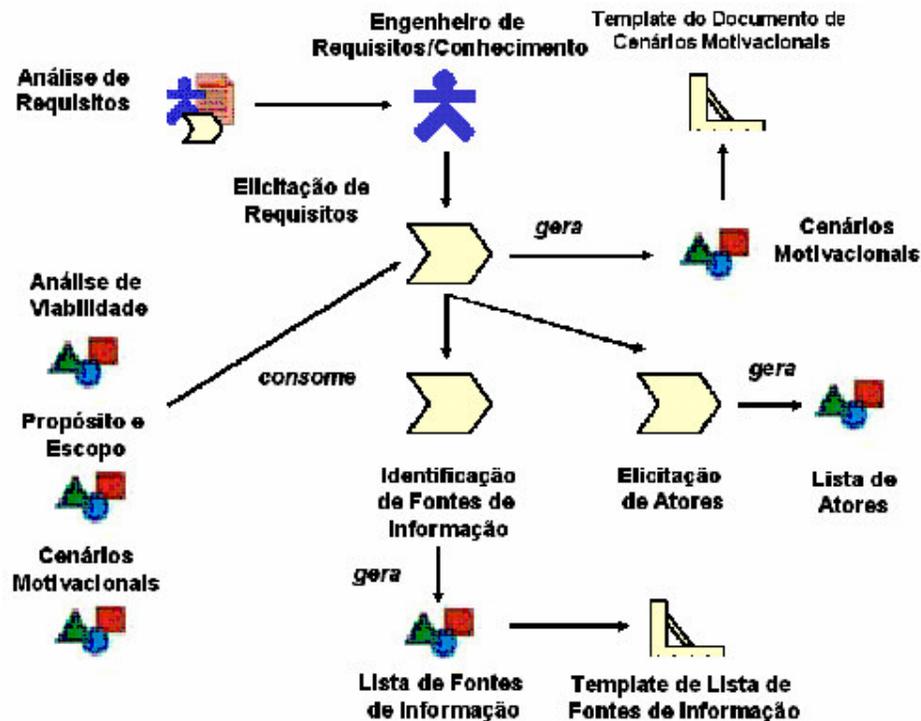


Figura 20 - Análise de Requisitos – Identificação de Fontes de Informação durante a Elicitação de Requisitos

Em seguida, parte-se para a atividade de Elicitação de Atores. Devem ser identificados os papéis de usuários típicos, outras aplicações e agentes de software que farão uso da ontologia ou das aplicações que forem desenvolvidas a partir dela. Estes papéis permitirão identificar e elaborar os cenários de uso da ontologia, a próxima atividade.

O KUP orienta ao projetista a escolha da abordagem mais conhecida por sua equipe na elaboração dos casos de uso da ontologia e de suas possíveis aplicações, ou aquela que achar mais adequada. Cada abordagem apresenta, obviamente, vantagens

e desvantagens, enfatizando alguns pontos em detrimento de outros – como o foco em algum tipo de aplicação - ou mesmo possuindo uma maior complexidade, o que exige um maior dispêndio de recursos. As três abordagens para a elaboração dos Cenários e Casos de Uso propostas como alternativa no KUP são:

1. Casos de Uso do Rational Unified Process estendido por KUP (Modelo apresentado no Anexo I)
2. Léxico Ampliado da Linguagem e Cenários, proposto por (Leite,1997)
3. Desenvolvimento dos Diagramas de Interação com o Usuário (UIDs), propostos no Método OOADM (Güell et al., 2000)

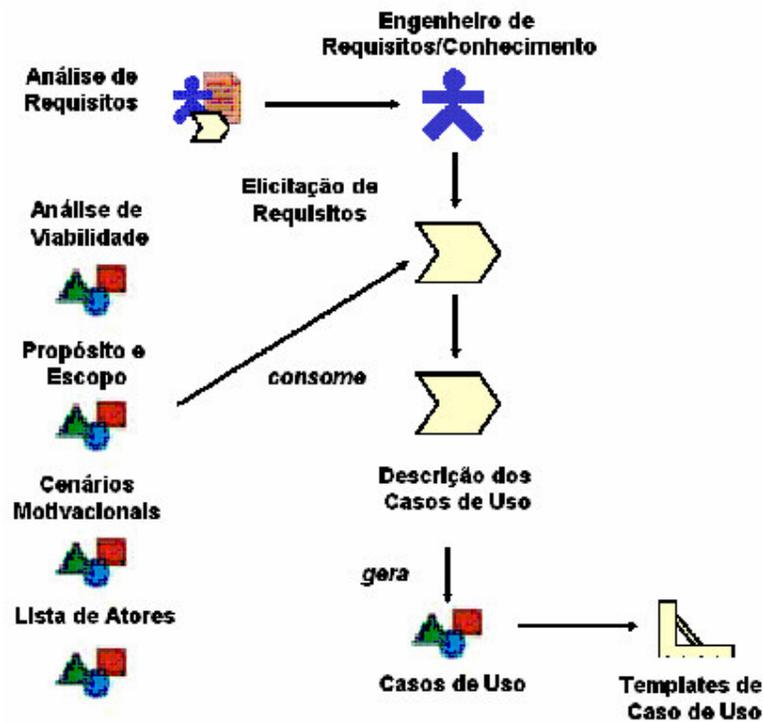


Figura 21 - Análise de Requisitos – Elaboração dos Casos de Uso durante a Elicitação de Requisitos

Estas atividades têm como objetivo facilitar o processo de identificação das tarefas que terão que ser resolvidas com o suporte da ontologia - seja por aplicações,

agentes de software ou *web services* - para posteriormente, orientar a identificação do escopo e delimitação do domínio de conhecimento a ser mapeado.

Algumas das abordagens avaliadas defendem a elaboração das chamadas Questões de Competência Informais (Ushold e Gruninger, 1996) para a descoberta de que conhecimento deve ser mapeado pela ontologia. Em (Ushold e Gruninger, 1996) as Questões de Competência informais são consideradas requisitos de expressividade da ontologia e são descritas na forma de consultas. No entanto, percebe-se uma certa falta de critério para identificação destas consultas, uma vez que estas não estão mapeadas na resolução de problemas específicos das aplicações em potencial, e são escolhidas de uma forma significativamente subjetiva.

O principal problema que o KUP tenta resolver é facilitar o trabalho de identificação dessa expressividade necessária, a partir da identificação das tarefas que terão que ser resolvidas a partir desta ontologia. Esta identificação deve ser feita através da análise das respostas dadas pela ontologia ou base de conhecimento com base seqüências de ações identificadas nos casos de uso que terão como atores suas aplicações em potencial.

Essas tarefas serão descritas a partir de cadeias de consultas – as já descritas Questões de Competência Informais, apresentadas como questões textuais. Essas questões também deverão permitir ao projetista decidir se a ontologia é mesmo necessária (de acordo com sua comparação com os Cenários Motivacionais) ou se podem ser totalmente respondidas com base em ontologias existentes.

As questões de competência devem ser apresentadas de forma estratificada, sendo representadas no seu limite superior pelas tarefas identificadas e no nível mais concreto por consultas simples. Entre cada nível, conforme representado na Figura 22, deve estar descrito como a resposta de cada questão de competência é usada para solucionar questões mais complexas e quais as Questões de Competência devem ser respondidas anteriormente para que seja possível resolvê-la. Os templates para os artefatos Lista de Tarefas e Lista de Questões de Competência - apresentados no Anexo I - devem auxiliar nestas atividades.

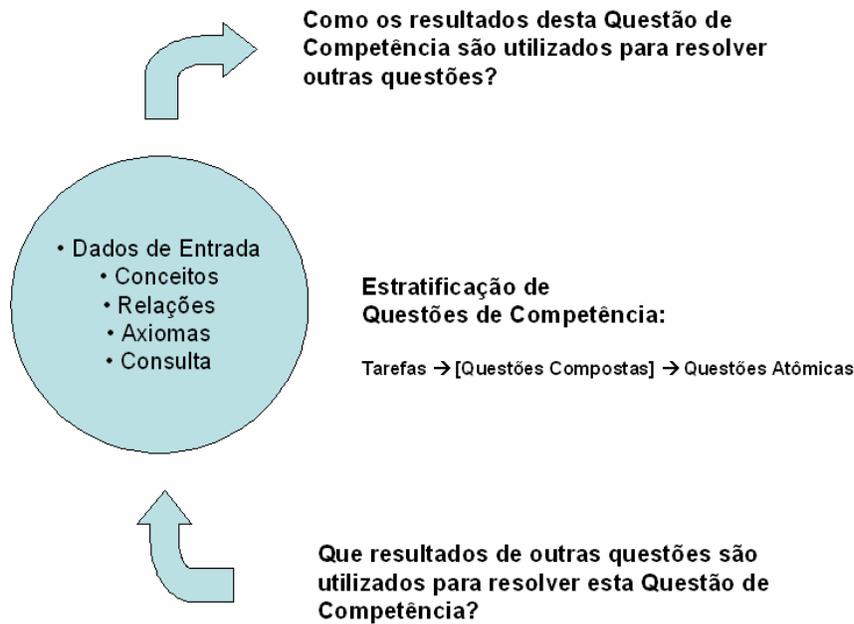


Figura 22 - Estratificação das Questões de Competência

Futuramente, a ontologia construída deve ser capaz de representar essas questões usando sua terminologia e gerar as respostas necessárias através de axiomas, definições e consultas. Nesta atividade, porém, o objetivo é representar as questões de maneira informal, uma vez que a linguagem formal na qual a ontologia será representada ainda não foi escolhida.

5.3.5.4. Projeto

A Disciplina de Projeto compreende as atividades que irão culminar com a geração da ontologia em uma linguagem formal ou mesmo em um modelo conceitual que permita sua implementação, através da identificação de seus conceitos, relações e axiomas. Esta disciplina mapeia - de acordo com o framework de avaliação apresentado neste documento - a etapa de Projeto pertencente aos sub-processos de Desenvolvimento.

De acordo com esse mesmo framework de avaliação, as seguintes metodologias, processos e métodos descrevem ou propõem atividades relacionadas a essa etapa:

- Método de Uschold & Gruninger
- Metodologia de Gruninger & Fox
- Método KACTUS
- Metodologia Methontology
- Metodologia On-To-Knowledge
- Método OOHDM
- Princípios de Noy e MacGuinness – Ontology 101 (Noy and McGuinness, 2001)

A metodologia de Gruninger e Fox (1995) aborda o problema propondo, como primeira atividade da disciplina de projeto, a Especificação da Terminologia da Ontologia em Linguagem Formal. Esta especificação parte da extração dos termos usados na definição informal das questões de competência e sua posterior formalização em alguma linguagem de representação de conhecimento. A próxima atividade compreende a Formulação das Questões de Competência usando a Terminologia definida na atividade anterior. A Especificação de Axiomas para a descrição da semântica desses termos conclui, segundo a metodologia de Gruninger e Fox, o projeto da ontologia. Neste ponto, os axiomas propostos em conjunto com a terminologia da ontologia devem ser avaliados para que seja possível verificar se todas as questões de competência podem ser então representadas e respondidas com base nesses elementos.

A metodologia Kactus (Bernaras et al., 1996) insere uma nova contribuição para a disciplina de Projeto, que reside na busca de outras ontologias já desenvolvidas que possam ser refinadas ou estendidas para dar suporte a questões de competência elaboradas no contexto da ontologia sendo construída.

No que tange o projeto da ontologia os autores da Methontology iniciam as atividades relacionadas a essa disciplina através da Conceitualização do conhecimento adquirido. Este conhecimento adquirido é representado através de um Modelo Conceitual que tem como objetivo descrever aspectos tanto do problema como de suas soluções (Gómez-Pérez et al., 1996). Este Modelo Conceitual deve então ser transformado em um artefato que seja computável, através de sua Formalização em alguma linguagem de representação escolhida.

Como tratam de um entendimento compartilhado sobre um domínio de conhecimento, uma ontologia bem modelada apresenta um alto grau de reusabilidade (Fernandez-López et al., 1997). A disciplina de Projeto é apontada como o momento mais adequado para a integração de outras ontologias que já tenham sido desenvolvidas para descrever o domínio ou parte do domínio em análise, diminuindo

o esforço necessário para a construção da ontologia em questão. Em (Gangemi et al., 1996) é apresentado um método que orienta este tipo de integração e demonstra sua aplicação no domínio médico. Foram identificados em (Farquar et al., 1995) quatro tipos de operações entre ontologias que podem ser integradas. São eles: inclusão, refinamento polimórfico, dependência circular e restrições. O objetivo da integração é, além de permitir o reuso de ontologias, a geração de um artefato consistente e sem conflitos entre a semântica de seus conceitos, símbolos, propriedades e axiomas.

Em (Noy e McGuinness, 2001) podem ser vistas uma série de orientações pragmáticas que devem ser seguidas durante o projeto da ontologia, que são resumidas a seguir:

- Ao construir uma ontologia, o projetista não deve se preocupar em abordar toda a informação possível sobre o domínio em questão, mas sim toda a informação que seja necessária para a resolução dos problemas levantados nas disciplinas anteriores a disciplina de projeto.
- Não é necessário nem mesmo recomendado especializar ou generalizar os conceitos da taxonomia que compõe a ontologia mais do que o necessário para a resolução dos mesmos problemas citados. Segundo as autoras, apenas um nível acima ou abaixo desses limites deve ser mapeado na ontologia.
- Da mesma forma, a ontologia não deve conter todas as relações possíveis – ou seja, existentes no domínio em questão – entre os conceitos que formam a ontologia. Deve conter, sim, apenas aquelas que façam sentido na resolução das questões de competência identificadas anteriormente. O mesmo vale para a distinção entre conceitos – que só devem existir quando forem de comprovada utilidade para as aplicações que farão uso da ontologia.
- Quando são mapeadas as relações taxonômicas da ontologia, para que seja justificada a criação de um sub-conceito, uma das situações a seguir deve ser verdadeira:
 - Existe pelo menos uma relação neste sub-conceito que o super-conceito não possui;
 - O sub-conceito possui restrições adicionais ao super-conceito;
 - A ontologia que está sendo construída será usada como hierarquia de termos, sendo importante, então, representar a diferença entre os conceitos.

A Metodologia On-To-Knowledge segue estes princípios, propondo como artefato dessa atividade uma descrição semi-formal da ontologia, expressa através de um grafo de nós (nomeados), representando os conceitos, e arestas (nomeadas ou não, direcionadas ou não) representando as relações entre os conceitos.

Em seguida passa-se a formalização dessa descrição semi-formal, através da conclusão dos seguintes passos:

- Constrói-se a taxonomia da ontologia, identificando-se as relações de especialização ou generalização entre os conceitos identificados na Conceitualização.
- Adicionam-se as relações não-taxonômicas entre os conceitos.
 - A melhor maneira de se identificar as relações não-taxonômicas está na análise das Questões de Competência Informais, que devem permitir a identificação tanto de atributos quanto de relações entre conceitos da ontologia.
 - Recomenda-se a adição primeiramente de atributos aos conceitos, seguida pela adição das relações entre os mesmos.

A estratégia de identificação dos conceitos pode ser enquadrada em três categorias:

- Estratégia Top-down:
 - Os conceitos e suas relações são identificados partindo-se dos Cenários de Uso e das Questões de Competência Informais.
 - Esse tipo de abordagem leva a um entendimento mais detalhado do problema e de seu domínio, sendo centrado na demanda por informação.
- Estratégia *Middle-out*:
 - Nesta estratégia, parte-se dos conceitos mais importantes para que depois sejam identificados os conceitos complementares. Isto também pode ser feito a partir dos artefatos gerados pela disciplina de Análise de Requisitos, mas o foco passa a ser dado nos conceitos centrais da ontologia.
 - Neste caso, a grande vantagem é que podem ser mais facilmente gerados subconjuntos da ontologia com os conceitos mais relevantes, possibilitando a geração de protótipos mais rapidamente.
- Estratégia Bottom-up
 - A estratégia bottom-up é bastante aplicada com base na análise automática, semi-automática ou manual das fontes de informação disponíveis. Procura-se, através desta análise, identificar termos freqüentes que possam estar relacionados a conceitos candidatos da ontologia. Se a estratégia *top-down* é focada na demanda por informação, nesta estratégia o foco se dá na disponibilidade de informação.

Obviamente, a escolha da abordagem depende da disponibilidade de fontes de informação e da clareza dos objetivos da ontologia. Os autores da Metodologia On-to-knowledge propõem uma mistura das abordagens, dependendo da disponibilidade, da qualidade e da credibilidade das fontes de informação, utilizando-

se, quando possível, de mecanismos para a extração do conhecimento intrínseco a essas fontes.

Para a nomenclatura dos conceitos e propriedades - na ocasião de Formalização da ontologia – pode-se seguir as recomendações da ISO 704 (ISO, 2000), que define princípios para construção de terminologias.

Com base nessas diferentes propostas apresentadas, o Processo Unificado (KUP) propõe a Disciplina de Projeto da ontologia organizada como a seguir:

A primeira atividade a ser executada compreende a Conceitualização da Ontologia. Esta atividade KUP engloba abordagens das metodologias de Gruninger e Fox e Methontology e pode deve ser composta pelas seguintes sub-atividades:

- Extração dos Termos (para conceitos e propriedades) a partir do artefato que descreve as Questões de Competência Informais
- Formulação das Questões de Competência usando a Terminologia definida na atividade anterior
- Construção de um Modelo Conceitual para a Ontologia – Modelagem Conceitual
 - Constrói-se a taxonomia da ontologia, identificando-se as relações de especialização ou generalização entre os conceitos identificados na Conceitualização.
 - Adicionam-se as relações não-taxonômicas entre os conceitos, conforme orientado pela Metodologia On-To-Knowledge.
- Identificação de ontologias que podem ser integradas de acordo com o Modelo Conceitual

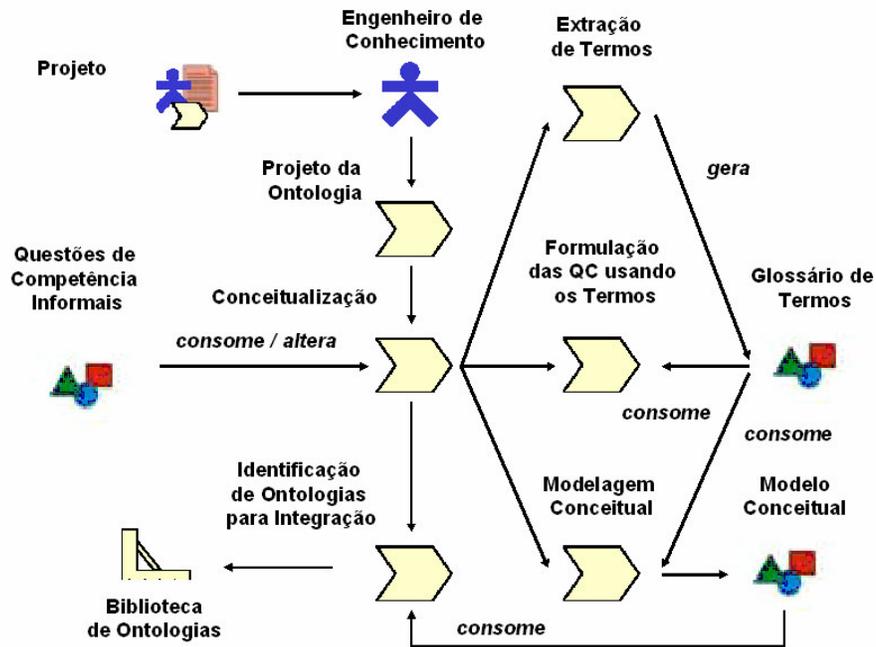


Figura 23 - Projeto da Ontologia - Conceitualização

A partir do Modelo Conceitual da Ontologia, parte-se para sua Formalização, que engloba os seguintes passos:

- Integração das ontologias identificadas na atividade de Conceitualização
- Especificação da Ontologia em Linguagem Formal
 - Especificação da Terminologia da ontologia em Linguagem Formal
 - Especificação dos Axiomas para a descrição dos termos da ontologia em Linguagem Formal
 - Especificação das Questões de Competência em Linguagem Formal

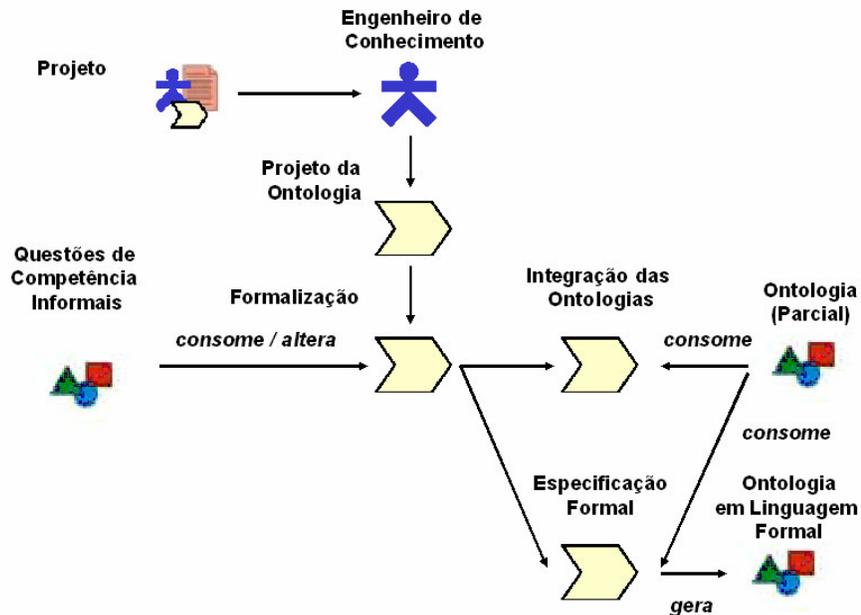


Figura 24 - Projeto da Ontologia - Formalização

Após a realização das atividades de Conceitualização e Formalização devem ser produzidos dois artefatos principais:

- Modelo Conceitual da Ontologia
 - A definição de um modelo conceitual para a ontologia tem como objetivo facilitar o entendimento da ontologia sem que seja necessário entender a linguagem de representação escolhida. Facilita, também, a troca de informações entre os membros da equipe. Um exemplo de modelo conceitual foi gerado no escopo do segundo estudo de caso e é apresentado no Anexo 3 a este documento.
- Ontologia descrita em Linguagem Formal, como, por exemplo, Description Logic ou Frames.

As atividades de Conceitualização e Formalização são seguidas por atividades de Avaliação da Ontologia Projetada e Refinamento dos resultados. Segundo Gomez-Perez (1996), avaliar uma ontologia é elaborar um julgamento técnico da mesma de acordo com alguma referência. Dependendo do tipo de ontologia sendo desenvolvida (Guarino, 1998), essa referência pode ser tanto o conjunto de requisitos levantados na disciplina de Análise de Requisitos (verificando-se, por exemplo, como os Cenários Motivacionais foram satisfeitos pela nova ontologia), as Questões de Competência desenvolvidas de acordo com as Tarefas a serem realizadas com o

suporte da ontologia, ou, no caso de uma ontologia de domínio, as características do mundo real que a ontologia tem como objetivo descrever.

Essa atividade deve ser realizada após a atividade de Formalização da Ontologia, para que seja possível identificar omissões e erros no artefato gerado e deve ser responsável pelo refinamento da ontologia. O foco da avaliação da ontologia projetada no KUP está na verificação da completude da ontologia. Isto é, verifica-se se todas as Questões de Competência propostas até esta Iteração podem ser representadas e respondidas com base nos termos e axiomas propostos para a ontologia. Caso isto não seja verdade, a ontologia deve ser refinada até que todas as omissões ou erros sejam solucionados.

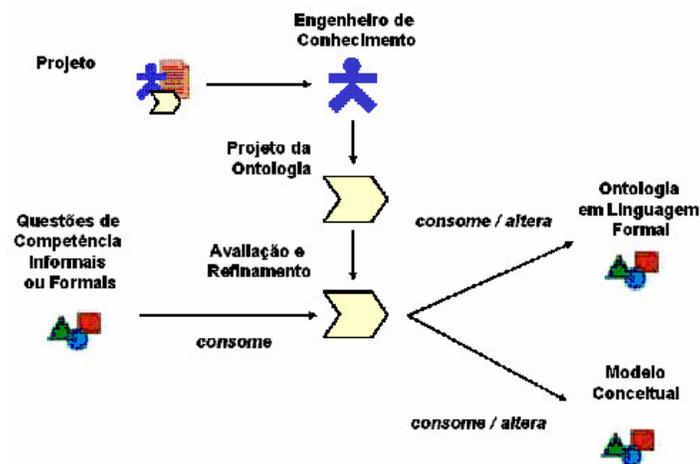


Figura 25 - Projeto da Ontologia - Avaliação e Refinamento

5.3.5.5. Implementação

Se o objetivo da Disciplina de Projeto é apresentar uma representação coerente e consistente da ontologia que atenda aos requisitos especificados, a Disciplina de Implementação transforma esta representação (formal ou conceitual, de acordo com as atividades executadas na instância do processo) em uma linguagem de implementação.

Seguindo a abordagem do KUP, a Disciplina de Implementação mapeia a etapa de Implementação dos sub-processos de Desenvolvimento relacionados ao Processo de Desenvolvimento da Ontologia propriamente dito. De acordo com o framework

de avaliação que está sendo utilizado como base para a elaboração do KUP, todas as metodologias, processos e métodos avaliados propõem ou descrevem atividades relacionadas a essa etapa. Isto já era esperado, pois o desenvolvimento da ontologia em uma linguagem de implementação é uma atividade que não pode ser omitida, uma vez que esta é crucial para o atendimento dos requisitos das aplicações desenvolvidas no contexto da Web Semântica.

A escolha da linguagem de representação de conhecimento deve ser, então, a primeira atividade a ser executada na Disciplina de Implementação.

Existem diversas linguagens de especificação de ontologias, que podem ser divididas em três grupos distintos (Gómez-Pérez et al., 2001):

- Linguagens de Ontologias Tradicionais
 - Ontolingua/KIF
 - LOOM
 - OKBC
 - OCML
 - F-Logic
- Padrões e Recomendações da W3C
 - XML
 - RDF(S)
- Linguagens de Especificação de Ontologias para a Web
 - SHOE
 - XOL
 - OML/CKML
 - OIL
 - DAML+OIL
 - OWL

Essas linguagens ainda podem ser divididas de acordo com o tipo de formalismo que adotam (Gómez-Pérez et al., 2001):

- Frames
 - Ontolingua/KIF
 - OKBC
 - OCML
 - F-Logic
 - RDF(S)
 - SHOE
 - XOL

- OIL
- DAML+OIL
- OWL

- Lógica de Primeira Ordem (First order Logic)
 - Ontolingua/KIF
 - OCML
 - F-Logic

- Description Logic
 - LOOM
 - OIL
 - DAML+OIL

- Conceptual Graphs
 - OML/CKML

- Semantic Nets
 - RDF(S)

Diversos critérios podem ser usados na escolha das linguagens. Em (Gómez-Pérez et al., 2001) as três categorias de linguagens apresentadas (tradicionais, padrões e web) e seus formalismos adotados são associados a quatro tipos diferentes de demandas de projeto:

Se o projetista identificar a necessidade de uma linguagem com grande expressividade, a categoria mais indicada é a de linguagens tradicionais de representação. Caso as aplicações necessitem de mecanismos de classificação automática, indica-se a utilização de linguagens que adotem o formalismo de Description Logic. Se a troca de informações através de ontologias for um requisito fundamental, deve ser utilizada alguma alternativa entre as linguagens de especificação para a web. Em arquiteturas baseadas em agentes de software, as necessidades podem estar mais presentes na troca de informações através de ontologias, caso no qual repete-se a indicação das linguagens de especificação para a web. Se, no entanto a demanda estiver concentrada na existência de mecanismos de *reasoning*, as linguagens tradicionais são novamente mais adequadas.

Outro estudo desenvolvido por Ribière e Charlton (2000) apresenta uma tabela comparativa (Tabela 4) entre diversas linguagens, apontando os elementos fundamentais de representação de conhecimento e o suporte dado por cada linguagem a esses elementos fundamentais:

	KIF	OKBC	XOL	RDF(S)	OIL	DAML + OIL	Requisitos
Elementos principais							
Conceitos	+	+	+	+	+	+	
Relações	+	+	+	+	+	+	
Funções	+	-	-	-	+	+	
Instâncias	+	+	+	+	-	+	
Axiomas	+	-	-	-	+	+	
Axiomas aceitáveis							
Negação	+	-	-	-	+	+	
Conjunção	+	-	-	-	+	+	
Disjunção	+	-	-	-	+	+	
Taxonomia dos conceitos							
Sub-classes	+	+	+	+	+	+	
Herança múltipla	+	+	+	+	-	+	
Meta-classes	+	+	-	+	-	+	
Slots							
Slots multi-valorados	+	+	+	+	-	+	
Hierarquia dos Slots (subslot-of)	+	+	-	+	-	+	
Slot inverso	+	+	+	-	+	+	
Facetas							
Valor de slot default	-	+	+	-	-	-	
Restrições de tipo	+	+	+	+	+	+	
Restrições de cardinalidade	+	+	+	-	+/-	+	
Outras restrições de slots	+	+	+	-	+	+	
Outras características							
Inclusão de meta informação	+	-	+	-	+	+	
Inclusão de outras ontologias	-	-	-	-	+	+	
Tipos de dados primitivos	+	+	+	-	-	-	

Tabela 4 - Tabela Comparativa entre Linguagens de Implementação de Ontologias – Elementos Fundamentais de Representação do Conhecimento

O projetista pode utilizar a coluna Requisitos para marcar quais as necessidades que ele identifica na linguagem de ontologia que pretende usar, de acordo com as demandas do projeto em questão. Pode separar as demandas em “Críticas”, “Necessárias”, “Úteis” e “Dispensáveis”, por exemplo. Isso pode ajudar a rejeitar linguagens que não tenham elementos fundamentais, permitindo também um

ranqueamento das linguagens disponíveis com base no suporte dado a esses elementos, com relação às necessidades do projeto.

Xiaomeng Su e Lars IJebreke (2002) realizaram também um estudo comparativo com base em um Framework para Avaliação de Qualidade de Ontologias e de Linguagens de Ontologias, descrito em (Krogstie et al., 1999), (Lindland et al., 1995). Este framework é baseado na idéia de que uma ontologia descreve um domínio de conhecimento e é representada através de uma linguagem para que possa ser então interpretada tanto por atores humanos quanto tecnológicos (agentes, aplicações e ferramentas). Com esse intuito, três aspectos de qualidade foram definidos:

- *Domain Appropriateness* – Esse aspecto expressa a capacidade da linguagem de capturar o conhecimento relacionado a um domínio. Os dois critérios que determinam essa capacidade são o poder de expressividade da linguagem e as perspectivas de modelagem suportadas.
 - Poder de Expressividade
 1. Suporta Axiomas?
 2. Permite meta-classes?
 3. Suporta instâncias?
 4. As propriedades são consideradas elementos de primeiro nível?
 5. É permitida hierarquia de propriedades?
 - Perspectivas suportadas
 1. Qual o suporte dado pela linguagem para as 7 diferentes perspectivas de modelagem?
 - (S)tructural – Permite descrever entidades e relacionamentos entre elas?
 - (F)unctional – Permite descrever processos, atividades e transformações?
 - (B)ehavioral – Permite descrever estados e transições?
 - (R)ule – Permite descrever regras para processos, atividades, ou outros elementos da linguagem?
 - (O)bject – Permite a descrição de objetos, seus métodos e atributos, processos e classes?
 - (C)ommunication – Permite descrever ações de linguagem, significados e entendimentos?
 - Actor-Role (AR) – Permite descrever atores, papéis, organizações e sociedades?
- *Comprehensibility Appropriateness* – Aborda a facilidade com a qual a linguagem pode ser entendida, através da utilização de mecanismos de abstração e a existência de elementos de construção consistentes e uniformes.
 - Qual a abrangência dos elementos de construção da linguagem?

- Quais os mecanismos de abstração disponíveis?
 1. Classification
 2. Generalization
 3. Aggregation
 4. Association

- *Technical Actor Interpretation Appropriateness* – Qual o suporte dado pela linguagem a agentes de software e aplicações, por exemplo?
 - A linguagem apresenta sintaxe formal?
 - A linguagem apresenta semântica formal?
 - A linguagem apresenta mecanismos de inferência? Como estes podem ser classificados?
 - A linguagem apresenta mecanismos de controle de restrições? Como estes podem ser classificados?

Com base nesses critérios, os autores avaliaram as seguintes linguagens:

- CycL
- Ontolingua (KIF)
- F-Logic
- OCML
- LOOM
- Telos
- RDF(S)
- OIL
- DAML+OIL
- XOL
- SHOE

O resultado da avaliação pode ser resumido através da Tabela 5.

		Cycl.	Omologua	F-Logic	OCML	LOOM	Telos	RDF(S)	OIL	DAML+OIL	XOL	SHOE
Domain appropriateness	Expressive Power	High	High	Medium	Medium+	Medium	High	Medium-	Medium-	Medium+	Medium	Medium
	Perspectives	S,O-R	S,O+,R	S,O+,R	S-,O,R, F,	S,O+,R,	S,O,R+ F,AR-	S,O,R-	S,O,R-	S,O,R-	S,O,R-	S,O,R
Comprehensibility appropriateness	Number of Constructs	Large	Large	Medium	Medium+	Medium	Medium+	Small	Small	Medium-	Medium	Small
	Abstraction Mechanism	Cla Gen+ Agg Ass	Cla Gen+ Agg Ass	Cla Gen+ Agg Ass	Cla Gen+ Agg- Ass	Cla Gen+ Agg- Ass	Cla Gen+ Agg- Ass	Cla Gen- Agg- Ass	Cla Gen- Agg- Ass	Cla Gen- Agg- Ass	Cla Gen- Agg- Ass	Cla Gen- Agg- Ass
Technical actor interpretation appropriateness	Formal Syntax	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	Formal Semantics	Yes	Yes	Yes	Yes	Yes	Yes	Yes-	Yes	Yes-	No	Yes-
	Inference Engine	Weak	No	Good+	Good	Good+	Good	No	Good+	Possible	No	Good
	Constraint Checking	Good	Good	Good	Good	Good	Good	Weak	Weak	Weak	Weak-	Weak

Tabela 5 - Tabela Comparativa entre Linguagens de Implementação de Ontologias - Framework de Avaliação de Qualidade

Com base nesses três estudos e orientações apresentadas cabe ao projetista identificar, primeiramente, quais as necessidades do projeto em questão – através da análise de requisitos – associando-as às categorias de linguagens e seus formalismos. Em seguida, é preciso identificar as demandas existentes para a linguagem de representação de ontologia. Essa identificação – associada com a classificação das necessidades na forma descrita anteriormente – permite ao projetista ranquear as linguagens de acordo com sua adequação ao projeto. O projetista pode incrementar essa análise através da avaliação de qualidade das linguagens ranqueadas, para, enfim, decidir qual a linguagem que será efetivamente usada.

Escolhida essa linguagem, o projetista deve escolher o ambiente de engenharia de ontologia mais adequado. Em (Gómez-Pérez et al., 2001) um estudo comparativo sobre as alternativas de ambientes é apresentado baseado em diversas dimensões. São elas:

- Características Gerais : desenvolvedores, *releases*, usos atuais, etc.
- Arquitetura de software e mecanismos de evolução da ferramenta
- Interoperabilidade com outras ferramentas, sistemas, mecanismos de tradução, etc.
- Modelo de conhecimento utilizado
- Bibliotecas integradas
- Serviços pra execução de inferências

- Suporte metodológico
- Cooperações
- Aspectos de usabilidade: Sistemas de *help*, edição e visualização

Em (Fernandez-Lopez, 2002) é apresentado um extenso estudo sobre diversos tipos de ferramentas para a Web Semântica, baseado em frameworks de avaliação específicos para cada um dos tipos escolhidos. Este estudo é discutido na seção de Implantação do KUP, a seguir, e pode auxiliar na atividade de escolha do ambiente necessário.

A Codificação da ontologia na linguagem e no ambiente escolhidos é a última atividade da disciplina de Implementação, devendo gerar ao seu final um artefato que represente a ontologia nesta linguagem.

Em resumo, a disciplina de Implementação compreende as seguintes atividades:

1. Identificação das demandas para a linguagem de representação da ontologia:
 - a. Quanto às necessidades da aplicação
 - Quais as categorias e formalismo mais adequados?
 - b. Quanto ao suporte dado aos elementos fundamentais de acordo com a classificação dos requisitos
 - Ranqueamento das linguagens de representação de ontologia de acordo com os requisitos identificados
 - c. Quanto a qualidade da linguagem em geral
 - Como os critérios de qualidade são atendidos?
2. Escolha da linguagem de representação
3. Escolha do ambiente de engenharia de ontologias
4. Codificação da ontologia na linguagem e ambiente escolhidos

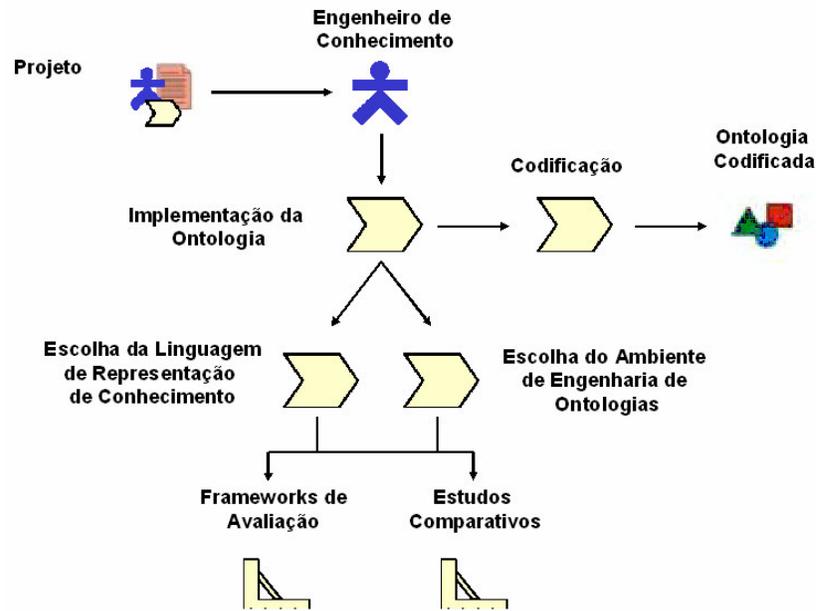


Figura 26 - Implementação da Ontologia

5.3.5.6. Implantação

Após a disciplina de Implementação, chega a hora de aplicar a ontologia desenvolvida. A disciplina de Implantação compreende os esforços necessários para integrar esta ontologia às diversas aplicações identificadas nas atividades anteriores do processo.

Esta Disciplina mapeia a etapa de Instalação dos sub-processos de Pós-Desenvolvimento, de acordo com o framework de avaliação. Apenas a metodologia On-To-Knowledge propõe a existência de atividades de Instalação da ontologia, através de sua utilização em aplicações denominadas *ontology-based applications* (Sure et al., 2002c). No entanto, esta metodologia não descreve como essa instalação – ou implantação - deve ser feita.

A primeira atividade desta disciplina corresponde à Identificação das aplicações a serem desenvolvidas com base nas ontologias.

Durante a Disciplina de Análise de Requisitos foram identificados espaços de uso e potenciais aplicações para a ontologia sendo desenvolvida. A cada iteração, novas aplicações podem ser identificadas e adicionadas à lista de potenciais aplicações. Essas aplicações deverão ser então desenvolvidas seguindo um processo bem estabelecido de projeto e desenvolvimento de software, mas deverão estar alinhadas a existência de elementos comuns de manipulação da ontologia e das bases

de conhecimento. Esses elementos são identificados na atividade de elaboração do Diagrama de Deployment das aplicações, nas quais a arquitetura de software, de hardware e de canais de comunicação é identificada.

A próxima atividade compreende, então a elaboração do Diagrama de Deployment das aplicações relacionando-as com as ontologias e ferramentas de suporte (storage, consultas, anotação) necessárias.

O Diagrama de Deployment da UML permite a representação – estática – dos elementos de hardware e dos componentes de software necessários em cada um desses elementos para que a aplicação entre em produção (Ambler, 2003). Esse tipo de diagrama descreve o hardware envolvido, o software instalado e o middleware necessário para conectar os diferentes elementos dessa arquitetura. Os objetivos de se construir um Diagrama de Deployment são:

- Explorar questões envolvidas na instalação do sistema para produção
- Explorar as dependências entre diversos sistemas e aplicações planejadas ou em produção
- Ter uma visão mais abrangente dos conjuntos de sistemas e aplicações em questão
- Permitir o projeto e a configuração de hardware e software do sistema.

No contexto do KUP, o principal objetivo é identificar componentes de software que sejam compartilhados por várias aplicações, permitindo a implantação de serviços que gerenciem o acesso e a manipulação da ontologia e das bases de conhecimento relacionadas.

Exemplos de diagramas de deployment para as aplicações que utilizam a ontologia E-BTS e Skill-O podem ser consultados nos Anexos 2 e 3.

A atividade seguinte compreende a seleção das ferramentas para implantação dos serviços de manipulação e consulta das ontologias e bases de conhecimento, descrita a seguir:

Após a identificação dos possíveis pontos de interseção entre os serviços necessários para manipulação e consulta das ontologias e bases de conhecimento é necessário selecionar as ferramentas que serão utilizadas na sua implantação.

Em (DAML) temos uma lista de diversas categorias de ferramentas para ontologias, base de conhecimento e a web semântica – citadas anteriormente neste trabalho. Em (Gómez-Pérez, 2002) são descritos frameworks para a avaliação de diferentes categorias de ferramentas, como construção, fusão e integração, avaliação de ontologias, anotação de documentos, armazenamento e consulta a ontologias e

bases de conhecimento. Além dos critérios gerais de avaliação, como os responsáveis pelo desenvolvimento da ferramenta, a versão atualmente disponível e a sua disponibilidade (código aberto, freeware, licenciamento, acesso remoto pela web, versão de avaliação, entre outras), existem critérios específicos para cada uma dessas categorias, descritos a seguir:

Para ferramentas de construção de ontologias são analisados critérios como sua arquitetura, interoperabilidade, mecanismos de representação de conhecimento e suporte a metodologias, mecanismos de inferência e usabilidade. Com relação a arquitetura, esse critério se sub-divide em arquitetura do software - que pode ser uma aplicação isolada (*standalone*), cliente/servidor ou em três camadas (*3-tier*) - extensibilidade, forma de armazenamento das ontologias - em arquivos binários, textuais, XML ou bases de dados - e o suporte a gerência de *backup*.

Além dos critérios gerais, as ferramentas de fusão e integração de ontologias são avaliadas de acordo com sua arquitetura, com as fontes e tipos de informações utilizadas durante os processos de fusão ou integração, sua interoperabilidade, seu modo de operação, com o suporte a gerência de diferentes versões de uma ontologia, os componentes suportados na fusão, sugestões apresentadas e conflitos detectados, suporte de técnicas e metodologias, sistema de *help*, formas de edição e visualização e maturidade.

As ferramentas de avaliação são analisadas de acordo com os tipos de propriedades das ontologias que elas permitem avaliar e o suporte dado a avaliação de aspectos tecnológicos como performance, alocação de memória e escalabilidade, entre outros.

Os critérios para avaliação de ferramentas de anotação englobam os tipos de fontes utilizados para as ontologias – arquivos locais, URLs, servidores de ontologias – as linguagens suportadas, o destino da ontologia populada, os elementos da linguagem utilizados, sua interoperabilidade, os tipos de fontes utilizados para os documentos a serem anotados – novamente, arquivos locais, URLs, entre outros – a linguagem de marcação, o destino do documento anotado, o grau de automatização do processo, o suporte a colaboração e a usabilidade da ferramenta.

O mesmo estudo apresenta ainda uma comparação entre linguagens de consulta (*query languages*) e ferramentas para armazenamento e consulta. Estas ferramentas são analisadas sob a ótica dos seguintes critérios: linguagem de consulta suportada, linguagem de implementação da ferramenta, o SGBD utilizado para

armazenamento, o suporte a inferências, o suporte a atualizações na ontologia e na base de conhecimento, a existência de APIs, escalabilidade, performance e linguagens nas quais exporta os dados armazenados.

As ferramentas analisadas nessas diferentes avaliações foram:

- Ferramentas para construção de ontologias
 - Apollo
 - LinkFactory©
 - OIEd
 - Ontoedit (Versão gratuitas e profissionais)
 - Ontolingua Server
 - OntoSauros
 - OpenKnoME
 - Protégé-2000
 - SymOntoX
 - WebODE
 - WebOnto

- Ferramentas para fusão e integração de ontologias
 - Chimaera
 - FCA-Merge (Método)
 - PROMPT
 - ODEMerge

- Ferramentas para avaliação de ontologias
 - OntoAnalyser
 - OntoGenerator
 - OntoClean/WebODE
 - ONE-T

- Ferramentas para anotação de documentos
 - AeroDAML
 - COHSE
 - MnM
 - OntoAnnotate
 - OntoMat-Annotizer
 - SHOE Knowledge Annotator

- Ferramentas para armazenamento e consultas
 - ICS-FORTH RDFSuite
 - Sesame
 - Inkling
 - rdfDB
 - RDFStore

- EOR – Extensible Open RDF
 - Redland
 - Jena
 - RDF Gateway
 - TRIPLE
 - KAON Tools Suíte
 - Cerebra®
 - Empolis K42
 - Ontopia Knowledge Suite
-
- Linguagens de consulta
 - ICS-FORTH RQL
 - ILRT SquishQL
 - Intellidimension RDFQL
 - RDFPath
 - VERSA RDF Query Language
 - TRIPLE
 - DAML+OIL Query Language
 - Topic Maps Query Language
 - Ontopia Tolog

Diversas conclusões foram tiradas pelos autores desse estudo, as quais são resumidas a seguir:

No que diz respeito a ferramentas de construção de ontologias, apesar de existirem uma série de alternativas similares, estas apresentam um baixo grau de interoperabilidade entre si. Além disso, abrangem apenas as disciplinas de projeto e implementação da ontologia, deixando de lado outras etapas de seu ciclo de vida. As ferramentas de fusão e integração de ontologias, segundo as conclusões do estudo, ainda precisam de muita evolução. Tais ferramentas não suportam a fusão de regras e axiomas, e exigem um alto grau de intervenção humana. No que tange as ferramentas que permitem a avaliação de ontologias sugere-se a definição de *benchmarks* e processos de padronização, para que estas avaliações possam ser melhor efetuadas. As ferramentas de anotação de documentos são orientadas principalmente para a criação de conteúdo para a web semântica, e dão suporte a diversas linguagens de ontologias para a web. Apesar de resolverem bem o problema de anotação de páginas estáticas, a anotação de páginas dinâmicas ainda deve ser melhor explorada. A anotação automática é também um problema ainda em aberto. Finalmente, foi detectado que no que tange as linguagens de consulta a ontologias e bases de

conhecimento ainda não é dado um suporte adequado ao conjunto completo das construções de modelagem apresentados pelos padrões da Web Semântica.

Como uma conclusão geral, as ferramentas são apresentadas como módulos isolados que resolvem apenas um conjunto limitado de problemas, não estando bem integradas entre si. Os autores propõem a construção de ambientes para desenvolvedores de ontologias que dêem suporte ao seu ciclo de vida completo e que facilitem seu uso. Propõem, também, a integração de serviços de *middleware*, que permitam, por exemplo, a busca de ontologias apropriadas aos requisitos de determinada aplicação, que permitam a comparação semântica entre ontologias e acesso remoto a bibliotecas de ontologia, entre outras funcionalidades.

Identificadas as relações entre as diversas aplicações a serem desenvolvidas e o compartilhamento dos serviços que devem ser oferecidos através das ferramentas selecionadas, devem ser iniciados os desenvolvimentos dessas aplicações. Estas podem ser desenvolvidas de acordo com diversos processos existentes na literatura e praticados na indústria – como desenvolvimento em cascata, RUP ou *extreme programming* (Beck, 1999) - para serem então integrados ao ambiente de produção das aplicações de web semântica.

A última atividade desta disciplina refere-se à implantação das aplicações propriamente dita. As diversas ferramentas selecionadas devem ser então implantadas seguindo-se o Diagrama de Deployment desenvolvido e integradas com as aplicações de Web Semântica ou ontologias geradas.

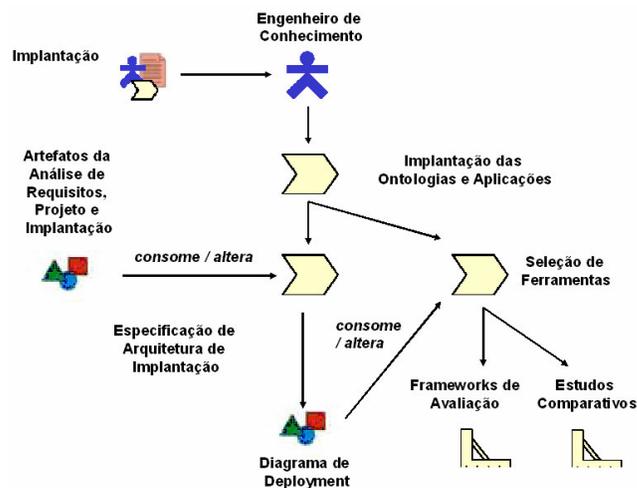


Figura 27 - Implantação – Especificação de Arquitetura e Seleção de Ferramentas

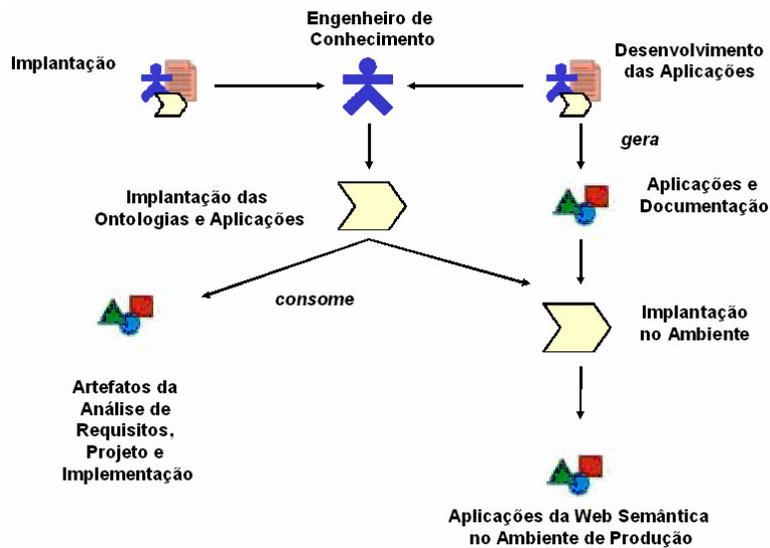


Figura 28 – Implantação - Desenvolvimento das Aplicações e Implantação no Ambiente de Produção

PUC-Rio - Certificação Digital Nº 0115632/CA

5.3.5.7. Processos de Suporte

No que diz respeito aos processos de pós-desenvolvimento, de acordo com o framework de avaliação, o KUP não descreve, nesta versão, como devem ser executadas as atividades das disciplinas de operação, suporte, manutenção e arquivamento da ontologia e suas aplicações. Apenas propõe sua inclusão na Disciplina de Processos de Suporte. A metodologia On-to-Knowledge propõe atividades de instalação e manutenção, estas também propostas pela Methontology, e descreve atividades de operação e suporte.

5.3.5.8. Processos Complementares

Os chamados Processos Complementares (*Integral Processes*) também são propostos, mas não são descritos nesta versão do KUP. São eles:

- Aquisição de Conhecimento
- Verificação e Validação
- Gerência de Configuração da Ontology
- Documentação
- Treinamento