

eSEE – Ambiente de apoio a experimentação em larga escala em Engenharia de Software

Paulo Sérgio M. dos Santos, Guilherme H. Travassos

PESC/COPPE – Universidade Federal do Rio de Janeiro (UFRJ) – Caixa Postal 68.511
21.945-970 – Rio de Janeiro – RJ – Brasil

{pasemes,ght}@cos.ufrj.br

Abstract. Currently, the software industry has limited knowledge regarding how software can be designed, implemented and evolved throughout its life cycle. It can be observed software engineers demanding improved models to allow the analysis of software processes and products and the understanding of the limits of a specific software technology. Besides, the ability to manage and share the scientific knowledge produced by their experiences and observations when exploring software technologies complements such demand. Considering the previous scenario, this paper describes eSEE (experimental Software Engineering Environment) that has been proposed to support software engineers in their experimentation and scientific knowledge management needs.

Resumo. Atualmente, a indústria de software tem conhecimento limitado sobre como software é projetado, construído e evoluído ao longo do tempo. Sente-se a necessidade de possuir modelos adequados para analisar processos e produtos de software; entender os limites das tecnologias de software em determinados contextos; e possuir habilidade de análise e de experimentação. Além disso, poder gerenciar e compartilhar o conhecimento científico gerado a partir das experiências com as tecnologias de software complementa esta necessidade. Neste sentido, este artigo descreve os conceitos gerais de eSEE – um ambiente para apoiar experimentação e gerencia de conhecimento científico em engenharia de software.

1. Introdução

A Engenharia de Software tem como um de seus propósitos fornecer meios para melhorar o desenvolvimento de software e sistemas. Para atingir este objetivo, métodos e ferramentas têm sido criados para apoiar abordagens de engenharia em diferentes estágios do processo de desenvolvimento e manutenção do software. No entanto, nota-se pequena aplicação de métodos investigativos para compreender o que torna bom e como construir bem um software [Pfleeger, 1999].

Existe um entendimento crescente na comunidade de Engenharia de Software de que estudos experimentais são necessários para desenvolver ou melhorar processos, métodos e ferramentas de desenvolvimento e manutenção de software [Tichy, 1998]. O método clássico para identificar relacionamentos de causa-efeito é conduzir estudos experimentais (experimentos controlados) nos quais somente algumas variáveis são alteradas. Estes estudos experimentais em engenharia de software envolvem, freqüentemente, estudantes resolvendo tarefas simples em sala de aula. Uma das principais críticas a tais estudos experimentais é sua falta de realismo, que pode deter a transferência de tecnologia da comunidade de pesquisa para a indústria. Os experimentos seriam mais realistas se fossem executados em sistemas reais, com profissionais de software que são representantes da população alvo da tecnologia, usando sua tecnologia usual de desenvolvimento em seu ambiente de funcionamento normal, o que amplia sobremaneira as dificuldades de sua realização.

A necessidade de experimentação em Engenharia de Software foi enfatizada pela primeira vez na década de 1980 em [Basili, 1986], e desde então, muitos outros artigos foram publicados para ressaltar a importância desse tipo de atividade: [Tichy, 1998], [Zelkowitz, 1998], dentre outros. Devido ao crescente interesse no campo da experimentação, muitos pesquisadores vêm investindo esforços na definição de tecnologia para apoiar a aplicação da metodologia científica em Engenharia de Software [Arisholm et al., 2002, Jedlisthka e Pfhal, 2005].

Técnicas de investigação científica já vêm sendo utilizadas em outras ciências, tais como Física, Engenharia, Biologia e Medicina, há várias décadas. Assim, os esforços iniciais de utilização do método experimental em Engenharia de Software tomaram como ponto de partida trabalhos oriundos dessas áreas de pesquisa, conforme pode ser observado em [Juristo e Moreno, 2001].

Desta forma, a construção de métodos e tecnologias de apoio à experimentação em Engenharia de Software é um ramo de pesquisa em ascensão e várias questões de pesquisa ainda estão em aberto. Os principais esforços de pesquisa nesta área envolvem, por exemplo, a definição de um processo de experimentação, mas ainda não há um consenso nesse sentido e um longo caminho ainda deverá ser percorrido [Jedlisthka e Pfhal, 2005], o que inclui a necessidade de realizar uma abstração um pouco mais elevada e raciocinar em termos do desenvolvimento e implantação de uma infraestrutura que apóie experimentação, incluindo os processos relacionados, a fim de tornar possível a assimilação e distribuição do conhecimento adquirido durante a condução de estudos experimentais em Engenharia de Software, considerando equipes distribuídas e experimentação em larga escala. Elas seriam premissas importantes no contexto do projeto eSEE, objeto deste artigo.

As próximas seções deste artigo trazem uma visão geral sobre um ambiente para apoiar experimentação em engenharia de software. A Seção 2 comenta sobre os aspectos relevantes em ciência em larga escala em engenharia de software. A Seção 3 apresenta conceitualmente o ambiente de apoio a experimentação, enquanto que a Seção 4 traz a sua arquitetura. Finalmente, a Seção 5 descreve o estado atual de construção do ambiente e cita alguns trabalhos em andamento.

2. Ciência em Larga Escala em Engenharia de Software

Há um crescente interesse em estudos experimentais em Engenharia de Software. Isto pode ser evidenciado pelo grande número de artigos descrevendo resultados de estudos experimentais publicados em vários veículos. Executar um estudo experimental não é simples. Geralmente várias atividades clíricas, como preenchimento de formulários, por exemplo, fazem parte de todo o processo de experimentação, tornando-o tedioso e repetitivo. Em complemento, Shull et al. (2001) ressaltam que a execução de estudos consome muito tempo e gera uma grande quantidade de informação que precisa ser gerenciada. Desta forma, observa-se uma demanda por ferramental que apóie a execução de estudos experimentais em Engenharia de Software.

Não importando o contexto, estudos experimentais são fortemente dependentes de uma infra-estrutura computacional, principalmente quando consideramos a necessidade da construção de modelos (simulação) e ciência em larga escala. Áreas como Física Nuclear, Informática biomédica, e Ciências da Terra já demonstraram

depender de uma infra-estrutura computacional integrada e distribuída pela *Internet* para gerência do processo de experimentação.

Em áreas mais recentes, como biologia computacional, experimentação também é fundamental. Nesta área, alguns experimentos são representados por estudos que envolvem a combinação múltipla de programas e dados, com gerenciamento complexo se realizado manualmente. O comportamento esperado de um participante em um estudo desta natureza está diretamente relacionado à sua habilidade de realizar todas as combinações viáveis dos dados e chamadas de programas necessárias pelo estudo. Existe um enorme volume de informação necessária, normalmente distribuída através da web, para serem utilizadas através de scripts ou programas que executam o experimento. A interação humana é restrita ao ponto inicial do processo de experimentação, onde o pesquisador configura o workflow científico que guiará o processo de experimentação, e ao final, quando novos dados estarão disponíveis para análise [Cavalcanti et al., 2003].

No entanto, diferentemente de outros meios de produção, desenvolvedores de software não produzem sempre os mesmos produtos. O software é produzido e cada produto é diferente do anterior [Basili et al., 1999]. Além disto, por ser a Engenharia de Software impactada pelo fator humano, existem outras inúmeras variáveis que tornam mais crítica a execução de estudos tanto no ambiente acadêmico quanto no industrial.

Do ponto de vista experimental, por exemplo, podemos ressaltar a representatividade do número de participantes em uma amostra da população. Alguns estudos executados não possuem um número de participantes que garanta a representatividade de uma população de desenvolvedores de software com um intervalo de confiança estatística adequado. Há também a questão da aleatoriedade, pois tanto na academia quanto na indústria, os pesquisadores não contam com a possibilidade de selecionar indivíduos aleatoriamente dentro de uma população. Estes participantes são geralmente selecionados por conveniência, mais especificamente, por estarem participando de algum treinamento, curso, projeto, ou outro tipo de atividade.

Do ponto de vista organizacional, fica difícil alinhar a execução de um estudo experimental aos objetivos e tarefas cotidianas da academia ou indústria. Por exemplo, é um risco para um projeto, acadêmico ou industrial, alocar desenvolvedores, tempo e/ou equipamentos para executar um estudo se este está fora de seu escopo. No contexto acadêmico, existe a necessidade de alinhar os objetivos de um estudo ao conteúdo, o objetivo e cronograma de uma disciplina acadêmica, para que se possam utilizar estudantes como participantes. A ausência de informações sobre esses e outros fatores dificulta a execução de um estudo e se não forem bem gerenciados podem constituir um risco à sua repetibilidade e, conseqüentemente, à validade de seus resultados.

Segundo Arisholm et al. (2002), a utilização de recursos da Web para apoiar a execução de estudos experimentais pode ajudar a tornar a experiência mais próxima das situações encontradas na indústria e a contornar vários problemas de logística, tais como a execução de tarefas por equipes geograficamente distribuídas, ou em seu próprio ambiente de trabalho. Isso facilitaria também o recrutamento, seleção e participação de participantes, permitindo o que os autores denominam de experimentos em larga escala, vislumbrando diminuir o problema de representatividade de amostras e aumentar a confiabilidade dos resultados observados.

Diante deste cenário, na tentativa de tratar os problemas citados na aplicação de experimentação em Engenharia de Software e com objetivo de atender a demanda por ferramenta por parte da comunidade científica, a equipe de Engenharia de Software Experimental da COPPE/UFRJ tem realizado esforços na definição e formalização do eSEE – *experimental Software Engineering Environment*. O eSEE foi inicialmente idealizado por Travassos et al. (2003) e, posteriormente, tratado em Mian et al. (2004), Dias Neto et al. (2004) e Chapetta et al. (2005).

3. Proposta Conceitual do eSEE e os seus Requisitos

A estrutura conceitual proposta para a infra-estrutura eSEE é baseada nos processos e pacotes de experimentação descritos em Costa et al. (2004) e Chapetta (2006). O objetivo de se tratar um estudo experimental como um processo tem base em alguns princípios do processo de software. Segundo Shull et al. (2001) um processo bem definido pode ser observado e medido, e desta forma melhorado. A Figura 1 apresenta o processo proposto por Costa et al. (2004), trabalho no qual a definição das etapas do processo pode ser encontrada com detalhes.

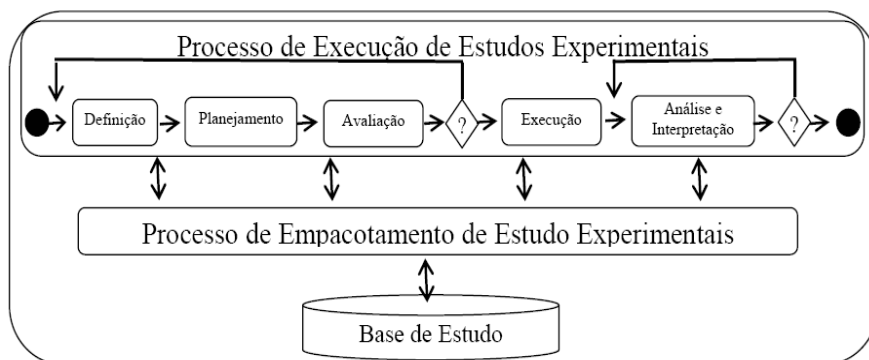


Figura 1. Processo de execução e empacotamento de estudos experimentais
[Costa et al., 2004]

A estrutura conceitual do eSEE também é fundamentada nos processos de experimentação específicos de pesquisa de opinião e revisão sistemática. Observando estes processos, podem-se observar várias atividades em comum. Todos eles possuem atividades de definição e/ou planejamento do estudo, sua execução, análise dos dados obtidos durante a execução e o armazenamento dos produtos gerados ao longo do processo. Isto nos permite dizer que as etapas citadas podem ser consideradas fases padrão do processo de execução de qualquer estudo experimental. Assim, um processo genérico poderia ser definido para servir como base para criação de vários processos específicos de experimentação.

Desta forma, três níveis de organização do conhecimento sobre o processo de experimentação devem ser tratados: conhecimento geral para qualquer tipo de estudo experimental (meta-nível), conhecimento para cada tipo de estudo experimental (nível de configuração), e conhecimento especializado para um estudo experimental específico (nível instanciado). Estes níveis foram refletidos na estrutura conceitual de eSEE. As camadas que compõem a estrutura conceitual de eSEE (Figura 2), são:

- **Meta-eSEE:** o meta-nível do ambiente contém o conhecimento geral sobre experimentação, incluindo conhecimento sobre engenharia de software. Neste

nível, está definido um processo de experimentação padrão, refletindo a generalidade comum dos estudos experimentais.

- **eSEE Configurado:** no nível de configuração, encontra-se o conhecimento sobre os tipos de experimentação específicos [Travassos e Barros, 2003]. Nesse nível, é realizada a instanciação de ambientes de experimentação específicos. No momento da instanciação, deve-se definir qual o tipo de experimento será executado. Em sua versão corrente, o eSEE provê apoio para estudos primários *in vitro*. Trabalhos tem sido realizados no sentido de estender o eSEE para tratar estudos *in virtuo* e *in silico*, bem como para apoiar estudos secundários (revisões sistemáticas).
- **eSEE:** ambiente de apoio à definição, planejamento e execução de experimentos. A cada novo experimento, um eSEE deve ser instanciado para gerenciar o experimento em particular.

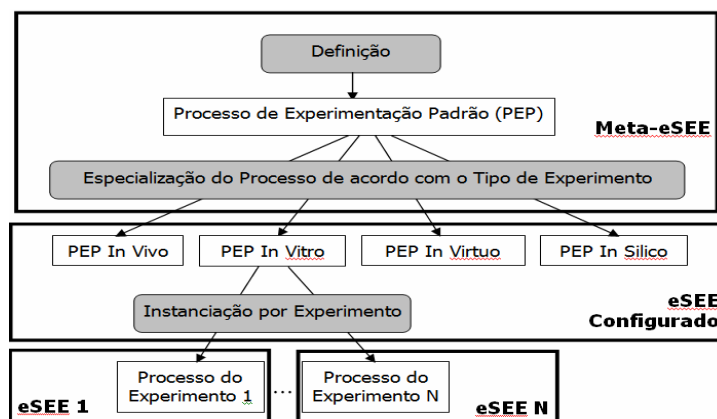


Figura 2. Os níveis da infra-estrutura de experimentação [Mian, 2006]

Com base na estrutura conceitual e com objetivo de dar continuidade ao desenvolvimento do ambiente eSEE, em Chapetta et al. (2004) são definidos os requisitos para o ambiente. Neste trabalho não detalharemos os requisitos do eSEE, mas de forma bem ampla, foram consideradas algumas pré-condições julgadas primordiais para garantir um melhor apoio às atividades de acompanhamento, execução e empacotamento de estudos experimentais. Essas pré-condições, listadas abaixo, nortearam a definição dos requisitos para o ambiente de experimentação:

- Ser um sistema Web, para permitir sua utilização em diferentes localidades, incluindo pesquisadores de diferentes instituições.
- Integrar ferramentas de apoio à experimentação, atuando de forma similar a um Ambiente de Desenvolvimento de Software (ADS).
- Ter uma arquitetura baseada no paradigma de serviço de software para facilitar a integração de ferramentas no ambiente.
- Possuir mecanismos que apoiem a Gestão de Conhecimento Científico.

Deste modo, a infra-estrutura eSEE fornece mecanismos de colaboração entre pesquisadores permitindo o acesso aos artefatos e ao conhecimento relativos a um estudo experimental.

4. Arquitetura do eSEE

A arquitetura definida para o eSEE é composta por três macro-componentes, de acordo com um agrupamento das características de seus sub-componentes identificadas através dos requisitos da infra-estrutura. Esses componentes são: Meta-Configurador, Ambiente de Instanciação e Ambiente de Execução (Figura 3). A seguir são descritos os macro-componentes que compõe a arquitetura.

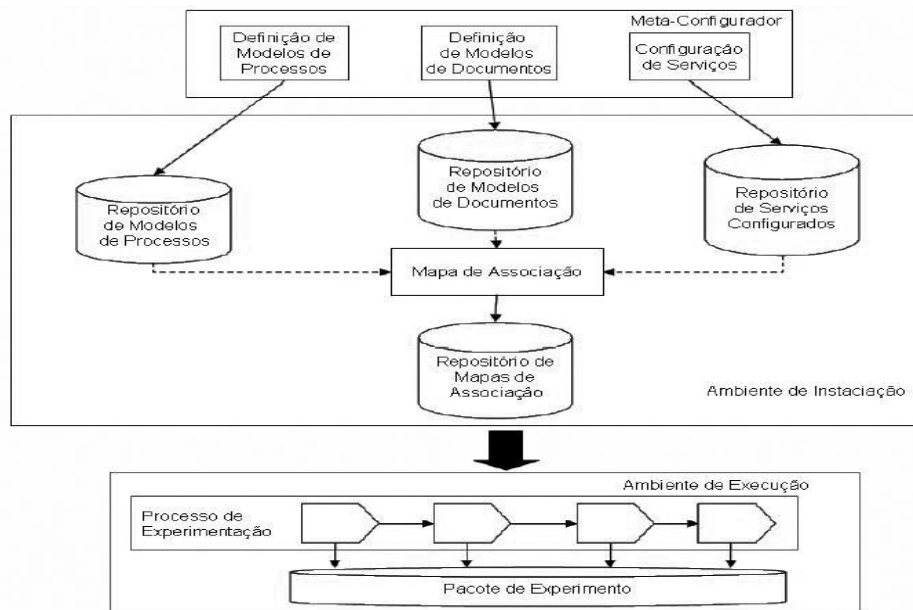


Figura 3 - Arquitetura eSEE e seus macro-componentes [Chapetta et al., 2005]

O Meta-Configurador possui como funcionalidade definir e configurar todos os elementos básicos que serão utilizados pelos dois outros componentes, que são:

- **Modelos de documentos:** são descrições que servem como base para instanciação de artefatos produzidos ao longo do Processo de Experimentação.
- **Modelos de Processo:** serve como base para a definição e instanciação de processos de execução de estudos experimentais.
- **Serviços Configurados:** informações relativas à utilização de serviços que possam apoiar diferentes atividades do Processo de Experimentação.

Neste nível os processos e os artefatos são descritos em XML segundo seus respectivos meta-modelos. Note que os conceitos (processo, artefatos e serviços) são descritos, mas nenhum relacionamento entre esses conceitos é descrito. Logo nesse componente estamos somente definindo o tipo, organização e granularidade das informações dos elementos básicos. Todos os conceitos descritos no componente Meta-Configurador são armazenados em repositórios específicos do Ambiente de Instanciação. Os repositórios estão sobre uma infra-estrutura de armazenamento XML nativa. Isto significa que tanto os dados quanto os meta-dados estão armazenados utilizando XML e que consultas ao repositório são realizadas através de tecnologias XML. O banco de dados utilizado no eSEE é o eXist (<http://exist.sourceforge.net/>).

O segundo componente, o Ambiente de Instanciação, possui como objetivo prover meios para instanciar um ambiente que permita a execução do Processo de

Experimentação, baseado em um modelo de processos e de documentos definidos no componente anterior. Para tal, as seguintes ações são realizadas:

- Definir o relacionamento entre o modelo do processo a ser seguido, o modelo de documento que vai ser preenchido durante o processo e os serviços configurados que devem ser disponibilizados para apoiar o preenchimento do modelo de documento. Este conjunto de relacionamentos é denominado mapa de associação. O mapa de associação define para cada atividade qual artefato será utilizado e qual serviço será disponibilizado.
- Baseando-se nestes relacionamentos o Ambiente de Instanciação deve permitir instanciar um Ambiente de Execução. Nesse momento o modelo de processo e o modelo de documento são instanciados em um Processo de Experimentação e um Pacote de Documentos, respectivamente.

A idéia é criar uma customização pré-instanciação do ambiente de execução. Isto é feito através dos relacionamentos descritos em um Mapa de Associação. Cada Mapa de Associação é então armazenado em um repositório específico, permitindo várias instanciações de uma determinada configuração do Ambiente de Execução. A Figura 4 apresenta o componente responsável pela criação dos mapas de associação.

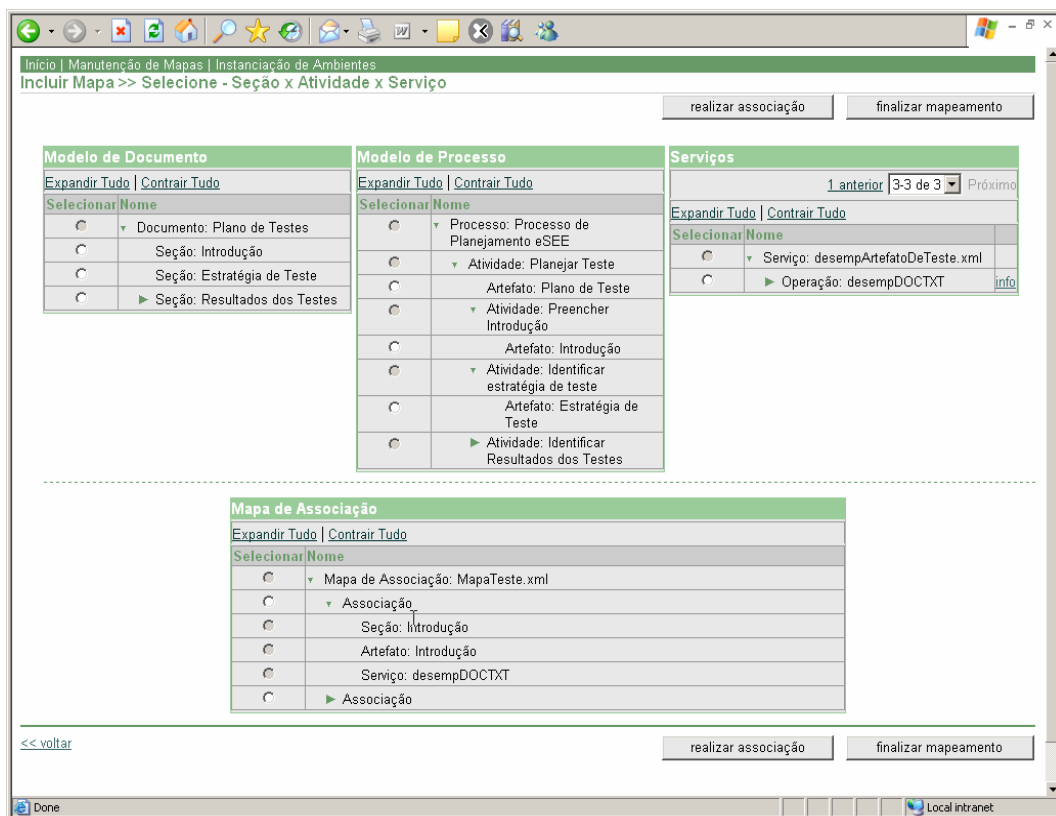


Figura 4 - Criação de um Mapa de Associação [Santos, 2007]

O Ambiente de Execução é o componente responsável pelo acompanhamento e execução das atividades do Processo de Experimentação. O acompanhamento do estudo será realizado através do processo definido pelo modelo de processo, de acordo com as restrições de acesso às informações e entre atividades do processo. O acesso às informações de um documento será realizado através dos serviços configurados, onde os relacionamentos descritos no Mapa de Associação são responsáveis pelo link entre o

artefato, uma determinada atividade do processo e o serviço configurado para realizar uma determinada tarefa.

Quando um Ambiente de Execução é instanciado seus repositórios são preenchidos. O modelo de processo selecionado no mapa de associação dá origem ao processo instanciado. A principal diferença entre um modelo de processo e um processo instanciado, é que o processo instanciado irá conter informações intrínsecas à execução do Ambiente de Execução, como datas efetivas de início e fim. Dessa forma, o processo instanciado não pode ser reaproveitado para utilização em outros ambientes. O modelo de processo, por outro lado, pode ser reaproveitado. O modelo de documento é usado para gerar os artefatos que deverão ser preenchidos durante a execução do processo. Ambos os modelos são armazenados no repositório de modelo utilizado para serem consultados posteriormente. O mapa utilizado na instanciación também é disponibilizado para acesso pelo Gerenciador de Serviço. A Figura 5 mostra a arquitetura detalhada do macro-componente Ambiente de Execução.

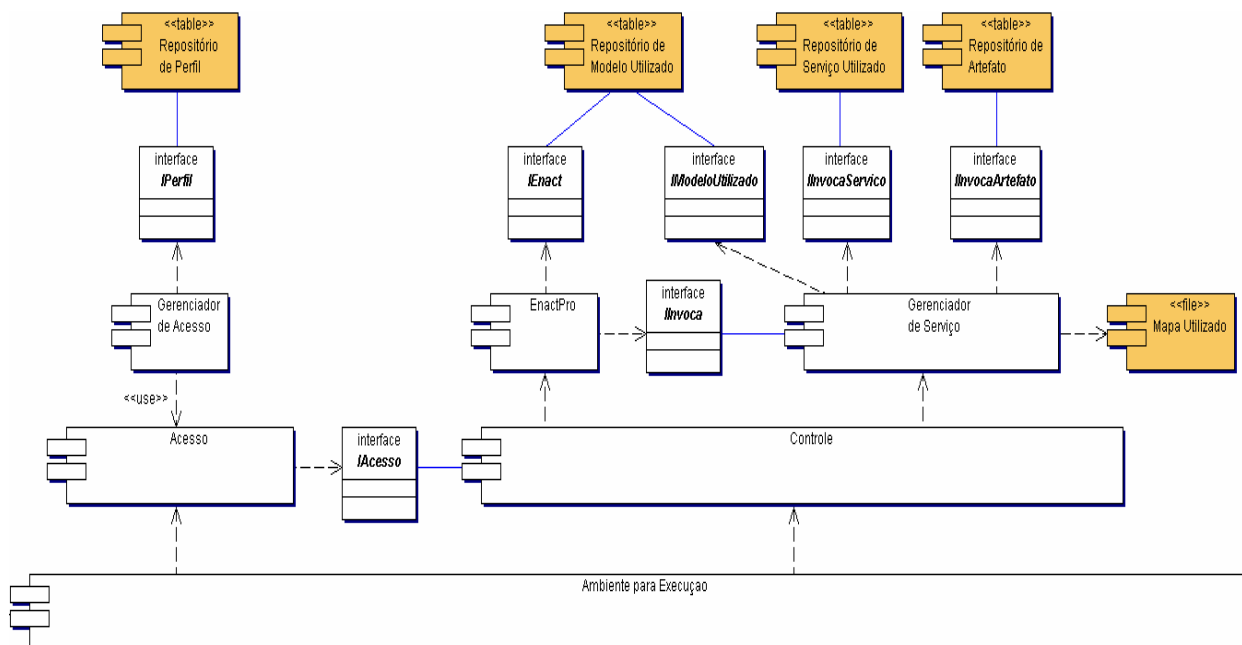


Figura 5 – Arquitetura detalhada do Ambiente de Execução do eSEE [Santos, 2007]

O objetivo do componente enactPro [Mafra et al., 2004] é prover um conjunto de facilidades visando automatizar a execução de processos de engenharia de software, permitindo o acompanhamento e o controle dessa execução através da visualização dos elementos do processo sob a perspectiva de atividades a serem realizadas, de artefatos a serem produzidos e de pessoas participantes dos processos. Para a utilização de enactPro, é necessária apenas a descrição do processo armazenada em um arquivo em XML.

O componente Gerenciador de Serviço tem o objetivo de prover facilidades para o apoio à construção dos artefatos previstos em um processo de engenharia de software. Entre essas facilidades, incluem-se a recuperação e persistência dos artefatos a serem manipulados e a gerência dos serviços que irão interagir sobre esses artefatos. O componente Gerenciador de Serviço trabalha em conjunto com o componente enactPro, à medida que trata as requisições de acesso aos artefatos produzidos durante a execução de um processo em enactPro. Ao executar enactPro, o usuário tem acesso aos

artefatos clicando duas vezes sobre o artefato. Quando isso ocorre, enactPro gera um evento que é capturado e tratado pelo componente Gerenciador de Serviço. O Gerenciador de Serviço, então, executa o serviço de edição, visualização ou processamento do artefato de acordo com o que estiver mapeado no Mapa. Dessa forma, ao final da execução do processo, os artefatos previstos encontram-se empacotados no repositório de artefato.

5. Considerações Finais e o Estado Atual do eSEE

Este artigo apresentou uma visão geral do ambiente de apoio à execução de estudos experimentais em engenharia de software – eSEE. Nesta apresentação foram explorados os aspectos de apoio inicial à ciência em larga escala (*e-science*), tais como utilização de infra-estrutura de *Internet* para a execução e acompanhamento do processo experimental e o uso de uma arquitetura orientada a serviços fundamentada em tecnologias XML para integração de ferramentas no ambiente.

Entretanto, atualmente, o eSEE ainda se encontra em construção. O Ambiente de Instanciação e o Ambiente de Execução estão parcialmente construídos, enquanto que o Meta-Configurador está totalmente concluído [Chapetta et al., 2005]. Novos tipos de estudos experimentais estão sendo descritos e inseridos na infra-estrutura. Estes incluem estudos primários *in virtuo* e *in silico* [Araújo e Travassos, 2005], bem como estudos secundários, notadamente neste momento revisões sistemáticas [Mian et al., 2005].

Para dar continuidade à construção do eSEE, os esforços de pesquisa têm se concentrado em três frentes: 1) Na definição e construção de uma primeira versão do componente Gerenciador de Serviços; 2) Na extensão da arquitetura do eSEE, através da realização de uma Engenharia de Domínio, para que o eSEE passe a apoiar diferentes tipos de estudo (ver Figura 2); e 3) Na construção de uma Ontologia de Pesquisa Científica em Engenharia de Software [Biolchini et al., 2007] visando facilitar a criação, acesso e reuso do conhecimento e a integração de ferramentas no ambiente.

Agradecimentos

O eSEE está inserido no contexto dos projetos CNPq – Engenharia de Software Experimental (472135/2004-0) e Infra-estrutura para experimentação em Engenharia de Software (302154/2004-3). Os resultados até o momento são consequência de um trabalho realizado em equipe pelo Grupo ESE (<http://www.cos.ufrj.br/~ese>), o qual os autores reconhecem e agradecem.

Referências

- Araujo, M. A. P., Travassos, G. H. (2005) Aplicação das Leis de Evolução de Software em Manutenção Evolutiva. Revista Tecnologia da Informação, Brasília, v. 5, n. 2.
- Arisholm, E., Sjøberg, D.I., Carelius, G.J., Lindsjörn, Y. (2002) "A Webbased Support Environment for Software Engineering Experiments", In: Nordic Journal of Computing, v. 9, n. 3 (September), pp. 231-247.
- Basili, V.R., Selby, R.W., Hutchens, D.H. (1986) "Experimentation in Software Engineering". In: IEEE Trans. on Software Engineering, 12 (7), pp.1728-1298.
- Basili, V.R., Shull, F., Lanubile, F. (1999) "Building Knowledge through Families of Experiments", In: IEEE Trans. on Software Engineering, vol. 25, No. 4.
- Biolchini, J. ; Mian, P. ; Conte, T. U. ; Natali, A. C. C. ; Travassos, G. H. (2007) "Scientific research ontology to support systematic review in Software Engineering", Advanced Engineering Informatics, v. 21, p. 133-151.

- Cavalcanti, M.C., Baião, F., Rössle, C., Bisch, P.M, Targino, R., Pires, P.F., Campos, M.L., Mattoso, M. (2003) "Structural Genomic Workflows Supported by Web Services", In: 1st International Workshop on Biological Data Management, Prague.
- Chapetta, W. A. (2006) "Uma Infra-Estrutura para Planejamento, Execução e Empacotamento de Estudos Experimentais em Engenharia de Software", Dissertação de Mestrado, PESC/COPPE, UFRJ, Rio de Janeiro, RJ, Brasil.
- Chapetta, W.A., Conte, T.U., Travassos, G.H. (2004) "Requisitos para um Ambiente de Experimentação em Engenharia de Software", PESC/COPPE/UFRJ, Relatório Técnico do Projeto eSEE. COPPE/UFRJ.
- Chapetta, W. A., Santos, P.S.M. and Travassos, G. H. (2005) "Supporting Meta- Description Activities in Experimental Software Engineering Environments", 2nd Experimental Software Engineering Latin American Workshop, Brasil.
- Costa, H.R.; Mian, P.G. Travassos, G.H. (2004) "Estendendo um Modelo de Processo e de Empacotamento de Estudos Experimentais", Relatório Técnico do Projeto eSEE. COPPE/UFRJ.
- Dias Neto, A.C., Barcelos, R., Chapetta, W.A., Santos, P.S.M., Mafra, S.N., Travassos, G.H. (2004) "Infrastructure for SE Experiments Definition and Planning". In: 1st Experimental Software Engineering Latin American Workshop, Brasília, Brazil.
- Jedlitschka, A., Pfahl, D. (2005) "Reporting Guidelines for Controlled Experiments in Software Engineering", In: Proceedings the 4th IEEE/ACM International Symposium on Empirical Software Engineering - Late Breaking Paper. Australia, November, 2005.
- Juristo, N., Moreno, A. (2001) "Basics of Software Engineering Experimentation", Kluwer Academic Press, 1st edition.
- Mafra, S. N. ; Barros, M. O. ; Travassos, G. H. (2004) "enactPro: Automatizando Processos de Software", In: 18 SBES - Sessão de Ferramentas, Brasília. Anais do 18 SBES - Sessão de Ferramentas. Porto Alegre : SBC.
- Mian, P., Travassos, G. H., Rocha, A.R.C., Natali, A.C.C (2004) "Towards a computerized infrastructure for managing experimental software engineering knowledge", In: Iv Jornadas IberoAmericanas en Ingenieria del Software e Ingenieria del Conocimiento, Madrid, v. 2. p. 475-487.
- Mian, P., Natali, A. C. C., Biolchini, J., Travassos, G. H., Conte, T. U. (2005) "A Systematic Review Process for Software Engineering", In: 3rd ESELAW - Experimental Software Engineering Latin American Workshop, Uberlandia.
- Mian, P.G. (2006) "Proposta de Infra-Estrutura Computacional para Apoiar Experimentação em Engenharia de Software". Exame de Qualificação, COPPE/UFRJ
- Pfleeger, S.L. (1999) "Albert Einstein and Empirical Software Engineering", IEEE Computer.
- Santos, P.S.M. (2007) "Ferramenta para Alocação de Serviços WEB em Ambientes de Engenharia de Software Experimental", B.Sc. Projeto Final de Curso, DCC/IM, Universidade Federal do Rio de Janeiro, Rio de Janeiro, RJ, Brasil.
- Shull, F., Carver, J., Travassos, G.H. (2001) "An Empirical Methodology for Introducing Software Processes", In: Proc. of 8th European Software Engineering Conference (ESEC) and 9th ACM SIGSOFT Symposium on the Foundations of Software Engineering (FSE-9). Viena.
- Tichy, W.F. (1998) "Should Computer Scientists Experiment More?" In: IEEE Computer, 31 (5), p. 32-39.
- Travassos, G.H., Barros, M.O. (2003) "Contributions of In Virtuo and In Silico Experiments for the Future of Empirical Studies in Software Engineering", In: Proc. of the WSESE03, Fraunhofer IRB Verlag, Roma.
- Travassos, G. H., Silva, L. F. S., Spinola, R. O., Kalinowski, M., Chapetta, W. (2003) "Tools and Facilities for Experimentation: Can we get there?" In: Panel presentation on the 11th International Software Engineering Research Network Meeting, Italia.
- Zelkowitz, M.V, Wallace, D.R. (1998) "Experimental Models for Validating Technology", In: IEEE Computer, 31 (5), p. 23-31.