

# Gerência de Conhecimento em ODE

*Ana Candida Cruz Natali  
Ricardo de Almeida Falbo*

Departamento de Informática, Universidade Federal do Espírito Santo – UFES  
Av. Fernando Ferrari s/n, Centro Tecnológico  
CEP 29.060-900, Vitória – ES, Brasil, +55 27 3335-2167

## Resumo

*No contexto do desenvolvimento de software, a gerência de conhecimento pode ser utilizada para capturar o conhecimento gerado durante o processo de software. Embora cada projeto de software seja único, experiências similares podem auxiliar desenvolvedores a executarem suas atividades. Neste artigo, discute-se a importância da gerência de conhecimento no desenvolvimento de software e é apresentada uma infra-estrutura de gerência de conhecimento integrada ao ambiente de desenvolvimento de software ODE. Esta infra-estrutura considera os seguintes serviços de gerência de conhecimento: captura, armazenamento, recuperação, disseminação, uso e manutenção.*

**Palavras-chave:** gerência de conhecimento, ambientes de desenvolvimento de software, ontologias.

## Abstract

*In the context of software development, knowledge management can be used to capture the knowledge and experience generated during the software process. Although every software development project is unique in some sense, similar experiences can help developers to perform their activities. In this paper we discuss the importance of knowledge management in software development and we present an infrastructure to enable knowledge management in ODE, an Ontology-based software Development Environments, which considers knowledge capture, store, retrieval, dissemination, use and maintenance.*

**Key-words:** knowledge management, software engineering environments, ontologies.

## 1. Introdução

Recentemente, a gerência do conhecimento – isto é, a forma como as organizações geram, disseminam e usam seu capital intelectual – vem sendo reconhecida como uma fonte de vantagem competitiva para organizações de desenvolvimento de software. A gerência de conhecimento apresenta-se como uma forma de apoiar os desenvolvedores, de modo que esses realizem suas atividades melhor e mais rapidamente, aproveitando experiências anteriores para auxiliar a tomada de decisão.

Conceitualmente, a gerência de conhecimento consiste em coletar e armazenar sistematicamente o conhecimento adquirido, compartilhar este conhecimento através de uma memória organizacional e promover o surgimento de novos conhecimentos. Para atingir estes objetivos, a gerência de conhecimento envolve recursos humanos, organização e cultura, além de tecnologia de informação, métodos e ferramentas para o seu apoio [1].

Porém, a adoção de gerência de conhecimento envolve medidas organizacionais, medidas relacionadas a recursos humanos e principalmente medidas relacionadas ao processo de trabalho. A integração com o processo da organização é um dos fatores mais determinantes para o sucesso da implantação de gerência de conhecimento em uma organização [2].

No caso de organizações de software, há muita dificuldade envolvida na tarefa de estabelecer um processo de software e fornecer meios para seu acompanhamento. Esses

problemas, porém, são minimizados com o auxílio de Ambientes de Desenvolvimento de Software (ADSs), sistemas computacionais que provêm suporte para o desenvolvimento e a manutenção de produtos de software e para o gerenciamento destas atividades [3].

Uma vez que, para ser efetivamente utilizado, um sistema de gerência de conhecimento deve estar integrado ao ambiente de trabalho existente, ADSs e gerência de conhecimento complementam um ao outro no apoio ao desenvolvedor durante o processo de software para a criação de produtos de melhor qualidade e com maior produtividade. Vários benefícios podem ser obtidos com a Gerência de Conhecimento e com o uso de Ambientes de Desenvolvimento de Software. Porém, benefícios maiores podem ser esperados quando essas duas tecnologias são usadas em conjunto, ou seja, quando o compartilhamento de conhecimento e experiência é utilizado como mecanismo de apoio à realização de atividades do processo de software apoiadas por um ADS. Através da disponibilização do conhecimento acumulado em vários projetos, espera-se auxiliar as atividades do processo de software.

Com base nessas necessidades, este trabalho apresenta a infra-estrutura de gerência de conhecimento desenvolvida para o ambiente de desenvolvimento de software ODE [4]. A seção 2 discute aspectos relacionados à gerência de conhecimento em engenharia de software. A seção 3 faz uma breve apresentação do ambiente ODE. Na seção 4, é apresentada a infra-estrutura de gerência de conhecimento proposta. A seção 5 aponta pontos fortes e fracos da proposta apresentada. Na seção 6, discutem-se trabalhos correlatos. Finalmente, a seção 7 apresenta as conclusões do trabalho.

## **2. Gerência de Conhecimento em Engenharia de Software**

A gerência de conhecimento pode ser aplicada a qualquer tipo de organização que necessita gerenciar seu conhecimento. A Fábrica de Experiências [5] é um exemplo de uma abordagem de gerência de conhecimento especialmente projetada para organizações de software. A fábrica de experiências reconhece que as organizações precisam aprender através de suas experiências anteriores, de forma a desenvolver produtos com maior rapidez, menor custo e com maior qualidade. Ou seja, o objetivo proposto tanto pela Fábrica de Experiências quanto pela gerência de conhecimento é definir uma maneira estruturada de gerenciar o conhecimento e tratar as pessoas que o possuem como recursos importantes, de forma a ajudar organizações a aumentarem sua produtividade e competitividade.

Mas, se a gerência de conhecimento é a solução para estes problemas, por que todas as organizações não adotam esta medida? A razão é que há alguns desafios envolvidos na implantação da gerência de conhecimento, especialmente na engenharia de software.

A engenharia de software é uma disciplina complexa, que envolve um grande número de pessoas trabalhando em diferentes fases e atividades. Constantes mudanças de tecnologia tornam o trabalho dinâmico: novos problemas são solucionados e novo conhecimento é criado todos os dias. O conhecimento na engenharia de software é diverso e cada vez mais crescente. Organizações de software têm dificuldade em controlar qual conhecimento é esse, onde ele se encontra e quem o possui.

### **2.1 Desafios da Gerência de Conhecimento na Engenharia de Software**

Implementar gerência de conhecimento em qualquer organização é um desafio por causa do tempo e do esforço necessários antes que se comece a obter retorno do investimento. Organizações de software parecem ter ainda menos tempo que outras, por causa da urgência da área de desenvolvimento de software. A falta de tempo é uma ameaça direta à gerência de conhecimento. Um exemplo é que um especialista de software pode estar tão envolvido em seu projeto, que não há tempo para apoiar outro projeto [6]. As pessoas freqüentemente não possuem tempo nem para procurar pelo conhecimento. Este comportamento cultural faz com

que investimentos a longo prazo, como a gerência de conhecimento e o aprendizado para um projeto futuro, não sejam priorizados. Enquanto gerentes não incentivarem a mudança dessa cultura e permitirem que seus funcionários gerenciem seus conhecimentos, a gerência de conhecimento pode não ser efetiva.

Um outro desafio é a característica do produto de software: diferente de produtos de outros domínios, produtos de software não são visíveis. Esta invisibilidade leva a um menor reuso. Um desenvolvedor, ao implementar ou modificar um produto de software, não consegue visualizar se este trabalho já foi feito anteriormente. Muitas vezes, desenvolvedores reinventam soluções ao invés de reutilizá-las e isto resulta em menor produtividade. Ou ainda, alguns desenvolvedores não estão acostumados ao reuso, o que é um problema, pois a idéia central da gerência de conhecimento é o reuso de recursos.

O mais problemático desafio, portanto, é que a maior parte do conhecimento na engenharia de software é conhecimento tácito. E pode permanecer tácito pela ausência de tempo de torná-lo explícito ou mesmo pela dificuldade de explicitá-lo. Uma maneira de tentar contornar este problema pode ser o desenvolvimento de uma cultura de compartilhamento eficiente, assim como apoio tecnológico para a gerência de conhecimento, sem esquecer que os recursos mais importantes de uma organização são seus funcionários.

## **2.2 Necessidades de Conhecimento em Organizações de Software**

Cada produto e processo de software é diferente em termos de objetivos e contexto. Uma única abordagem de desenvolvimento de software não pode ser assumida para todos os projetos ou produtos. Essas características do desenvolvimento de software são apenas algumas dificuldades enfrentadas em organizações de software e justificam a necessidade de se gerenciar o conhecimento obtido.

Assim é possível detectar algumas das necessidades das organizações de software relacionadas à gerência de conhecimento:

- **Necessidade de capturar e compartilhar conhecimento do processo de software:** Desenvolvedores de software são expostos a uma grande diversidade de características, objetivos e contextos e constantemente obtêm experiência em cada novo projeto. O conhecimento emerge na prática, frequentemente sendo definido pelo primeiro projeto que trata as novas questões envolvidas. Idealmente, deveria-se aplicar esta experiência em projetos futuros de maneira a evitar erros e aumentar o potencial de sucesso. Isto não acontece sempre, porque frequentemente essas práticas de trabalho não são capturadas [7]. Equipes de desenvolvimento trabalham em projetos similares sem saberem que resultados seriam alcançados mais facilmente se seguissem uma prática adotada por um projeto anterior [5]. Ou seja, equipes de desenvolvimento não se beneficiam da experiência existente. Pelo contrário, elas cometem erros repetidamente [6], pois muitos membros da equipe demonstram falta de conhecimento em projeto específico, conhecimento este que se encontra na organização. Esta situação também se repete quando é necessário transferir conhecimento a um novo membro na organização. A transferência de conhecimento de especialistas para novatos é facilitada se o conhecimento já foi capturado, armazenado e organizado, tornando-se disponível ao novo membro da organização de software.
- **Necessidade do conhecimento do domínio:** O desenvolvimento de software não requer apenas conhecimento sobre seu próprio domínio, mas também sobre o domínio para o qual o software está sendo desenvolvido. Desenvolver software para domínios complexos, como portuário ou controle aéreo, pode causar uma dificuldade extra: o entendimento sobre o problema em questão. Ou ainda, pode ser necessário o aprendizado de uma técnica específica ou de uma nova linguagem de programação. Assim, adquirir experiência e habilidades necessárias a um projeto pode tomar bastante tempo [6]. Não há atalho para o

aprendizado. O conhecimento sobre um domínio que nenhum membro da organização possui deve ser adquirido através de treinamento ou contratando recursos humanos que possuam esse conhecimento. A gerência de conhecimento, entretanto, pode ajudar a organizar a aquisição desse novo conhecimento e auxiliar a identificação de especialistas, assim como a captura, o armazenamento e o compartilhamento do conhecimento já existente na organização.

- **Necessidade de adquirir conhecimento sobre novas tecnologias:** O desenvolvimento de software torna-se cada vez mais complexo em razão das constantes mudanças e do surgimento de novas tecnologias. O resultado é que se torna difícil manter uma organização de software na posição de liderança competitiva [8]. O surgimento de novas tecnologias torna os produtos de software mais poderosos, mas, para isso, pode ser necessária a mudança de uma tecnologia até no meio do desenvolvimento de um software. Quando desenvolvedores utilizam uma tecnologia que é nova para todos os membros da equipe, o aprendizado geralmente ocorre através de erros e acertos [6]. Isto freqüentemente gera sérios atrasos no projeto em questão. A gerência de conhecimento incentiva uma cultura de compartilhamento de conhecimento dentro da organização, que pode facilitar o aprendizado de novas tecnologias. A gerência de conhecimento também defende que o tempo deve ser gasto na busca do conhecimento já construído pela organização e não criando-o novamente.
- **Necessidade de conhecer quem sabe o quê:** Muito conhecimento pode ser registrado, mas os recursos mais importantes de uma organização são seus empregados e o conhecimento tácito que eles possuem. A gerência do conhecimento deve envolver quem detém qual conhecimento [8]. Muito tempo e esforço são gastos para determinar quem na organização pode auxiliar a desempenhar uma determinada tarefa. Assim, além de capturar o conhecimento, é igualmente importante determinar quem o possui. Uma outra razão para se ter controle de quem sabe o quê, é que uma organização de software, especialmente, é bastante dependente do conhecimento tácito, que é bem móvel. Se um funcionário que possui um conhecimento crítico abandona a organização, falta de conhecimento é gerada [8]. O problema é que provavelmente ninguém na organização sabe qual conhecimento ele possui [5]. A gerência de conhecimento não poderá nunca substituir o cérebro dos funcionários, mas pode ajudar a construir estruturas para capturar informações chave que ajudam a reter algum conhecimento quando o funcionário se desligar da organização. Esta informação pode, pelo menos, ajudar a entender o que o funcionário que saiu conhecia e qual o perfil que seu sucessor deverá ter para preencher a posição vaga.

### 3. O Ambiente ODE

Para ser efetiva, a gerência de conhecimento tem de estar integrado ao processo da organização, permitindo que o conhecimento relevante seja coletado e armazenado à medida que ele é gerado no trabalho. Conseqüentemente, o sistema de gerência de conhecimento deve estar integrado ao ambiente de trabalho existente.

No contexto de desenvolvimento de software, esse ambiente de trabalho é exatamente um Ambiente de Desenvolvimento de Software (ADS). Durante a utilização do ADS, surge o conhecimento envolvido nas atividades do processo de desenvolvimento de software.

ODE (*Ontology-based software Development Environment*) [4] é um Ambiente de Desenvolvimento de Software Centrado em Processo, desenvolvido no Laboratório de Engenharia de Software da Universidade Federal do Espírito Santo (LabES/UFES), tendo por base ontologias. Em seu estágio atual, há duas ontologias centrais para ODE: a ontologia de processo de software [9] e a ontologia de qualidade de software [10]. Estas ontologias são

usadas como a fundação do ambiente, a partir da qual são definidos e acompanhados os processos de software no ambiente.

Uma das preocupações centrais no projeto de ODE é a integração de conhecimento. A integração do conhecimento de domínio é importante para prover apoio de gerência de conhecimento de domínio e evoluir ODE para um Ambiente de Desenvolvimento de Software Orientado a Domínio (ADSOD). Sendo assim, foi identificada a necessidade de prover suporte ao desenvolvimento de ontologias de domínio em ODE, como forma de permitir a descrição do conhecimento de domínio no ambiente. Para atender a esta necessidade, foi desenvolvido ODEd, um editor de ontologias [11].

ODEd apóia a definição de conceitos e relações utilizando representações gráficas e promove a geração automática de alguns tipos de axiomas. A partir de uma ontologia, ODEd gera automaticamente *frameworks* de objetos e permite a instanciação das ontologias através da criação automática de bases de dados e interfaces customizadas [11].

Além do apoio à captura do conhecimento sobre o domínio, é necessário apoiar a captura do conhecimento gerado sobre desenvolvimento de software em ODE. Cada novo projeto de software desenvolvido no ambiente gera conhecimento relevante e útil a novos projetos.

Para atender a esse requisito, foi desenvolvida uma infra-estrutura de gerência de conhecimento incorporada a ODE de forma a torná-lo um ambiente capaz de prover aos desenvolvedores de software conhecimento acumulado no contexto do desenvolvimento e da manutenção de software.

#### 4. A Infra-Estrutura para Gerência de Conhecimento de ODE

A infra-estrutura de gerência de conhecimento de ODE, mostrada na figura 1, posiciona a memória organizacional no centro, apoiando o compartilhamento e reuso de conhecimento, sendo cercada pelos seguintes serviços de gerência de conhecimento: criação e captura, recuperação e acesso, disseminação, uso e manutenção do conhecimento [12]. Estes serviços são agrupados em duas categorias: serviços gerais, que são incorporados ao ambiente e estão disponíveis tanto diretamente no ambiente como a partir de suas ferramentas, e os serviços específicos de ferramentas, aqueles que precisam levar em conta o funcionamento e as características de uma ferramenta específica. A seguir, cada um dos componentes da infra-estrutura proposta é apresentado.

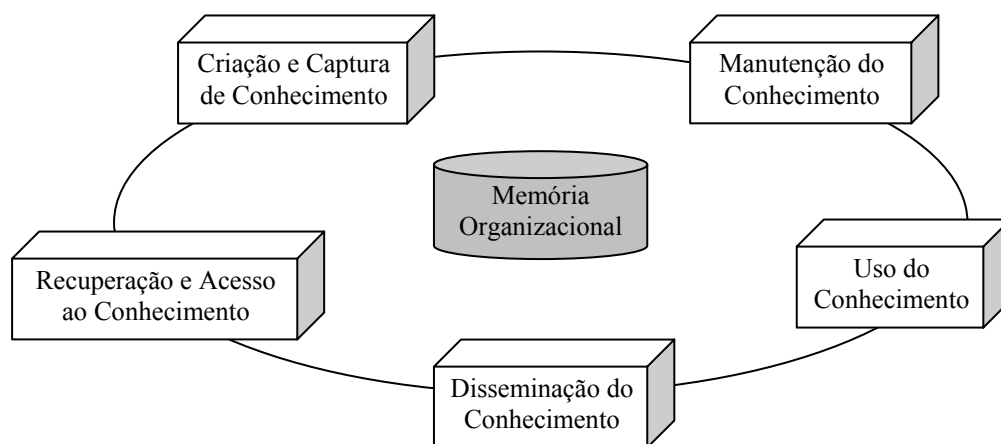


Figura 1 – A Infra-estrutura de Gerência de Conhecimento de ODE.

#### 4.1 Memória Organizacional

A memória organizacional é uma representação explícita e persistente do conhecimento e das informações cruciais para uma organização, cuja finalidade é facilitar o acesso, compartilhamento e reuso pelos diversos membros da organização [13].

A memória organizacional deve armazenar todos os tipos de conhecimento necessários ao desenvolvimento de software. Em ODE, foram definidos, inicialmente, três tipos de conhecimento a serem capturados e armazenados na memória organizacional: *instâncias de ontologia*, *artefatos gerados pelas ferramentas* e *lições aprendidas*.

Sendo assim, é importante fornecer meios de armazenar tanto o conhecimento formal, ou seja, os artefatos do processo de software e as instâncias de ontologia, assim como o conhecimento informal, isto é, as lições aprendidas. A figura 2 mostra a estrutura da memória organizacional de ODE.

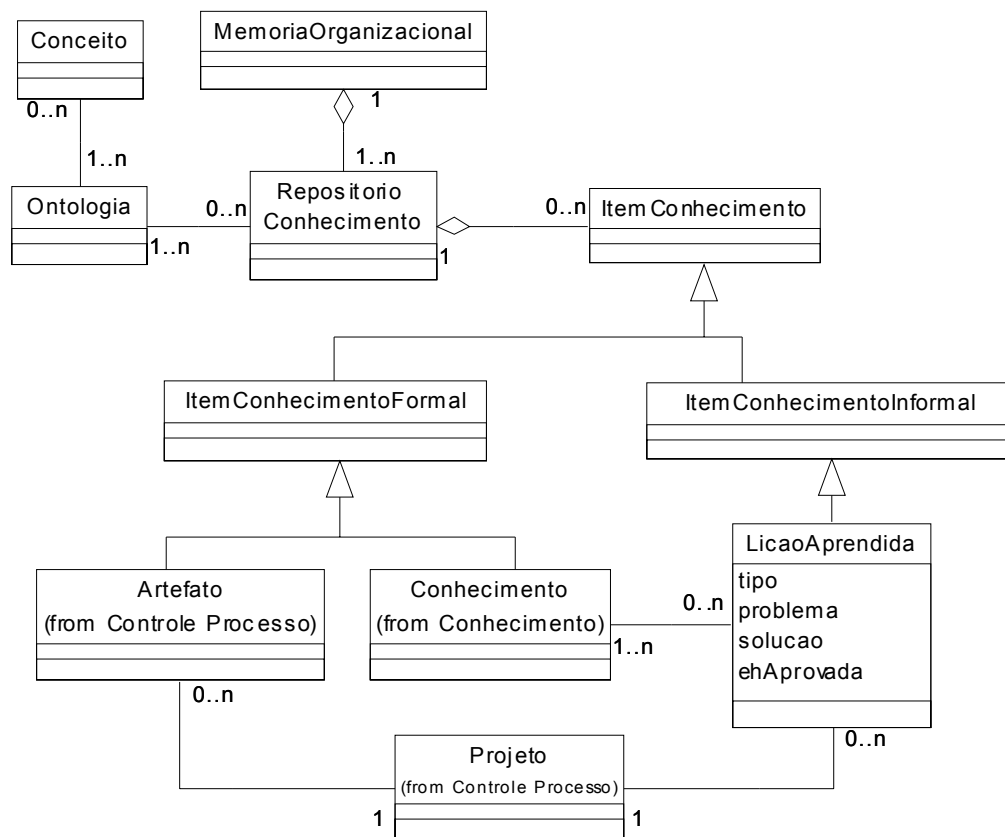


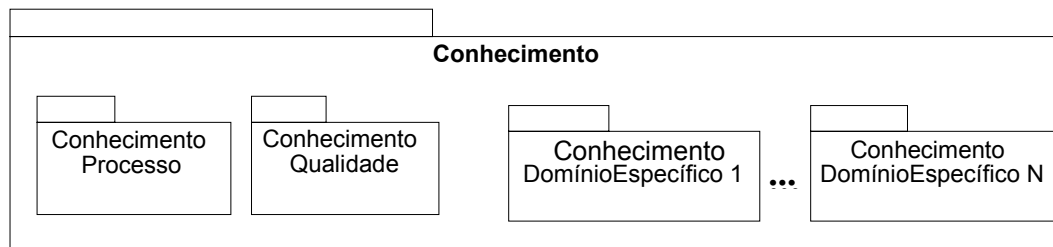
Figura 2 – A Estrutura da Memória Organizacional de ODE.

A memória organizacional é composta pelos vários *repositórios de conhecimento*. Cada um dos repositórios possui um conjunto de itens de conhecimento e está baseado em pelo menos uma *ontologia*, que é composta por *conceitos*.

Os *itens de conhecimento* podem ser de dois tipos: itens de conhecimento *formais* e *informais*. Os itens de conhecimento formais são os *artefatos* e as *instâncias de ontologia*, estas últimas tratadas nas subclasses da classe *Conhecimento*, oriunda do pacote *Conhecimento*, representando os conceitos das ontologias básicas de ODE e das ontologias de domínio construídas usando ODEd, como mostra a figura 3.

Como *itens de conhecimento informal*, no estágio atual, têm-se as *lições aprendidas*, que surgem no desenvolvimento de *projetos*. *Lições aprendidas* são relatos de sucesso ou oportunidades de melhoria detectados durante o desenvolvimento de um produto de software, utilizando o processo definido para o projeto. Uma lição sempre está associada a um *objeto de Conhecimento*, tal como um conhecimento sobre uma atividade, um artefato, um

procedimento, um método, um recurso, uma ferramenta, etc, que são instâncias de conceitos das ontologias utilizadas. Uma lição deve ser adaptada e aprovada antes de ser considerada uma lição aprendida para a organização. Nesta adaptação, feita pelo gerente do conhecimento, a lição pode ser alterada, aumentando, assim, o seu potencial de reuso.



**Figura 3 – O pacote Conhecimento.**

#### **4.2 Serviços Gerais da Gerência de Conhecimento de ODE**

Os serviços gerais estão diretamente disponíveis em ODE e podem ser utilizados a qualquer momento por seus usuários, estejam eles utilizando o próprio ambiente ou enquanto estiverem utilizando uma de suas ferramentas. Estes serviços são: coleta de conhecimento e aprovação de lição aprendida, busca por itens de conhecimento, caracterização de uso e manutenção do conhecimento armazenado. Porém, é importante ressaltar que, apesar de serem serviços acessíveis no ambiente, é preciso respeitar serviços que são exclusivos de usuários que possuem o papel de Gerente de Conhecimento. O Gerente de Conhecimento é responsável pela aprovação de lições aprendidas registradas por um desenvolvedor e é responsável pela manutenção dos itens armazenados na memória organizacional. Os serviços gerais da gerência de conhecimento disponíveis em ODE são apresentados a seguir.

#### **Coleta de Conhecimento e Aprovação de Lições Aprendidas**

Já que ODE lida com três tipos de conhecimento em sua memória organizacional, deve haver facilidades para apoiar a coleta de cada um destes itens de conhecimento. Os conhecimentos formais (artefatos e instâncias de ontologia), contudo, não são coletados diretamente pela gerência de conhecimento de ODE.

Os artefatos produzidos ao longo do processo de software são armazenados em um repositório central do ambiente, sendo submetidos à gerência de configuração. Desta forma, no contexto da gerência de conhecimento, a coleta dos artefatos não é necessária, podendo-se acessá-los no repositório de artefatos do ambiente, onde se encontram armazenados.

A criação de instâncias das ontologias básicas de ODE, ontologias de processo e de qualidade de software, está disponível como uma funcionalidade básica de ODE. É possível, ainda, instanciar ontologias de domínio desenvolvidas usando ODEd [11], conforme discutido anteriormente. Uma vez criadas, as instâncias das ontologias ficam disponíveis como itens de conhecimento.

Assim, apenas o conhecimento informal, isto é, as lições aprendidas, é coletado pelo serviço de gerência de conhecimento, como mostra a Figura 4. O esquema de caracterização de uma lição aprendida no repositório envolve os seguintes dados:

- Projeto: é o projeto no qual a lição foi gerada;
- Objetos selecionados: objetos de conhecimento aos quais a lição está associada. Estes objetos possuem um *tipo*, que é oriundo da lista de conceitos das ontologias que estruturam o repositório onde a lição será armazenada. E, ainda, a partir do tipo, é possível escolher o *objeto*, ou seja, uma instância deste conceito da ontologia. Por exemplo, é possível associar uma lição ao objeto *Documento de Especificação de Requisitos*, que é do tipo *Artefato*;

- Tipo de lição aprendida: identifica se a lição é considerada uma boa prática (ponto positivo), uma oportunidade de melhoria (ponto negativo) ou é uma lição informativa;
- Contexto: descrição da situação em que se deu a lição;
- Problema: descrição do problema ocorrido;
- Solução: descrição da solução adotada para o problema ou a descrição de uma solução recomendada para esta situação;
- Resultado esperado: descrição do resultado.

**Figura 4 – Coleta de Lição Aprendida.**

Uma vez que nem todas as lições aprendidas são úteis à organização, cabe ao gerente do conhecimento fazer uma triagem dos relatos a serem disseminados e adaptar esse conhecimento, categorizando o problema e a solução adotada. Assim, uma lição aprendida pode ser registrada por um desenvolvedor a qualquer momento. Mas, para se tornar uma lição de nível organizacional e estar, assim, acessível a todos os membros da organização, é preciso que esta seja aprovada pelo Gerente de Conhecimento.

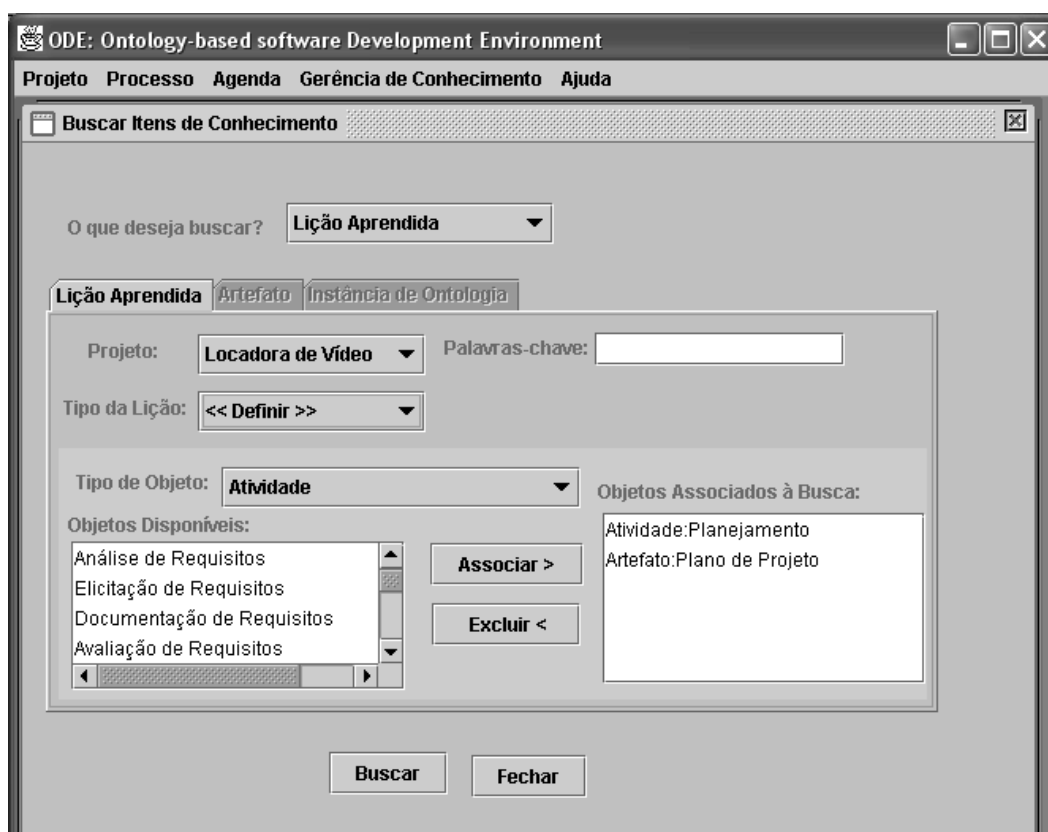
### **Recuperação e Acesso a Itens de Conhecimento Através da Busca**

O acesso ao conhecimento em ODE se dá através da busca pelos itens de conhecimento armazenados na memória organizacional. Em qualquer momento, um usuário de ODE pode buscar qualquer tipo de conhecimento: instâncias de ontologias, artefatos de software e lições aprendidas. Esta busca é de iniciativa do usuário, já que cabe a ele definir quais as suas



necessidades, ou seja, qual o tipo de conhecimento ele deseja buscar. Estas necessidades do usuário se tornam uma consulta à memória organizacional e os itens de conhecimento recuperados são apresentados.

Para recuperar lições aprendidas é possível buscar por um *projeto* ao qual pertença, pelos *objetos* aos quais está associada, pelo seu *tipo* ou por *palavras-chave* que estejam presentes em seus campos descritivos como contexto, problema, solução ou resultado esperado. A seleção dos objetos envolve definir o *tipo* de objeto, estabelecido a partir dos conceitos utilizados pela ontologia que fundamenta o repositório. A partir do tipo, os objetos disponíveis são exibidos sendo permitida a sua seleção, como mostra a Figura 5.



**Figura 5 – Buscar Itens de Conhecimento: Lições Aprendidas.**

Para a busca de artefatos, também é possível elaborar uma combinação de critérios. Pode-se estabelecer um *projeto* ao qual esteja relacionado, um tipo de *atividade* a qual esteja associado (seja como insumo ou produto) e ainda pelo *tipo* de artefato. O critério *tipo* para a busca de um artefato é definido no *Pacote Conhecimento*, pela classe *Conhecimento Artefato*. Através dela, é possível determinar uma lista de tipos de artefatos reconhecidos pelo ambiente. Ou seja, todo artefato produzido em ODE é classificado através dos tipos definidos no ambiente, ressaltando que esta classificação pode ser estendida a qualquer momento.

A busca por instâncias de ontologia permite a definição dos seguintes critérios: a ontologia a partir da qual a instância foi criada, a escolha de um dos conceitos desta ontologia e a instância que se deseja buscar.

### **Caracterização de Uso de Itens de Conhecimento**

Após a busca ser efetuada, os itens de conhecimento que satisfizeram os critérios definidos são apresentados. O usuário pode navegar pelo resultado da busca efetuada e abrir um item de conhecimento que desperte seu interesse.

Caso deseje, o usuário pode avaliar a utilidade do item de conhecimento para sua tarefa, caracterizando o uso desse item, como mostra a Figura 6. Devem ser informados:

- *Classificação*: caso o reuso tenha sido bem-sucedido ou não;
- *Problemas encontrados*: descrição dos problemas surgidos no uso deste item, ou ainda, descrição de discordâncias com item de conhecimento relatado;
- *Soluções adotadas*: descrição de soluções adotadas para os problemas ou situações derivadas do uso deste item;
- *Dicas para um próximo uso*: o que foi aprendido para que este item de conhecimento possa evoluir, auxiliando o próximo usuário a reutilizá-lo.

ODE: Ontology-based software Development Environment

Projeto Processo Agenda Gerência de Conhecimento Ajuda

Caracterizar Uso de Item de Conhecimento

Item de Conhecimento Reutilizado: Artefato

Descrição do Item Outras Caracterizações de Reuso deste Item

\* ARTEFATO: Plano de Riscos  
-Projeto: Locadora de Vídeo  
-Risco: Custos Ultrapassados  
-Probabilidade: Média  
-Plano de Mitigação: Usar ferramentas de estimativa de projeto de software;  
Usar ferramentas de planejamento de projetos;

Classificação:  Bom  Regular  Ruim

Problemas Encontrados:

Soluções Adotadas:

Foi adotado o plano de contingência descrito para um dos riscos deste plano.

Dicas para Próximo Reuso:

Um bom plano de riscos abrangendo um conjunto de riscos muito comuns em qualquer projeto.

Salvar Fechar

Figura 6 – Caracterização de Uso de um Item de Conhecimento.

### Manutenção dos Itens de Conhecimento

A memória organizacional precisa evoluir. É preciso atualizar o conhecimento antigo, eliminar o conhecimento defasado e abrir espaço para conhecimento novo, que esteja mais de acordo com a situação atual da organização. Estes são os objetivos do serviço de manutenção do conhecimento oferecido pela infra-estrutura de gerência de conhecimento de ODE.

A manutenção do conhecimento em ODE é realizada levando-se em conta o *feedback* do usuário. Baseado na caracterização de uso dos itens de conhecimento feita pelos usuários, o Gerente de Conhecimento pode selecionar os itens que devem ser excluídos. A partir da definição de seus critérios, uma busca é realizada e os itens recuperados são exibidos, podendo ser excluídos pelo Gerente de Conhecimento.

A manutenção da memória organizacional de ODE pode, ainda, ser apoiada por um agente de software. O agente de manutenção pode ser configurado para lembrar o gerente de conhecimento da necessidade de se realizar a manutenção em um repositório de conhecimento, no intervalo de tempo definido pelo gerente, ou para auxiliar a busca de itens de conhecimento, a partir de critérios especificados pelo gerente.

Para a exclusão de itens de conhecimento, o gerente de conhecimento pode escolher um repositório específico para realizar a manutenção ou operar sobre todos os repositórios, ou seja, a memória organizacional. Após definir o local da manutenção, é preciso estabelecer os critérios para exclusão. Esses critérios podem ser baseados nas caracterizações de reuso dos itens de conhecimento feitas pelo usuário. A partir das caracterizações de reuso de um item, é possível determinar, por exemplo, sua importância (dada pela classificação feita pelos usuários), frequência de uso ou data do último uso. Com base nos critérios definidos, os itens candidatos à exclusão são recuperados, apresentados e, caso selecionados, excluídos.

### **4.3 Os Serviços Gerais em ODE**

ODE é um ambiente composto por várias ferramentas que trabalham conjuntamente para apoiar o desenvolvimento de software. Dentre estas ferramentas, pode-se citar ferramenta para planejamento e controle da qualidade (ControlQ), ferramenta de auxílio à análise por ponto de função, ferramentas para estimativas, ferramenta de gerência de recursos humanos, ferramentas de apoio à modelagem de sistemas, ferramenta de análise de risco e ferramenta de apoio à documentação.

É necessário, portanto, que os serviços de gerência de conhecimento propostos sejam providos para todo o ambiente ODE, incluindo suas ferramentas. Para entender melhor o funcionamento de ODE e como a gerência de conhecimento se insere no ambiente, é possível descrevê-lo como um conjunto de classes representando as várias funcionalidades do sistema, como mostra a Figura 7.

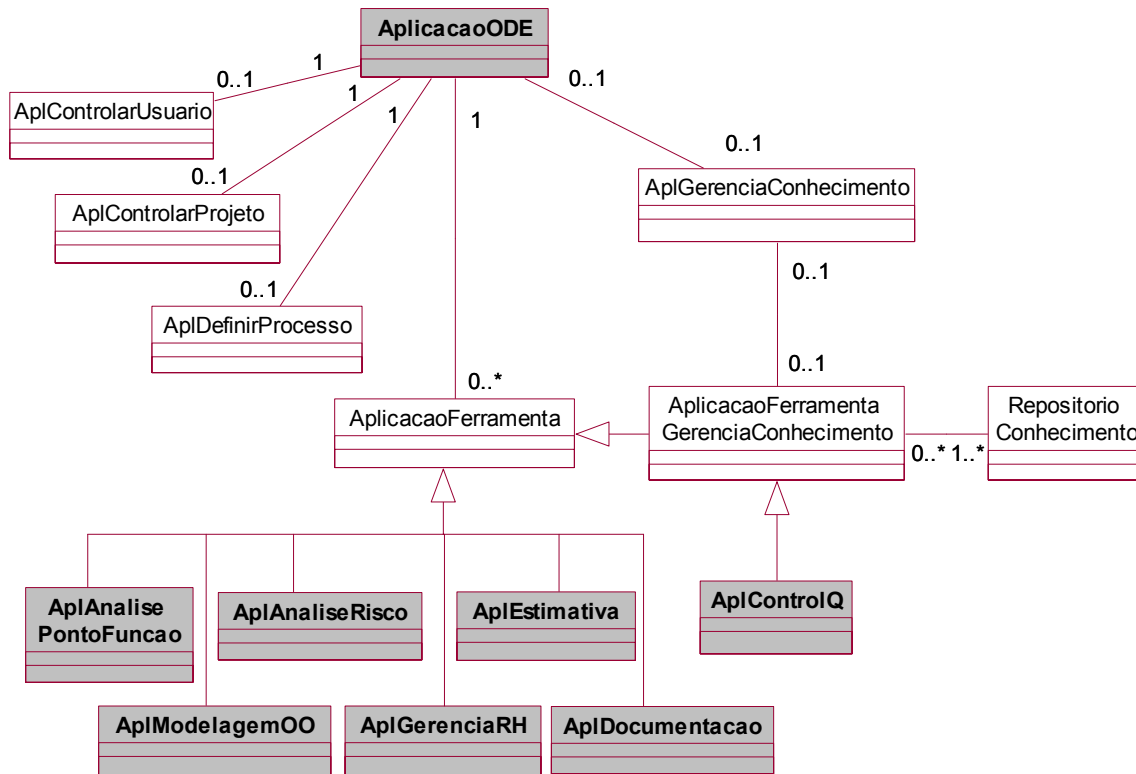
A classe *AplicacaoODE* representa o sistema como um todo, isto é, o ambiente ODE. Os objetos desta classe representam as várias sessões (execuções) do sistema. Devido à complexidade do ambiente, algumas de suas funcionalidades foram isoladas em outras classes, tais como *AplControlarUsuario*, que identifica o usuário no sistema e controla suas ações permitidas, *AplControlarProjeto*, responsável por apoiar a criação e o controle de projetos desenvolvidos no ambiente, *AplDefinirProcesso*, que permite a definição de um processo a ser seguido por um projeto e acompanha sua execução, e a classe *AplGerenciaConhecimento*, que trata os serviços gerais definidos para a infra-estrutura de gerência de conhecimento do ambiente. Finalmente, as ferramentas internas de ODE são representadas como subclasses da classe *AplicacaoFerramenta*. As classes da Figura 7 destacadas com sombreado representam aplicações executáveis, ou seja, o próprio ODE e suas ferramentas internas. Pode-se notar, então, que os serviços gerais da infra-estrutura de gerência de conhecimento pertencem ao ambiente ODE e podem se estender a suas ferramentas.

### **4.4 Serviços Específicos da Gerência de Conhecimento em Ferramentas de ODE**

Os serviços apresentados anteriormente estão presentes no ambiente ODE e podem ser utilizados a qualquer momento. Há, porém, um serviço que é provido para auxiliar ferramentas específicas do ambiente, levando em conta seu funcionamento e características particulares: a disseminação de conhecimento.

Ao contrário da busca, em que o usuário deve ter a iniciativa de procurar conhecimento, na disseminação, esta iniciativa é do próprio sistema. Ou seja, os itens de conhecimento que o

sistema julgar relevantes são apresentados ao usuário como uma sugestão de ajuda na realização de uma atividade. Mas não é possível oferecer ajuda pró-ativa sem conhecer detalhes sobre a tarefa que está sendo realizada e, conseqüentemente, sobre a ferramenta que apóia a atividade. Assim, o serviço de disseminação tem ser implementado ferramenta a ferramenta.



**Figura 7 –ODE e suas Ferramentas.**

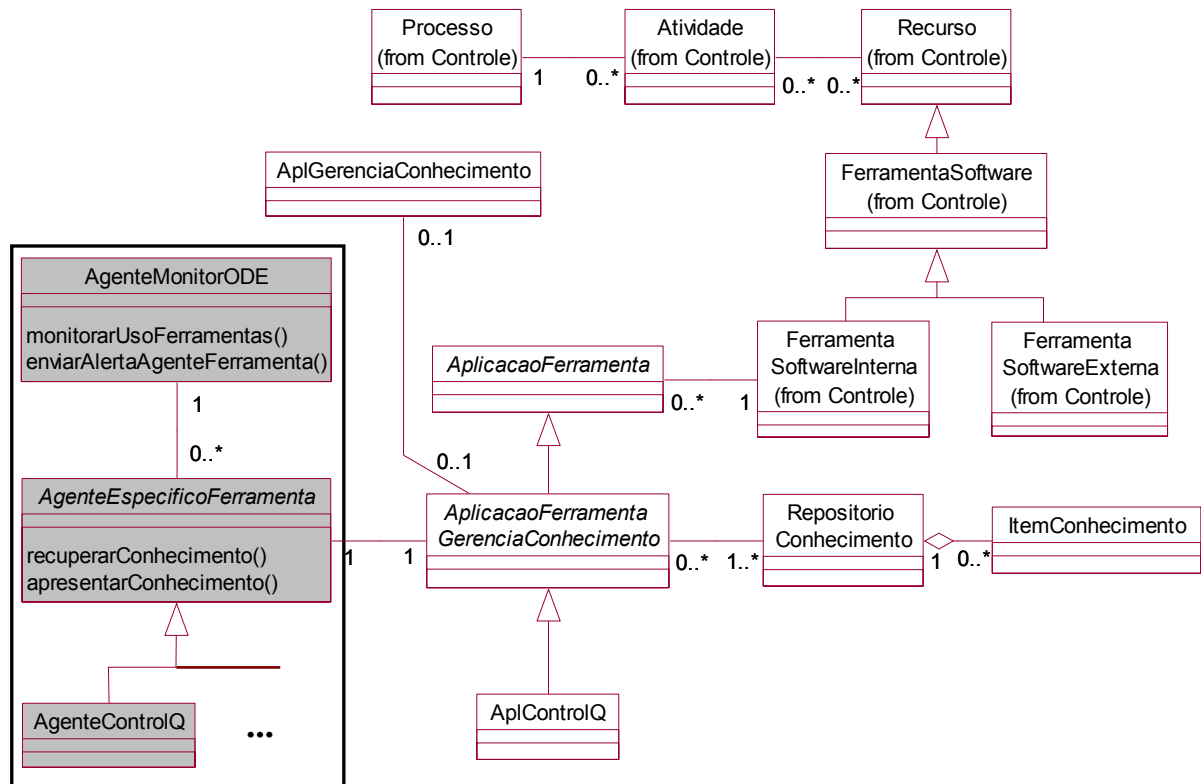
Em ODE, a proposta é utilizar agentes de software para monitorar as ações dos usuários enquanto eles utilizam uma determinada ferramenta. Quando o usuário está executando uma atividade do desenvolvimento de software, agentes de software específicos devem agir, identificando as necessidades de conhecimento do usuário e recuperando experiências anteriores. Esta disseminação de conhecimento é particularmente importante quando os usuários não estão motivados a buscar uma informação ou não sabem da existência de conhecimento relevante à tarefa que estão executando.

Para integrar a gerência de conhecimento diretamente a uma ferramenta de ODE, é preciso realizar alguns passos. Primeiro, deve-se determinar o local onde o conhecimento relevante para a ferramenta estará armazenado, isto é, a que repositórios de conhecimento a ferramenta terá acesso. É importante lembrar que cada repositório está associado a uma ou mais ontologias que estruturam o conhecimento armazenado.

O ponto mais importante, contudo, é oferecer conhecimento relevante à atividade que o usuário está realizando de forma pró-ativa. O serviço de disseminação é realizado a partir do funcionamento de um agente geral de ODE, que monitora as ações de usuário e verifica quando uma ferramenta com apoio de gerência de conhecimento começou a ser utilizada. A partir deste momento, é preciso que o agente específico para esta ferramenta acompanhe as ações do usuário e saiba *quando* apresentar conhecimento e *o quê* é considerado conhecimento relevante para essa ferramenta. Como um único agente de software não conhece o funcionamento interno de todas as ferramentas existentes em ODE, é preciso que

cada ferramenta com apoio de gerência de conhecimento tenha seu agente específico, definido e configurado em tempo de desenvolvimento da ferramenta.

Como não é possível adotar uma estrutura única para a disseminação de conhecimento nas ferramentas de ODE, a solução adotada foi construir um *framework* para facilitar a definição deste serviço nas ferramentas de ODE. No contexto desse trabalho, um *framework* deve ser entendido como uma estrutura computacional passível de extensão para aplicações dentro de um domínio específico. O *framework* apresentado na Figura 8 tem o objetivo de facilitar a implementação do serviço de disseminação de conhecimento nas ferramentas de ODE.



**Figura 8 - Framework Para Disseminação de Conhecimento em Ferramentas de ODE.**

Em ODE, cada *ferramenta de software interna* possui uma aplicação principal através da qual é executada. Para a respeitar a política de integração de ferramentas de ODE, estas aplicações de ferramentas de software internas devem pertencer à hierarquia de classes, cuja superclasse é *AplicacaoFerramenta*. Dessa forma, a aplicação principal de cada ferramenta deve herdar de *AplicacaoFerramenta*. As ferramentas de software internas de ODE que possuem o apoio da gerência de conhecimento definido, por sua vez, têm sua aplicação principal herdando da classe *AplicacaoFerramentaGerenciaConhecimento*. A Figura 8 mostra a aplicação ControlQ, uma ferramenta de apoio ao planejamento e controle da qualidade [12], herdando dessa classe.

Tanto as instâncias da classe *AplicacaoFerramentaGerenciaConhecimento*, ou seja, aplicações de ferramentas de software, quanto as instâncias da classe *AplicacaoODE*, isto é, o ambiente ODE como um todo, têm acesso a *repositórios de conhecimento*.

Além disso, é preciso criar uma especialização da classe *AgenteEspecificoFerramenta* associando-a à ferramenta que deseja apoiar. Nessa especialização, deve-se definir o *quê* apresentar (tipo de conhecimento) e *quando* apresentar o conhecimento, implementando os métodos *recuperarConhecimento()* e *apresentarConhecimento()*, respectivamente. Ou seja, é preciso obter respostas para as duas perguntas que orientam o desenvolvimento do agente específico: o que é considerado conhecimento relevante? E quando o agente deve apresentá-

lo ao usuário? A figura 8 mostra a classe *AgenteControlQ* que implementa o agente específico para atuar na ferramenta ControlQ.

Este agente específico é ativado pelo *AgenteMonitorODE* quando ControlQ for iniciada. Desta forma, o agente específico criado é capaz de recuperar e disseminar pró-ativamente conhecimento ao usuário da ferramenta no momento de sua utilização.

## 5. Pontos Fortes e Pontos Fracos da Infra-estrutura Proposta

A abordagem de gerência de conhecimento integrada a ODE reflete uma perspectiva dinâmica de gerência de conhecimento, na qual os usuários criam constantemente novo conhecimento durante o uso do ambiente. Alguns pontos fortes da abordagem proposta podem ser citados:

- **A memória organizacional não é fechada e está em constante evolução:** a abordagem proposta neste trabalho assume que a memória organizacional não contém todo o conhecimento necessário para apoiar seus usuários a entenderem e solucionarem seus problemas. Desta forma, os usuários de ODE podem criar conhecimento novo e este novo conhecimento pode ser armazenado e disseminado pela organização. A evolução da memória organizacional é feita obedecendo a um controle de crescimento, exercido sob duas formas: antes de estar disponível aos usuários, o novo conhecimento criado é validado (por exemplo, lições aprendidas aprovadas previamente) e, após armazenados, os itens de conhecimento sofrem manutenções constantes.
- **A memória organizacional é integrada ao processo de desenvolvimento de software:** a maior preocupação para a gerência de conhecimento em ODE é capturar informações do processo, enquanto seus usuários desempenham suas atividades de desenvolvimento de software habituais. Na abordagem aqui proposta, a memória organizacional está ativamente integrada ao processo de desenvolvimento de software e às práticas dos usuários que constroem e consomem o conhecimento. Desta forma, o novo conhecimento que surge é armazenado em repositórios, no momento que surge, para que possa ser compartilhado com outros usuários.
- **Usuários de conhecimento não são apenas receptores passivos:** os usuários de ODE não são apenas receptores passivos do conhecimento, mas são ativos pesquisadores, construtores e comunicadores do conhecimento. O conhecimento é construído no contexto do desenvolvimento de software.
- **Disseminação de conhecimento relevante:** abordagens de gerência de conhecimento devem prover informação que seus usuários necessitam, quando eles necessitam. A gerência de conhecimento de ODE pode desempenhar um papel ativo na disseminação do conhecimento. Agentes de software podem monitorar as ações do usuário e informar sobre potencial conhecimento relevante, já que procurar conhecimento relevante à tarefa que o usuário está desempenhando pode consumir bastante tempo.
- **Gerência de conhecimento auxiliando a melhoria de processo:** o compartilhamento de conhecimento pode auxiliar a melhoria do processo de software através da divulgação das experiências. Todo o conhecimento gerado relativo ao processo de software que possa contribuir para sua melhoria deve ser reutilizado, pois indica os defeitos e os pontos positivos que foram detectados durante a aplicação do processo no desenvolvimento dos projetos.

Contudo, há alguns requisitos para gerência de conhecimento que não foram atendidos em ODE ou que precisam ser apoiados de maneira mais efetiva. Podemos citar alguns pontos fracos da abordagem proposta:

- **Apoio ao Trabalho Cooperativo:** a construção de um produto de software é um processo que envolve grupos de pessoas com interesses e visões distintas, mas em direção a um só objetivo. Considerando um nível mais operacional, colaboração ocorre em uma grande variedade de contextos. Pessoas colaboram durante reuniões, através de meios de comunicação, como correio eletrônico, no trabalho trocando informações, documentos, bases de dados, etc. É preciso que a infra-estrutura para gerência de conhecimento proposta apóie o trabalho cooperativo, incluindo novos espaços para geração de conhecimento como fóruns, perguntas e respostas, etc.
- **Outros Tipos de Conhecimento:** A memória organizacional de ODE armazena apenas três tipos de conhecimento (lições aprendidas, artefatos e instâncias de ontologia). Porém, durante o desenvolvimento de software mais tipos de conhecimento podem ser capturados, tal como conhecimento tácito.
- **Apresentação da Informação:** a apresentação dos itens de conhecimento recuperados em ODE é realizada de maneira simples. Isso é particularmente um problema quando a uma memória organizacional é muito grande e, portanto, deveria ser provida uma apresentação mais efetiva dos itens armazenados, como, por exemplo, permitindo que o usuário pudesse navegar visualmente pela memória organizacional e verificar como os itens de conhecimento se relacionam.

## 6. Trabalhos Correlatos

Diversos autores propõem diferentes abordagens para a gerência de conhecimento atender às necessidades da engenharia de software, cada uma delas com um diferente enfoque.

*NetDE* [2] é um ambiente de desenvolvimento de software que possui um sistema de gerência de conhecimento para apoio ao domínio de projeto e administração de redes de áreas locais. *NetDE* apóia o acesso a informação por busca e pode ter um papel ativo na disseminação de conhecimento. O sistema *AnswerGarden* [5] atende às necessidades de compartilhamento de conhecimento em organizações de suporte ao cliente e *helpdesks*, capturando o conhecimento de especialistas e enviando rapidamente respostas conhecidas aos clientes. Em ambos os casos, o serviço de gerência de conhecimento é voltado para um propósito específico. Neste trabalho, a ênfase é prover uma infra-estrutura de gerência de conhecimento geral, que possa ser especializada, conforme discutido na seção 4.

Mais em linha com nosso trabalho está o trabalho de Villela et al. [3], que propõe uma infra-estrutura de conhecimento com o intuito de atender às necessidades de ambientes de desenvolvimento de software orientado à organização, configurados pela Estação TABA. Contudo, ainda que os propósitos sejam similares, a forma de prover os serviços de gerência de conhecimento e a estruturação dos repositórios é bastante diferente. Em ODE, os repositórios de conhecimento são sempre estruturados com base em ontologias, o que não ocorre na Estação TABA. No que tange aos serviços, a Estação TABA ainda não possui apoio à disseminação, além de serem diferentes as abordagens para tratar o uso e manutenção.

## 7. Conclusões e Perspectivas Futuras

Este artigo apresentou uma proposta para gerência de conhecimento em um ambiente de desenvolvimento de software. A infra-estrutura de gerência de conhecimento definida para ODE é organizada da seguinte forma: a memória organizacional está no centro da arquitetura e, em torno dela, estão posicionados serviços de apoio da gerência de conhecimento.

É importante ressaltar que o uso estruturado e eficiente da memória organizacional requer que os usuários tenham um mesmo entendimento dos conceitos e da taxonomia utilizados. Desta forma, a memória organizacional apresentada foi construída fundamentada em ontologias e alguns itens de conhecimento são instâncias dessas ontologias. Duas ontologias

têm destaque na versão atual de ODE: a ontologia de processo de software desenvolvida em [9] e a ontologia de qualidade de software definida em [10].

Buscando-se melhorar e expandir a abordagem de gerência de conhecimento proposta, alguns trabalhos estão sendo desenvolvidos, tais como a melhoria dos mecanismos de identificação de similaridade, através de técnicas de raciocínio baseado em casos e a integração dos serviços de gerência de conhecimento e de configuração de ODE, dotando uma perspectiva mais apropriada para a gerência de conhecimento baseada em documentos.

## Agradecimentos

Os autores agradecem à CAPES e ao CNPq pelo apoio financeiro a este trabalho.

## Referências

- [1] O’Leary, D.E., Studer, R., “Knowledge Management: An Interdisciplinary Approach”, *IEEE Intelligent Systems*, Vol. 16, No. 1, Janeiro/Fevereiro de 2001.
- [2] Fischer, G., Ostwald, J., “Knowledge Management: Problems, Promises, and Challenges”. *IEEE Intelligent Systems*, v. 16, n. 1 (Janeiro/Fevereiro), pp. 60-72 2001.
- [3] Villela, K., Santos, G., Travassos, G., Rocha, A.R., “Melhoria de Processos de Software e Evolução de Ambientes de Desenvolvimento de Software com base no Conhecimento do Domínio e na Cultura Organizacional”, *Simpósio Brasileiro de Qualidade de Software*, Gramado, Brasil, 2002.
- [4] Bertollo, G., Ruy, F.B., Mian, P.G., Pezzin, J., Schwambach, M., Natali, A.C.C., Falbo, R.A., “ODE – Um Ambiente de Desenvolvimento de Software Baseado em Ontologias”, *Anais do XVI Simpósio Brasileiro de Engenharia de Software - Caderno de Ferramentas*, Gramado, Outubro de 2002.
- [5] Basili, V. R., Tesoriero, R., Costa, P., Lindvall, M., Rus, I., Shull, F., And Zelkowitz, M. V., “Building an Experience Base for Software Engineering: A report on the first CeBASE eWorkshop”, Bomarius, Frank and Komi-Sirviö, Seija, Springer, In *Proceedings of Product Focused Software Process Improvement*, pp. 110-125, 2001.
- [6] Brossler, P., “Knowledge Management at a Software Engineering Company - An Experience Report, Workshop on Learning Software Organizations”, *LSO’99*, Kaiserslautern, Germany, pp. 163-170, 1999.
- [7] Henninger, S., “Case-Base Knowledge Management Tools for Software Development”, *Automated Software Engineering*, vol. 4, pp. 319-340, 1997.
- [8] Tiwana, A., *The Knowledge Management Toolkit: Practical Techniques for Building a Knowledge Management System*. Prentice Hall PTR, 2000.
- [9] Falbo, R. A. *Integração de Conhecimento em um Ambiente de Desenvolvimento de Software*. Tese de Doutorado, COPPE/UFRJ, Rio de Janeiro, RJ. Dezembro de 1998
- [10] Duarte, K. C., Falbo, R. A., “Uma Ontologia de Qualidade de Software”, In: *Anais do VII Workshop de Qualidade de Software*, João Pessoa – Paraíba. Brasil. Outubro 2000.
- [11] Mian, P.G., Falbo, R.A., “Supporting Ontology Development with ODEd”, *Proceedings of the 2<sup>nd</sup> Iberoamerican Symposium on Software Engineering and Knowledge Engineering*, Salvador, Brazil, 2002.
- [12] Natali, A. C. C., Falbo, R. A., “Knowledge Management in Software Engineering Environments”, In *Simpósio Brasileiro de Engenharia de Software (SBES’02)*, Gramado, Outubro de 2002.
- [13] Dieng, R., Corby, O., Giboin, A., Ribière, M., “Methods and Tools for Corporate Knowledge Management”, In: *Proceedings of the 11th Knowledge Acquisition, Modeling and Management Workshop*, KAW’98, Banff, Canada, April 1999.