

De maneira geral o formato da linguagem criada e utilizada como entrada pela ferramenta ficou aquém do esperado de uma linguagem de especificação. Um formato gráfico seria muito mais adequado para a correta compreensão dos diagramas. Neste trabalho, essa necessidade foi contornada através da utilização de uma ferramenta gráfica para a modelagem, o JUDE Community versão 5.3 [35], realizando-se a posterior conversão para o formato textual dos *scripts* da ferramenta. Tal conversão, embora simples de ser realizada, insere o ser humano em um procedimento que de outra forma poderia ser realizado automaticamente, tornando-o propenso a erros. Também foi necessária a atenção da equipe para manter o sincronismo entre os diferentes documentos, o que torna ainda maior a possibilidade de falhas nesse processo. Idealmente, o problema da co-evolução poderia ser resolvido com a utilização de uma linguagem de entrada da ferramenta que possuísse editores gráficos, de modo que por construção o esforço de conversão e sincronização entre os documentos não existiria. Em uma abordagem um pouco mais simples, pode ser construído um conversor que receba como entrada um formato de arquivo de alguma ferramenta gráfica e gere como saída o formato de entrada da ferramenta, textual. Essa abordagem, embora ainda torne necessária a interação humana, reduz muito a possibilidade de erros provenientes do processo de conversão, mas que ainda poderiam ocorrer devido a defeitos no conversor, equívocos na utilização do conversor, entre outros. Algum esforço de co-evolução ainda existiria, porém o processo automatizado seria uma barreira a menos para a manutenção dos documentos. A ferramenta de modelagem escolhida para esse trabalho, o JUDE Community versão 5.3 [35], provavelmente não seria uma boa opção para se realizar a conversão automática, pois gera os arquivos dos modelos em um formato binário e proprietário, o que dificulta a leitura dos modelos para a conversão.

Um ponto interessante seria a integração do *framework* a outras ferramentas, como sistemas de suporte a integração contínua, ambientes integrados de

desenvolvimento, ambientes de testes, entre outros, de forma que a execução dos testes fosse realizada de forma mais integrada à metodologia de desenvolvimento da organização. A ferramenta foi construída tendo em vista sua integração a outras ferramentas, de forma que a lógica de testes é bem separada dos módulos de entrada e saída. A integração a ferramentas mais populares no mercado provavelmente contribuiria positivamente para diminuir a barreira para a sua utilização, tornando mais simples para as organizações avaliá-la devido à maior familiarização de sua equipe técnica ao ambiente que a executa.

O maior objetivo desse trabalho foi a criação de uma ferramenta de testes que utilizasse uma linguagem para a criação dos testes que auxiliasse o máximo possível na especificação do que está sendo construído. Acredita-se que a linguagem proposta contribua bastante positivamente nesse sentido, entretanto, seria bastante interessante encontrar métodos mais empíricos e quantitativos de se concluir algo nesse sentido. Portanto, uma sequência natural para o trabalho seria conduzir um estudo com uma quantidade maior de projetos e integrantes, a fim de se medir o quanto a linguagem utilizada foi mais bem compreendida pelos membros dos projetos em comparação a formas mais tradicionais de se especificar testes, e de que forma esta maior compreensão contribuiu para reduzir o número de falhas encontrados no projeto por causa de defeitos de especificação, e o conseqüente custo para se realizar as correções. Apenas realizando um estudo dessa natureza seria possível afirmar com mais certeza a eficácia destas práticas para um aumento da qualidade e redução de custos no desenvolvimento de sistemas com as características estudadas neste trabalho.

De forma ainda mais geral, seria interessante verificar a eficácia de diferentes linguagens para diferentes naturezas de projetos, além da estudada nesse trabalho. Tal estudo contribuiria para se concluir sobre as vantagens de se utilizar linguagens específicas para cada domínio na criação dos testes, mostrando a utilidade da característica de facilidade de substituição de linguagem da ferramenta construída.

Uma maneira interessante de realizar esses experimentos seria medir projetos em que sistemas já em produção fossem traduzidos para outras linguagens. Embora nesses casos a diferença de linguagem fosse uma barreira para a medição de custos, seria possível obter uma medição bastante confiável de número de defeitos de especificação encontrados em cada fase do projeto. Para

verificar o entendimento da linguagem, seria possível também realizar um experimento onde se aplicasse questionários fechados a certa quantidade de integrantes dos projetos, distribuídos por diversos níveis e funções. Assim, seria possível verificar o grau de compreensão da ótica dos envolvidos na construção do sistema, permitindo conclusões sobre a clareza e a abrangência das especificações para cada um dos diversos papéis desempenhados no projeto.

A criação de módulos de integração da ferramenta criada a outras ferramentas já estabelecidas e que são amplamente utilizadas no mercado facilitaria bastante a condução desse tipo de experimento, uma vez que provavelmente aumentaria as chances de organizações utilizarem a ferramenta e as possibilidades de se obter população para o estudo.

Embora o *framework* criado tenha se mostrado suficientemente simples de ser instanciado durante a criação da linguagem de máquinas de estado objeto desse estudo, tal afirmação enfrenta uma grande ameaça a sua validade externa. Todo o trabalho de extensão da ferramenta foi realizado com a participação da própria equipe que a desenvolveu, o que remove a barreira de compreensão das interfaces e pontos de extensão. Para se concluir algo mais concreto sobre a simplicidade e clareza dos pontos de extensão disponíveis seria necessário um estudo que envolvesse diversas equipes, com uma maior quantidade de desenvolvedores, com diferentes formações, em várias organizações ou de forma independente, que empregassem outras práticas e metodologias diferentes das utilizadas na organização em que foi realizado o estudo objeto deste trabalho.

O contato com essas equipes externas poderia se dar através de um fórum de discussão disponível *on-line* onde os interessados na ferramenta se cadastrassem e encontrassem um espaço para apresentar suas dúvidas, que seriam respondidas por membros da equipe do projeto da ferramenta. Dessa forma seria possível perceber quais foram as maiores dificuldades encontradas pelos desenvolvedores usuários da ferramenta, sendo possível classificá-las por critérios definidos a partir do cadastro, identificando, por exemplo, erros mais comuns entre grupos com menor grau de instrução, possíveis barreiras de idioma, entre outros. Esse contato também ajudaria a popularizar a ferramenta, pois além de diminuir a barreira para a sua utilização, criaria uma comunidade ao seu redor, contribuindo para encontrar mais rapidamente as falhas que precisam ser corrigidas, criando um ambiente para o intercâmbio de módulos de linguagem e apresentação entre os

usuários, servindo como fonte de requisitos para novas versões através de críticas e sugestões apresentadas.

Novamente, iniciativas que contribuam para a popularização da ferramenta são essenciais para que se consiga realizar estudos mais abrangentes, e a criação de uma comunidade em torno da ferramenta, seja através do fórum de discussão ou qualquer outra medida nesse sentido, se faz necessária para que isso aconteça. Essa comunidade poderia ser iniciada através da criação do fórum de discussão no âmbito do grupo de pesquisa e desenvolvedores da ferramenta, afim de se criar uma documentação que auxilie novos trabalhos que venham a ser realizados por pessoas que não participaram do desenvolvimento da ferramenta e possam vir a enfrentar o mesmo tipo de barreiras que desenvolvedores de outras organizações encontrariam.