

UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação
ISSN 0103-2569

Resultados Experimentais da Avaliação do SASHE

Débora Maria Barroso Paiva
Maria das Graças Volpe Nunes

Nº 131

RELATÓRIOS TÉCNICOS



São Carlos – SP
Jan./2001

SYSNO	124510
DATA	1 / 1
ICMC - SBAB	

Resultados Experimentais da Avaliação do SASHE

Débora Maria Barroso Paiva
Maria das Graças Volpe Nunes

*Programa de Pós-Graduação em Ciências da Computação
Instituto de Ciências Matemáticas e de Computação - São Carlos
Universidade de São Paulo*

Resumo

Este relatório tem por objetivos detalhar o processo de avaliação realizado no módulo de autoria do sistema SASHE, registrar os resultados relativos à avaliação de um conjunto de requisitos especialmente propostos para sistemas hipermídia educacionais, registrar resultados adicionais obtidos a partir dessa avaliação e apresentar as pendências de implementação no contexto do SASHE, observadas durante a realização da avaliação

ICMC/USP
Janeiro 2001

Sumário

1. INTRODUÇÃO	1
2. ASPECTOS RELATIVOS À AVALIAÇÃO DO SASHE.....	2
3. RESULTADOS RELATIVOS AOS REQUISITOS PROPOSTOS EM (PAIVA, 2001)	6
3.1 REQUISITOS GERAIS DE AUTORIA HIPERMÍDIA (H)	6
3.2 REQUISITOS PARA SISTEMAS DE AUTORIA HIPERMÍDIA EDUCACIONAL (E).....	18
3.3 REQUISITOS ESPECÍFICOS DO PROJETO SASHE (S).....	22
3.4 REQUISITOS QUE NÃO FORAM AVALIADOS POR USUÁRIOS-AUTORES	24
4. ASPECTOS RELATIVOS À USABILIDADE DO SASHE.....	26
4.1 FACILIDADE DE APRENDIZADO	26
4.2 FAMILIARIDADE	28
4.3 CONFORTABILIDADE.....	29
4.4 CLAREZA E INTUITIVIDADE.....	29
4.5 CUSTOMIZAÇÃO	30
5. PROBLEMAS RELACIONADOS À INTERFACE DO SISTEMA	31
5.1 JANELAS DO SISTEMA.....	31
5.2 MENSAGENS DO SISTEMA	32
5.3 MENUS DO SISTEMA.....	32
5.4 OUTRAS OBSERVAÇÕES RELACIONADAS À INTERFACE DO USUÁRIO.....	32
6. SUGESTÕES DE MELHORIA DO SISTEMA.....	33
7. PROBLEMAS RELACIONADOS À EXECUÇÃO DO SISTEMA	34
8. ASPECTOS POSITIVOS DO SISTEMA	34
9. PENDÊNCIAS DE IMPLEMENTAÇÃO RELACIONADAS AO SASHE.....	35
10. CONCLUSÕES	36
REFERÊNCIAS BIBLIOGRÁFICAS.....	37

1. Introdução

O sistema SASHE é um ambiente de autoria e navegação de hiperdocumentos para aplicações educacionais desenvolvido no Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo com o objetivo de servir como ferramenta para usuários da comunidade universitária no âmbito do processo de ensino e aprendizagem (Nunes et al., 1997a, 1997b; Santos et al., 1997). O modelo de dados utilizado em sua implementação foi o Modelo de Contextos Aninhados (MCA) (Casanova et al., 1991).

A camada de autoria é composta, basicamente, pelos módulos de Modelagem, Importação e Edição. No módulo de Autoria o autor pode criar hiperdocumentos com vários níveis de aninhamento, sendo que a qualidade da navegação encontrada pelo aprendiz está diretamente relacionada a essa estruturação organizacional proposta pelo autor.

O módulo de Importação do SASHE implementa o protótipo da ferramenta Html2Hip (Haber, 1999), que tem por objetivo permitir a importação e adaptação de documentos HTML para a representação interna do SASHE, que se baseia na estruturação de objetos multimídia segundo a hierarquia de classes propostas pelo MCA.

A inclusão do módulo de Modelagem foi motivada pelas dificuldades advindas da utilização do MCA. Com o objetivo de fornecer uma base para o desenvolvimento da ferramenta de modelagem, foi definido o *Educational Hyperdocuments Design Method* (EHDM) (Pansanato, 1999), que fornece uma série de etapas para o desenvolvimento de aplicações hipermídia para ensino, além de oferecer modelos de representação adequados para descrever o domínio de conhecimento e os aspectos navegacionais dessas aplicações.

O EHDM é o método desenvolvido para permitir o projeto de aplicações no contexto do SASHE. A característica principal do EHDM é a utilização de primitivas de categorias de conhecimento e relacionamentos para modelar o domínio de conhecimento do hiperdocumento.

O EHDM é composto por três fases distintas: (1) modelagem conceitual hierárquica, (2) projeto navegacional de contextos e (3) construção e teste.

A fase de modelagem conceitual hierárquica consiste em estruturar o domínio de conhecimento, através da classificação de partes da teoria em categorias de conhecimento pré-definidas, e do estabelecimento de relações entre estas partes. O projeto navegacional de contextos deriva um modelo navegacional de contextos do modelo conceitual hierárquico criado na fase anterior. Contextos navegacionais, nós, elos e estruturas de acesso são definidos nessa etapa. Na fase de construção, um hiperdocumento real é obtido através de um processo de tradução do modelo navegacional de contextos para uma plataforma hipermídia. Nessa fase também ocorrem os testes, que procuram identificar os erros ocorridos na construção do hiperdocumento.

O EHDM forneceu a base metodológica para o desenvolvimento do EHDT (*Educational Hyperdocuments Development Tool*) (Pansanato, 1999), desenvolvido como uma ferramenta de suporte automatizado integrada ao SASHE que proporciona um ambiente de modelagem para auxiliar a autoria de hiperdocumentos para ensino neste sistema.

No módulo de navegação, o aprendiz encontra botões associados às funções de navegação disponíveis, que utilizam, em sua maioria, informações provenientes dos atributos e da contextualização dos nós de informação.

Este relatório tem por objetivos detalhar o processo de avaliação realizado no módulo de autoria desse sistema, registrar os resultados relativos à avaliação dos requisitos propostos em (Paiva, 2001), registrar resultados adicionais obtidos a partir dessa avaliação e apresentar as pendências de implementação no contexto do SASHE, observadas durante a realização da avaliação.

2. Aspectos relativos à avaliação do SASHE

Para a realização da avaliação do SASHE, buscou-se, inicialmente, por documentos que indicassem os requisitos estabelecidos para sua implementação. Observou-se, contudo, que os requisitos considerados na implementação não haviam sido formalmente definidos em uma etapa específica de análise de requisitos. Dessa forma, foi realizada a revisão dos documentos disponíveis relacionados ao sistema (dissertações, artigos, relatórios técnicos), de forma a recuperar essas informações. Foram realizadas também entrevistas com os projetistas e desenvolvedores do sistema. Nesta etapa foram obtidos 13 requisitos, sendo que 9 deles haviam sido estabelecidos considerando-

se referências de outros autores. Os requisitos recuperados foram registrados em Paiva (Paiva, 2001).

Além disso, foram pesquisadas publicações da literatura que indicassem requisitos cujas implementações fossem consideradas importantes em sistemas pertencentes a este domínio, de forma a complementar o conjunto de requisitos que deveria ser avaliado no SASHE. Diversas publicações foram revisadas, podendo-se destacar os trabalhos de Campos (Campos, 1994; Campos et al., 1999), Britain (Britain, 1999), Balagopalan e Macknight (Balagopalan & Macknight, 1989), Nanard e Nanard (Nanard & Nanard, 1995), Papatheodorou (Papatheodorou et al., 1998), Kearsley (Kearsley, 1982), Díaz e Isakowitz (Díaz & Isakowitz, 1995), Zellwegger (Zellwegger, 1989), Hall e Mendes (Hall & Mendes, 1999), Santos Jr. (Santos Jr., 1998), Pansanato (Pansanato, 1999), Nunes e Pansanato (Nunes & Pansanato, 1999).

Considerando-se o volume de dados obtido a partir da revisão destes trabalhos, optou-se por seleccionar requisitos que haviam passado por algum processo de validação ou que foram apresentados mais frequentemente nas publicações, tendo sido priorizado o primeiro caso. Dessa forma, foram considerados 17 requisitos, sendo que 11 deles foram classificados como *Requisitos Gerais para Sistemas de Autoria Hipermídia* e 6 deles foram classificados em *Requisitos para Sistemas de Autoria Hipermídia Educacional*. A primeira classe refere-se aos recursos que devem ser oferecidos por um sistema de autoria para possibilitar a construção de aplicações hipermídia sobre qualquer domínio, podendo ser utilizados, portanto, na etapa de engenharia de requisitos ou de avaliação de sistemas destinados a esta finalidade. A segunda classe refere-se aos recursos que devem ser oferecidos por um sistema de autoria para possibilitar a construção de uma aplicação do domínio específico de ensino. Analogamente, estes requisitos podem ser utilizados na etapa de engenharia de requisitos ou de avaliação de sistemas de autoria de aplicações para ensino.

Conforme sugere Sommerville (Sommerville, 1996) os requisitos foram definidos e especificados de forma a fornecer uma descrição abstrata em alto nível (definição) e uma descrição detalhada sobre o que o sistema deve possuir (especificação). Além disso, foram definidas questões que facilitam a avaliação desses requisitos em situações de teste. Estas questões consideram (a) a importância da implementação dos requisitos, segundo a opinião dos avaliadores; (b) a existência ou não dos recursos que satisfazem os requisitos; (c) a satisfação do usuário no que se refere à qualidade da implementação e (d) a flexibilidade da implementação do recurso (possíveis extensões

ou adaptações à implementação). Ressalta-se que outras métricas, por exemplo, a atribuição de notas à implementação dos requisitos, podem ser elaboradas e utilizadas.

Realizou-se, em seguida, a etapa de *Especificação da Avaliação*, de acordo com a recomendação da norma ISO/IEC 14598-5 (ISO/IEC 14598-5, 1996). Segundo esta norma, o propósito de especificar a avaliação é definir seu escopo e as medições a serem executadas no produto que será avaliado. Nesta etapa foram definidos os seguintes itens:

a) *Escopo da avaliação*: foi avaliado o módulo de autoria de aplicações hipermídia do SASHE (em particular, a ferramenta EHDT), considerando-se o conjunto de requisitos proposto neste trabalho.

b) *Objetivos da avaliação*: obtenção de dados que retratem as condições atuais da funcionalidade do módulo de autoria do SASHE, que apontem os problemas da implementação e que indiquem sugestões para o sistema.

c) *Funções atribuídas aos participantes*: esta avaliação envolveu a participação de alguns avaliadores (os usuários-autores, ou seja, os usuários finais do módulo de autoria) e de um supervisor. Os usuários-autores construíram aplicações hipermídia utilizando a ferramenta EHDT e os demais recursos a eles oferecidos. Dessa forma, eles puderam avaliar o sistema respondendo as questões elaboradas para os requisitos. O supervisor teve como funções (1) estabelecer as tarefas a serem cumpridas pelos avaliadores; (2) direcionar a execução da avaliação, esclarecendo dúvidas e auxiliando os avaliadores e (3) analisar e divulgar os resultados obtidos.

Além disso, foi elaborado um material que foi entregue aos usuários-autores anteriormente à execução da avaliação descrevendo as principais características e funcionalidades do módulo avaliado do SASHE.

O *Planejamento da Avaliação* foi a etapa realizada em seguida. Segundo a Norma ISO/IEC 14598-5 (ISO/IEC 14598-5), o propósito do planejamento da avaliação é documentar os procedimentos a serem usados pelo avaliador para executar as medições. Devem ser descritos os recursos necessários para executar a avaliação especificada bem como a distribuição desses recursos nas várias ações a serem executadas. Dessa forma, foram definidos os seguintes itens nesta etapa:

a) *Treinamento*: Para aprender a usar o SASHE, em particular o módulo de autoria de aplicações, os usuários-autores receberam material explicativo sobre as principais características do EHDM e do

EHDT, conforme citado. Além disso, foi oferecida ao usuário-autor uma demonstração de modelagem e construção de uma aplicação hipermídia. Após explicá-lo as etapas mais importantes do processo de desenvolvimento, o supervisor construiu uma aplicação sobre um tema específico (Evolução dos Computadores) com os objetivos de (1) facilitar o aprendizado dos avaliadores em relação ao sistema e, dessa forma, aumentar a possibilidade de se obter dados que retratassem as reais condições do sistema e (2) diminuir a influência nos resultados da inexperiência dos usuários com o sistema.

b) *Método de Avaliação*: Os requisitos propostos foram avaliados no SASHE através da execução de experimentos supervisionados realizados com usuários não experientes na utilização do sistema e que possuem a característica de serem professores. Após a etapa de treinamento, os avaliadores puderam começar a desenvolver suas próprias aplicações sobre um tópico específico da área de Ciências de Computação (Recursão), cumprindo tarefas pré-estabelecidas pelo supervisor, em um plano de avaliação (Paiva, 2001). As respostas às perguntas formuladas para avaliação dos requisitos e as opiniões dos avaliadores sobre a implementação das funções do sistema foram sendo registradas à medida que a aplicação foi evoluindo. Os experimentos foram gravados (apenas o áudio) para agilizar o processo de avaliação e assegurar que as informações obtidas seriam integralmente consideradas na etapa de análise dos resultados. Os experimentos foram realizados em um laboratório de Computação.

Foi utilizada a abordagem de avaliação proposta por Nielsen (Nielsen, 2000), sendo que somente a primeira etapa de testes experimentais foi realizada (foram realizados cinco experimentos). Após o reprojeto do sistema (considerando-se os resultados obtidos neste trabalho), outra etapa de avaliação deverá ser realizada (em trabalhos futuros), seguida por novo reprojeto do sistema e nova etapa de avaliação.

c) *Avaliação piloto*: O primeiro usuário do sistema executou todas as etapas propostas para a realização dos experimentos, porém, sua importância no contexto da avaliação esteve vinculada à identificação de possíveis problemas do processo de avaliação utilizado. A importância desta avaliação está relacionada à possibilidade de descobrir os problemas do planejamento da avaliação antes de executar todos os experimentos (Preece et al., 1994). Neste trabalho, em particular, os seguintes critérios foram analisados durante a avaliação piloto: (1) Qualidade (clareza, objetividade) das perguntas elaboradas para avaliação dos requisitos bem como a adequação da quantidade de

perguntas a serem respondidas; (2) Suficiência e qualidade do treinamento oferecido; (3) Abrangência das tarefas estabelecidas, no sentido de se conseguir ou não avaliar todos os requisitos propostos na etapa de análise de requisitos; (4) Adequação da hiperbase (contendo arquivos de texto, de imagem e de áudio) oferecida aos avaliadores.

d) *Classe de usuários*: Foram convocados professores (atuantes ou não) da área de Ciências de Computação, pois esta é a principal classe de usuários do módulo de autoria do SASHE. Os experimentos foram executados individualmente.

e)- *Recursos oferecidos aos usuários-autores*: Os usuários tiveram acesso aos seguintes recursos: (1) Material explicativo sobre o sistema; (2) Sistema SASHE em funcionamento; (3) Plano de avaliação contendo as tarefas a serem cumpridas; (4) Hiperbase composta por documentos (arquivos) referentes ao assunto (tema) da aplicação a ser construída. Esses arquivos foram organizados de forma a possibilitar boa exploração do tema, oferecendo várias possibilidades aos avaliadores; (5) Endereços de *sites* sobre o tema escolhido, que permitiram a utilização de outras informações sobre o tema da avaliação disponíveis na *web*.

A etapa seguinte ao processo de avaliação do SASHE foi a realização da avaliação piloto. Os resultados obtidos com a realização dessa avaliação geraram modificações no processo de avaliação, conforme descrito em (Paiva, 2001). Após implementadas as modificações necessárias, outros usuários-autores avaliaram o sistema. Os resultados são apresentados nas seções seguintes.

3. Resultados relativos aos requisitos propostos por Paiva (Paiva, 2001)

Para a obtenção dos resultados apresentados a seguir, foram considerados os requisitos propostos por Paiva (Paiva, 2001) e as etapas de especificação e planejamento descritas acima.

3.1 Resultados Referentes aos Requisitos Gerais de Autoria Hipermídia (H)

Os requisitos pertencentes a esta classe referem-se aos recursos que devem ser oferecidos por um sistema de autoria para possibilitar a construção de aplicações hipermídia sobre qualquer domínio.

Os resultados obtidos a partir da avaliação do SASHE referentes a esses requisitos são apresentados a seguir.

H.A.1. O sistema deve fornecer recursos que suportam laços de realimentação (*feedback loops*)

O sistema de autoria suporta laços de realimentação entre as fases de projeto da aplicação?

O EHDT suporta laços de realimentação ao permitir que (1) o autor possa mover-se facilmente pelas três fases do processo de modelagem da aplicação, (2) os diagramas gerados na modelagem estejam interconectados através do mapeamento dos elementos conceituais em elementos navegacionais, (3) protótipos sejam gerados a partir da tradução da especificação do diagrama navegacional para uma aplicação automaticamente.

Os recursos oferecidos auxiliam o desenvolvimento e a avaliação das aplicações?

Segundo os usuários-autores, estes recursos permitem aos autores avançar rapidamente pelas três fases, desde que estejam funcionalmente satisfatórios. Em relação ao EHDT, a implementação dos recursos (1) e (3) precisa ser aprimorada, pois a etapa de criação da aplicação (geração e avaliação dos protótipos até a obtenção da aplicação final) apresenta algumas limitações (descritas na resposta do requisito seguinte) que prejudicam o suporte a laços de realimentação. O item (2) foi citado por quatro usuários-autores como sendo importante para permitir mudar rapidamente entre as fases. Três usuários comentaram também a importância de o sistema mostrar exatamente o mesmo nível de modelagem quando o usuário troca entre modelagem conceitual e navegacional, conforme foi implementado no EHDT. Segundo eles, isso evita ter que procurar pelo diagrama requerido.

H.A.2. O sistema deve fornecer recursos que possibilitam prototipação do material

O SASHE fornece o recurso de prototipação?

Sim.

A implementação desses recursos é importante para o autor?

Quatro usuários-autores acharam importante esta funcionalidade, pois pode-se observar (em nível de instância) como a navegação foi projetada antes que a aplicação esteja pronta. Além disso, o usuário pode observar se a lógica que ele utilizou na modelagem confere com o que ele gostaria de apresentar ao usuário.

Um dos usuários-autores não soube opinar sobre a importância dessa funcionalidade.

Os mecanismos de prototipação disponíveis no sistema são satisfatórios?

Foram citadas algumas restrições da ferramenta em relação à prototipação:

- 1)- Se o usuário possui um protótipo da aplicação cujos arquivos referentes aos nós (arquivos de texto, imagem, áudio, vídeo) foram editados, e se ele necessita modificar a modelagem e gerar novo protótipo, todo o conteúdo dos arquivos é perdido. Dessa forma, o autor precisa concluir toda a modelagem para poder começar a editar os arquivos, o que prejudica a praticidade da ferramenta.
- 2)- Somente uma âncora é gerada em cada nó pertencente ao protótipo. Se determinado nó é modelado no EHDT possuindo mais de uma âncora, somente uma delas será gerada no protótipo.
- 3)- Os elos não são gerados quando os protótipos são criados. Isso dificulta muito o processo de desenvolvimento das aplicações porque os usuários têm que retornar ao módulo de modelagem para lembrarem os elos que foram criados e então criá-los novamente, o que acarreta executar a mesma função (criação de elos) mais de uma vez.

É fácil para o usuário gerar vários protótipos antes de decidir pelo melhor?

Tendo em vista as dificuldades citadas anteriormente, os usuários-autores não acharam fácil a realização desta etapa. Eles se sentiram desanimados a continuar o desenvolvimento das aplicações principalmente pelo fato de terem que criar novamente os elos, conforme citado anteriormente. No entanto, eles ressaltaram a utilidade desse módulo de criação de protótipos, devido à possibilidade de gerar vários protótipos e modificá-los antes de decidir pelo melhor.

H.A.3. O sistema deve fornecer recursos que permitam a utilização de abordagens de desenvolvimento *bottom up* e *top down*

*O sistema de autoria fornece recursos para utilização de abordagens de desenvolvimento *bottom up* e *top down*? Os recursos fornecidos pelo SASHE para utilização de abordagens de desenvolvimento *bottom up* e *top down* são satisfatórios?*

Os recursos oferecidos pelo sistema para utilização dessas abordagens de desenvolvimento se referem principalmente aos módulos de modelagem e prototipação. Todos os usuários-autores acharam importante e satisfatório¹ o processo de criação de modelos (abstração da aplicação) e posterior geração da aplicação (instanciação). Segundo eles, é importante ter a possibilidade de pensar na aplicação em alto nível de abstração e acrescentar detalhes posteriormente (abordagem

top down). Os usuários-autores confirmaram também que a prototipação de hiperdocumentos facilita os movimentos entre as camadas abstrata e concreta da aplicação, permitindo ao autor alternar entre as abordagens *bottom up* e *top down*, de acordo com o que foi proposto para o desenvolvimento da ferramenta.

Quanto à utilização da abordagem *bottom up*, dois usuários-autores solicitaram que o sistema atualizasse automaticamente os modelos criados na fase de modelagem quando o autor modificasse o protótipo da aplicação, de forma a manter protótipo e modelagem consistentes.

H.A.4. O sistema deve fornecer recursos que proporcionam geração de roteiros

O SASHE permite a geração de diversos roteiros de forma fácil?

Atualmente o sistema permite a geração de apenas um roteiro por aplicação. Este foi um aspecto negativo do sistema mencionado por três usuários-autores.

A implementação desses recursos é importante para o autor?

Quatro usuários-autores acharam importante a funcionalidade dos roteiros, porque permite sugerir uma ordem de navegação que facilite o aprendizado e que auxilie usuários autores e leitores a alcançarem seus objetivos. Foi citado que a construção de roteiros é válida desde que o leitor possa seguir também o caminho dos *links* e que ele tenha acesso a todo o conteúdo.

Um dos usuários-autores entende que a navegação através de elos seria suficiente, mas acha importante a restrição do acesso ao material. Por exemplo, o autor deve poder planejar um exercício mas deixá-lo disponível somente quando desejar.

Alguma dificuldade foi encontrada na atividade de criação de roteiro?

As dificuldades encontradas foram:

1)- O EHDT permite a criação automática de roteiros para os nós pertencentes ao modelo conceitual hierárquico, mas essa funcionalidade está incompleta (por exemplo, o usuário não é capaz de trocar a ordem de apresentação de um nó). Além disso, esse roteiro gerado no EHDT não é transportado para o SASHE (no qual a aplicação de fato é criada). Observou-se, dessa forma, a má integração do EHDT ao SASHE em relação à criação de roteiros.

Os itens relacionados a seguir referem-se à criação de roteiros utilizando-se o SASHE.

¹ O termo satisfatório refere-se ao processo genérico de modelar e, em seguida, criar a aplicação.

2)- Interface não é intuitiva. Sugeriu-se que a interface fosse semelhante à utilizada pelo *Windows Explorer*, que mostra a lista de arquivos pertencentes a determinado diretório. No caso da criação de roteiros, seria mostrada a lista sequencial de nós pertencentes ao roteiro, e o usuário poderia inserir, apagar e mover nós;

3)- Informações importantes para o autor, por exemplo, nome dos nós inseridos no roteiro, aparecem apenas como *hints*, o que dificulta o trabalho de criação e alteração de roteiros. Dessa forma, a visualização dos nós pertencentes ao roteiro torna-se bastante problemática.

4)- O sistema não mostra as âncoras pertencentes aos nós na janela de criação de roteiros. As tarefas de criação de roteiro e estabelecimento do grau de liberdade seriam mais facilmente realizadas se os usuários pudessem visualizar as âncoras.

A flexibilidade oferecida para a geração desses roteiros atende às necessidades do usuário-autor?

Não. Devido às dificuldades mencionadas acima, e à geração de apenas um roteiro, os usuários-autores sentiram-se desconfortáveis na realização dessa etapa.

Citou-se que seria muito interessante que a navegação pelos roteiros fosse implementada fornecendo-se aos usuários comentários (em um rodapé, por exemplo) sobre os objetivos do nó seguinte ao nó que está sendo visualizado. O usuário poderia escolher, então, por continuar seguindo o roteiro ou por fazer um desvio (navegar pelos *links*). Caso ele optasse pela segunda opção, seria interessante que o sistema marcasse com um *flag* o último nó do roteiro visitado, e então, quando o usuário necessitasse ou desejasse, poderia solicitar ao sistema que retornasse àquele nó, e ele continuaria seguindo o roteiro sugerido. Obviamente, ler as mensagens sobre os objetivos dos nós poderia gerar sobrecarga cognitiva, mas ficaria como uma opção ao usuário.

H.A.5. O sistema deve fornecer recursos que permitam especificar todos os aspectos do domínio de conhecimento da aplicação

As primitivas oferecidas pelo EHDT são suficientes, no que se refere à quantidade/qualidade, para a realização da modelagem de uma aplicação hipermídia para o domínio do ensino?

Os usuários-autores observaram que a categoria *conceito* foi a mais importante para a modelagem. Para a aplicação desenvolvida apenas um usuário utilizou a categoria *resultado* e três utilizaram a categoria *exemplo*. Dois usuários-autores citaram que a categoria *resultado* seria útil somente em alguns domínios, por exemplo, o domínio da Física. A categoria *exemplo* foi citada como sendo importante para o domínio de Ensino.

Quais seriam possíveis extensões ou adaptações para esse conjunto de primitivas?

Os usuários-autores não citaram extensões para as categorias de conhecimento e relacionamentos. Um deles comentou o fato de que criar categorias de conhecimento é um trabalho muito complexo e envolve muito estudo, por isso, é difícil sugerir categorias.

Em relação às adaptações:

- 1)- Tornar a categoria *exemplo* hierarquizável, pois o usuário pode desejar colocar diversos exemplos em um mesmo contexto ou detalhar um exemplo e assim necessitar criar um novo nível;
- 2)- Possibilitar maior liberdade de escolha dos relacionamentos, de forma que a ligação entre categorias não fique restrita a alguns relacionamentos. Um exemplo seria permitir ligar duas instâncias da categoria *conceito* com o relacionamento de *dedução*².

A modelagem do domínio de conhecimento pode ser estruturada hierarquicamente?

O EHDT permite a estruturação hierárquica da modelagem. Quatro usuários-autores acharam válida e fundamental a idéia de hierarquizar um modelo, o que o torna mais simples. Além disso, foi citado que hierarquizar facilita o entendimento de um modelo, o que é especialmente importante quando um usuário-autor precisa compreender um modelo criado por outro usuário para estendê-lo ou adaptá-lo, pois, possuindo-se maior nível de detalhes, a pessoa poderá compreender melhor o que terá que fazer. Entre esses quatro usuários-autores que acharam importante a estruturação hierárquica, um deles citou que gostaria que o sistema permitisse que vários níveis fossem mostrados simultaneamente na tela.

No entanto, um dos usuários-autores não sentiu necessidade de criar diferentes níveis hierárquicos (utilizou somente o primeiro nível – genérico – e criou o segundo nível, com toda a modelagem), mas percebeu também certa deficiência do sistema em relação à visibilidade de níveis. O usuário-autor citou, ainda, que se o sistema mostrasse os níveis de forma *top down*, como se fosse uma árvore, em uma única tela, ficaria mais fácil fazer a estruturação. Segundo ele, uma ferramenta que oferece um editor que possui essa funcionalidade é o *ConcurTaskTrees* (Paternò et al., 1997). Foi sugerido que o EHDT adotasse um editor semelhante.

A implementação desse recurso atende às necessidades do usuário-autor?

² Os autores entendem, no entanto, que toda dedução implica um resultado e não outro conceito.

Em relação à modelagem do domínio de conhecimento, dois usuários-autores citaram como restrição a rigidez envolvida nesta etapa, pois o usuário precisa selecionar categorias e relacionamentos e, quando o sistema gera o modelo navegacional de contextos, todos os relacionamentos se tornarão *elos* e todas as categorias se tornarão *nós*.

H.B.1. O sistema deve fornecer recursos que informam as referências dos nós ao usuário-autor

O sistema de autoria fornece a lista de ligações ou mapa local do nó que está sendo editado?

O SASHE não oferece este recurso.

A implementação desses recursos é importante para o autor?

Três usuários-autores acham a implementação deste requisito importante porque facilita a manutenção das aplicações. Caso ele queira modificar uma categoria (por exemplo, mudar uma categoria conceito para a categoria resultado), essas informações são relevantes porque permitiriam ao autor visualizar quais são os tipos de relacionamentos existentes com a categoria atual, e se esses relacionamentos deverão ser modificados (porque existem relacionamentos específicos para cada tipo de categoria). Caso o autor queira apagar uma categoria, essas informações permitiriam que ele visualizasse quais as consequências dessa decisão em relação às outras categorias, antes de apagar efetivamente.

Dois usuários-autores citaram que, apesar de importante, este não é um requisito primordial.

H.B.2. O sistema deve fornecer recursos que facilitam a correção (manutenção) das aplicações

A manutenção das aplicações (modificação dos dados referentes aos componentes das aplicações - nome, arquivo a que está vinculado; inserção, alteração ou remoção de nós e ligações) ocorre de forma simples? Alguma dificuldade foi encontrada na realização desta atividade?

Segundo quatro usuários-autores, a manutenção das aplicações não ocorre de forma simples. Algumas dificuldades foram destacadas:

1)- Na etapa de modelagem, o sistema não permite que uma categoria ou relacionamento seja modificado diretamente. O usuário precisa apagar o item atual e inserir um novo item. Além disso, quando o usuário seleciona a opção *Alterar*, mas não seleciona uma categoria, o sistema mostra a

janela de alteração mas não permite a interação do usuário. Sugeriu-se que o sistema alertasse o usuário (através do envio de uma mensagem) sobre a necessidade de selecionar a categoria; ou que a janela de alteração não fosse mostrada;

2)- Na etapa de criação de roteiros, pelo fato de a janela *Roteiros* não apresentar uma interface intuitiva, torna-se difícil executar atividades, como excluir um nó do roteiro, mudar o grau de liberdade de um nó, alterar a ordem de apresentação entre dois ou mais nós, etc;

3)- Citou-se novamente o fato de o sistema apagar o conteúdo dos arquivos quando o usuário precisa alterar algum aspecto da modelagem. Segundo o usuário, dificilmente alguém faria alguma manutenção nos modelos criados, pois isso implicaria em editar novamente o conteúdo dos *nós*.

Um dos usuários-autores não teve dificuldades em relação à manutenção da aplicação.

H.B.3. O sistema deve fornecer recursos que verificam a coerência das ligações

Quando um nó é modificado, as referências a este nó são atualizadas automaticamente?

Este item não foi avaliado devido ao sistema não permitir a alteração direta de uma categoria. Quando o usuário precisa fazer uma alteração, ele tem que apagar a categoria existente e inserir uma nova, conforme citado anteriormente. Ao apagá-la, seus relacionamentos são automaticamente apagados.

Quando um nó é excluído, o sistema remove automaticamente todas as referências a ele?

Sim.

A implementação desses recursos é importante para o autor?

Três usuários-autores opinaram que, caso um sistema permita a atualização de categorias, esse é um requisito importante somente se as aplicações que estão sendo construídas forem grandes ou complexas. Dois usuários-autores não souberam opinar.

H.B.4. O sistema deve fornecer recursos que promovam a propagação de modificações

As modificações realizadas em um nó são automaticamente atualizadas em todos os níveis em que ele aparece?

Este requisito não foi avaliado devido ao fato de o sistema não implementar estes recursos. Quando o usuário escolhe o mesmo nome para dois *nós* (categorias) diferentes, o sistema trata-os distintamente. Logo, as modificações realizadas em um não podem ser propagadas para o outro.

A implementação desses recursos é importante para o autor?

Três usuários-autores citaram a importância de o sistema suportar a utilização do mesmo *nó* mais de uma vez e não fazer distinção entre eles. Sugeriu-se a utilização de um *flag*, que indicaria que determinado *nó* está sendo utilizado mais de uma vez, e tudo o que fosse editado/modificado em um, por *default*, seria atualizado no outro.

H.B.5. O sistema deve fornecer recursos que facilitam a edição de nós

As informações relativas aos nós e contextos da aplicação (nome do nó, tipo de mídia relacionada à apresentação do nó, escolha do arquivo que estará associado ao nó) são fáceis de editar? Alguma dificuldade foi encontrada na realização desta atividade?

No geral, os usuários acharam fácil a edição das informações relativas aos *nós* e não tiveram dificuldades em relação a esta tarefa. Sugeriu-se, no entanto, que seja modificada a forma de seleção de origem e destino de relacionamentos porque, à medida que aumenta o número de categorias, torna-se cansativo selecioná-las.

H.B.6. O sistema deve fornecer recursos que permitam a edição de textos, sons e gráficos a partir do próprio sistema de autoria

O sistema de autoria permite a edição de textos, sons e gráficos externa ou internamente ao sistema?

O SASHE possui um editor de textos próprio. O sistema não permite a utilização direta (a partir do próprio sistema) de editores de sons e gráficos, porém, os autores concordam que deve ser possível utilizá-los a partir do sistema. Alegou-se que, devido à complexidade de desenvolvimento de um editor e tendo em vista a existência de diversos editores, seria suficiente que o SASHE permitisse o acesso a editores externos.

A implementação desses recursos é satisfatória às necessidades do usuário-autor?

Foram citadas algumas sugestões em relação ao editor de textos:

- 1)- Adotar o critério WYSIWYG (*what you see is what you get*) pois, durante a edição, o autor não visualiza o documento da mesma forma que ele será apresentado aos usuários. Ele tem disponível apenas *tags* de formatação e textos;
- 2)- Habilitar os *hints* dos botões;
- 3)- Permitir aumentar e diminuir o tamanho das letras do texto que é editado.

H.C.1. O sistema deve fornecer recursos que suportam o gerenciamento e armazenamento de versões das aplicações hipermídia

O sistema de autoria registra as diversas versões da aplicação?

O SASHE não suporta o gerenciamento e armazenamento das versões das aplicações.

É importante para o autor o suporte ao gerenciamento de versões?

Todos os usuários-autores acham importante a implementação desse requisito. Segundo eles, essa é uma funcionalidade importante para qualquer sistema, mas a implementação não requer rigidez.

H.C.2. O sistema deve fornecer recursos que apóiam o trabalho cooperativo

O sistema de autoria prevê o trabalho cooperativo dos autores?

O SASHE não suporta o trabalho cooperativo.

A implementação desses recursos é importante para o autor?

Para os usuários-autores, a implementação desses requisitos é importante, tendo-se em vista os benefícios que podem ser obtidos a partir do desenvolvimento cooperativo de aplicações.

H.D.2. O sistema deve fornecer recursos que possibilitam ao autor criar aplicações hipermídia de acordo com suas necessidades e preferências

A implementação e a interface da aplicação foram influenciadas pelas características do sistema de autoria?

As aplicações geradas por quatro usuários-autores não foram influenciadas pelas características do sistema. Somente alguns ajustes precisaram ser feitos no sentido de adequar as necessidades do autor à funcionalidade da ferramenta. Estes ajustes estavam relacionados a não permissão do sistema à expansão da categoria de *exemplos* e à quantidade excessiva de categorias que precisariam ser inseridas quando o usuário necessitou colocar vários exemplos para motivar um conceito, o que tornaria o diagrama sobrecarregado (optou-se por inserir os nós utilizando-se a janela de inserção de nós essenciais e complementares). Foi citado que, em determinados momentos, teve-se que mudar a forma de pensar para continuar o desenvolvimento da aplicação (o que o usuário achou normal), mas conseguiu-se alcançar o objetivo.

Há necessidade de se fazer alguma extensão ou adaptação no módulo de autoria de forma que este permita ao usuário-autor criar aplicações de acordo com suas necessidades e preferências?

Um dos usuários citou que gostaria de poder utilizar *applets* Java, porque as aplicações hipermídia não podem ser totalmente passivas, ou seja, devem ser interativas, como acontece em www.cut-the-knot.com, que oferece diversos problemas e curiosidades matemáticas e permite ao usuário interagir diretamente com figuras até que ele obtenha uma solução. O usuário gostaria, portanto, de poder utilizar recursos semelhantes.

H.D.3. O sistema deve fornecer recursos que detectam a presença de referências cegas e ligações incorretas

O sistema de autoria detecta referências cegas e ligações incorretas?

No caso do EHDT este requisito não foi avaliado porque a ferramenta somente permite que um relacionamento seja inserido quando o usuário seleciona uma origem e um destino (não permite referências cegas). Em relação às ligações incorretas, o sistema permite selecionar o relacionamento somente depois que forem escolhidas a origem e o destino, ou seja, o próprio sistema filtra os relacionamentos que podem ser escolhidos e apresenta-os ao autor. Naturalmente, ligações incorretas não são inseridas. No entanto, percebeu-se que, quando o usuário escolhe origem e destino iguais, o sistema habilita o botão *Ok*, o que não é um procedimento correto. Se este botão é pressionado, o usuário recebe a mensagem que o diagrama não pode ser mostrado, mas os dados são salvos em arquivos (o que é um *bug* do sistema).

A implementação desses recursos é importante para o autor?

Segundo os usuários-autores, caso o sistema permita referências cegas ou ligações incorretas, é importante que o sistema detecte esses problemas e comunique-os ao autor.

H.E.1. O sistema deve fornecer recursos que promovam auxílio ao usuário

O sistema de autoria possui informações explicativas sobre o processo de autoria no instante em que o usuário desejar?

O sistema não fornece informações explicativas sobre o processo de autoria. Esses recursos foram solicitados diversas vezes durante os experimentos. Os usuários acham muito importante a disponibilização dos mesmos. Sugeriu-se o *help* do *windows 2000* como modelo, pois este oferece

alguns recursos simples e interessantes, como a utilização de janelas flutuantes para definições breves, glossário e tópicos relacionados por assunto.

Os experimentos não foram prejudicados pela falta de um *help* porque foram realizados de forma supervisionada, conforme citado anteriormente.

H.E.2. O sistema deve fornecer recursos de tutorial para autoria

O sistema de autoria fornece um tutorial para o processo de autoria?

O SASHE não oferece um tutorial.

A implementação desses recursos é importante para o usuário-autor?

Todos os usuários-autores acharam importante a implementação desse recurso. Mencionou-se que, especialmente para usuários novatos, é importante que o sistema possua um tutorial que explique em detalhes como o sistema funciona, quais são seus objetivos, quais são os benefícios obtidos a partir de sua utilização e o que ele gera, explicando também como construir uma aplicação, pois os usuários têm diferentes níveis de habilidade e um tutorial auxiliaria a suprir as dificuldades de usuários menos experientes ou usuários que não sejam da área de Ciências de Computação. Citou-se como exemplo de ferramentas que oferecem bons tutoriais (e que poderiam ser utilizados como modelo) o *Autocad* (Autocad, 2000) e o *CorelDraw* (CorelDraw, 2000).

H.F.1. O sistema deve fornecer recursos que possibilitam o armazenamento de informações sobre os nós

O sistema de autoria armazena informações sobre os nós?

O sistema permite armazenar informações a respeito da função didática e dificuldade do nó. No entanto, não permite armazenar informações como data de criação, atualização, etc. Uma sugestão para implementação dessa funcionalidade no EHDT é a utilização de um arquivo de *log* que registre todas as modificações que forem realizadas, como foi implementado em <http://www.nondot.org/sabre/os/> e em <http://www.extremeprogramming.org/>, que oferecem a possibilidade de autores e leitores verem o que foi modificado em todas as atualizações e nas mais recentes atualizações da página, respectivamente.

Este recurso é importante para o autor?

Quatro usuários-autores acham importante o armazenamento de informações, principalmente se muitos usuários estiverem trabalhando na mesma modelagem. Outra justificativa se referiu à importância de armazenar dados relativos a quem alterou ou teve a responsabilidade de alterar uma entidade, porque houve necessidade de alterar e como foram afetadas as demais entidades após a alteração, como forma de facilitar o trabalho em grupo e torná-lo organizado. Foi citado como sendo mais importante o armazenamento da data de criação do nó e da data de última alteração.

Um dos usuários não acha interessante o armazenamento de dados devido ao fato de aumentar o trabalho de usuários-autores. Segundo ele, isso aumentaria o número de campos a serem preenchidos.

H.F.2. O sistema deve fornecer recursos de documentação

O sistema de autoria possui documentação?

O SASHE possui documentação para a versão 1.0 (Feltrim & Fortes, 1999). Essa documentação não engloba o EHDT.

Ela é satisfatória às necessidades de usuários novatos e experientes? Quais seriam possíveis melhorias da documentação existente?

Estes itens não foram avaliados porque o EHDT não possui documentação. Pôde-se observar que todos os usuários-autores requisitaram este recurso e expressaram a sua relevância no contexto de qualquer sistema, ou seja, ela serve principalmente como auxílio para novos usuários e como referência técnica sobre a implementação do sistema para futuras manutenções.

3.2 Requisitos para Sistemas de Autoria Hipermídia Educacional (E)

Os requisitos pertencentes a esta classe referem-se aos recursos que devem ser oferecidos por um sistema de autoria para possibilitar a construção de aplicações hipermídia pertencentes ao domínio específico de ensino. Os resultados obtidos a partir da avaliação do SASHE referentes a esses requisitos são apresentados a seguir.

E.A.1. O sistema deve fornecer recursos que proporcionam liberdade de projeto dos aspectos navegacionais da aplicação

O professor possui liberdade para projetar os aspectos navegacionais da aplicação em relação à modelagem do domínio de conhecimento?

Os usuários-autores preferiram gerar automaticamente o modelo navegacional de contextos. Todos eles acharam a inserção de dados necessários para a geração automática fácil e simples. Algumas dúvidas estavam relacionadas à nomenclatura utilizada pelo sistema e às etapas a serem seguidas, devido à inexperiência dos usuários com o mesmo.

Ao tentar modificar o modelo navegacional gerado, sentiu-se a necessidade de o sistema indicar de forma mais clara que, ao inserir um novo elemento (contexto navegacional ou nó), o autor pode fazer um mapeamento desse novo elemento com algum elemento do modelo conceitual.

Alguma dificuldade foi encontrada na realização dessa atividade?

As dificuldades estavam relacionadas principalmente à interface. Foi sugerido que o sistema indique que o usuário deve clicar o botão de *Opções*, para setar as categorias e os relacionamentos para os quais ele deseja criar o modelo de navegação, o que seria bastante útil para usuários novatos. Além disso, foi sugerido que ficassem setadas todas as categorias e relacionamentos na janela *Opções*, de forma a diminuir o trabalho do usuário.

E.A.2. O sistema deve fornecer recursos que permitam o estabelecimento do tipo de controle sobre a navegação

O professor pode estabelecer o tipo de controle sobre a navegação?

Sim. Durante a criação de roteiros, o usuário-autor estabelece o grau de liberdade de cada nó que é inserido.

A implementação destes recursos é importante para o professor?

Para dois usuários-autores a navegação não deve ser controlada, ou seja, para eles, em um ambiente de aprendizado o aprendiz tem direito de explorar o assunto pelo qual surgiu interesse. Deve-se estabelecer um roteiro para evitar que o usuário navegue em ordem não desejada, mas que serviria apenas como um esquema de navegação preferencial. Esse roteiro permitiria ao aprendiz investigar o material de acordo com seu interesse, como acontece em <http://www.extremeprogramming.org/>, que oferece botões (*xp*) que compõem o roteiro de navegação sugerido e *links* através dos quais o usuário pode navegar. Em contrapartida, um desses usuários citou que pode ser importante em algum momento determinar um caminho para que o aprendiz veja um determinado assunto de uma forma, e então, o controle sobre a navegação torna-se importante. Ambos alegaram também que

essa questão é mais complexa que apenas responder sim ou não, porque deve envolver estudo e conhecimento pedagógico.

Dois usuários-autores foram favoráveis ao controle sobre a navegação porque, para eles, é uma tarefa difícil fazer com que o aprendiz concentre-se em algo delimitado. Isso se tornaria mais difícil se o aprendiz tivesse a chance de visualizar tudo.

Um usuário não soube responder à questão devido à falta de conhecimento de pesquisas nessa área.

A forma segundo a qual o usuário-autor estabelece o tipo de controle sobre a navegação no SASHE é satisfatória?

Foram citadas as seguintes restrições em relação à implementação dessa funcionalidade:

1)- Na janela *Liberdade de Navegação*, o autor tem a idéia de que a liberdade da navegação estará restrita a nós precedentes ao nó atual (que será inserido no roteiro) mas, na realidade, estabelece-se a liberdade de navegação em relação a todo o contexto, a toda a aplicação ou ao nó atual somente.

2)- A mesma janela não oferece as opções de *Cancelar* ou *Fechar*, ou seja, o usuário não tem a chance de desistir de inserir um nó no roteiro enquanto utiliza esta janela. Para isso ele deverá inserir o nó e depois apagá-lo. Apesar de o usuário perceber que essa janela foi projetada dessa forma devido à obrigatoriedade de escolha da liberdade de navegação, citou-se que esse não é o padrão das janelas do ambiente *windows*.

E.B.1. O sistema deve fornecer recursos que permitam fixação de material específico

O sistema de autoria permite ao usuário-autor especificar material para estudantes ou grupos de estudantes?

O sistema não oferece esta funcionalidade.

A implementação desses recursos é importante para o professor?

Os usuários-autores não consideram a implementação desse requisito importante.

E.B.2. O sistema deve fornecer recursos que possibilitam inclusão de material essencial e complementar

O sistema de autoria permite a inclusão de material essencial e complementar?

Sim.

A implementação desses recursos é importante para o professor?

Os usuários-autores consideram essa funcionalidade importante, porque permite ao autor fazer comentários adicionais sobre o assunto que está sendo ensinado, colocar material relacionado ao

tema mas que não seja primordial, mostrar o mesmo assunto sob outra perspectiva, sugerir outras referências bibliográficas, enfim, enriquecer o material básico.

É fácil fazer a inclusão de material essencial e complementar? Alguma dificuldade foi encontrada na realização dessa atividade?

O sistema permite facilmente fazer a distinção de material essencial e complementar porque o autor precisa somente selecionar (clicar) os *nós* que serão complementares. No entanto, algumas restrições relacionadas a essa funcionalidade foram citadas:

- 1)- A distinção de material não é intuitiva e a ferramenta não indica como o usuário deve proceder para realizar a tarefa;
- 2)- O EHDT permite a distinção de material mas, ao gerar a aplicação para o domínio do SASHE, todos os nós (essenciais e complementares) são tratados igualmente, ou seja, a distinção de material é feita somente em nível de modelagem. O usuário gostaria que o SASHE indicasse se determinado *nó* foi modelado como essencial ou complementar;
- 3)- Ao inserir um *nó* complementar, utilizando-se a janela de inserção de nós essenciais e complementares, o usuário gostaria que o EHDT explicitasse-o no diagrama.

E.C.1. O sistema deve fornecer recursos que proporcionam visualização da estrutura da aplicação hipermídia

O SASHE fornece recursos para visualização da estrutura da aplicação?

Sim. O usuário pode modelar primeiramente a aplicação utilizando o EHDT e criar a aplicação no SASHE, que permitirá a visualização da estrutura da aplicação.

A implementação desses recursos é satisfatória?

Todos os usuários-autores acharam muito bons os recursos disponíveis para visualização da estrutura do SASHE. A interface, por ser bastante familiar (semelhante à interface do *Windows Explorer*), é totalmente satisfatória.

E.C.2. O sistema deve fornecer recursos que possibilitam ao autor qualificar os elementos (componentes) do material

Foram definidos atributos para qualificação dos elementos do material?

O SASHE permite que o professor qualifique os elementos da aplicação de acordo com a *função didática* (motivação, introdução, definição, exemplo, exercício, resumo) e com a *dificuldade* (baixa, regular, alta).

Em relação a qualidade/quantidade, esses atributos são satisfatórios? Quais seriam as possíveis extensões ou adaptações para esse conjunto de atributos?

Em relação aos atributos de função didática:

Três usuários-autores acharam as opções de atributos oferecidas pelo sistema satisfatórias e não sugeriram novos atributos.

Os demais usuários-autores citaram a falta de um atributo que indicasse um experimento ou uma prática, que seriam especialmente importantes para disciplinas como Química e Física. Outra extensão sugerida seria a inclusão de um atributo que indicasse uma aplicação prática de um assunto.

Em relação aos atributos de dificuldade:

Três usuários-autores acharam satisfatórios e pertinentes os atributos relacionados à dificuldade. No entanto, dois usuários-autores citaram a questão da subjetividade na escolha de um dos atributos. Um assunto pode ser fácil para um leitor, mas difícil para outro; ou pode ser fácil para o autor da aplicação (que já estudou sobre o tema) e difícil para aqueles que estarão estudando o assunto pela primeira vez. Além disso, a transição de um assunto considerado difícil para um assunto considerado fácil pode acontecer muito rapidamente, ou seja, a medida que um autor ou leitor se interessa por um assunto e começa a pesquisar sobre ele, os conceitos vão se tornando mais fáceis.

3.3 Requisitos Específicos do Projeto SASHE (S)

S.1. O sistema deve fornecer recursos que possibilitam a contextualização de informações e o aninhamento de contextos

Os recursos de contextualização de informações e aninhamento de contextos foram implementados no SASHE?

Sim.

A implementação desses recursos é importante para o autor?

Todos os usuários-autores acharam importante a contextualização e o aninhamento das informações, pois, segundo eles, o material pode ser facilmente organizado e estruturado. Dois usuários-autores citaram também que a importância em contextualizar e aninhar informações está diretamente relacionada ao controle sobre a navegação, porque o autor pode decidir facilmente como será o acesso do leitor ao material, baseando-se no conteúdo dos *nós* pertencentes aos contextos.

A implementação desses recursos atende às necessidades dos usuários?

A implementação dessa funcionalidade foi considerada satisfatória pelos usuários-autores.

S.2. O sistema deve fornecer recursos que possibilitem ao usuário-autor gerar diferentes modelos de navegação baseados no mesmo modelo conceitual

Utilizando-se o modelo gerado na fase de modelagem conceitual hierárquica, o usuário pode construir diferentes modelos navegacionais?

Sim.

A implementação desses recursos é importante para o autor?

Dois usuários-autores acharam que esta é uma questão difícil de responder, alegando que não conhecem suficientemente o assunto. Um deles mencionou que a geração de diferentes modelos navegacionais e a criação de roteiros apresenta a mesma funcionalidade, o que dispensaria esta etapa de geração de vários modelos navegacionais, pois podem ser gerados vários roteiros.

Dois usuários-autores acharam essa funcionalidade útil. Eles concordam que a possibilidade de criar diferentes aplicações de acordo com diferentes classes de usuários e de acordo com as necessidades destas classes é bastante interessante.

Um dos usuários foi indiferente à geração de mais de um modelo navegacional para um modelo conceitual. Ele acha difícil que alguém queira gerar um modelo navegacional apenas para uma parte do modelo conceitual, se ele pode gerar para todo o modelo. No entanto, ele também não eliminaria essa funcionalidade.

Os modelos gerados são satisfatórios às necessidades dos usuários?

Sim.

Quais seriam possíveis extensões ou adaptações para os recursos de geração de modelos navegacionais oferecidos?

Os usuários-autores observaram que, apesar de o sistema permitir a geração de diferentes modelos navegacionais, ao tentarem criar a aplicação (no contexto do SASHE) apenas para o modelo navegacional atual (o que está selecionado), o sistema gera a aplicação para todos os modelos, ou seja, o EHDT permite criar o modelo navegacional de apenas uma parte do modelo conceitual, mas o SASHE não cria a aplicação somente dessa parte. Segundo os usuários, essa seria uma importante adaptação a ser feita no sistema.

S.3. O sistema deve fornecer recursos que proporcionam replicação (*cloning*) ao nível de instância

Os recursos oferecidos pelo sistema para realização dessa tarefa são suficientes? Quais seriam possíveis extensões ou adaptações para os recursos disponíveis?

O EHDT permite que diversos modelos navegacionais sejam criados a partir de um mesmo modelo conceitual hierárquico. O módulo de importação de hiperdocumentos *web* não está disponível na versão 2.0 e o sistema não permite que sejam criadas diferentes aplicações modificando-se apenas o roteiro de navegação.

Segundo os usuários-autores, os recursos oferecidos atualmente não são satisfatórios. Foi sugerido que sejam disponibilizadas as funcionalidades citadas acima e que o sistema possibilite a geração de um modelo navegacional a partir de outro modelo navegacional.

3.4 Requisitos que não Foram Avaliados por Usuários-Autores

Conforme mencionado na seção 6.1, os requisitos H.D.1 e S.4 não foram avaliados durante a execução dos experimentos com usuários-autores. Essa seção apresenta informações referentes a estes requisitos.

H.D.1. O sistema deve fornecer recursos que promovam redução de custos (tempo) envolvidos na criação de aplicações

Comparando-se o processo de modelagem da aplicação à estruturação direta da aplicação, o usuário-autor pode perceber algum ganho ao utilizar a primeira abordagem?

Este requisito não pôde ser avaliado porque os usuários não desenvolveram aplicações adotando a abordagem da estruturação direta (sem utilizar a ferramenta de modelagem). No entanto, pode-se fornecer o tempo gasto no desenvolvimento das aplicações utilizando-se a ferramenta de modelagem para posterior geração da aplicação. É importante observar, contudo, que os dados se referem ao tempo total gasto em cada experimento, ou seja, considerou-se o tempo envolvido no treinamento³ oferecido e na discussão a respeito da implementação dos requisitos. Logo, esses dados devem ser cuidadosamente analisados caso sejam utilizados em futuras comparações relacionadas ao custo de desenvolvimento.

Usuário 1 - 14h 05m divididas em quatro sessões (Avaliação Piloto)

Usuário 2 - 5h 10m divididas em duas sessões

Usuário 3 - 5h 30m divididas em duas sessões

Usuário 4 - 4h 45m divididas em duas sessões

Usuário 5 - 4h 30m divididas em duas sessões

S.4. O sistema deve fornecer recursos que possibilitam que aplicações sobre quaisquer assuntos possam ser geradas

O SASHE fornece recursos para construção de aplicações de outros domínios?

Foram desenvolvidos alguns trabalhos relacionados à extensão dos recursos oferecidos pelo SASHE, de forma a tornar possível a construção de aplicações pertencentes ao domínio de Engenharia de Software:

1)- Adequação do SASHE ao domínio da documentação de engenharia reversa de software (Feltrim, 1999). Neste trabalho a autora propõe mudanças nas funções de navegação, nos atributos de qualificação de *nós* terminais e na nomenclatura utilizada na interface. Além

disso, percebeu-se que recursos como criação de roteiros e controle sobre a navegação não são relevantes para o novo domínio.

2)- Adaptação do SASHE à tarefa de documentação de software (Silva, 2000). Este trabalho resultou no desenvolvimento do SASHDoc - *Sistema de Autoria e Suporte Hipermídia para Documentação de Software*, em que foram reutilizados diversos componentes do SASHE e definidas novas funções para atenderem ao domínio de documentação de software.

Quais foram as principais limitações encontradas quando aplicações de outros domínios foram modeladas?

Este item não pôde ser avaliado porque, para o desenvolvimento deste trabalho de avaliação do SASHE, foram consideradas apenas aplicações relacionadas ao domínio de ensino.

Algumas informações adicionais foram obtidas durante a realização dos experimentos. Essas informações referem-se a aspectos relativos à usabilidade do SASHE e do EHDT, sugestões de melhoria para o sistema, problemas relacionados à sua execução e aos seus aspectos positivos. Os avaliadores foram explicitamente questionados em relação aos aspectos relacionados à usabilidade. Os demais itens referem-se a comentários dos usuários sobre o sistema durante a execução dos experimentos.

4. Aspectos relativos à usabilidade do SASHE

As questões relacionadas à usabilidade da ferramenta enfatizaram a facilidade de aprendizado do sistema, a familiaridade, a confortabilidade, a clareza e intuitividade do sistema e as possibilidades de customização do ambiente de trabalho. Os resultados obtidos são apresentados nas subseções seguintes.

4.1 Facilidade de Aprendizado

No contexto da avaliação de sistemas de autoria hipermídia, a *facilidade de aprendizado* pode ser avaliada investigando se o sistema permite aos autores construir soluções rápida e facilmente de forma a refletirem seus objetivos (Balagopalan & Macknight, 1989; Campos, 1994; Kearsley, 1982; Nanard & Nanard, 1995; Papatheodorou, 1998). Os usuários puderam expressar suas opiniões a respeito da facilidade de aprendizado do SASHE ao responderem às seguintes questões:

É fácil aprender a usar o sistema para fins de autoria?

Três usuários acharam fácil aprender a usar o sistema. Um deles citou que o fato do experimento ser realizado de forma supervisionada pode ter influenciado sua resposta e que, apesar de ter achado fácil, devem ser considerados também os comandos e funções não intuitivos do SASHE.

³ O tempo médio de duração dos experimentos foi de quarenta minutos.

Um dos usuários classifica a facilidade de aprendizado como sendo média, pois algumas ações que foram executadas para realizar determinadas tarefas na primeira sessão não foram facilmente reconhecidas na segunda sessão. O usuário alegou, inclusive, que não conseguiria executar novamente algumas tarefas que já havia executado anteriormente.

O outro usuário não achou fácil aprender a usar o sistema.

O sistema de autoria permite a fácil construção de aplicações?

Todos os usuários concordaram que, após a etapa de aprendizado de utilização do sistema, foi fácil construir a aplicação. Alguns comentários foram feitos:

1)- "Não é difícil gerar a aplicação, mas é trabalhoso. Em especial, o preenchimento dos dados de cada nó foi cansativo. Mas devemos considerar que a atividade de gerar uma aplicação hipermídia é algo trabalhoso. Então a ferramenta deve ajudar no sentido de facilitar esse processo. Alguns problemas de implementação, por exemplo, travamento do sistema, problemas na geração automática de elos, dentre outros, também foram empecilhos";

2)- "O processo é muito bom. A idéia do sistema é muito boa e leva a uma implementação rápida".

Em relação à facilidade de uso, é vantajoso utilizar um sistema de autoria em relação à utilização de uma linguagem de programação (por exemplo, Java) para o desenvolvimento de uma aplicação hipermídia?

Todos os usuários preferem utilizar um sistema de autoria a utilizar uma linguagem de programação para construir suas aplicações. Segundo eles, em relação ao SASHE, a grande vantagem é que ele suporta a criação da aplicação em alto nível (através da modelagem) e a parte mais trabalhosa (programar, gerar a aplicação) é executada automaticamente. Alguns comentários se referiram às facilidades oferecidas pelo sistema em relação à modelagem da navegação: o sistema auxilia muito nas decisões sobre navegação, porque o autor tem uma visão completa da estrutura da aplicação, o que não acontece quando o usuário precisa programar utilizando uma linguagem de programação. Além disso, destacou-se que é vantajoso utilizar um sistema de autoria porque o tempo e o conhecimento necessários para gerar uma aplicação são consideravelmente menores se compararmos com o desenvolvimento através da utilização de uma linguagem de programação, o que é particularmente importante para autores de áreas diferentes de Ciências de Computação, que não precisarão aprender a programar. Apesar de os usuários se mostrarem favoráveis à utilização do SASHE em relação a uma linguagem de programação, foram enfatizados alguns problemas que devem ser solucionados a fim de favorecer a utilização do sistema:

- 1)- O sistema gera aplicações que podem ser visualizadas apenas no *browser* do SASHE, ou seja, não gera aplicações que possam ser disponibilizadas na Internet (extensão html), que é uma tendência atual;
- 2)- O sistema apaga o conteúdo dos arquivos. Isso acontece quando o usuário já possui um protótipo da aplicação, resolve modificar algum aspecto da modelagem e gera novamente o protótipo.

4.2 Familiaridade

A *familiaridade de um sistema* refere-se a atributos do software que evidenciam o esforço do usuário para aplicar a determinado software os conhecimentos e experiências obtidas com a utilização de outros softwares (Campos, 1994; Dix et al., 1998). Os usuários puderam expressar suas opiniões a respeito da familiaridade do SASHE ao responderem à seguinte questão:

O sistema de autoria é familiar durante todo o processo de desenvolvimento de aplicações?

Os usuários não acharam o sistema familiar. Segundo eles, o sistema usa uma linguagem própria e desconhecida. Alguns problemas foram destacados:

- 1)- O EHDT não permite que vários modelos sejam criados, ou seja, suporta apenas uma modelagem;
- 2)- As funções de *Salvar*, *Abrir* e *Fechar* não foram implementadas no EHDT. Destacou-se o fato de o autor não utilizar (como é o usual) a função salvar. Isso gera insegurança em relação ao sistema, pois não se sabe quando exatamente os dados são salvos em disco (a avaliação piloto indicou que os dados são salvos apenas quando a execução do sistema é encerrada).
- 3)- Geralmente, quando o usuário começa a utilizar um sistema, ele precisa criar um projeto ou abrir um projeto existente (como acontece, por exemplo, na utilização de um ambiente de programação). Em relação ao SASHE, o usuário começa a utilizar diretamente a ferramenta de modelagem sem criar uma aplicação, ou seja, o sistema fornece um modelo inicial que deverá ser expandido, e o usuário não precisa sequer dar um nome a esse modelo. Comparou-se a utilização da ferramenta de compressão de dados *Winzip* com o EHDT: a primeira oferece diversos procedimentos que podem ser executados e que o usuário deve selecionar (*check out*, *extract*, etc); a segunda não permite ao autor fazer escolhas sobre os procedimentos a serem executados.

4.3 Confortabilidade

A *confortabilidade de um sistema* refere-se a atributos do software que evidenciam sua capacidade de ser confortável ao usuário (Campos, 1994). Os usuários puderam expressar suas opiniões a respeito da confortabilidade do SASHE ao responderem à seguinte questão:

O sistema de autoria é confortável ao usuário?

Em relação a este item, citou-se como aspecto positivo do sistema a geração automática do modelo navegacional de contextos e da aplicação final. Citou-se como aspecto negativo a falta de padronização. Às vezes o usuário precisa utilizar o botão direito do mouse, às vezes o esquerdo, às vezes o esquerdo e o direito e às vezes duplo clique. Além disso, os problemas de implementação (*bugs* do sistema) prejudicam o aspecto de confortabilidade.

4.4 Clareza e Intuitividade

A *clareza e intuitividade de um sistema* referem-se a atributos do software que evidenciam o esforço do usuário para lembrar-se dos comandos (Campos, 1994). Os usuários puderam expressar suas opiniões a respeito da clareza e intuitividade do sistema ao responderem à seguinte questão:

Os comandos fornecidos pelo sistema de autoria são auto-explicativos e intuitivos?

Todos os usuários responderam que o sistema possui alguns comandos que não são auto-explicativos. Foram citados como exemplos:

- 1)- Na janela de criação de elos do SASHE, o usuário precisa escolher o nó origem, clicar com o botão direito do mouse sobre o nó (para que as âncoras pertencentes àquele nó sejam apresentadas) e então escolher o nó destino. Na mesma janela, o sistema solicita a escolha de uma âncora para o nó destino;
- 2)- A forma como são apresentados os nós pertencentes ao roteiro na janela *Roteiros* não é intuitiva;
- 3)- Alguns termos utilizados não são auto-explicativos, por exemplo, *Gerar para:* e *No contexto:* na janela *Gerar Diagrama Navegacional*;
- 4)- Na criação do modelo navegacional, quando o autor deseja criar um modelo para o primeiro nível, deve-se deixar a opção do *combobox* relacionada ao campo *No contexto:* em branco.

5)- O EHDT cria o arquivo mapp.hip no diretório C:\Sashe32\data, que é aplicação, mas não indica isso ao usuário.

Alguma melhoria pode ser feita de forma a torná-los mais claros e intuitivos?

As melhorias sugeridas referem-se basicamente à utilização de termos mais claros e significativos e à reestruturação das janelas de criação de elos e roteiros.

4.5 Customização

A *customização de um sistema* refere-se a atributos do software que evidenciam a possibilidade do usuário personalizar a interface do ambiente de trabalho através, por exemplo, da escolha da posição das janelas na tela, da possibilidade de criar ícones e definir tipos e tamanhos de letras (Campos, 1994; Dix et al., 1998). Os usuários puderam expressar suas opiniões a respeito das possibilidades de customização do sistema ao responderem às seguintes questões:

É importante para o autor poder customizar o ambiente de trabalho?

Todos os usuários acham importante que o sistema seja flexível em relação à customização do ambiente de trabalho, porque é importante que o usuário possa tomar decisões em relação à interface de acordo com suas preferências, experiências de utilização de sistemas e aptidões técnicas. Dessa forma, segundo os usuários, a possibilidade de customização favorece a utilização do sistema.

O ambiente de trabalho do SASHE pode ser facilmente customizado?

O sistema não oferece essa possibilidade. Ressaltou-se que este não é um problema que deve ser resolvido primordialmente.

Quais seriam possíveis extensões do conjunto de possibilidades de customização que o módulo de autoria oferece ao usuário?

Um dos usuários citou a importância da personalização realizada pelo próprio autor de aplicações, ao interagir com o sistema (adaptabilidade) e das modificações automáticas que o sistema faz considerando-se o conhecimento que tem dos usuários (adaptividade).

As sugestões dos demais autores foram relacionadas apenas à adaptabilidade do SASHE e do EHDT. Eles gostariam que o sistema oferecesse o recurso de personalização da barra de tarefas já existente. Dessa forma, poderiam ser utilizados ícones para usuários experientes e opções textuais

do menu para usuários novatos. Requisitou-se também que a ferramenta permita o redimensionamento da janela de edição de diagramas.

As seções seguintes referem-se a comentários e sugestões dos usuários-autores durante a execução dos experimentos.

5. Problemas Relacionados à Interface do Sistema

Alguns tópicos relacionados à interface do SASHE e do EHDT foram especialmente discutidos durante a execução dos experimentos. Estes tópicos e seus problemas relacionados são mencionados a seguir.

5.1 Janelas do sistema

No geral, os nomes dados às janelas do sistema são pouco significativos. Os termos utilizados não fornecem informações sobre o objetivo das janelas. Podem-se citar os exemplos:

- 1)- A janela de edição de roteiros no EHDT e a janela de edição de propriedades dos nós do modelo navegacional de contextos possuem o nome *Editar*, que não é auto-explicativo;
- 2)- A janela de inserção de informações complementares e essenciais recebe, a cada execução, o nome de um dos *nós* essenciais da aplicação.

Sugeriu-se que os nomes de todas as janelas do sistema sejam reprojatados.

Outras características relacionadas às janelas do EHDT foram discutidas:

- 1)- Observou-se que, quando um diagrama torna-se grande, o sistema não permite ao autor visualizá-lo completamente, e não fornece também barras de rolagem. Logo, algumas categorias do modelo ficam inacessíveis.
- 2)- As janelas do EHDT são modais preemptivas (enquanto uma janela está sendo mostrada, nenhuma ação do usuário pode ser executada na região exterior a ela), o que gerou descontentamento enquanto o usuário interagiu simultaneamente na ferramenta de modelagem e no SASHE.

5.2 Mensagens do sistema

Em relação ao envio de mensagens, notou-se que o sistema utiliza esse recurso de forma excessiva. Em determinadas situações, é suficiente que o usuário perceba que as tarefas de fato foram realizadas. Isso acontece, por exemplo, na criação de elos, no contexto do SASHE. Quando o usuário cria novo elo, o sistema envia mensagem comunicando que a tarefa foi realizada com sucesso. No entanto, o usuário pode perceber que isso aconteceu quando o novo elo é acrescentado à lista de elos existentes. Para a exclusão de componentes da aplicação é importante a confirmação do usuário, como ocorre no sistema.

Por outro lado, o sistema não envia mensagem de confirmação de salvamento dos dados em arquivos quando o sistema é encerrado, o que é muito importante, segundo os usuários. Além disso, ao gerar a aplicação, o sistema não avisa aos usuários que o arquivo da aplicação (mapp.hip) será sobrescrito.

5.3 Menus do sistema

A funcionalidade de algumas opções disponíveis nos menus não foi implementada. No EHDT isso acontece no menu Arquivo, opções Novo, Abrir, Salvar e *Salvar como* e no menu ?, opções Assistente e Conteúdo. No SASHE isso acontece no menu Visualizar, opções *Browser Gráfico* e *Browser Estrutural*; Ferramentas, opções *Importação*, *Editor HTML* e *Opções* e no menu Ajuda, opção *Conteúdo*. Sugeriu-se a reformulação desses menus caso a funcionalidade de suas opções não seja implementada.

5.4 Outras observações relacionadas à interface do usuário

Outros problemas relacionados à interface do EHDT foram citados pelos usuários:

- 1)- Os nomes dos nós não aparecem completos nas categorias;
- 2)- O sistema não disponibiliza botão para escolha do diretório dos arquivos de conteúdo na janela de edição de propriedades dos nós;
- 3)- O menu *pop up* utiliza o mesmo termo (*Explodir*) no Diagrama Conceitual Hierárquico e no Diagrama Navegacional de Contextos para funcionalidades diferentes;

4)- O campo *Gerar para:* na janela *Gerar Diagrama Navegacional* não é editável. Como consequência disto, todos os modelos navegacionais gerados para o mesmo nível do modelo conceitual possuirão o mesmo nome (padrão) indicado pelo sistema, o que dificulta a seleção de uma determinada modelagem.

Em relação ao SASHE, alguns problemas foram detectados:

- 1)- a visualização do conteúdo dos nós a partir do SASHE deveria ser facilitada, permitindo, por exemplo, que o conteúdo pudesse ser visualizado apenas clicando sobre o *nó*;
- 2)- *Hints* estão disponíveis apenas em algumas janelas.

6. Sugestões de Melhoria do Sistema

Além das sugestões citadas em (Paiva, 2001), outras observações dos usuários podem ser destacadas:

- 1)- Ao iniciar a execução do EHDT, o usuário gostaria de fornecer o nome da aplicação, e este nome estaria automaticamente ligado à categoria do primeiro nível da modelagem;
- 2)- O sistema deveria fornecer o recurso de impressão da modelagem e da aplicação final;
- 3)- Seria muito importante que o sistema fornecesse heurísticas para construção dos modelos. Segundo o usuário, o sistema oferece os tipos de “blocos” e como eles podem ser ligados, mas não oferece um conjunto de regras de como fazer o modelo, ou seja, de como representar o conhecimento utilizando estes “blocos”. Heurísticas auxiliariam no sentido de tornar o processo de criação da aplicação mais sistemático;
- 4)- A interface do sistema deveria ser reprojetaada, oferecendo, por exemplo, um fichário com as principais funcionalidades e também barras de tarefas que possam ser personalizadas;
- 5)- A função *undo* do sistema deveria estar disponível;
- 6)- Seria interessante que o EHDT permitisse criar um nível entre dois níveis existentes, pois, segundo o usuário, a hierarquia vai surgindo ao longo do desenvolvimento da modelagem, e o sistema pressupõe que quando o usuário começa uma nova modelagem ele já tem uma visão do modelo completo.

7. Problemas Relacionados à Execução do Sistema

Alguns problemas foram detectados durante a execução dos experimentos, tendo sido relacionados a seguir:

- 1)- Quando o sistema trava, toda a modelagem é perdida. Isso acontece porque não existe a possibilidade do próprio autor salvar seus modelos. Dessa forma, a atividade de modelagem tem que ser reiniciada;
- 2)- Alguns relacionamentos são inseridos pelos autores no diagrama conceitual hierárquico mas não são mostrados na tela, o que impossibilita a continuação da atividade de modelagem, pois o usuário não consegue visualizar o diagrama completo. Isto indica que há inconsistências entre o que é mostrado na tela e o que os arquivos de dados contém;
- 3)- Alguns elos que já haviam sido inseridos desapareceram quando novos componentes do diagrama foram inseridos;
- 4)- A opção *Apresentar todos os arquivos do contexto simultaneamente* é apresentado em duas janelas diferentes do EHDT, mas a maneira segundo a qual essa opção é habilitada não é consistente;
- 5)- Na criação de aplicações utilizando-se o EHDT ou o SASHE, os arquivos têm que estar, necessariamente, no diretório C:\SASHE32\data, o que obriga aos usuários a copiarem arquivos de outros diretórios ou dispositivos (CD, disquete, etc.);
- 6)- O usuário não consegue navegar satisfatoriamente pelos protótipos porque o sistema permite a navegação por apenas uma das âncoras, ou seja, quando o usuário visualiza o primeiro nó do hiperdocumento e clica em uma âncora, o nó que está ligado a essa âncora é apresentado. No entanto, quando o usuário clica em uma âncora deste segundo nó, o sistema trava.

8. Aspectos Positivos do Sistema

São mencionadas a seguir as vantagens de utilização do SASHE ressaltadas pelos usuários durante a execução dos experimentos:

- 1)- A utilidade da ferramenta em relação à facilidade de criação de aplicações hipermídia e à possibilidade de conjugação de sons, textos, figuras e áudio;
- 2)- A possibilidade de começar o desenvolvimento de uma aplicação a partir da modelagem permite ao usuário estruturar as informações em categorias de conhecimento e estabelecer relacionamentos entre estas categorias. Em seguida o especialista pode decidir sobre como mostrar

esse conhecimento aos leitores, utilizando recursos como criação de diferentes modelos navegacionais e diferentes roteiros, de acordo com a necessidade do público alvo;

3)- A criação de protótipos da aplicação antes da geração da aplicação final facilita a avaliação e o refinamento da aplicação;

4)- Na edição dos diagramas conceitual hierárquico e navegacional de contextos, o próprio sistema distribui as categorias (nós) e os relacionamentos (elos) na tela. Em outras ferramentas *CASE* (*Computer-Aided Software Engineering*) o usuário tem que arrastar os componentes de forma a tornar o grafo mais simples, o que pode ser uma tarefa muito trabalhosa;

5)- O sistema cria automaticamente o modelo de navegação a partir da modelagem do conhecimento. Para o usuário é muito mais fácil trabalhar com os aspectos do domínio que ele conhece ou é especialista do que trabalhar com aspectos de navegação (nós e elos);

A seção seguinte apresenta algumas pendências de implementação relacionadas ao SASHE, registradas com o objetivo de motivar trabalhos futuros e, conseqüentemente, contribuir para o aprimoramento do sistema.

9. Pendências de Implementação relacionadas ao SASHE

Durante o processo de avaliação do SASHE foram observadas as seguintes pendências de implementação:

Em relação ao SASHE:

- a) Implementação dos *nós* de Ajuda e Exercícios;
- b) Implementação e disponibilização do Browser Estrutural;
- c) Adaptações funcionais e disponibilização do Browser Gráfico e da Ferramenta Html2Hip;
- d) Flexibilização do armazenamento dos arquivos (textos, sons, vídeos, imagens) pertencentes à hiperbase. Atualmente, estes arquivos devem ser obrigatoriamente armazenados em C:\SASHE32\data\textolsnd\videolimg, conforme citado anteriormente;
- e) Permissão de criação de múltiplos roteiros.

Em relação ao EHDT

- f) Implementação da funcionalidade de criação automática de elos (ao gerar a aplicação para o domínio do SASHE) que foram previamente criados na etapa de modelagem da aplicação;
- g) Suporte à criação de roteiros.

10. Conclusões

Durante aproximadamente cinco anos, alguns trabalhos foram realizados no contexto do SASHE. Em particular, o trabalho de avaliação do sistema foi motivado pela necessidade de se obter informações que indicassem as condições atuais de sua implementação e que servissem como apoio para decisões em relação à continuidade do projeto SASHE.

Este relatório técnico registrou, portanto, informações relativas à avaliação a qual foi submetido o sistema SASHE.

A abordagem de avaliação utilizada mostrou-se satisfatória. A utilização do método experimental trouxe grandes benefícios à avaliação, pois os resultados obtidos refletem a opinião de avaliadores pertencentes à principal classe de usuários do sistema. Logo, as adaptações a serem realizadas beneficiarão, especialmente, a esta classe de usuários.

Em relação ao sistema avaliado, pode-se dizer que os resultados obtidos a partir da realização desta avaliação poderão auxiliar futuras decisões em relação à evolução do sistema.

Referências Bibliográficas

- (Balagopalan & Macknight, 1989) Balagopalan, S.; Macknight, C.B. An Evaluation Tool for Measuring Authoring System Performance. *Communications of the ACM*, v.32, n.10, p.1231-1236.
- (Britain, 1999) Britain, S. A Framework for Pedagogical Evaluation of Virtual Learning Environments. *Relatório Técnico n.41*, Bangor University Centre for Learning Technology, Outubro, 1999. Disponível na Internet <http://www.leeds.ac.uk/educol/documents/00001237.htm>
- (Campos, 1994) Campos, F. C. A. *Hipermídia na Educação: Paradigmas e Avaliação da Qualidade*. Rio de Janeiro, 1994. 136 p. Dissertação (Mestrado) - Engenharia de Sistemas e Computação, COPPE / Universidade Federal do Rio de Janeiro.
- (Campos et al., 1999) Campos, F.C.A; Rocha, A.R.C; Campos, G.H.B. Qualidade de Software Educacional: uma Proposta. In: WORKSHOP DE QUALIDADE DE SOFTWARE (WQS), Florianópolis, 1999. *Anais*. Universidade Federal de Santa Catarina.
- (Casanova et al., 1991) Casanova, M.A., Tucherman, L.; Lima, M.J.; Rangel, J.L.; Rodriguez, N.L.R.; Soares, L.F.G. The Nested Context Model for Hyperdocuments. In: THIRD ACM CONFERENCE ON HYPERTEXT, San Antonio, 1991. *Proceedings*. p.193-201.
- (Díaz & Isakowitz, 1995) Díaz, A.; Isakowitz, T. RMCASE: Computer-Aided Support for Hypermedia Design and Development. In: International Workshop on Hypermedia Design, Montpellier, France. *Proceedings*, 1995.
- (Dix et al., 1998) Dix, A.J.; Finlay, J.E.; Abowd, G.D.; Beale, R. *Human-Computer Interaction*, Prentice Hall, 1998.
- (Hall & Mendes, 1999) Hall, W.; Mendes, M.E.X. Hyper-authoring for education: a qualitative evaluation. *Computers & Education*, v.32, n.1, p.51-64, 1999.
- (Habber, 1999) Habber, R.R. *Uma ferramenta de importação de documentos HTML para um ambiente de ensino*. São Carlos, 1999. 104p. Dissertação (Mestrado) - Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo.
- (ISO/IEC 14598-5, 1996) ISO/IEC 14598-5, International Standard. Information Technology - Software product evaluation - Part 5: Process for evaluators; 1996 (DIS).
- (Kearsley, 1982) Kearsley, G. Authoring Systems in Computer Based Education. *Communications of the ACM*, v.25, n.7, p. 429-437, 1982.
- (Nanard & Nanard, 1995) Nanard, J.; Nanard, M Hypertext Design Environments and the Hypertext Design Process. *Communications of the ACM*, v.38, n.8, p.49-56, 1995.
- (Nielsen, 2000) Nielsen, J. *Why you Only Need to Test With 5 Users*. [Online] [20/06/2000] Disponível na Internet <http://www.useit.com/alertbox/20000319.html>

- (Nunes et al., 1996) Nunes, M.G.V.; Hasegawa R.; Vieira, F.M.C. *HIP/Windows: um ambiente de autoria de hiperbases multimídia*. São Carlos, ICMSC, 1996. (Relatório Técnico ICMC-USP, 38, 34 p).
- (Nunes et al., 1997a) Nunes, M.G.V.; Fortes, R.P.M.; Nicoletti, M.C. Flexible guided-tours in hypertexts: a way of controlling the user in learning applications. In: WORLD MULTICONFERENCE ON SYSTEMICS, CYBERNETICS NA INFORMATICS (SCI'97). Caracas, 1997. *Anais*. p.184-191.
- (Nunes et al., 1997b) Nunes, M.G.V.; Hasegawa, R.; Vieira, F.M.C.; Santos, G.H.R.; Fortes, R.P.M. *SASHE: Sistema de Autoria e Suporte Hipermissão para Ensino*. São Carlos, Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 1997. (Notas ICMSC-USP, n.33, 22 p.).
- (Nunes & Pansanato, 1999) Nunes, M.G.V.; Pansanato, L.E. Autoria de Aplicações Hipermissão para Ensino. *Revista Brasileira de Informática na Educação*, n.5, p.103-124, 1999.
- (Paiva, 2001) D.M.B. *Proposta e Avaliação de um Conjunto de Requisitos para Sistemas de Autoria Hipermissão Educacional*. São Carlos, 2001. Dissertação (Mestrado) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo.
- (Pansanato, 1999) Pansanato, L.T.E. *EHDM: Um Método para o Projeto de Aplicações Hipermissão para Ensino*. São Carlos, 1999. 104p. Dissertação (Mestrado) - Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo.
- (Papatheodorou, 1998) Papatheodorou, T. S.; Styliaras, G. D.; Christodoulou, S. P. Evaluation of Hypermedia Application Development and Management Systems. The Ninth ACM Conference on Hypertext and Hypermedia. *Proceedings*, Pittsburgh p.1-10, 1998.
- (Preece et al., 1994) Preece, J.; Rogers, Y.; Sharp, H.; Benyon, D.; Holland, S.; Carey, T. *Human-Computer Interaction*, Addison-Wesley, Reading, MA, 1994.
- (Santos et al., 1997) Santos, G.H.R.; Vieira, F.M.C.; Hasegawa, R.; Nunes, M.G.V. SASHE: Autoria de Aplicações Hipermissão para o Ensino. In: VIII SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO. São José dos Campos, 1997. *Anais*. ITA, p.425-440.
- (Santos Jr, 1998) Santos Jr, J.B. *Documentos Estruturados para o Domínio de Aplicação Ensino: Modelagem, Autoria e Apresentação na WWW*. São Carlos, 1998. 107p. Dissertação (Mestrado) - Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo.
- (Sommerville, 1996) Sommerville, I. *Software Engineering*. 5th ed. England, Addison-Wesley, 1996.
- (Zellwegger, 1989) Zellwegger, P. T. Scripted Documents: A Hypermedia Path Mechanism. In: Hypertext'89. Pittsburgh, Pennsylvania, November 1989. *Proceedings*. New York, ACM Press, 1989, p.1-14.