Uma Abordagem Semi-Automática para a Avaliação Comparativa de Software Educacional de Matemática

Ma. de Fátima C. de Souza[1,*] José Aires C. Filho[2] Mauro C. Pequeno[1]
fatimasouza@lia.ufc.br j.castro@ufc.br mauro@vdl.ufc.br

[1]Universidade Federal do Ceará
Departamento de Computação
Fortaleza, Ce, Brasil. 60455-760

[2]Universidade Federal do Ceará
Faculdade de Educação
Fortaleza, Ce, Brasil. 60020-110

Resumo

Atualmente tem-se observado uma intensa produção de softwares educacionais que, segundo seus fabricantes, podem auxiliar os professores no trabalho de aprendizagem dos alunos. Desse modo, a seleção de softwares educacionais pelos professores tem se tornado uma tarefa cada vez mais difícil. Argumentamos nesse trabalho que mesmo existindo inúmeras estratégias de avaliação elas não são efetivamente usadas para realizar uma escolha baseada na comparação precisa das características dos software avaliados. Nesse sentido, apresentamos nesse trabalho uma nova abordagem para a avaliação de softwares educacionais. Nessa abordagem propomos uma metodologia que tem como objetivo principal facilitar a seleção de softwares educacionais que possuam características funcionais semelhantes através da utilização de métricas definidas pelo próprio avaliador baseadas na teoria dos campos conceituais e extraídas automaticamente dos softwares avaliados.

1 Introdução

Tem-se observado, de forma cada vez mais intensa, o lançamento de softwares no mercado que, segundo seus fabricantes, poderiam auxiliar o trabalho de professores e facilitar a aprendizagem dos alunos. No entanto, grande parte destes programas é de baixa qualidade tanto técnica quando pedagógica, fato explicado pela dificuldade de se expressar conceitos pedagógicos na produção de software educativo (SE) [Tch02, Sta90]. Desta forma, é fundamental que se faça uma avaliação sistemática da qualidade e dos efeitos de tais softwares antes de aplica-los na sala de aula. Porém, tal procedimento não ocorre com freqüência, visto que muitas instituições de ensino adquirem programas que são utilizados pelos alunos sem uma avaliação prévia [Sil02]. Nestes termos, de forma simplificada, podemos dizer que avaliar um software educativo significa analisar as características de sua interface e suas implicações para o uso pedagógico [Gom02].

A seleção dos SE a serem trabalhados em sala de aula deve ser feita com a participação do professor que está em contato direto com o processo de ensino e aprendizagem, pois é ele quem vai identificar as dificuldades dos alunos, por meio da análise de suas ações, e vai propor o uso de materiais mais adequados a criar as situações favoráveis a aprendizagem dos conceitos mal compreendidos [Oli01b]. Contudo, pelo fato de existir uma grande variedade de SE disponíveis atualmente que podem ser utilizados em um mesmo conteúdo, essa tarefa de avaliação torna-se bastante difícil, principalmente se considerarmos as técnicas de avaliação propostas atualmente na literatura. De fato, essas técnicas têm como ênfase a análise tanto de aspectos cognitivos como de aspectos de usabilidade, normalmente utilizando métricas simplificadas de forma a categorizar um determinado software ou mostrar que certos aspectos estão ou não presentes. Desse modo, em virtude da expressividade limitada dessas métricas (por exemplo, pontuações que variam de 0 e 4) essas metodologias acabam não sendo eficazes quando a tarefa que está sendo realizada é a comparação de produtos de software.

Nesse artigo propomos uma nova metodologia de apoio à avaliação de SE. Nossa abordagem se baseia na idéia de formalizar tanto aspectos pedagógicos como técnicos de forma a possibilitar a realização de uma avaliação comparativa precisa entre SE. Nossa metodologia propõe a adoção e a especificação de métricas para aspectos relevantes definidas pelo próprio professor a serem aferidos nos SE que estão sendo avaliados. De modo a automatizar nosso processo de avaliação, propomos uma ferramenta, que denominamos de MeS (*Metrics Evaluator System*), que dá

*Trabalho apoiado pela FUNCAP: Processo 1212/04.

suporte à definição e à elicitação das métricas sugeridas pelo professor. Esse software realiza a comparação entre as avaliações fornecendo informações para que o professor possa realizar a seleção dos softwares avaliados com base nas métricas que ele mesmo definiu.

Além disso, também propomos nesse trabalho, a utilização da técnica de engenharia de requisitos de Pontos de Caso de Uso [Kar93], largamente utilizada para estimar tempo de desenvolvimento de software, juntamente com a teoria dos campos conceituais de Vergnaud [Ver90], de modo a capturar formalmente os requisitos funcionais baseados em fatores pedagógicos dos SE, permitindo a geração de métricas para dar suporte a avaliação precisa de SEs de matemática.

Nesse trabalho, além dessa seção, apresentamos na seção 2 uma visão geral sobre as técnicas utilizadas atualmente na avaliação de SE, além de uma breve descrição sobre a teoria dos campos conceituais. Na seção 3 apresentamos alguns conceitos importantes sobre engenharia de requisitos e sobre a técnica de pontos de caso de uso. Na seção 4 apresentamos uma nova metodologia de avaliação de SE cuja função principal é dar suporte à avaliação comparativa e precisa de SE de mesmos objetivos pedagógicos. Na seção 5 apresentamos um estudo de caso simples, onde aplicamos a metodologia proposta para avaliar softwares de ensino de matemática. Por fim, na seção 6 apresentamos algumas discussões sobre esse trabalho.

2 Avaliação de Software Educacional

2.1 Modelos de Avaliação

Existem vários modelos de avaliação de SE propostos na literatura, de uma maneira geral, podemos destacar os modelos de avaliação objetiva, formativa e os baseados em teorias da aprendizagem [Oli01a]. Na avaliação objetiva uma equipe multidisciplinar analisa diferentes aspectos a serem considerados na avaliação da qualidade do produto através de listas de critérios que representam a posição teórica de cada autor sobre a aprendizagem. Já a avaliação formativa é realizada durante a utilização do SE pelos próprios usuários através de entrevistas, questionários ou o acompanhamento de perto da interação do usuário com o software. Os modelos que utilizam teorias de aprendizagem, se propõem a justificar a concepção do SE de acordo com alguma teoria de aprendizagem. Em [Jún02] são levantadas várias formas de concepção como a empirista, por exemplo, que é baseada no condicionamento de estímulos.

Contudo, tradicionalmente, os SE são analisados seguindo-se grades de categorias oriundas da engenharia de software que focalizam parâmetros gerais relativos à qualidade da interface, à coerência de apresentação dos conceitos e aos aspectos ergonômicos gerais dos sistemas [Gom02]. Esta avaliação é feita a partir da aplicação de tabelas de critérios nas quais aspectos como: consistência da representação, usabilidade, qualidade da interface, qualidade do feedback, etc., são considerados segundo uma escala de três ou quatro níveis (regular, bom, ótimo; ou regular, bom, muito bom e ótimo) [Gom02].

A literatura sobre avaliação de SE é abundante no emprego de tabelas que ora se adaptam ao tipo de software (independentemente do conteúdo veiculado), ora se adaptam ao tipo de ferramenta (software ou site). Esta literatura busca pontuar aspectos importantes na análise de um SE como: idioma, conteúdos abordados, público alvo, documentação (ficha técnica clara e objetiva, manual do professor com sugestões para o uso, ajuda online), aspectos pedagógicos (facilidade no acesso às informações, adequação a faixa etária, clareza nas informações, tipo de exercícios), entre outros [Gom02].

Como já citamos, as metodologias para avaliação de SE são basicamente voltadas à verificação da presença de determinados requisitos nos softwares avaliados. Em alguns casos há a utilização de algum tipo de pontuação simples onde esses pontos são levantados basicamente a partir de observações superficiais e informais. Por conta dessa informalidade, podemos concluir que essas metodologias não são tão eficazes quando o objetivo principal é comparar SE de mesmo domínio a partir de requisitos funcionais e não funcionais, principalmente no que tange a aspectos pedagógicos. Esse fato fica mais evidente se tratarmos da avaliação de SE na área de matemática. Para se avaliar SE para essa disciplina devemos utilizar teorias e modelos capazes de responder por toda a complexidade por trás de um conceito matemático. Sobre esses aspectos, a teoria dos campos conceituais proposta por Vergnaud é uma das teorias mais atuais nesse campo [Jún02].

Argumentamos que a definição de métricas precisas para a avaliação de requisitos funcionais e não funcionais de SE de matemática utilizando os conceitos de campos conceituais de Vergnaud pode nos fornecer um mecanismo bastante eficiente para permitir a classificação de SE de matemática.

Em [Gom03] é fornecida uma metodologia de elicitação de requisitos baseados na abordagem dos campos conceituais de Vergnaud. Essa metodologia é composta por um conjunto de etapas que vão desde a identificação do domínio até a geração de casos de uso com os requisitos baseados nas necessidades dos usuários. Contudo, nesse trabalho, não são tratados aspectos da formalização das métricas baseadas nos casos de usos, de modo a fornecer um ferramental básico para a realização da avaliação precisa dos softwares.

2.2 A Teoria dos Campos Conceituais

A teoria dos campos conceituais, proposta por Vergnaud [Ver90], é uma teoria cognitivista que oferece um referencial ao estudo do desenvolvimento cognitivo e da aprendizagem de competências complexas, particularmente aquelas implicadas nas ciências, levando em conta os próprios conteúdos do conhecimento e a análise conceitual de seu domínio. Embora Vergnaud esteja especialmente interessado nos campos conceituais das estruturas aditivas e das estruturas multiplicativas [Ver83], a teoria dos campos conceituais não é específica desses campos, nem da matemática.

Vergnaud considera o campo conceitual como uma unidade de estudo para dar sentido às dificuldades observadas na conceitualização do real e a teoria dos campos conceituais supõe que a conceitualização é a essência do desenvolvimento cognitivo.

3 Engenharia de Requisitos

3.1 Requisitos de Software

Fundamental para entendermos a disciplina Engenharia de Requisitos é entendermos exatamente o que são requisitos de um sistema. Grosso modo, podemos descrever requisitos como as funções que deverão ser incorporadas pelo software, quando inserido em seu contexto de funcionamento. No entanto, necessidades como desempenho, integridade, disponibilidade e segurança seriam difíceis de acomodar nessa definição preliminar de requisitos. Apesar de tais qualidades transformarem-se, em algum nível de abstração, em funcionalidades do novo sistema, no nível de abstração no qual requisitos estão sendo tratados, tais qualidades não são funcionalidades do sistema em especificação. Dessa forma, segundo [Som04] podemos classificar os requisitos de um software da seguinte forma:

1. **Requisitos Funcionais:** São declarações de funções que o sistema deve fornecer, como o sistema deve agir a entradas específicas e como deve se comportar em determinadas situações. Em alguns casos, os requisitos funcionais podem também explicitamente declarar o que o sistema não deve fazer.

2. **Requisitos Não funcionais:** São restrições sobre os serviços ou as funções oferecidos pelo sistema. Entre eles destacam-se restrições sobre o processo de desenvolvimento, padrões, entre outros.

Normalmente, os requisitos funcionais são verificados em um determinado software apenas pela sua presença ou ausência. Já os requisitos não funcionais, não podem ser avaliados da mesma forma que os funcionais. De fato, a verificação de requisitos não funcionais não é uma tarefa simples, pois muitas vezes esses requisitos são também tratados como funções a serem fornecidos pelo sistema, sendo essas funções difíceis de serem testadas nos softwares. Para resolver esses problemas, os requisitos não funcionais devem ser expressos quantitativamente [Som04], utilizando métricas que possam ser efetivamente testadas.

Adicionalmente, para que possamos verificar de forma precisa a presença e a efetividade de um requisito funcional em um SE de matemática apenas a especificação de casos de uso não é suficiente. De fato, casos de usos são apenas descrições das atividades que um conjunto de atores podem realizar em um software. Nesse trabalho, abordamos esse problema em duas etapas. Na primeira propomos a utilização de algumas restrições na produção de casos de usos baseados nos conceitos de campos conceituais. Em seguida propomos a utilização de pontos de casos de uso para extrair informações das especificações a serem usadas na avaliação de SE.

3.2 Pontos de Caso de Uso

Atualmente, a análise de sistemas orientados a objetos utiliza diagramas de casos de uso para descrever as funcionalidades do sistema de acordo com a forma de utilização por parte dos usuários. A técnica de análise de dimensão

por casos de uso foi criada para permitir que seja possível estimar o tamanho de um software ainda na fase de levantamento de requisitos, utilizando-se dos próprios documentos gerados nesta fase de análise como subsídio para o cálculo dimensional. A técnica de estimativa por pontos de caso de uso foi proposta em 1993 por Gustav Karner [Kar93]. Essa técnica trata de estimar o tamanho de um software de acordo com o modo como os usuários o utilizarão, a complexidade de ações requerida por cada tipo de usuário e uma análise em alto nível dos passos necessários para a realização de cada tarefa.

Os passos necessários para a geração da estimativa são:

1. Classificar os atores envolvidos em cada caso de uso, de forma a obter um somatório de pontos não-ajustado. O peso total dos atores do sistema (*Unadjusted Actor Weight*, ou UAW) é calculado pela soma dos produtos do número de atores de cada tipo pelo respectivo peso;

2. Calcular o peso bruto dos casos de uso (*Unadjusted Use Case Weight*, ou UUCW). O cálculo do UUCW é realizado como no cálculo de peso dos atores, somando-se os produtos da quantidade de casos de uso classificados em cada tipo pelo peso nominal do tipo em questão. O peso total não ajustado (*Unadjusted Use Case Points*, ou UUCP) é calculado pelo somatório entre os pesos de atores e casos de uso: $UUCP = UAW + UUCW$;

3. Calcular os fatores de ajuste. O método de ajuste é constituído de duas partes - um cálculo de fatores técnicos *Technical Complexity Factor*, ou TCF), cobrindo uma série de requisitos funcionais do sistema; e um cálculo de fatores de ambiente (*Environment Factor*, ou EF), requisitos não-funcionais associados ao processo de desenvolvimento;

4. Finalmente, podemos calcular o valor total do sistema em (*Use Case Points*, ou UCP) utilizando-se da seguinte fórmula: $UCP = UUCP * TFC * EF$.

4 Avaliação Comparativa de Software Educacional

O objetivo principal da metodologia de avaliação que estamos propondo nesse trabalho é permitir que um professor que esteja escolhendo quais softwares utilizar em sala de aula para um determinado domínio da matemática (campo conceitual), seja capaz de realizar a escolha de forma mais precisa. Por escolha mais precisa, queremos dizer que não somente aspectos da qualidade da interface gráfica, que em muitos casos acabam escondendo outros aspectos importantes com relação à usabilidade do software, possam ser decisivos no processo de escolha. Estamos propondo nesse trabalho, uma forma mais aguçada de se aferir requisitos funcionais e não funcionais que sejam a priori relevantes para os softwares que estão sendo avaliados.

4.1 Metodologia de Avaliação Comparativa

Na Figura 1 apresentamos de forma simplificada a metodologia de avaliação que estamos propondo. Nesse diagrama podemos observar as atividades envolvidas no processo de avaliação e os resultados utilizados como saída e como entrada dessas atividades.

As seguintes são atividades da metodologia:

1. **Especificar os Requisitos Funcionais:** A especificação dos requisitos funcionais deve ser realizada através da construção de diagramas de casos de uso considerando as idéias dos campos conceituais de Vergnaud. Em [Gom03] é apresentada uma metodologia que pode ser aplicada para realizar essa tarefa. Contudo, na descrição final dos diagramas de casos de uso devemos impor algumas restrições. Essas restrições permitirão uma extração de informações mais precisa na próxima fase do processo.

2. **Calcular Métricas dos Requisitos Funcionais:** Depois de devidamente especificados os casos de usos, deve ser realizado o cálculo das métricas baseadas nos requisitos funcionais. Esse cálculo é realizado através de pontos de caso de uso é deve ser orientado à fatores pedagógicos baseados nas teorias da aprendizagem. No fim dessa etapa é gerada uma especificação XML com os resultados das avaliações.

3. **Definir os Requisitos Não Funcionais:** Nessa etapa são identificados os requisitos não funcionais que devem ser utilizados para avaliar o software. Esses requisitos devem ser escolhidos se observando os objetivos definidos para o software no contexto dos conteúdos que estão sendo utilizadas na sala de aula.

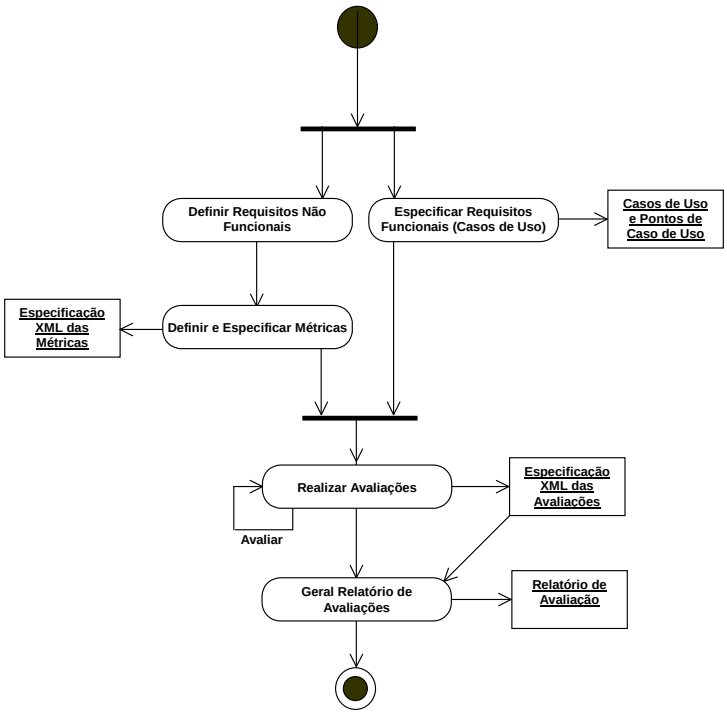

Figura 1: Atividades da Metodologia.

4. **Definir e Especificar Métricas para os Requisitos Não Funcionais:** Baseado nos requisitos não funcionais levantados na atividade anterior, devemos realizar a definição e especificação das métricas a serem utilizadas para aferir os requisitos escolhidos. A especificação dessas métricas, pode ser realizada diretamente através da construção de um arquivo XML onde todas as métricas devem ser relacionadas ou, de forma a automatizar esse processo, descrevemos mais adiante nesse trabalho, uma ferramenta que está em fase de construção para dar suporte à metodologia aqui proposta.

5. **Realizar Avaliações:** As avaliações dos softwares candidatos são realizadas nessa fase. Para cada software que está sendo avaliado, deverá ser criada uma entrada no arquivo XML com os dados capturados. Essa tarefa também deve ser automatizada. Apresentaremos esse software mais adiante. Assim, ao invés de anotar tudo que está acontecendo com relação às métricas definidas, o avaliador deverá interagir com o software que é automaticamente configurado com as métricas escolhidas pelo avaliador. Essa fase deve ser repetida para cada software que está sendo avaliado afim capturar as informações necessárias para realizar comparações. Como resultado dessa fase, será gerada uma especificação XML com os valores das métricas para cada software.

6. **Gerar Relatório de Avaliações:** Depois de devidamente extraídas, as métricas devem ser relacionadas de modo a facilitar a comparação dos softwares. Essa etapa tem como resultado um relatório com as métricas que foram aferidas em cada software tanto no que diz respeito aos requisitos não funcionais como aos requisitos funcionais extraídos nas etapas iniciais, e que retratam também fatores pedagógicos. Esse relatório poderá ser configurado para ser mostrado através de valores numéricos, de forma gráfica, entre outras opções.

7. **Analisar Relatório e Realizar Escolha:** Já com os resultados das avaliações, o avaliador poderá escolher os softwares que venham melhor se adequar às métricas que foram aferidas.

4.2 Especificação de Casos de Usos e Cálculo de Métricas Funcionais

Para a especificação dos casos de usos do SE que está sendo avaliado, deve-se aplicar a técnica proposta em [Gom03] que considera fatores relacionados à teoria dos campos conceituais para a produção de especificações de requisitos que tratem de fatores pedagógicos. Contudo, para a geração dos casos de usos, que é a última etapa do processo, algumas restrições devem ser observadas:

1. Devem ser utilizados apenas dois atores (**Professor**, **Aluno**), com pesos 1 e 5 respectivamente;
2. Os casos de uso devem ser classificados da seguinte forma:
   (a) Simples: define um caso de uso cuja função não é importante para o software e tem peso 1;
   (b) Médio: define um caso de uso cuja função é importante, mas não é essencial para o software e tem peso 2;
   (c) Importante: define um caso de uso cuja função é essencial para o software e tem peso 3;

Para a metodologia que estamos propondo não usaremos os fatores de ajuste, pois esses fatores normalmente se referem a requisitos não funcionais do software e, no nosso caso, esses requisitos serão definidos em uma etapa subsequente. Desse modo, o cálculo final para as métricas dos requisitos funcionais dos pontos de casos de uso será composto apenas pelas *Unadjusted Use Case Points* (UUCP), seguindo a fórmula apresentada anteriormente ($UUCP = UAW + UUCW$).

4.3 Definição e Especificação de Métricas Não Funcionais

Na fase de definição dos requisitos não funcionais são escolhidos que requisitos não funcionais são importantes para os softwares que estão sendo avaliados. Por questão de simplificação será utilizada nesse trabalho apenas a usabilidade como requisito funcional a ser avaliado. Para o requisito de usabilidade devem ser definidas e especificadas as métricas a serem utilizadas. Particularmente para usabilidade podemos utilizar métricas como [Nie93]:

1. Tempo para completar uma tarefa;
2. Razão entre sucessos e falhas;
3. Frequência da utilização do *help* ou da documentação;

Para cada métrica devem ser definidos valores que essas métricas podem assumir. A Tabela 1 apresenta alguns valores possíveis para as métricas definidas anteriormente.

Métrica	**Tipo de Valor**	**Formato**
Tempo para completar uma tarefa	Tempo	hh:mm:ss
Razão entre sucessos e falhas	Real	xxx.xx
Frequência da utilização do *help* ou da documentação	Tempo	hh:mm:ss

Tabela 1: Valores de Métricas.

De forma a permitir a especificação das métricas pelo avaliador, essas deverão ser descritas através de um arquivo XML (Figura 2) que segue uma gramática especificamente definida para esse fim. Por questão de espaço, a seguir apresentamos apenas um exemplo de um documento XML para a definição de métricas não funcionais. É válido observar que também os requisitos funcionais extraídos pelos diagramas de casos de uso e calculados pelos pontos de casos de uso na etapa anterior, também estão especificados nesse mesmo arquivo (essa especificação foi suprimida do exemplo por questão de espaço).

Através do documento XML anterior podemos realizar a extração das métricas dos softwares que estão sendo avaliados, sendo que, para cada software será criado um elemento do tipo `Software` com os dados relativos às métricas para serem avaliadas. Cada elemento `Software` por sua vez, possui um atributo `name` pela qual o software é reconhecido. Como elementos-filhos temos o elemento `description` pelo qual pode-se realizar uma descrição sucinta do software. A partir daí, as métricas a serem avaliadas para esse software serão definidas. Um elemento `NFMetrics` é utilizado para esse fim. Esse elemento é composto por um conjunto de elementos `Met`, cada um definindo uma métrica não funcional diferente. Sendo que esse elemento possui um atributo `name` que define o nome da métrica a ser avaliada. Como elemento filho de `Met` temos o elemento `Value` que define os valores que essa métrica pode assumir. No exemplo, mostramos uma métrica para medir a quantidade de "clicks" de mouse para realizar uma operação.

```
<?xml version="1.0"?>
<SoftwareEvaluation>
   <Document>
      <author name="M. de Fátima C. de Souza"
             email="fatimasouza@lia.ufc.br"/>
      <lastUpdate date="2004-07-12"/>
   </Document>
   <Softwares>
      <Software name="Venn">
         <description>
          Software que manipula diagramas Venn.
         </description>
         <NFMetrics>
          <Met name="Quantidade de Clicks de Mouse">
           <Value type="Int" rangeInit="0" increment="1">
          </Met>
          <Met name="Tempo para realizar uma operação">
           <Value type="Time">
          </Met>
         </NFMetrics>
      </Software>
   </Softwares>
</SoftwareEvaluation>
```

Figura 2: Código XML gerado pelo MeS.

4.4 Automatização do Processo de Avaliação

A característica básica da metodologia que propomos nesse trabalho é permitir a avaliação de SE de matemática através de um levantamento preciso de requisitos funcionais e não funcionais e pela definição de métricas pelo próprio avaliador. Contudo, para que esse processo seja de fato produtivo, e não apenas mais uma metodologia sem fins práticos, estamos desenvolvendo uma ferramenta de automatização do processo de avaliação. Essa ferramenta, que denominamos de MeS (*Metrics Evaluator System*), ou Sistema Avaliador de Métricas, que ainda está em fase preliminar de implementação, realiza a automatização de todas as etapas relativas à avaliação comparativa de SE. Essa ferramenta realiza as seguintes tarefas:

1. **Especificação de Casos de Uso e Cálculos de Pontos de Caso de Uso:** Através de uma interface gráfica, os casos de uso são descritos. Cada caso de uso tem uma anotação onde deve ser especificado o tipo do caso de uso para fins de cálculo das métricas funcionais. Os valores dos pontos de caso de uso para cada software são gerados automaticamente e inseridos em uma especificação XML que é utilizada nas etapas posteriores;

2. **Especificação de Métricas:** Através de uma tela, as métricas a serem avaliadas nos softwares são definidas. Essas métricas são manipuladas internamente no software em formato XML. Por exemplo, o arquivo XML mostrado em 4.3 é gerado automaticamente através do preenchimento dos campos de um formulário disponibilizado no MeS;

3. **Extração de Métricas Não Funcionais:** A partir das definições das métricas não funcionais é gerada uma interface padronizada para que o avaliador interaja com o MeS enquanto está utilizando o software que está sendo avaliado. Essa interação ocorre pelo registro na interface dos eventos relativos às métricas que estão sendo avaliadas. Nesse caso, será gerada automaticamente uma interface gráfica baseada nas informações do MeS para que o usuário possa realizar o registro das métricas enquanto avalia o software;

4. **Geração de Relatórios:** Através dos dados armazenados nos arquivos XML com as avaliações dos softwares, são gerados os relatórios através de *scripts* XSLT. Esses relatórios podem ser customizados de modo a serem mais ou menos detalhados de acordo com as necessidades do avaliador.

5 Estudo de Caso

Para validar a abordagem apresentada nesse trabalho, apresentaremos um exemplo simplificado da avaliação comparativa de dois SE utilizados para a conversão entre diversas medidas. Não apresentaremos aqui, por questão de espaço, os arquivos XML utilizados na avaliação nem as interfaces geradas pelo MeS.

5.1 Descrição dos Softwares

Os softwares escolhidos para mostrar a nossa abordagem de avaliação comparativa são dois softwares que apresentam características funcionais semelhantes, ou seja, os dois realizam as mesmas tarefas. Esse fator dificulta a decisão de qual software usar pelos professores. Com a nossa abordagem permitimos a automatização da aferição de requisitos, tanto funcionais como não funcionais, desses softwares, facilitando e formalizando a escolha do "melhor" software.

O primeiro software utilizado foi o **Unit Convertor 1.3** [WEB]. Esse software permite que após estabelecida a medida que se quer transformar, possa se indicar de onde se quer transformar para o que se quer transformar. O valor encontrado pode ainda ser armazenado, podendo se fazer outras transformações, mas sem perder o resultado anterior. A interface desse software, é mais atrativa visualmente, assim, esse fator pode ser decisivo para sua escolha prematura.

O segundo software foi o **CC Units 3.0** [WEB]. Esse software permite que após estabelecida a medida e o valor que se quer transformar, obtenhamos como resultado não apenas uma transformação como faz o software anterior, mas sim, todas as transformações possíveis dentro da escala que se escolheu. A diversidade na escolha das transformações é muito maior nesse software, do que no Unit Convertor 1.3, que permite apenas a transformação de apenas quatro medidas (áreas, aceleração, consumo de gasolina e tamanhos).

5.2 Especificação de Casos de Uso e Cálculo de Pontos de Caso de Uso

Para um software de conversão de medidas, além dos valores a serem convertidos, aspectos como a apresentação do processo de conversão, as operações que são realizadas, entre outras, são de suma importância para o processo de aprendizagem. Desse modo, o campo conceitual da álgebra deve ser utilizado na especificação dos casos de uso, de forma a capturar de forma mais precisa os fatores pedagógicos necessários para um SE desse tipo. Assim, aplicamos a metodologia apresentada em [Gom03] para realizar a elaboração dos casos de uso baseados nas idéias de Vergnaud.

Em seguida, os diagramas desenvolvidos devem ser especificados no MeS e os pontos para cada caso de uso devem ser fornecidos. Por questão de espaço, não apresentaremos a representação gráfica dos casos de uso nem todos os resultados especificados, mas apenas alguns dos valores (Tabela 2).

Caso de Uso	**Peso**
Converter Medidas de Área	3
Apresentar Processo de Conversão	3
Converter Medidas de Aceleração	1

Tabela 2: Exemplo de Casos de Uso.

Nesse exemplo, apresentamos somente alguns casos de uso relativos ao ator Aluno. Esses casos de uso devem ser verificados para cada software que está sendo avaliado. Isso é realizado pelo MeS, que gera uma tela com perguntas a serem respondidas pelo avaliador baseadas nas especificações dos casos de usos desenvolvidas. Dessa forma, são calculados os pontos de casos de uso para cada software da seguinte forma:

$$UAW = 5 * 1 = 5$$
$$UUCW = 2 * 3 + 1 * 1 = 7$$
$$UUCP = UAW + UUCW = 12$$

Assim, podemos concluir que os dois softwares avaliados possuem os mesmo valores funcionais. Isso implica dizer que eles realizam tarefas semelhantes no que tange aos aspectos que elicitamos nos casos de uso e aos valores de importância que demos para cada caso de uso. É bom observar que não estamos concluindo que os softwares

são absolutamente iguais funcionalmente, mas sim que, do ponto de vista pedagógico, eles atingem os mesmos objetivos funcionais.

5.3 Definição das Métricas Não Funcionais

Na definição das métricas não funcionais para a avaliação utilizamos duas métricas simples. A primeira é o tempo de execução. Nesse caso, levamos em consideração a realização de 10 operações, visto que o tempo para realizar uma única operação é muito pequeno. A outra métrica estabelecida foi o número de "clicks" no mouse para a realização de uma única operação. Vale ressaltar que existem diversas outras métricas que poderíamos ter utilizado, contudo resolvemos colocar nesse trabalho, apenas essas duas métricas que já são suficientes para dar uma boa visão da diversidade existente entre os softwares.

A definição dessas métricas pode ser realizada através da ferramenta MeS de forma bem simples. Sendo que um arquivo XML com o resultado da especificação das métricas é gerado no final dessa atividade. Logo em seguida utilizamos a interface gerada a partir das métricas que definimos para iniciar a avaliação dos softwares. Dessa forma, a medida que vamos utilizando o software usamos a interface de avaliação customizada para esse avaliação de modo a capturar os valores para as métricas. No final dessa operação teremos gerado um arquivo XML com os dados capturados das métricas para ambos os softwares.

Para realizar as avaliações utilizamos como operações a transformação da medida "Área", de quilômetro (quadrado) para metro (quadrado).

5.4 Resultados

A Tabela 3 apresenta os resultados obtidos no relatório gerado pela ferramenta MeS.

Métrica \ Software	**Unit Convertor**	**CCUnits**
Avaliação Funcional	12	12
Tempo para executar 10 operações	2 minutos	1 minuto
Total de "clicks" de mouse para realizar uma operação	5 clicks	3 clicks

Tabela 3: Resultados de Avaliação.

Os resultados funcionais, como apresentados anteriormente, são iguais, não sendo possível realizar uma escolha precisa do software somente a partir desse critério. Já com relação aos requisitos não funcionais no Unit Convertor levamos um tempo de 2 minutos para realizar as 10 operações de conversão, incluindo nesse tempo a operação de limpeza manual dos dados armazenados, uma vez que não existe a opção de limpeza automática nesse software. Já no CCUnits realizamos essas mesmas operações em apenas 1 minuto.

Quanto ao número de "clicks" no mouse para realizar 1 operação temos que no Unit Convertor foram necessários 4 "clicks", já incluindo o armazenamento, mais 1 "click" para se realizar a limpeza dos resultados armazenados, perfazendo um total de 5 "clicks". No CCUnits foram necessários um total de 3 "clicks", já incluindo o "click" para se realizar a limpeza automática dos resultados.

Constatamos através dos resultados obtidos no MeS, que apesar dos dois softwares terem a mesma funcionalidade em termos pedagógicos, e a interface do Unit Convertor 1.3 ser a priori mais atrativa, ainda assim o CC Units 3.0 é mais indicado, levando-se em conta o tempo utilizado e o número de "clicks", para se realizar o mesmo número de operações. Dessa forma, podemos concluir que existem características embutidas nos requisitos não funcionais que só podem ser fielmente analisadas se forem corretamente identificadas e extraídas dos softwares.

6 Conclusão

Para que todo o ferramental tecnológico disponível atualmente possa de fato fazer parte do cotidiano das salas de aula, não basta que os professores aprendam a utilizar o computador. Os inúmeros SE disponíveis atualmente fornecem mecanismos que podem auxiliar de modo efetivo a aprendizagem de conceitos, dos mais simples aos mais elaborados. Contudo, essa disponibilidade de SE tem causado a proliferação de produtos dos mais diversos tipos. Atualmente, podemos encontrar inúmeros softwares que trabalham conceitos semelhantes. Dessa forma, a tarefa

de se escolher um determinado software para se utilizar na sala de aula vem se tornando uma tarefa cada vez mais difícil.

As metodologias de avaliação de SE disponíveis atualmente realizam suas análises seguindo-se grades de categorias oriundas do campo da engenharia de software que focalizam parâmetros gerais relativos à qualidade da interface, à coerência de apresentação dos conceitos e aos aspectos ergonômicos gerais dos sistemas. Contudo, essas metodologias não se aprofundam em fornecer mecanismos para a realização de uma avaliação comparativa de SE que forneçam características funcionais semelhantes.

Nesse artigo, argumentamos que é importante o tratamento preciso de aspectos funcionais e não funcionais como um modo de realizar uma avaliação voltada para a seleção de produtos de SE de mesmo domínio de aplicação. Dessa forma, apresentamos nesse trabalho uma metodologia semi-automatizada no suporte a avaliação comparativa de SE. Essa metodologia considera que o próprio avaliador deve escolher que requisitos são importantes para ele e, a partir daí, realizar a definição e especificação de métricas para serem aferidas nos softwares que estão sendo comparados. Nesse trabalho, mostramos que é possível se automatizar e, portanto, tornar mais rápida e viável a aplicação da avaliação de produtos de SE, considerando tanto aspectos técnicos, como pedagógicos, através da adoção das idéias dos campos conceituais na geração de especificação de casos de uso.

Referências Bibliográficas

[Gom02] Gomes, Alex S. and Castro Filho, José A. and Gitirana, Verônica. Avaliação de Software Educativo para o Ensino de Matemática. In *Workshop em Informática na Educação.*, Florianópolis, Brasil, 2002.

[Gom03] Gomes, Alex S. and Wanderley,Eduardo G. Elicitando Requisitos em Projetos de Software Educativo. In *Workshop Brasileiro de Informática Educativa (WIE 2003).*, Campinas, SP, 2003.

[Jún02] Júnior, Amauri F. L. Avaliação de Software Educacionais de Matemática de Acordo com a Teoria dos Compos Conceituais: O Caso da Álgebra. *Monografia de Especialização em Ensino de Matemática. Universidade Estadual do Ceará*, 2002.

[Kar93] Karner, G. Metrics for Objectory. *Diploma thesis,University of Linkoping,Suécia*, 1993.

[Nie93] Nielsen, J. *Usability Engineering.* Chestnut Hill, 1993.

[Oli01a] Oliveira, C. C. and Costa, J. W. and Moreira, M. *Ambientes Informatizados de Aprendizagem - Produção e Avaliação de Software Educativo.* Papirus Editora.Campinas,SP., 2001.

[Oli01b] Oliveira, Silvia S. de and Gomes, Alex S. and Borges Neto, Hermínio. Avaliação de Software Educativo para o Ensino de Matemática - o Caso das Estruturas Aditivas. In *XV Encontro de Pesquisa Educacional do Nordeste.*, São Luís, Brasil, 2001.

[Sil02] Silva, Christina M. T. da. Avaliação de Software Educacional. *Revista on-line de Educação a Distância*, 2002.

[Som04] Sommerville, I. *Software Engineering. 7th. ed.* Addison-Wesley, 2004.

[Sta90] Stahl, M. M. Software Educacional: Características dos Tipos Básicos. In *I Simpósio Brasileiro de Informática na Educação.*, Rio de Janeiro, RJ, 1990.

[Tch02] Tchounikine, P. Pour une ingénierie des Environnements Informatiques pour l'apprentissage humain. *Information-Interaction-Intelligence*, 2002.

[Ver83] Vergnaud G. *Multiplicative structures*. Acquisition of Mathemtics Concepts and Processes. New York: Academic Press Inc, 1983.

[Ver90] Vergnaud G. *La théorie des champs conceptuels.* Recherches en Didactique des Mathématiques, 1990.

[WEB] WEB. Só Matemática. Disponível em: <http://www.somatematica.com.br/softwares.phtml>. Acesso em: 14/05/2004.