

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

RODRIGO CLARO ZEMBRZUSKI

**Ferramenta de etiquetação de vídeos para o
padrão brasileiro de objetos de
aprendizagem.**

Trabalho de Graduação.

Profa . Dra. Renata Galante
Orientadora

Luiz Henrique Longhi Rossi
Coorientador

Porto Alegre, julho de 2013

CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Claro Zembrzuski, Rodrigo

Ferramenta de etiquetagem de vídeos para o padrão brasileiro de objetos de aprendizagem. / Rodrigo Claro Zembrzuski. – Porto Alegre: Graduação em Ciência da Computação da UFRGS, 2013.

57 f.: il.

Trabalho de Conclusão (bacharelado) – Universidade Federal do Rio Grande do Sul. BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO, Porto Alegre, BR-RS, 2013. Orientadora: Renata Galante; Coorientador: Luiz Henrique Longhi Rossi.

1. Objetos de Aprendizagem. 2. Vídeo. 3. Etiqueta semântica. I. Galante, Renata. II. Luiz Henrique Longhi Rossi, . III. Título.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Carlos Alexandre Netto

Vice-Reitor: Prof. Rui Vicente Oppermann

Pró-Reitora de Graduação: Prof^a. Valquiria Link Bassani

Diretor do Instituto de Informática: Prof. Luís da Cunha Lamb

Coordenador do CIC: Prof. Raul Fernando Weber

Bibliotecária-Chefe do Instituto de Informática: Beatriz Regina Bastos Haro

*“O que é bom, Fedro,
e o que não é bom —
Acaso precisamos pedir a alguém que nos ensine essas coisas?”*
— ZEN E A ARTE DA MANUTENÇÃO DE MOTOCICLETAS

AGRADECIMENTOS

Agradeço a minha família, por me ajudar a encontrar meus valores e por proverem minha educação.

A meus amigos que cantaram e tocaram o violão comigo.

Aos mestres que se dedicaram à verdadeira educação. Sobretudo à professora Renata, minha orientadora, e ao Preto, meu coorientador, que viabilizaram a realização desse trabalho.

SUMÁRIO

LISTA DE ABREVIATURAS E SIGLAS	9
LISTA DE FIGURAS	11
LISTA DE TABELAS	13
RESUMO	15
1 INTRODUÇÃO	17
2 EMBASAMENTO TEÓRICO	19
2.1 Definições	19
2.1.1 Objetos de Aprendizagem	19
2.1.2 Repositórios	20
2.1.3 Segmento multimídia	20
2.2 Metadados de educação	21
2.2.1 <i>Dublin Core</i> - DC	22
2.2.2 LOM	23
2.2.3 OBAA	24
2.3 Interação Humano-Computador	31
2.3.1 Desafios de interação com dados multimídias	31
2.3.2 Ferramentas básicas	33
2.3.3 Estudo de um protótipo alternativo	35
3 FERRAMENTA DE SEGMENTAÇÃO	39
3.1 Visão Geral	39
3.2 Requisitos	40
3.3 Servidor	41
3.3.1 Geração de thumbnails	41
3.3.2 Criação dos metadados OBAA	42
3.4 Cliente	42
3.4.1 Desenvolvimento do player	42
3.4.2 Interface de botões	43
3.4.3 Interface com storyboard	44
3.5 Integração da interface o sistema	46
3.6 Considerações finais	47

4	AVALIAÇÃO EXPERIMENTAL	49
4.1	Preparação dos experimentos	49
4.2	Protocolo de testes	50
4.3	Resultados dos experimentos	51
4.3.1	Avaliação subjetiva	51
4.3.2	Avaliação objetiva	54
5	CONCLUSÃO	55
	REFERÊNCIAS	57

LISTA DE ABREVIATURAS E SIGLAS

DC	Dublin Core
DCMI	Dublin Core Metadata Initiative
FEB	Federação Educa Brasil
IEEE	Institute of Electrical and Electronics Engineers
JSON	JavaScript Object Notation
LTSC	Learning Technology Standards Committee
LOM	Learning Object Metadata
MPEG	Moving Picture Experts Group
OA	Objeto de Aprendizagem
OBAA	Objetos de Aprendizagem Baseados em Agentes
USP	Universidade de São Paulo
XML	Extensible Markup Language

LISTA DE FIGURAS

Figura 1.1:	O aluno quer aprender o cálculo integral, mas somente a parte escura indicada pela linha de tempo trata do assunto.	18
Figura 2.1:	Solução para aumentar o nível de detalhamento mantendo a orientação espacial.	32
Figura 2.2:	Interface típicas para segmentação de vídeo: uma timeline sem possibilidade de refinamento e caixas de texto e em que o usuário coloca os pontos de entrada e saída do segmento.	34
Figura 2.3:	Componentes do protótipo de interface	35
Figura 2.4:	Aplicação de um corte, de uma segmentação, utilizando os recursos	35
Figura 2.5:	Refinamento sucessivo	37
Figura 3.1:	Visão geral da ferramenta: relação existente entre usuário, interface, cliente e servidor	39
Figura 3.2:	Visão geral do servidor: existe um método que extrai os <i>thumbnails</i> do vídeo e grava no disco, um método que busca no disco o vídeo e os <i>thumbnails</i> para o cliente e outro que recebe os dados de segmentação do usuário em JSON e gera metadados no padrão OBAA.	41
Figura 3.3:	Protótipo de interface baseada no estado-da-arte.	43
Figura 3.4:	Protótipo de interface com storyboard. Tela inicial.	44
Figura 3.5:	Protótipo de interface com storyboard após o primeiro corte.	45
Figura 4.1:	Na esquerda, os usuários dizem ter alto conhecimento de internet e na direita dizem ter média ou pouca experiência com vídeo.	51
Figura 4.2:	Na esquerda, os usuários afirmam que a <i>storyboard</i> ajuda o usuário a se contextualizar e na direita afirmam que a possibilidade de refinar a marcação aos poucos também ajuda.	52
Figura 4.3:	Na esquerda, 40% dos usuários dizem que que a interface de botões é melhor. 30% diz que não existe uma interface melhor que a outra, 20% diz que a interface com storyboard é um pouco melhor e 10% diz que ela é muito melhor. Já na direita, vemos que somente 30% diz que é melhor selecionar trechos arrastando o mouse, 40% diz que utilizar botões é melhor que arrastar e 30% diz que é indiferente.	53
Figura 4.4:	Observamos que para vídeos curtos, os usuários preferem utilizar a interface com botões, sem refinamento. Já para vídeos mais longos, eles preferem a interface com possibilidade de refinamento.	53

LISTA DE TABELAS

2.1	Conjunto de metadados propostos como extensão do padrão LOM a partir do padrão TV-Anytime.	26
Tabela 4.1:	Média e desvio padrão do erro para cada interface.	54

RESUMO

Existem diversos repositórios educacionais, que são utilizados para catalogação e busca de objetos de aprendizagem, que podem se apresentar em diversas formas, como textos ou vídeos. Esse trabalho consiste na proposta e implementação de uma ferramenta para segmentação de vídeos educacionais.

Uma deficiência comum aos repositórios educacionais é o suporte que se dá à catalogação de objetos multimídia, embora existam padrões de metadados que possuem suporte à segmentação de vídeo. Essa carência se deve ao fato de não existirem interfaces adequadas para que esse recurso seja utilizado.

A proposta do trabalho é o estudo e o desenvolvimento de uma ferramenta com interface de boa usabilidade que permita ao usuário a aplicação de etiquetas semânticas a segmentos de vídeo, utilizando o suporte dado pelo padrão OBAA. Para isso, é feito um estudo sobre os metadados oferecidos por esse padrão e um estudo sobre a forma de interação com dados multimídia. A partir disso, são realizados experimentos com usuários e integração da interface com um repositório que utiliza o padrão OBAA. Espera-se, com isso, dar início à obtenção de um número maior de metadados multimídias para que em trabalhos futuros seja efetuada uma busca mais efetiva por esse tipo de conteúdo.

Através da avaliação com usuários, conclui-se que as ferramentas para segmentação de vídeo devem oferecer recursos para que o usuário consiga se contextualizar facilmente em relação ao vídeo e que possa trocar o nível de detalhamento temporal para marcação de segmentos mais precisos. Esses recursos devem ser incorporados com elementos de fácil interação como utilização de botões para marcar os pontos de início e fim do segmento.

Palavras-chave: Objetos de Aprendizagem, Vídeo, Etiqueta semântica.

1 INTRODUÇÃO

A tecnologia sempre esteve presente no contexto educacional. Existem diversos repositórios educacionais — isto é, locais de armazenamento, catalogação e busca de objetos de aprendizagem — em que os autores de materiais de ensino podem submeter seus trabalhos para que sejam recuperados posteriormente. Entre os tipos de materiais educacionais que podem ser submetidos, encontram-se os objetos de vídeo. Esse trabalho tem como objetivo oferecer um suporte para que sejam feitas anotações semânticas a determinados segmentos de vídeos educacionais.

Os objetos de aprendizagem que antes se resumiam a textos e imagens — como artigos e dissertações — passaram a dar mais espaço para conteúdo multimídia devido a diversos avanços tecnológicos, como o aumento da capacidade de armazenamento e da banda das redes. O FEB — Federação Educa Brasil — uma das principais ferramentas de busca em repositórios educacionais do país, já possui mais de 2500 vídeos disponibilizados, que correspondem a cerca de 3% do conteúdo disponibilizado. A USP — Universidade de São Paulo — lançou recentemente um portal com aulas e conferências com mais de 800 vídeos.

Os principais sistemas que oferecem conteúdo de vídeo da atualidade não oferecem suporte à busca por determinados trechos dos vídeos. Tanto os sistemas não-educacionais, como Youtube ¹ ou Vimeo ² quanto os repositórios educacionais, como o portal de vídeos da USP e o FEB oferecem a busca somente por vídeos inteiros.

O grande problema envolvido nesse tipo de busca é que a pessoa interessada em recuperar o vídeo não sabe de antemão em qual parte está o conteúdo que ela de fato está interessada. Por exemplo, se um aluno está aprendendo trigonometria e já fixou bem o conteúdo de seno e cosseno, mas se encontra com dificuldade na parte da tangente, ele pode fazer uma busca por “tangente trigonometria”. As respostas apontarão para as aulas de trigonometria que possuem o conteúdo de tangente, o que é correto. Entretanto, pode ser que a parte de tangente encontre-se no final do vídeo e o aluno tenha que gastar tempo vendo a parte do vídeo que não lhe interessa, procurando manualmente pela parte da tangente ou até mesmo que deixe de ver o vídeo antes que chegue a parte que lhe é importante.

A busca por conteúdo multimídia é feita baseada principalmente em etiquetas semânticas aplicadas ao conteúdo. As soluções oferecidas normalmente são a aplicação de etiquetas ao vídeo inteiro. Diversas entidades já atentaram para o fato de dar suporte à segmentação de vídeo, como o padrão OBAA (Vicari et al, 2009). Contudo, esses recursos oferecidos por esses padrões normalmente não são utilizados devido à falta de

¹<http://www.youtube.com>

²<http://www.vimeo.com>

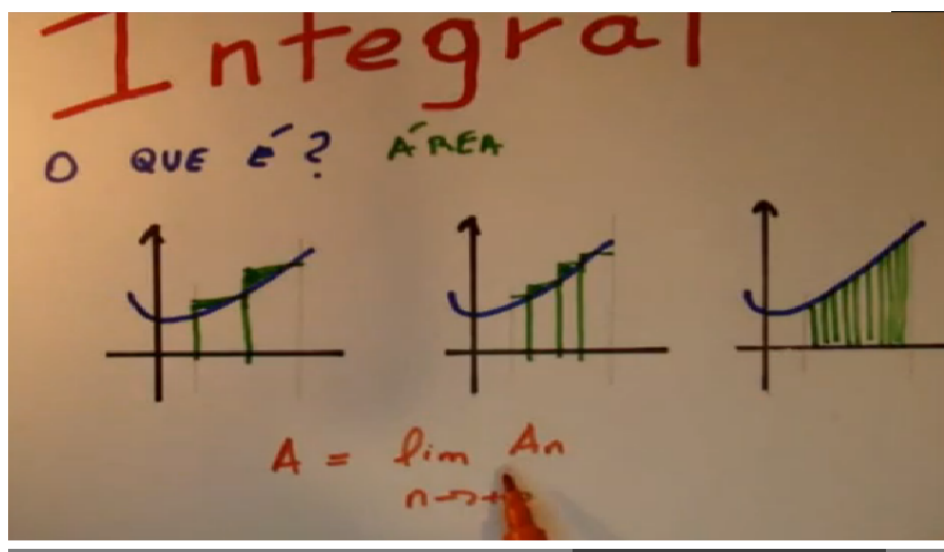


Figura 1.1: O aluno quer aprender o cálculo integral, mas somente a parte escura indicada pela linha de tempo trata do assunto.

ferramentas que dêem ao usuário suporte à utilização desses recursos. Comercialmente, não se encontram essas ferramentas e academicamente existem algumas abordagens que não vingaram.

Este trabalho tem como objetivo principal a proposta e o desenvolvimento de uma ferramenta que ofereça ao usuário o suporte necessário à obtenção de metadados semânticos a segmentos de vídeo e que estimule os usuários a fazer esse tipo de anotação. Para que a ferramenta possua boa usabilidade, serão feitos experimentos com usuários de modo que eles possam comparar 2 tipos de interface: a primeira é uma interface mais tradicional, que apresenta além da tela que exibe o vídeo, uma linha de tempo e botões para marcar os trechos; a segunda interface oferece ao usuário alguns outros recursos que supostamente melhoram a usabilidade, como a exibição de uma *storyboard* construída através da extração de miniaturas de determinados instantes do vídeo e a possibilidade de um refinamento iterativo. As interfaces coletam os dados providos pelos usuários e criam metadados educacionais no padrão OBAA, que dá suporte à segmentação de vídeos.

Espera-se, com isso, que sejam feitas anotações mais ricas ao conteúdo multimídia para que, em trabalhos futuros, seja feita uma busca mais rica ao conteúdo. Além disso, o trabalho testa o suporte que é dado à segmentação de vídeo pelo OBAA, que é bastante fundamentado mas que até então não foi colocado em prática.

O texto está organizado da seguinte forma: o capítulo 2, de embasamento teórico, apresenta as principais definições relevantes ao desenvolvimento do trabalho, um estudo sobre os principais padrões de metadados adotados para objetos de aprendizagem — com maior enfoque nos metadados de vídeo do padrão OBAA — e um estudo que aponta os principais elementos e dificuldades envolvidas na interação humano-computador no que tange à manipulação de conteúdos multimídias. No capítulo 3, de solução, serão apresentadas duas interfaces gráficas supostamente adequadas para manipulação de vídeos. No capítulo 4, serão apresentados resultados relevantes extraídos dos experimentos com usuários e o capítulo 5 finaliza o trabalho com as conclusões e os trabalhos futuros.

2 EMBASAMENTO TEÓRICO

Nesse capítulo serão apresentadas algumas definições que são relevantes para o desenvolvimento do trabalho, como objetos de aprendizagem, repositórios e segmentos de vídeo. Após as definições, será realizado um estudo comparativo sobre a forma como são catalogados os objetos de aprendizagem e a apresentação detalhada do suporte oferecido pelo padrão OBAA à segmentação de vídeo, pois é o único dos padrões estudados que oferece esse recurso. Em seguida é feito um estudo sobre os principais problemas envolvidos na interação entre usuários e dados multimídias e o estudo de uma interface que se propõe a solucionar esses problemas, que será utilizada como base para implementação do trabalho.

2.1 Definições

2.1.1 Objetos de Aprendizagem

Para o *Learning Technology Standard Committee (LTSC)* do *Institute of Electrical and Electronics Engineers (IEEE)* (LTS Committee, 2004), um objeto de aprendizagem é qualquer entidade, digital ou não, que possa ser usada, reusada e referenciada durante algum processo de aprendizagem. O objeto de aprendizagem é um conteúdo digital ou não (maquete, imagem, filme, etc.) que possa ser usado com algum propósito educacional e que inclui, internamente ou via associação, sugestões de contextos, nos quais ele deve ser utilizado.

A tecnologia de objetos de aprendizagem (OA) fundamenta-se na hipótese de que é possível criar componentes de material pedagógico e organizá-los de forma a possibilitar sua reutilização, promovendo economia de tempo e de custo na produção de cursos *on-line*, ou na sua utilização como parte do plano pedagógico de uma aula presencial.

A principal propriedade de um objeto de aprendizagem é sua reusabilidade. Tal característica pode ser alcançada através da modularidade, interoperabilidade e recuperação. Modularidade descreve o grau de separação e consequente recombinação dos componentes do OA. Interoperabilidade é a capacidade de operar em plataformas heterogêneas. A recuperação está relacionada à capacidade de poder ser encontrado em função de sua descrição de propriedades e funcionalidades.

Além da reusabilidade, outro aspecto importante que pode ser usado para avaliar e comparar padrões de metadados, para objetos de aprendizagem (ou padrões de metadados para outros tipos de objetos e conteúdos) é o grau de abrangência e cobertura das informações armazenadas nos metadados, em relação às aplicações que se pretendem dar aos objetos.

2.1.2 Repositórios

Os repositórios podem ser entendidos como uma casa de OAs, onde eles podem ser localizados (McGreal, 2004). Os repositórios também normalmente facilitam a localização dos materiais para que as buscas possam ser muito precisas.

Para atribuir aos OAs a característica de reusabilidade, é importante armazená-los e catalogá-los para que possam ser localizados. Para atender a essa demanda, surgem os repositórios de objetos educacionais (Richards, 2002).

Os repositórios tornam possível a busca por títulos, temas, autores, tipo de conteúdo entre outros. Em algumas situações, dados mais específicos, tais como a idade para quem foi desenvolvido o material, se o OA foi feito para professores, alunos ou educadores. Às vezes em buscas ainda mais refinadas, como uma consulta por materiais que não contém informação visual para pessoas com necessidades especiais. As possibilidades de busca estão intrinsicamente ligadas a seus metadados. Os metadados são informações sobre cada objeto. Por exemplo, um filme contém um título, diretor, atores, duração, entre outros. Esses são os metadados de um filmes.

Cada objeto de aprendizagem deve conter uma declaração de seus direitos de *copyright* (que devem ser expostos pelo repositório). Existem alguns formatos de licenças de *copyright* que permitem a modificação de conteúdo dos objetos, criando novas versões dos materiais. Essas atualizações podem ser geridas pelo repositório ainda que muitos dos que estão disponíveis hoje em dia não o façam. Repositórios que somente contém OAs abertos são chamados Recursos Educativos Abertos, que é uma iniciativa apoiada pelas Nações Unidas (ONU).

2.1.3 Segmento multimídia

Um segmento é um fragmento contínuo de um programa. Um segmento particular pode pertencer a um único programa, mas pode ser membro de vários grupos de segmentos. Um grupo de segmentos denota uma coleção de segmentos que são associados por uma finalidade particular ou devido a uma propriedade em comum. Um grupo de segmentos pode conter segmentos ou outros grupos de segmentos, de um ou mais programas. (Vicari, 2009)

2.2 Metadados de educação

Como se pode saber a quantidade de vitaminas e minerais que uma fruta (ou qualquer outro alimento) contém? Precisamos olhar sua tabela nutricional. Essa tabela representa seus metadados. Não é uma fruta em si e sim dados sobre ela (Jay, 2013). O mesmo se aplica a diversos sistemas que necessitam de uma padronização, como os recursos educacionais. Por isso, os padrões de metadados são tão importantes. Em cada área existe um conjunto de dados que são mais importantes. Para as frutas, é mais importante saber a quantidade de vitaminas; para os OAs, isso não se aplica. Precisamos, ali, saber os autores, palavras-chaves, etc. Por isso surgem os padrões de metadados de objetos educacionais.

É muito comum se ver grandes coleções de objetos mal catalogados. Isso acontece, principalmente, por não se dar o devido tempo e atenção ao processo de catalogação. Quando um autor faz um OA, ele não se dá conta de que há a necessidade de gerar seus metadados, e somente sente essa necessidade quando vai publicá-lo em um repositório, e às vezes quando está publicando também muitos OAs, aí os erros se sucedem e o processo se complica como um todo. Os erros na catalogação são de difícil percepção e trazem outros problemas, como dificuldade nas buscas, má publicação de direitos autorais, entre outros. Por essas situações que muitas vezes a definição dos padrões de metadados são feitas tentando minimizar os erros e facilitar para os autores, mesmo que com conjuntos menores se perca algumas capacidades interessantes desses recursos.

Para facilitar a catalogação e seja adquirida a principal característica de OA — reusabilidade —, diversos grupos de pesquisa e entidades visam a padronização desses objetos, a fim de compatibilizar o desenvolvimento e uso de OAs em escala mundial. Os padrões serão vistos a seguir.

2.2.1 Dublin Core - DC

O *Dublin Core* (DC) é um padrão de metadados para OAs desenvolvido e mantido pela DCMI (*Dublin Core Metadata Initiative*) e é definido por um conjunto pequeno de metadados (quinze) trazem informações gerais sobre cada OA, como os contribuintes, cobertura, criador, data, formato, identificador, idioma, publicadores, relações, direitos, fonte, tema, título e tipo.

- **Contribuinte:** a entidade (pessoa, organização ou serviço) responsável por fazer alguma contribuição ao projeto;
- **Cobertura:** relatorio ao tempo ou espaço onde o objeto faz sentido; uma jurisdição onde o objeto é relevante;
- **Criador:** o principal responsável pelo objeto; pode ser uma pessoa, organização ou serviço;
- **Data:** uma data específica ou período do ciclo de vida desse objeto, criação e modificação;
- **Descrição:** um resumo do conteúdo do objeto;
- **Formato:** formato serve tanto para o formato do arquivo, quanto para definições de tamanho e tempo de duração;
- **Identificador:** uma referência desambígua a esse objeto em seu contexto;
- **Idioma:** o idioma em que o objeto está feito;
- **Publicador:** entidade responsável por manter o objeto disponível;
- **Relação:** uma fonte relacionada;
- **Direitos:** a declaração de direitos autorais e de distribuição;
- **Fonte:** a fonte do qual o objeto é derivado (caso tenha);
- **Tema:** o tema de interesse do recurso;
- **Título:** o nome dado ao objeto;
- **Tipo:** a natureza o gênero do recurso; se é uma imagem, vídeo;

O preenchimento dos metadados é uma etapa muito cansativa e, por isso, por ser o padrão mais antigo, a adoção de um conjunto menor de metadados muitas vezes é a preferida. Logo o DC tem sido um dos padrões mais populares nos repositórios do mundo inteiro.

2.2.2 LOM

O padrão desenvolvido pelo grupo de trabalho IEEE *Learning Standards Committee* (LTSC), o *Learning Object Metadata* (LOM) é um padrão especializado em entidades digitais ou não-digitais usada para educação ou formação (LTS Commitee, 2004).

Dentre essas iniciativas, o LOM é considerado um padrão aberto e internacionalmente reconhecido para facilitar a busca, avaliação, construção e uso de OAs, provendo um modelo de dados normalmente codificado em XML, que segue a norma IEEE Std 1484.12.1 – 2002. O objetivo desse padrão é especificar a sintaxe e a semântica de informações (metadados) a respeito de um OA. Essas especificações permitem catalogar os materiais educacionais (através de seus metadados), tendo em conta a diversidade de contextos culturais e linguísticos da criação e reuso dos OAs. Dessa maneira, procura-se garantir modos eficientes de identificação, (re)utilização, gerenciamento, interoperabilidade, compartilhamento, integração e recuperação de objetos de aprendizagem (Vicari, 2009).

Diferentemente do DC, o LOM é um padrão com níveis de hierarquia. Ou seja, seus metadados estão divididos em subconjuntos, uns dentro dos outros. Pode-se afirmar também que o DC está contido dentro do conjunto de metadados do LOM, então afirmamos que há como transcrever um OA que está em DC para LOM, mas não no outro sentido (de LOM para DC).

No LOM há um conjunto total de 59 metadados separados em nove categorias: geral, ciclo de vida, meta metadados, técnicos, educacionais, direitos, relações, anotações e classificação. Cada conjunto com seus metadados específicos:

- **Geral:** conjunto com nove metadados para descrição do objeto como um todo, com títulos, palavras-chaves, entre outros;
- **Ciclo de Vida:** conjunto de cinco metadados relacionados à história do objeto, desde sua data de criação até todas suas modificações e o estado atual;
- **Meta metadados:** conjunto de sete metadados para a informação sobre os próprios metadados, como a pessoa que fez os registros dos OAs;
- **Técnico:** conjunto de dez metadados sobre os requisistos e características técnicas do OA, como o tipo (vídeo, imagem);
- **Educacional:** conjunto de onze metadados para uso educacional e pedagógico, com dados como a idade do público, se o OA é para professores (como plano de aula) ou direto para os alunos;
- **Direitos:** conjunto com três metadados para declarações sobre os direitos autorais e de uso;
- **Relações:** conjunto com quatro metadados para criar relações entre os OAs. Por exemplo OA-1 “faz parte de” OA-2 ou OA-1 “é uma versão de” OA-2;
- **Anotação:** conjunto de três metadados para que se ponham comentários sobre o uso educacional do OA; essa categoria permite ao educador compartilhar suas opiniões e sugestões de uso;
- **Classificação:** conjunto de sete metadados para uma especificação do objeto em cada sistema de classificação, aí se põe uma relação a uma taxonomia para classificação específica do objeto.

Cabe destacar aqui que o grau de abrangência e cobertura das informações armazenadas nos metadados do padrão DCMI é muito baixo em relação à abrangência que o padrão LOM oferece para aplicações educacionais e pedagógicas.

Isso evidencia uma clara diferença de objetivo entre os conjuntos de metadados apresentados. Enquanto o DC é um modelo para simplificar seu uso, o LOM tem por objetivo ser mais completo e mais específico quanto ao uso de cada um de seus objetos.

Com essa maior quantidade de metadados, se ganha em qualidade da descrição do OA, mas necessita de mais tempo para sua catalogação.

2.2.3 OBAA

No contexto de conversão e adaptação de conteúdos multimídia para distintas plataformas e disponibilização de conteúdos digitais, como por exemplo, *web*, dispositivos móveis, TV digital e consoles de jogos, pode-se verificar a relativa falta de cobertura que o padrão IEEE-LOM tem a oferecer quando comparado com o padrão MPEG-7 ou outros padrões similares de metadados para conteúdo multimídia.

Os metadados OBAA podem ser entendidos como uma extensão do padrão LOM, com metadados para suprir essas deficiências, bem como para oferecer uma extensão para os objetos inteligentes baseados em agentes de inteligência artificial (Vicari, 2010).

Os seguintes conjuntos de metadados não sofreram alterações em relação à proposta inicial do IEEE-LOM: geral, ciclo de vida, meta metadados, direitos, anotações e classificação. Já os grupos que tiveram modificações são os seguintes: técnico e educacional. Além disso, também houve a criação de dois novos grupos: acessibilidade e tabela de segmentação.

- **Técnico:** dezessete metadados para suporte multiplataforma e serviços de agentes artificiais para sistemas de inteligência artificial.
 - **Multiplataforma:** são definidos quais serão as plataformas suportadas (um metadado) e as especificações delas (oito metadados);
 - **Serviços:** são definidos oito metadados para implementação de serviços com agentes artificiais, todos os metadados são parâmetros para instanciar um agente em uma plataforma FIPA;
- **Educacional:** são adicionados seis novos metadados a esse conjunto; o modelo pedagógico é composto por uma arquitetura e contextualização epistemológica pedagógicos.
- **Acessibilidade:** este grupo foi criado no padrão IMS AcessForAll. São 23 metadados para especificações do OA quanto às alternativas e requisitos para portadores de necessidades especiais, seja áudio, texto ou recursos visuais.
- **Tabela de segmentação:** grupo criado com treze metadados para segmentação de vídeos, ou seja, identificação de partes de vídeos, tornando possíveis consultas como “momento da aula onde se resolveram os exercícios”, se o vídeo foi segmentado e se o sistema de consulta tem suporte, deve ser retornada uma lista de todos os momentos e não a aula toda. Essa seção é uma adaptação do MPEG-7 (Vicari, 2009).

Com isso, o OBAA totaliza 108 metadados para descrição dos materiais educacionais, e como é uma extensão do LOM, podemos dizer que o conjunto de dados do LOM está

contido no conjunto de dados do OBAA. Com isso, é evidente que há uma tradução de LOM para OBAA, porém o contrário não é verdade.

Como recomendação de metadados para o padrão educacional, com base no *TV-Anytime* (TVA,2003b), sugere-se a inclusão dos metadados de segmentação definidos pelo padrão MPEG-7, que compõe o *SegmentInformationTable*. A seguir, uma descrição dos metadados de segmentação a serem previstos, seguindo o padrão MPEG-7.

O padrão LOM também foi estendido com o conjunto de metadados *SegmentInformationTable*, obtido a partir do padrão *TV-Anytime* (TVA,2003b), padrão de metadados utilizado pela TV Digital na Europa, mas simplificado e adaptado para atender as necessidades do padrão proposto.

A tabela a seguir apresenta os metadados de segmentação, que serão utilizados para implementação do protótipo desse trabalho.

Tabela 2.1: Conjunto de metadados propostos como extensão do padrão LOM a partir do padrão TV-Anytime.

Nr.	Nome	Descrição	Card	Domínio	Tipo de dado	Exemplo
11	<i>SegmentInformationTable</i>	Grupo que conterá o conjunto de informações de segmentação dos objetos de aprendizagem e de grupos de segmentos dos objetos de aprendizagem.	0..1	-	<i>Container</i>	-
11.1	<i>SegmentList</i>	Conjunto de informações de segmentos.	0..N	-	-	-
11.1.1	<i>SegmentInformation</i>	Agrupamento das informações de um segmentos (<i>SegmentInformationType</i> do TV-Anytime)	1..N	<i>TV-Anytime</i>	<i>String(1000)</i>	-
11.1.1.2	<i>Identifier</i>	Identificador único do segmento nesse objeto de aprendizagem. É um campo alfanumérico	1	-	<i>String(100)</i>	1, 222, 12, a, b, exercício1, ...
11.1.1.3	<i>Title</i>	Título do segmento.	1	-	<i>String(1000)</i>	“Exercícios sobre receptores de TV Digital”
11.1.1.4	<i>Description</i>	Descrição do conteúdo do segmento.	1	-	<i>String(2000)</i>	“Seção de exercícios do módulo de introdução à TVD.”
11.1.1.5	<i>Keyword</i>	Palavras-chave referentes ao segmento.	1..N	-	<i>String(1000)</i>	“TV Digital, receptores, exercícios”
11.1.1.6	<i>SegmentMediaType</i>	Descreve se é um documento de texto (<i>document</i>), hiperdocumento (<i>hyperdocument</i>), arquivo multimídia (<i>audio</i> ou <i>video</i>), ou outros (<i>other</i>).	1	<i>Document, Hyperdocument, audio, video, other</i>	Vocabulário descrito no domínio	<i>video, hyperdocument</i>

11.1.1.7	<i>Start</i>	Se o segmento for originado de um arquivo multimídia, deverá indicar o tempo de início (<i>MPEG-7 MediaTimeType</i>). Se for um documento de texto, indica a página e, opcionalmente, a linha inicial. Se for um hiperdocumento, a página, seção de uma página ou mídia inclusa no hiperdocumento.	1	-	<i>String</i>	“PT00H05M” “Page20Line10”
11.1.1.8	<i>End</i>	Fim do segmento. Se o segmento for originado de um arquivo multimídia, deverá indicar o tempo de fim (<i>MPEG-7 MediaTimeType</i>). Se for um documento de texto, indica a página e, opcionalmente, a linha final. Se for um hiperdocumento, este metadado não será utilizado.	1	-	<i>String</i>	“PT00H05M” “Page20Line10”
11.2	<i>SegmentGroupList</i>	Conjunto de grupos de segmento.	0..1	-	<i>Container</i>	-
11.2.1	<i>SegmentGroupInformation</i>	Conjunto de informações do grupo de segmentos (<i>SegmentGroupInformationType</i> do <i>TV-Anytime</i>).	1..N	-	<i>Container</i>	-
11.2.1.1	<i>Identifier</i>	Identificador do grupo de segmento. Deve ser único no objeto de aprendizagem.	1	-	<i>String</i>	1, 23, 10000, Grupo1, Exercícios

11.2.1.2	<i>GroupType</i>	Tipo de agrupamento. Seguirá a mesma sintaxe de TVA: <i>Segment-GroupTypeType</i>	1	<i>highlight, bookmarks, theme-Group, preview, activities, other</i>	Vocabulário definido no domínio	<i>Highlight, Bookmarks</i>
11.2.1.3	<i>Title</i>	Título do segmento.	1	-	<i>String(1000)</i>	Atividades e Exercícios do OA.
11.2.1.4	<i>Description</i>	Descrição do conteúdo do segmento.	1	-	<i>String(2000)</i>	Grupo contendo os segmentos com os exercícios deste objeto de aprendizagem.
11.2.1.5	<i>Keyword</i>	Palavras-chave referentes ao segmento.	1..N	-	<i>String(1000)</i>	Exercícios, Atividades
11.2.1.6	<i>Segments</i>	Segmentos que fazem parte do grupo	1	-	<i>Container</i>	-
11.2.1.6.1	<i>Identifier</i>	Código identificador único do segmento (11.1.1.2: <i>Identifier</i>) que percente a esse grupo	1..N	-	<i>String(1000)</i>	1, 222, 12, a, b, exercício1

Fonte: Vicari, 2009

11 *SegmentInformationTable*

Este novo grupo de metadados que conterá o conjunto de informações de segmentação dos objetos de aprendizagem e de grupos de segmentos dos objetos de aprendizagem.

11.1 *SegmentList*

Conjunto de informações de segmentos

11.1.1 *SegmentInformation*

Agrupamento das informações de um segmento (*SegmentInformationType* do *TV-Anytime*)

11.1.1.1 *SegmentIdentifier*

Identificador único do segmento no objeto de aprendizagem. É um campo alfanumérico.

11.1.1.2 *Title*

Título do segmento.

11.1.1.3 *Description*

Descrição do conteúdo do segmento.

11.1.1.4 *Keyword*

Palavras-chave referentes ao segmento.

11.1.1.5 *SegmentMediaType*

Descreve se é um documento texto (*document*), hiperdocumento (*hyperdocument*), arquivo multimídia (*audio* ou *video*), ou outros (*other*).

11.1.1.6 *Start*

Localiza o início do segmento no objeto de aprendizagem.

Se o tipo de mídia do segmento for um documento geral (*SegmentMediaType* = “*document*”), o início do segmento será a página e, opcionalmente, a linha de início (por exemplo, “Page10Line15”).

Se o tipo de mídia for um áudio ou vídeo (*SegmentMediaType* = “*audio*” ou *SegmentMediaType* = “*video*”), será informado o tempo de início do segmento. O formato a ser utilizado é o MPEG-7 *datatype*, *MediaTimeType*, na seção 6.4.10 de (ISO,2001).

11.1.1.7 *End*

Localiza o final do segmento no objeto de aprendizagem.

Para objetos de aprendizagem do tipo documento (*document*) e multimídia (*multimedia*), segue a mesma sintaxe do item 11.1.1.7: *SegmentInformationTable* .*SegmentList* .*SegmentInformation.Start*. Caso não seja especificado, o final do segmento será o final da mídia (digital ou impressa).

Para os objetos do tipo hiperdocumento (*hyperdocument*), esse metadado não será preenchido, pois a seção indicada no Start já indica por si só seu início e fim, assim como uma página ou outro tipo de mídia inclusa na própria página.

11.2 *SegmentGroupList*

Conjunto dos grupos de segmento.

11.2.1 SegmentGroupInformation

Conjunto de informações do grupo de segmentos (*SegmentGroupInformationType* do *TV-Anytime*).

11.2.1.1 Identifier

Identificador único do segmento. É um campo alfanumérico.

11.2.1.2 GroupType

Tipo de agrupamento. Terá o seguinte domínio de valores: “*highlights*”, “*bookmarks*”, “*themeGroup*”, “*preview*”, “*activities*” ou “*other*”.

11.2.1.3 Title

Título do segmento.

11.2.1.4 Description

Descrição do conteúdo do segmento.

11.2.1.5 Keyword

Palavras-chave referentes ao segmento.

11.2.1.6 Segments

Identificadores dos segmentos que fazem parte do grupo.

11.2.16.1 Identifier

Código identificador único do segmento (11.1.1.2:Identifier) que pertence a este grupo.

2.3 Interação Humano-Computador

Nessa seção, serão abordados alguns dos principais problemas que frequentemente acontecem na interação dos usuários com ferramentas de manipulação multimídia. É feito um estudo do estado-da-arte, com a observação das principais ferramentas de manipulação de vídeo e o estudo de um protótipo alternativo que teoricamente é adequado para lidar com os problemas observados.

2.3.1 Desafios de interação com dados multimídias

Um dos clássicos desafios do design de interfaces para humanos, que há muito tempo vem sendo discutido, é a superação dos limites espaciais das pequenas e limitadas resoluções de tela (Mills, 1992). Por mais que as telas tenham crescido em tamanho e resolução, as workspaces dos usuários ainda são limitadas. Se precisarmos, por exemplo, manipular uma fotografia tirada com uma câmera comum de 10 *megapixels*, nas telas comerciais de melhor resolução só aparecerá uma visão parcial da fotografia. Do mesmo modo, se quisermos ler um livro e houver a necessidade de ir para determinados capítulos, precisamos de mecanismos que dêem ao usuário um *feedback* da parte que está sendo visualizada. O recorrente problema, então, é desenhar workspaces que permitam ao usuário trabalhar com visões parciais e detalhadas dos objetos enquanto se mantém o senso de contexto.

Existem diversas abordagens já consolidadas para resolver o problema de contextualizar a visão parcial que o usuário possui em relação à totalidade do conteúdo. O display bifocal é uma abordagem que ajuda as pessoas a trabalhar com documentos em diferentes níveis de granularidade e técnicas 3D também vêm sendo aplicadas para tornar a integração entre o nível detalhado e seu contexto mais natural.

Do mesmo modo como os usuários precisam de ferramentas adequadas para trabalhar com conteúdo estático, como imagens e textos, em diferentes níveis de granularidade, eles também precisam de ferramentas de detalhamento temporal para trabalhar com dados dinâmicos, como filmes, animações e áudio. Na segmentação de um vídeo, por exemplo, alguém pode querer dar um *zoom-in* em uma pequena parte do clipe com intuito de marcar alguma transição ou algum efeito de uma cena. No próximo momento, o usuário pode querer voltar para um nível menos detalhado para navegar até outra cena.

Mas, ainda assim, oferecer a possibilidade de alterar o nível de detalhamento de um objeto espacial e temporal não significa que se obtém uma workspace de boa usabilidade. O problema maior é encontrar uma maneira de dar ao usuário um senso de orientação, uma maneira de ajudá-lo a entender que parte do vídeo ele está observando e relacioná-lo ao todo. Trabalhando com objetos espaciais, o usuário pode apresentar dificuldades se não obtiver uma visão abrangente do objeto que está manipulando. Se um usuário quer retirar os olhos vermelhos de uma foto com várias pessoas, por exemplo, ele pode dar um *zoom* nos olhos de uma pessoa e recolorir os olhos. Entretanto, se ela não tiver uma informação global da imagem, ela poderá apresentar dificuldade em chegar aos olhos de outra pessoa para continuar fazendo a recoloração dos olhos.

Do mesmo modo, pode ser um desafio navegar com boa precisão em diferentes segmentos de um vídeo longo se não obtivermos uma ferramenta que dê a possibilidade de aumentar o nível de detalhamento temporal mantendo a consciência do contexto periférico. O usuário pode querer dar um *zoom-in* para marcar uma cena que dura apenas alguns segundos e voltar para uma visão global para buscar uma outra cena subsequente. Entretanto, se a interface não suporta adequadamente o usuário com a informação contextual, ele pode ter dificuldade de navegar entre essas partes do vídeo.

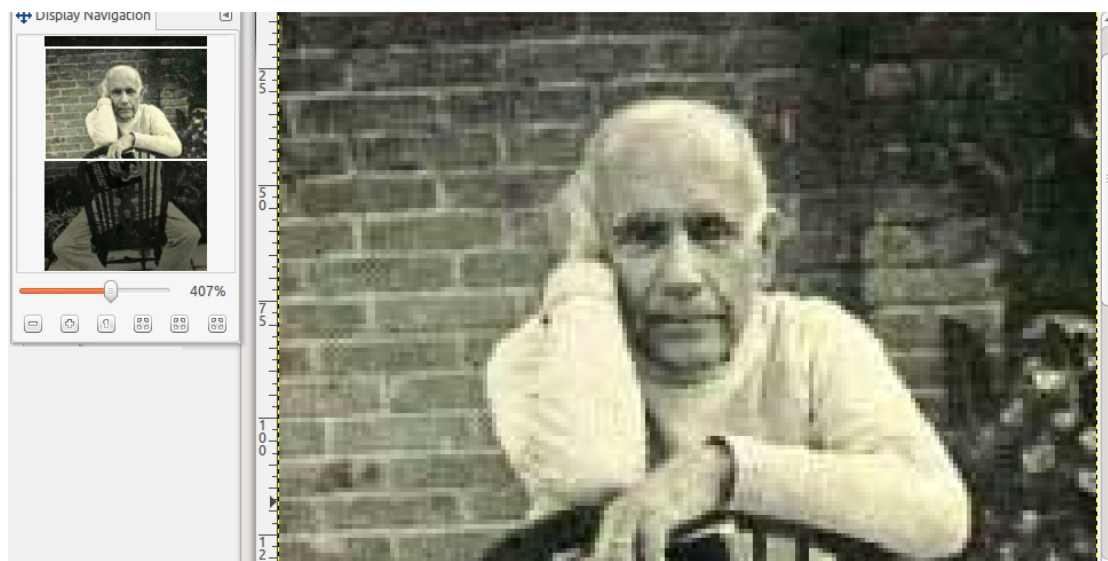


Figura 2.1: Solução para aumentar o nível de detalhamento mantendo a orientação espacial.

Para fornecer aos usuários um senso de orientação espacial quando se trabalha com grandes imagens, por exemplo, muitos programas de edição de imagens oferecem a ferramenta de “lupa”, que permite a visualização ampliada e parcial da imagem. Mas, ao mesmo tempo, mantém uma outra pequena janela com a imagem inteira separada. Essa janela contém a imagem em baixa definição e serve apenas para mostrar ao usuário qual é a parte da imagem que ele está observando em grande resolução, como se pode observar na figura 2.1 . Diversas interfaces para leitura de texto também apresentam o mesmo tipo de solução: exibem as páginas do livro em baixa definição para que o usuário possa navegar rapidamente enquanto mantém a página em alta definição em outra janela.

2.3.2 Ferramentas básicas

Muitas ferramentas de edição de vídeo na Web existentes atualmente fornecem um limitado senso de contexto temporal através da utilização de uma *timeline* espacial. Isso frequentemente se dá na forma de um *slider*, no qual o eixo horizontal representa a duração total de um evento, como no exemplo da figura 2.2. A execução de um vídeo em sua velocidade normal irá atualizar a posição horizontal do indicador ao longo da *timeline* em uma frequência apropriada. O usuário rapidamente pode ir de uma parte a outra do vídeo arrastando o *slider* pela *timeline*. Entretanto, se há menos *pixels* constituindo o tamanho horizontal da *timeline* e o usuário arrastar o *slider*, o player irá deixar de mostrar muitos frames — isso é uma forma de compressão temporal. Por exemplo, se o usuário está visualizando um clipe muito longo e ele resolve arrastar o *slider*, por mais sutil que seja seu deslocamento, resultará um salto temporal muito grande no clipe, e se o usuário quiser fazer um ajuste mais fino nesse clipe, ele apresentará dificuldade.

O *slider* funciona como um quadro de referência para o atual instante do vídeo que é exibido. A posição do indicador na *timeline* ajuda o usuário a visualizar a posição corrente do vídeo em relação à duração total do vídeo. Observando essa relação, o usuário pode responder a questões do tipo “Dado que estou olhando para a cena do céu, quanto do vídeo ainda está por vir?” ou ainda “Mostre-me o quadro que representa a metade do vídeo”(Mills, 1992). Além disso, a *timeline* oferece uma visão ampla que pode ser utilizada para navegação do vídeo.

Entretanto, ao passo que o *slider* oferece um contexto temporal, ele ainda pode ser visto como uma representação muito pobre do vídeo. Algumas questões como “A cena de luta acontece mais de uma vez no filme?” ou “A parte em que a moça acende o cigarro vem antes ou depois da cena de janta?”. Não existe informação explícita para esse tipo de questão que são úteis, nem existem pistas no *slider* de onde o usuário possa clicar para encontrar essas respostas.

Em suma, essa visão contextual oferecida pela *timeline* é muito pobre e a navegação apenas pela *timeline* limita a usabilidade de navegação. Ela não contém pistas sobre o conteúdo e estrutura do vídeo apresentado.

À parte dessas limitações, diversas ferramentas para segmentação de vídeo utilizam somente a *timeline* para situar o usuário. Uma das aplicações mais típicas de segmentação de vídeo é a aplicação de legendas a filmes, com intuito de transcrever a fala dos personagens do clipe. Na Figura 2.2 são apresentadas as interfaces do programa “Divx Land Media Subtitler”¹ e do “Captions Tube”², programas típicos para aplicação de legendas. Assim como eles, grande parte das ferramentas apresenta uma interface similar: possuem uma tela onde é exibido o vídeo e a *timeline* sem possibilidade de refinamento temporal. No primeiro exemplo existe uma lista de segmentos, na qual o usuário deve entrar com o frame de início do segmento, de fim de segmento e legenda do trecho. Para criação de um segmento, o usuário coloca o vídeo para ser reproduzido e quando o usuário pressiona uma tecla específica — tipicamente é a tecla “espaço” — é marcado como ponto de início do segmento o frame que é exibido no instante que ele pressionou a tecla. O usuário mantém a tecla apertada com o vídeo sendo executado e quando ele quer marcar o ponto de fim do segmento, ele solta a tecla. Esse trecho, então, é anotado em uma lista para que ele posteriormente aplique um texto ou um significado a ele. O segundo exemplo é similar, a diferença básica é que em vez de usar as teclas do mouse para obter os pontos de início e

¹<http://www.divxland.org/en/media-subtitler/>

²<http://captiontube.appspot.com/>

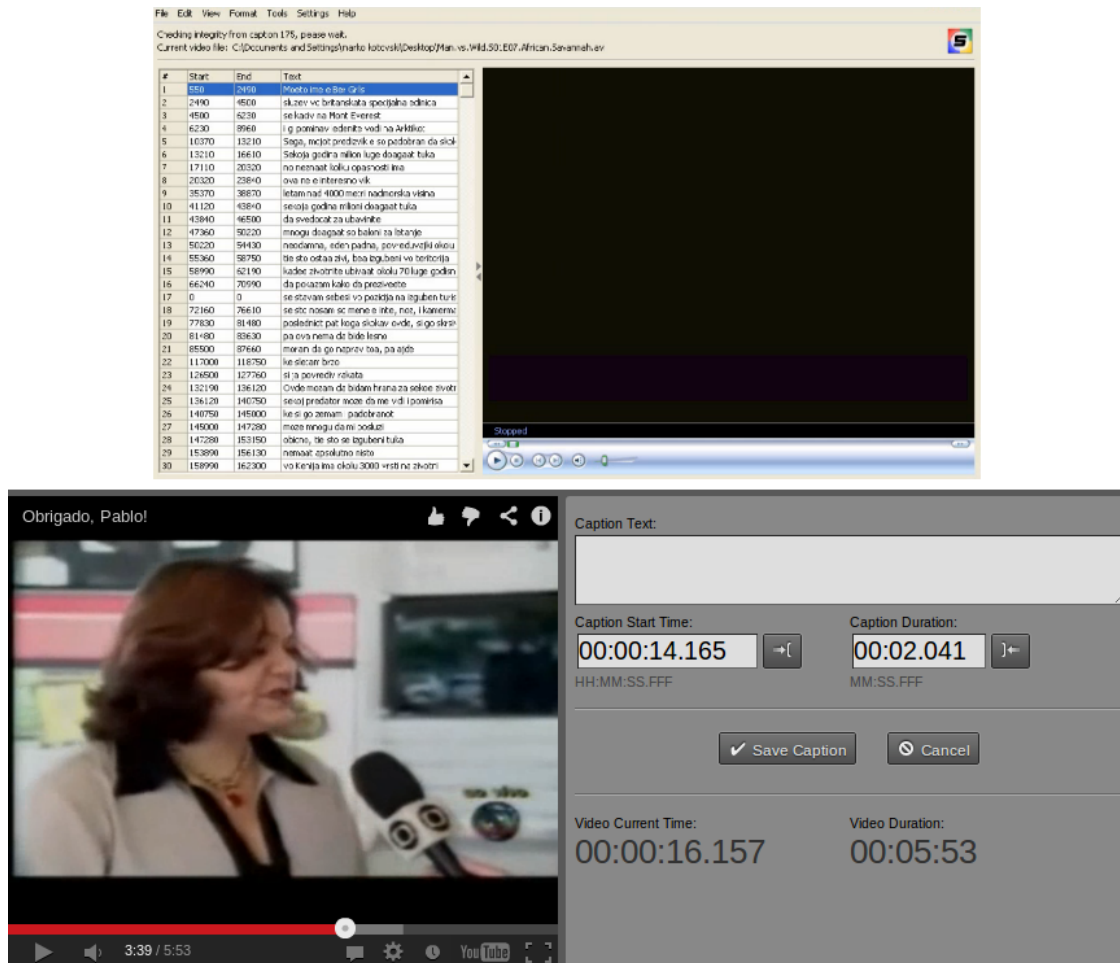


Figura 2.2: Interface típicas para segmentação de vídeo: uma timeline sem possibilidade de refinamento e caixas de texto e em que o usuário coloca os pontos de entrada e saída do segmento.

fim do trecho, são utilizados botões para isso (Wendy, 1998).

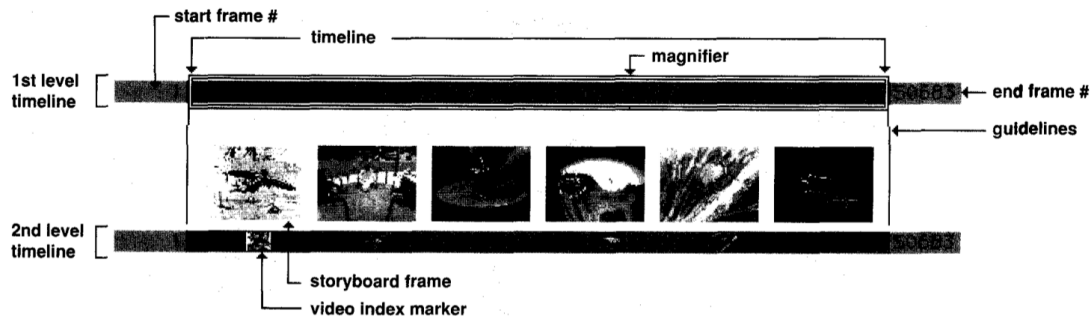


Figura 2.3: Componentes do protótipo de interface

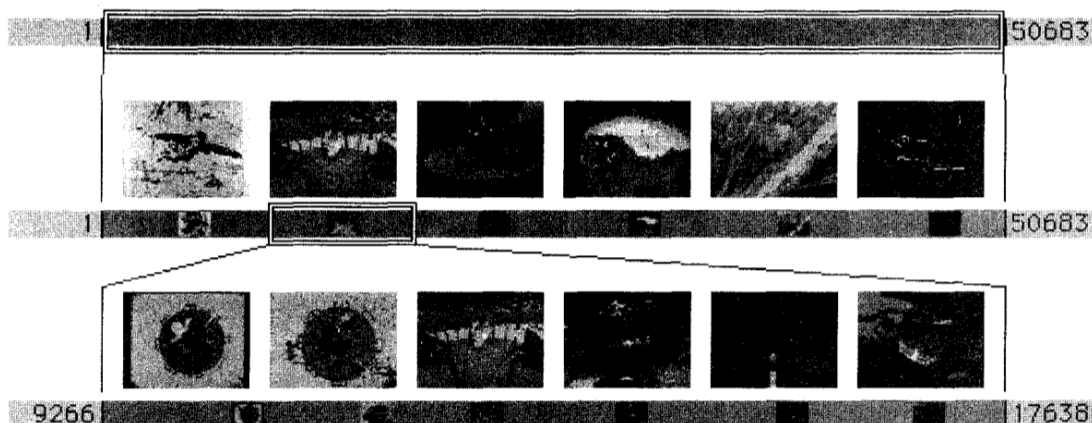


Figura 2.4: Aplicação de um corte, de uma segmentação, utilizando os recursos

2.3.3 Estudo de um protótipo alternativo

Existe um protótipo de interface que tenta resolver as limitações da timeline básica descritas anteriormente, proposto por Mills no artigo intitulado “*A magnifier Tool for Video Data*”. Este protótipo tem como proposta a apresentação de uma combinação de timeline navegável com uma representação mais rica do vídeo. O objetivo é fornecer ao usuário uma ferramenta poderosa para trabalhar com a estrutura e o conteúdo do vídeo. Sua idéia básica é a seguinte: é apresentada inicialmente uma timeline que representa a duração total do vídeo mas, em vez de apenas oferecer a navegação com um único nível de granularidade, é oferecido ao usuário também uma série de frames em baixa resolução que formam um array ao longo da timeline com o intuito de proporcionar um senso de contexto do vídeo, uma ferramenta de lupa, que pode ser utilizada para expandir ou diminuir a resolução temporal de qualquer parte da timeline e a manutenção de uma visão contextual — uma visão espacial explícita da estrutura do vídeo, que resulta da aplicação sucessiva da lupa temporal (Mills, 1992).

O protótipo de interface funciona da seguinte forma: existem inicialmente 3 janelas: na primeira janela é exibido o vídeo, na segunda tela é exibido um controlador de navegação do vídeo e a terceira contém uma ferramenta de lupa temporal.

Podemos observar na Figura 2.3 os elementos com mais detalhes: no topo existe uma *timeline* que representa a duração total da fonte do vídeo; na esquerda indica o frame inicial (1) e na direita o frame final (50,683). Existe uma lupa retangular que envolve a duração inteira do primeiro nível da *timeline*. Note que existem linhas que dos limites da timeline inicial para a segunda timeline, que exibida abaixo. A segunda *timeline*

representa um segmento da *timeline* acima, que no caso inicial corresponde ao vídeo inteiro. Existem também seis frames em baixa resolução igualmente espaçados ao longo da *timeline*, e cada um representa uma visão extremamente comprimida do vídeo original. Ela serve como um tipo de *storyboard*, que dá ao usuário um tipo de informação sobre a ordem em que ocorrem os eventos ao longo do vídeo.

O segundo nível da *timeline* contém miniaturas dos *frames* maiores que estão aparecendo na *storyboard*. Elas servem como “marcadores de vídeo” — indicadores de posição que podem ser arrastadas ao longo da *timeline* para visualizar o vídeo. Arrastando o marcador de frame para a direita, por exemplo, irá atualizar o conteúdo da janela que exibe o vídeo, o conteúdo dos próprios marcadores de vídeo e os frames da *storyboard* correspondentes. Através da manipulação dos marcadores de vídeo, o usuário pode rapidamente customizar o conteúdo da *timeline* de segundo nível.

No seu estado inicial, o intervalo da lupa temporal — o quadro que envolve a *timeline* — se estende ao longo de toda *timeline*. Ele mostra a visão mais abrangente do vídeo. Com a modificação da largura da lupa, o usuário foca em uma região menor da *timeline*, como mostra a Figura 2.4. Essa ação efetivamente baixa a compressão temporal e, além disso, oferece uma visão mais detalhada de uma pequena parte do vídeo. Nesse caso, antes da compressão havia 50.000 frames na *timeline* e após a compressão passou para apenas 25.000 frames. Note que os conteúdos dos frames, o índice dos marcadores e o número dos frames na segunda *timeline* foram atualizados para refletir o menor intervalo que a lupa restringiu.

Observe que o usuário tem a disponibilidade de variar a resolução temporal da *storyboard*: suponha que o usuário queira ver com um nível maior de detalhamento que tipo de evento cerca a cena de carrocéu, o segundo frame da esquerda na figura 2.3. Através do clique na *timeline* na região do carrossel, o usuário traz para a tela uma visão mais ampliada daquela região. Veja na figura 2.4, uma nova *storyboard* surge junto com uma nova *timeline* que corresponde à região delimitada pela lupa. As linhas que descendem dos limites da *timeline* anterior para a recém criada ajudam o usuário a visualizar a parte que foi ampliada — ele vê que o segundo nível da *storyboard* é uma expansão detalhada” da região selecionada acima.

Cabe destacar que nessa abordagem a primeira *storyboard* — a visão mais crua — não é apenas substituída pela segunda, a visão mais detalhada. A primeira *storyboard* permanece visível e pode ser inspecionada, servindo como um *frame* de referência para a parte expandida.

O usuário, evidentemente, não precisa parar após o primeiro nível de expansão. Nesse exemplo, o usuário pode continuar a expandir com a lupa o carrossel através da aplicação da lupa em níveis sucessivos. A figura 2.5 mostra os efeitos da aplicação da lupa por 2 vezes. Note que os 6 frames da visão de baixo correspondem a uma parte muito pequena do vídeo (cerca de 8 segundos), mas numa taxa muito maior.

O primeiro benefício cognitivo do uso dessa interface é que o usuário não depende apenas do seu “olho da mente”, ou seja, ele não precisa lembrar quais cenas acontecem antes ou depois da parte que ele está examinando. Outro benefício é que a visão hierárquica não serve apenas para prover informações sobre o vídeo: ela também pode ser usada para controlá-lo. Se o usuário quiser executar um segmento do vídeo, ele não precisa apontar para um frame nem colocar o código do tempo que ele deseja. Ele apenas arrasta o mouse pelo trecho que deseja visualizar.

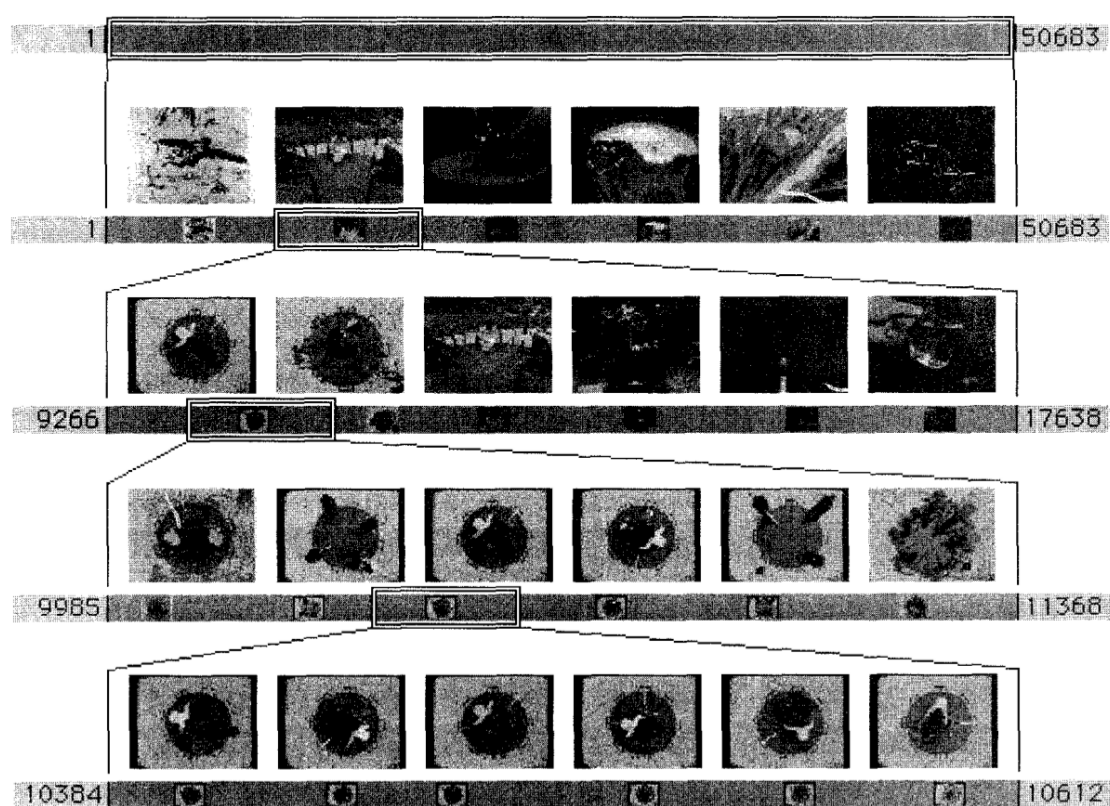


Figura 2.5: Refinamento sucessivo

3 FERRAMENTA DE SEGMENTAÇÃO

Neste capítulo, será apresentada a proposta e implementação de um protótipo de ferramenta para criação de metadados de segmentação de vídeo no padrão OBAA. Primeiramente é feita uma análise dos requisitos da ferramenta. Em seguida, é definido o ambiente implementado e as tarefas que o servidor da aplicação deve realizar. Em seguida, são apresentadas duas interfaces que são executadas do lado do cliente para que o usuário possa realizar a anotação dos vídeos e, para concluir, é demonstrada a integração da interface com o restante do sistema.

3.1 Visão Geral

O objetivo dessa ferramenta é que ela apresente a solução para segmentação de vídeo e que sua interface seja muito fácil de ser integrada aos repositórios já existentes. Basicamente o usuário deve informar os pontos de início e fim dos segmentos e uma etiqueta semântica para ele. A ferramenta, a partir desses dados, gera metadados no padrão OBAA.

O ambiente foi implementado como um protótipo de aplicação Web, seguindo uma arquitetura cliente-servidor. Esse formato foi escolhido porque a maioria dos repositórios educacionais seguem esse formato. A aplicação pode ser acessada pelos navegadores Web mais recentes que possuem suporte a HTML5. As novas versões dos principais navegadores, como Google Chrome, Firefox, Opera e Internet Explorer dão suporte a HTML5 video.

A implementação do lado do servidor foi desenvolvida com a utilização do *framework* web Grails, que utiliza a JVM (Máquina Virtual Java). O *framework* foi escolhido por ter

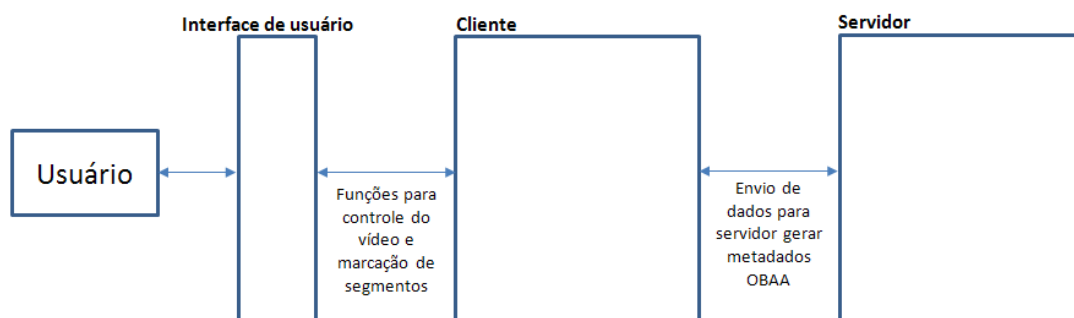


Figura 3.1: Visão geral da ferramenta: relação existente entre usuário, interface, cliente e servidor

arquitetura no padrão MVC (*Model-View-Controller*) e além de ser compatível com Java, possui os recursos da linguagem *Groovy*, uma linguagem dinâmica de alta produtividade.

No lado do cliente, foi utilizado HTML5, JavaScript e o *framework* jQuery, pois esse *framework* é interpretado da mesma forma em diferentes tipos de browser, ao contrário do JavaScript padrão, que varia muito a interpretação dependendo do navegador.

3.2 Requisitos

Como foi visto no capítulo de embasamento teórico, o padrão OBAA possui um grande conjunto de metadados, mas não é necessário se ocupar com a maioria deles nesse momento, pois os repositórios já oferecem suporte para que eles sejam preenchidos. A carência nos repositórios se dá na parte de segmentação de vídeo, ou seja na *SegmentInformationTable*, que é dividida em 2 partes: *SegmentList* e *SegmentGroupList*. A tabela *SegmentGroupList* serve para adicionar segmentos a grupos, ou seja, unir segmentos por uma propriedade comum. Não será dado suporte nesse trabalho à adição de grupos, somente a de segmentos isolados. Essa decisão se deve ao fato de que devemos oferecer ao usuário uma interface bastante simples e que seja fácil de usar. À medida que se vamos oferecendo mais recursos, a densidade informacional também vai aumentando e talvez ele fique muito complicado fazer uso de todos os recursos. Optamos, pois, por uma interface mais simples com um conjunto de metadados mínimo que se extraíam dados que agreguem valor a uma busca mais efetiva por vídeos. Nessa interface, o usuário deve marcar apenas o instante de início do segmento, o instante de fim do segmento e uma etiqueta para aquele trecho definido.

O restante dos metadados — *SegmentIdentifier* e *SegmentMediaType* — deverá ser preenchido pelo próprio repositório, então o usuário não deve se ocupar com esse tipo de informação.

Outro requisito importante é que a ferramenta deve ser Web, pois os repositórios tradicionalmente funcionam assim, e devem funcionar nos principais navegadores da atualidade — Google Chrome, Firefox e Internet Explorer. Ela deve preencher requisitos básicos de usabilidade, como facilidade de uso, facilidade de aprendizado, robustez e satisfação.

O usuário padrão que preenche os repositórios é uma pessoa com alto nível de escolaridade e está habituada com o uso de internet em computadores. Entretanto, os usuários não estão habituadas à manipulação de vídeo e a interface deve ser fácil o suficiente para que eles utilizem sem problemas. Devido a esse requisito retiramos o suporte à adição de grupos de segmentos.

Para que se possa avaliar e chegar a uma conclusão precisa sobre a forma como deve ser apresentada uma interface para esse tipo de usuário, foram projetadas 2 interfaces. A primeira é baseada nas interfaces vistas no estado da arte, que apresenta apenas uma timeline, botões para marcar o ponto de início e de fim e uma caixa de texto para digitar o valor da etiqueta. A segunda interface é desenvolvida a partir do protótipo visto no capítulo de embasamento teórico, que é teoricamente fundamentada mas que não foi validada empiricamente num contexto educacional.

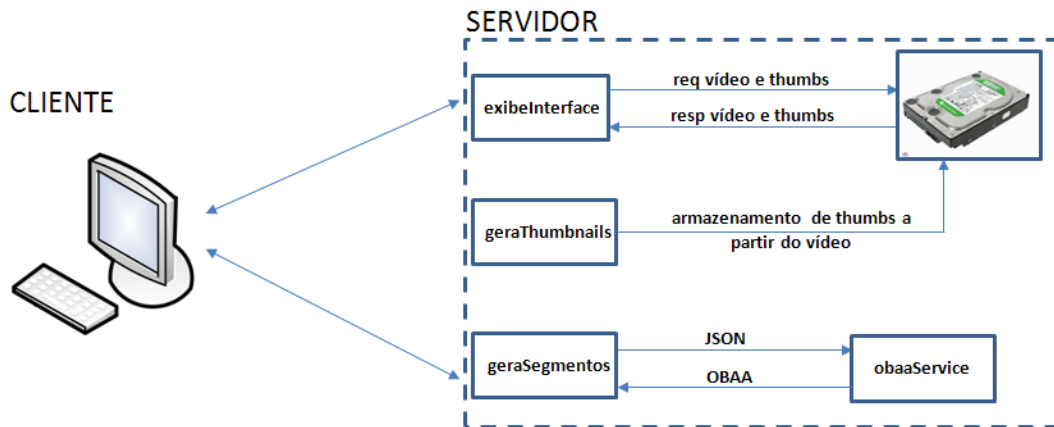


Figura 3.2: Visão geral do servidor: existe um método que extrai os *thumbnails* do vídeo e grava no disco, um método que busca no disco o vídeo e os *thumbnails* para o cliente e outro que recebe os dados de segmentação do usuário em JSON e gera metadados no padrão OBAA.

3.3 Servidor

O servidor da aplicação tem os seguintes objetivos: fornecer o vídeo para o usuário segmentar, gerar e fornecer os *thumbnails* que são exibidos na interface — seu uso será melhor detalhado na seção de cliente — e receber os dados enviados de cada segmento — pontos de início, fim e etiqueta semântica — e gerar metadados no padrão OBAA.

3.3.1 Geração de thumbnails

Quando um vídeo é enviado para o servidor, ele deve ser previamente processado para que sejam extraídos seus *thumbnails*, que serve para que o usuário possa se contextualizar melhor, conforme visto no capítulo de embasamento teórico. Nesse protótipo, o vídeo já é colocado diretamente no servidor — não será feito o upload do arquivo pelo usuário, pois essa tarefa já é suportada pelos repositórios. Porém, a extração das miniaturas não é suportada e é implementada pela ferramenta. Para isso, foi utilizado um programa chamado *ffmpegthumbnailer*, que extrai thumbnails de um vídeo e está disponível para uso no Linux. Esse programa é chamado por linha de comando e deve ser passado como parâmetro o endereço do vídeo, o instante de tempo que se quer extrair o *frame* e o tamanho do thumbnail. Então, é extraído 1 thumbnail para cada segundo do vídeo, e isso é feito com ShellScript. Primeiramente são extraídos quadros com 128 pixels de largura para que posteriormente seja formada a storyboard de mais alta definição e em seguida são extraídos os quadros com 43px de largura — ou seja, possui 1/3 dos quadros de maior definição — para formar a storyboard de menor resolução.

3.3.2 Criação dos metadados OBAA

Outra importante tarefa executada do lado do servidor da aplicação é o método que recebe os parâmetros enviados pelo cliente e cria os metadados no padrão OBAA. O cliente passa ao servidor, no formato JSON, um *array* de segmentos. Cada segmento é composto de três atributos: “início”, “fim” e “etiqueta”. O método deve, então, fazer um *parse* dos parâmetros para criar os objetos. No protótipo desenvolvido, o *parse* é feito com a classe *JsonSlurper* e os objetos são criados com uma biblioteca Java desenvolvida pela Cognitiva Brasil — empresa que possui um repositório que utiliza o padrão OBAA e oferece uma biblioteca *open-source* para isso — nas quais os metadados são facilmente criados através de objetos Java.

3.4 Cliente

Os objetivos da aplicação do lado do cliente são os seguintes: (1) oferecer um *player* de vídeo para que o usuário consiga ter o controle básico dele — funções de *play*, *pause* e trocar o instante de vídeo que é exibido —, (2) fornecer recursos para realizar a marcação de segmentos, armazenando os pontos de início, fim e etiqueta semântica dos trechos e (3) enviar ao servidor as informações dadas pelo usuário para que sejam gerados metadados no padrão OBAA.

3.4.1 Desenvolvimento do player

Para o desenvolvimento da aplicação do lado do cliente, não foi utilizado nenhum *player* de vídeo já existente, pois o desenvolvimento da aplicação exige radicais mudanças nos *players* de código aberto existentes, e o custo para aprender a utilizá-los e personalizá-los é maior que o desenvolvimento completo de um *player* simples. Optou-se, portanto, por desenvolver todo o código com uso apenas de HTML5 e jQuery.

Ambos os protótipos de interface partem do mesmo *player* básico: uma tela onde é exibido o vídeo, com botões de controle *play/pause*, uma *timeline* navegável que serve tanto para feedback do instante do trecho que é executado quanto para navegação — para o usuário ir para o instante do vídeo que deseja. Para isso, utiliza-se a tag “*video*” padrão do HTML5, que reproduz o vídeo. Nessa tag, são informadas duas fontes para o vídeo: uma em MP4, que é suportada pelo Internet Explorer e Google Chrome e outra em WebM para que seja reproduzida no Firefox. Dessa forma, o requisito que a aplicação seja suportada por diferentes browsers é atendida.

A tag “*video*” oferecida pelo HTML5 possui métodos que dão o controle sobre o vídeo, como “*play*”, “*pause*” e “*currentTime*”. Para que o usuário tenha controle sobre essas ações — de iniciar, parar e alterar o instante de execução do vídeo —, basta vincular um objeto ao método com funções do jQuery. Outro importante método oferecido é o “*timeupdate*”, que é constantemente invocado quando o tempo de execução do vídeo muda. Com ele, podemos atualizar a *timeline* à medida que o tempo avança para dar o feedback ao usuário.

Com o *player* básico implementado, podemos avançar para a interface de segmentação do vídeo em si.



Figura 3.3: Protótipo de interface baseada no estado-da-arte.

3.4.2 Interface de botões

A primeira interface proposta é baseada nas interfaces vistas no estado-da-arte. Ela possui além das funcionalidades básicas do *player* padrão, botões em que o usuário pode capturar o instante de tempo do vídeo que é exibido, como na figura 3.3.

Nessa interface, existem botões e para marcar o ponto de início do segmento e o ponto de fim de segmento. Esses botões capturam o instante de vídeo que é executado no instante para que posteriormente seja adicionada uma etiqueta ao segmento. Desse modo, o usuário pode escolher o ponto do vídeo que quer marcar na *timeline* e selecionar esses botões. Caso ele não consiga no primeiro momento selecionar o exato instante que deseja, existem botões de refinamento: botões para avançar e recuar os pontos marcados.

Quando o usuário já marcou o segmento a que deseja aplicar uma etiqueta, ele pode acionar o botão “Executar trecho”, que reproduz o trecho que ele marcou. Essa funcionalidade é oferecida para que ele possa conferir se a marcação ficou adequada. Para concluir a marcação, o usuário deve selecionar “Aplicar tag” e então digitar uma etiqueta para o trecho.

À medida que o usuário preenche os segmentos, as informações das marcações ficam guardadas em variáveis no lado do cliente. Quando ele finaliza a marcação, esses dados são enviados ao servidor como parâmetro e um XML é devolvido.



Figura 3.4: Protótipo de interface com storyboard. Tela inicial.

3.4.3 Interface com storyboard

A interface que possui a storyboard é baseada no protótipo visto no capítulo de embasamento teórico. Supostamente, elas ajudam o usuário a se contextualizar melhor no vídeo e oferece ao usuário a possibilidade de fazer um refinamento iterativo: ele pode aplicar sucessivos cortes — primeiramente menos exatos — e ir refinando até chegar num corte preciso.

Nessa interface, não existem botões para segmentar o vídeo. Em vez disso, miniaturas de quadros extraídos do vídeo são exibidas lado a lado formando uma storyboard. Essa storyboard é exibida em duas camadas: uma camada é feita com 7 quadros em maior resolução com frames mais espaçados temporalmente e outra com 21 frames em menor resolução e menos espaçados temporalmente. Acima da storyboard, existe uma *timeline* que corresponde ao segmento marcado e acima dela existe uma outra *timeline* que corresponde ao tempo total do vídeo, como se pode ver na figura 3.4. Na tela inicial, antes que seja feito algum corte para segmentar o vídeo, ambas *timelines* são iguais, ou seja, correspondem à mesma fatia de tempo, que é o vídeo inteiro.

O usuário deve, então apontar na *storyboard* o instante de início do segmento e apontar até o fim do segmento. Quando ele faz isso, é destacada numa cor mais clara a parte em que ele está selecionando, com intuito de dar o *feedback* imediato ao usuário.

Quando o usuário solta o botão do mouse, é feito um corte no vídeo. A storyboard que ele usou para realização desse corte desloca-se para baixo e surge uma nova *storyboard* que corresponde exatamente com o trecho que ele marcou. A opção por manter a storyboard mais recente acima — e não abaixo, como seria mais natural — se deve ao fato de manter o agrupamento dos objetos: a *storyboard* fica próxima à *timeline* e à tela onde é exibido o vídeo e, assim, o usuário não perde o foco e nem precisa rolar a tela para cima ou para baixo, que aconteceria se o usuário realizasse a segmentação após muitas iterações.

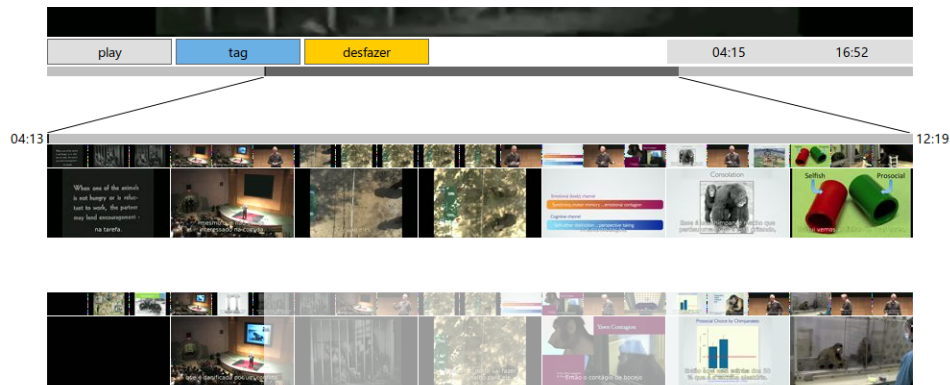


Figura 3.5: Protótipo de interface com storyboard após o primeiro corte.

Para que a nova storyboard surgisse exatamente no lugar da anterior e o usuário percebesse intuitivamente que ela foi substituída, foi feito um efeito de deslocamento da storyboard e o efeito de *fade-in* da nova storyboard.

Aplicado o corte, é destacada na *timeline* mais geral a parte que foi cortada, para que o usuário possa se situar no contexto do vídeo inteiro. Linhas que ligam a storyboard à parte destacada nessa timeline reforçam a idéia de contexto.

Para concluir os elementos da interface, existe também um botão para desfazer o corte — para que o usuário tenha controle total da aplicação e possa se recuperar de eventuais cortes mal feitos — e o botão de aplicar tag, para que surja uma caixa de diálogo e o usuário possa aplicar uma etiqueta ao trecho marcado.

Quando o usuário encerra suas atividades, uma requisição com os segmentos da mesma forma como foi visto na seção 3.4.2.

3.5 Integração da interface o sistema

Nesta seção, será vista a forma como são gerados os dados pela aplicação cliente a partir da marcação dos usuários através da interface e como são gerados os metadados do padrão OBAA.

A marcação dos segmentos é feita pelo usuário, conforme foi descrito anteriormente. A aplicação cliente armazena esses dados em um *array* e quando o usuário finaliza a marcação, é gerado um JSON com as informações anotadas. Por exemplo, se o usuário visualizou um vídeo de 10 minutos (ou seja, de 600 segundos) e marcou dois segmentos, sendo o primeiro segmento com a etiqueta “Metade inicial”, que tem início no segundo 0 e fim no segundo 300 e o segundo segmento com a etiqueta “Metade final”, com início no segundo 301 e fim no segundo 600, o JSON gerado é o seguinte:

```
[
  {
    "start":0.0,
    "end":300.0,
    "etiqueta":"metade inicial"
  },
  {
    "start":301.0,
    "end":600.0,
    "etiqueta":"metade final"
  }
]
```

O servidor, então, recebe esse JSON, efetua o *parse* dele e instancia uma classe OBAA. A partir dessa classe, é gerado um XML conforme foi descrito no capítulo 2 da seguinte forma:

```
<obaa:obaa xsi:schemaLocation="http://ltsc.ieee.org/xsd/LOM
http://ltsc.ieee.org/xsd/obaav1.0/lom.xsd"
xmlns:obaa="http://ltsc.ieee.org/xsd/LOM"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <segmentInformationTable>
    <segmentList>
      <segmentInformation>
        <obaa:identifier>0</obaa:identifier>
        <obaa:title>metade inicial</obaa:title>
        <obaa:description/>
        <obaa:segmentMediaType>video</obaa:segmentMediaType>
        <obaa:start>0.0</obaa:start>
        <obaa:end>300.0</obaa:end>
      </segmentInformation>
      <segmentInformation>
        <obaa:identifier>1</obaa:identifier>
        <obaa:title>metade final</obaa:title>
        <obaa:description/>
        <obaa:segmentMediaType>video</obaa:segmentMediaType>
        <obaa:start>301.0</obaa:start>
        <obaa:end>600.0</obaa:end>
      </segmentInformation>
    </segmentList>
  </segmentInformationTable>
</obaa:obaa>
```

```
</segmentList>  
<segmentGroupList/>  
</segmentInformationTable>  
</obaa:obaa>
```

Nesse XML, podemos notar que é gerado um grupo de informações *segmentInformation* para cada segmento. Em cada um deles, existe um identificador único, o título do segmento (que corresponde à etiqueta semântica aplicada pelo usuário), o tipo de dado (nessa ferramenta é sempre vídeo), o ponto inicial e o ponto final. O campo de descrição não é preenchido pela ferramenta, pois na implementação da interface optou-se pela não utilização desse recurso.

3.6 Considerações finais

Nesse capítulo foi apresentado então a implementação de uma ferramenta cliente-servidor que dá ao usuário a possibilidade da aplicação de etiquetas semânticas a segmentos de vídeo. Vimos duas interfaces que dão recursos para que o usuário faça as anotações e que enviam ao servidor esses dados no formato JSON para que sejam gerados os metadados no padrão OBAA.

A geração de metadados OBAA foi feita utilizando uma biblioteca Java, em que se instancia um objeto da classe OBAA. Essa classe oferece métodos correspondentes aos metadados do padrão para que eles sejam setados. Desse modo, a integração com o padrão é feita através de um *parser* dos dados recebidos pelo cliente e instanciação do objeto OBAA.

4 AVALIAÇÃO EXPERIMENTAL

A avaliação experimental consiste na comparação das duas interfaces propostas, utilizando *A-B testing*. A escolha desse tipo de avaliação foi baseada no fato de que ela é ideal para identificar mudanças em páginas web que melhoram ou pioram o interesse dos usuários (Dias, 2013). Nesse caso específico, o objetivo é avaliar se a implementação de uma interface de usuário mais rica, como a do protótipo estudado, melhora a experiência de usuário em relação às interfaces mais simples estudada no estado da arte.

4.1 Preparação dos experimentos

Para que as interfaces pudessem ser comparadas adequadamente, estabeleceu-se que cada usuário deve utilizar ambas interfaces. A composição da amostra de grupos utilizadas foi “*within-subjects*”, em que todos os usuários executam testes em ambas interfaces e a ordem com que elas são apresentadas é aleatória. Cerca de 50% dos usuários avaliam a interface A antes da B e cerca de 50% utilizam a interface B antes da A, com intuito de eliminar o efeito da ordem de execução.

Cada usuário, então vê um vídeo e experimenta cada uma das interfaces. O vídeo escolhido foi a palestra do professor Frans de Waal intitulada “Comportamento moral em animais¹”. Esta é uma palestra de cerca de 17 minutos em que o professor afirma que valores considerados humanos, como “cooperação” e “justiça” também são observados em animais. O vídeo apresenta, portanto, conteúdo educacional e algumas trocas de cenário — o professor expõe alguns vídeos em sua palestra — pois grande parte dos vídeos educacionais apresentam mudanças de cenário também, como troca de cenário ou variações no quadro negro. O vídeo foi colocado no servidor da aplicação web desenvolvida no trabalho e os *thumbnails* previamente processados através do método visto no Capítulo 3.

Na realização dos experimentos com usuário, os dados coletados são colocados em um banco de dados para que sejam computados posteriormente. Nessa avaliação, não são gerados os metadados no padrão OBAA, pois a avaliação refere-se somente à interação dos usuários com as duas interfaces propostas, e não à criação dos metadados em si.

O objetivo é fazer o teste com 10 usuários para se ter uma amostra suficiente para apontar os pontos fortes e fracos de cada interface e, se possível, afirmar qual delas é mais adequada para o conteúdo educacional.

¹http://www.youtube.com/watch?v=b_Lm49XVkgQ

4.2 Protocolo de testes

Para cada usuário é solicitado que responda um questionário prévio para caracterização. As perguntas feitas são “Idade”, “nível de escolaridade”, “familiaridade com uso de internet” e “experiência com manipulação de vídeo”, para que sejam avaliados somente usuários que se assemelham aos usuários reais de repositórios educacionais.

Em seguida, o usuário faz um treinamento com cada interface, solicitando que ele utilize os componentes de cada uma de forma adequada: basicamente mostrando a ele a forma de marcação dos segmentos com os botões e como se marcam os segmentos através do apontamento direto nos *thumbnails*.

Após estarem familiarizados com ambas interfaces, o usuário assiste ao vídeo educacional para que posteriormente possa marcar o trecho e aplicar as etiquetas semânticas.

Para cada interface, é solicitado que o usuário marque três segmentos

- um segmento de curta duração, com cerca de 2 segundos
- um segmento de média duração, com cerca de 30 segundos
- um segmento de longa duração, com cerca de 1 minuto e 30 segundos

À medida que o usuário vai marcando o vídeo, seus dados — início, fim e etiqueta semântica — são armazenados em banco de dados para que seja feita uma avaliação estatística posterior sobre a precisão dessas marcações: se existe alguma interface que permita ao usuário marcar segmentos mais precisos que outra.

Para concluir, é feito outro questionário, com intuito de se extrair respostas mais subjetivas. Essas perguntas são

- É melhor usar a primeira interface que a segunda?
- A storyboard com miniaturas ajuda o usuário a se situar no contexto do vídeo?
- O refinamento iterativo ajuda a criar segmentos mais precisos?
- É melhor clicar em botões do que arrastar para marcar os segmentos?
- Qual interface você preferiria para vídeos de 2 minutos?
- Qual interface você preferiria para vídeos de 10 minutos?
- Qual interface você preferiria para vídeos de 1 hora?
- Se você quisesse etiquetar assuntos de uma palestra em partes, qual interface você preferiria?
- Se você tivesse que etiquetar as cenas de luta em um filme de 2 horas, qual das interfaces você preferiria?
- Se você assistir a uma palestra sobre um livro, que é dividida em capítulos, qual das interfaces você preferiria?
- Comentários

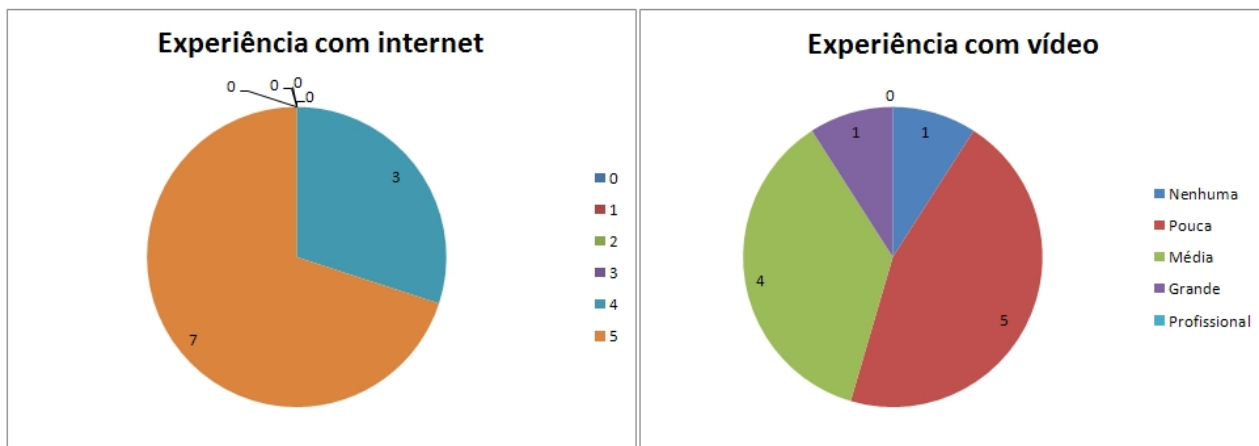


Figura 4.1: Na esquerda, os usuários dizem ter alto conhecimento de internet e na direita dizem ter média ou pouca experiência com vídeo.

4.3 Resultados dos experimentos

Serão avaliados primeiramente os resultados subjetivos, a partir da opinião dos usuários e posteriormente será feita uma avaliação objetiva, obtida a partir da observação de dados obtidos através de *logs*.

4.3.1 Avaliação subjetiva

Sobre os resultados do questionário de caracterização dos usuário, podemos destacar o alto nível de escolaridade — todos usuários possuem pelo menos o ensino médio completo e a maioria possui curso de graduação. Além disso, conforme podemos observar na Figura 4.1, todos usuários dizem ter bastante experiência com o uso de internet (70% se deu nota 5, a máxima, e 30% se deu nota 4) e a maioria diz ter pouca experiência com manipulação de vídeo. Somente 10% das pessoas dizem ter alto conhecimento de manipulação de vídeo, 40% dizem ter experiência média e 60% dizem ter pouca experiência. Esse resultado já era esperado e o questionário foi feito com o intuito de descartar da avaliação as pessoas que fossem muito diferentes dos usuários reais. Nenhum usuário foi descartado, pois foram procuradas justamente pessoas com essas características.

Sobre os dados obtidos do questionário feito após a segmentação do vídeo, com questões subjetivas dos usuários sobre as impressões obtidas acerca do experimento, serão apresentados primeiramente os pontos em que os usuários convergem sua opinião e depois serão apresentados os pontos em que eles divergem.

Foi feita a pergunta “A *storyboard* com miniaturas ajuda o usuário a se contextualizar melhor?”. Todos os participantes afirmaram que ajuda. 70% dos usuários responderam que ajuda muito e 30% deles responderam que ajuda um pouco, conforme a figura 4.2. Com isso, podemos afirmar que o apelo visual para a marcação de vídeos é essencial.

Também foi feita a pergunta “O refinamento iterativo ajuda a criar segmentos mais precisos?”. 90% dos usuários afirmaram que ajuda, enquanto somente 10% afirmam que não. Desses usuários que responderam afirmativamente, 77% dizem que ajuda muito e 33% afirmam que ajuda um pouco.

Essas duas primeiras questões referiam-se às características implementadas sobre a interface estudada no protótipo visto no capítulo de embasamento teórico. As respostas dos usuários comprovam que essas características se aplicam ao contexto educacional e

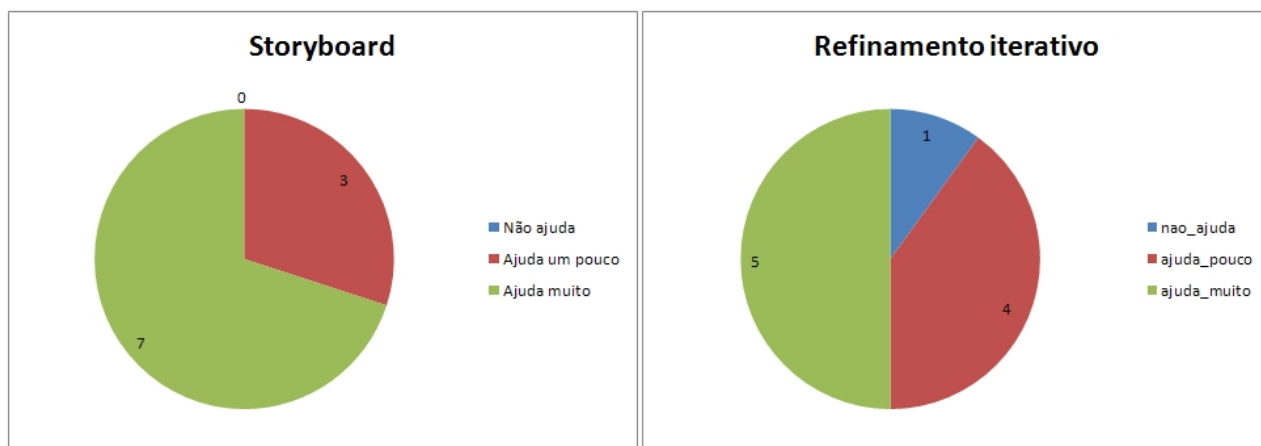


Figura 4.2: Na esquerda, os usuários afirmam que a *storyboard* ajuda o usuário a se contextualizar e na direita afirmam que a possibilidade de refinar a marcação aos poucos também ajuda.

que as interfaces devem possuir essas características.

Perguntamos aos usuários qual interface eles preferiram para segmentar o vídeo no teste. Nessa pergunta, houve uma discrepância enorme. Conforme a figura 4.3, podemos observar que 30% das pessoas dizem que não existe uma interface melhor que a outra. 30% dizem que a interface com *storyboard* é melhor e 40% dizem que a interface do botões é um pouco melhor.

Perguntamos aos usuários se eles preferem marcar os segmentos arrastando o mouse sobre a *storyboard* ou utilizando botões. A maioria dos usuários diz que a utilização de botões é mais adequada para marcação de segmentos. 60% diz que é um pouco melhor utilizar botões, 20% diz que é melhor arrastar e 20% diz que é indiferente. Cabe destacar aqui que nesse quesito, a interface simples estudada no estado-da-arte leva vantagem.

Grande parte das pessoas diz que a interface que apresenta botões é melhor que a interface em que se marca os segmentos diretamente na *storyboard*. A *storyboard* foi bem avaliada pelos usuários, como vimos, mas acreditamos que esses usuários preferiram a utilização de botões porque estão mais habituadas a manuseá-los em outras aplicações. Quando usuários vêem os botões, eles sabem imediatamente como proceder, ao contrário do modelo de marcação de segmentos através da marcação na *storyboard*, que parece ser menos intuitiva. Deve-se, portanto, tentar obter uma interface de melhor usabilidade, com a combinação do uso de botões com o recurso de visualização da *storyboard*.

Na sequência, foram feitas diversas perguntas aos usuários dando 2 opções a eles: se eles preferem utilizar a interface de botões ou a interface com *storyboard* para diversos casos. Para vídeos mais curtos, como pode-se observar no gráfico 4.4, os usuários preferem utilizar a interface de botões, que não oferece a possibilidade de refinamento. Entretanto, para vídeos mais longos, em que a *timeline* fica menos precisa, os usuários dizem preferir a interface de *storyboard*, que apresenta a característica de refinamento temporal.

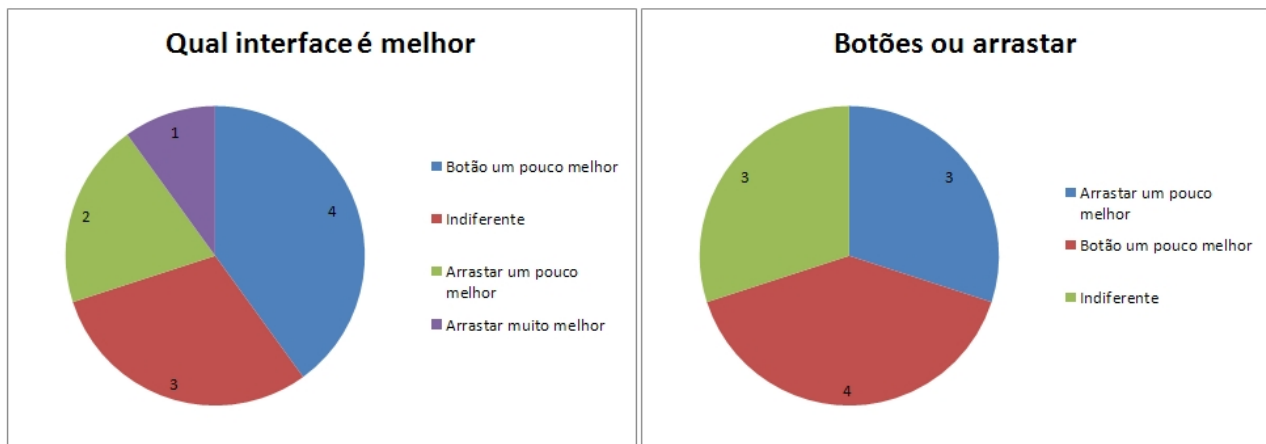


Figura 4.3: Na esquerda, 40% dos usuários dizem que que a interface de botões é melhor. 30% diz que não existe uma interface melhor que a outra, 20% diz que a interface com storyboard é um pouco melhor e 10% diz que ela é muito melhor. Já na direita, vemos que somente 30% diz que é melhor selecionar trechos arrastando o mouse, 40% diz que utilizar botões é melhor que arrastar e 30% diz que é indiferente.

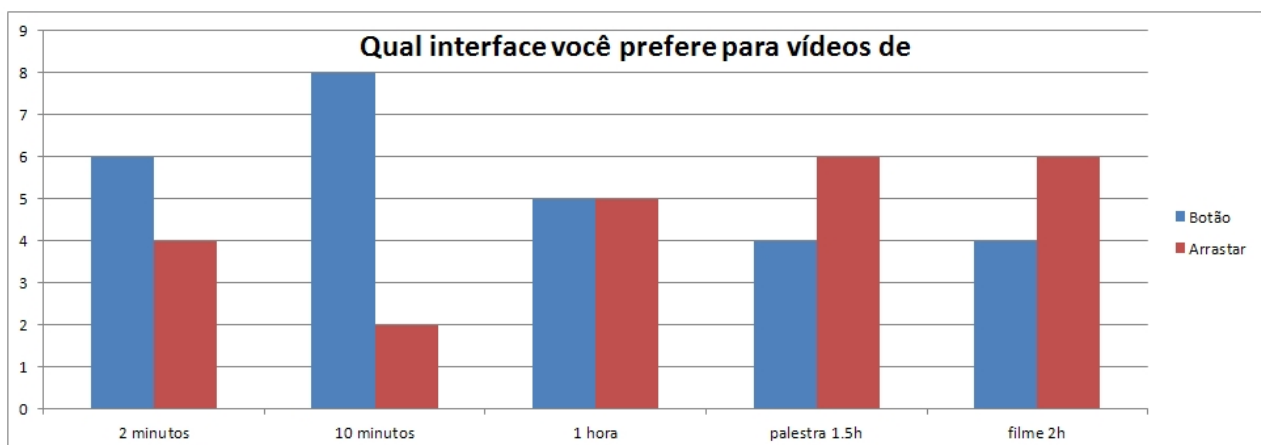


Figura 4.4: Observamos que para vídeos curtos, os usuários preferem utilizar a interface com botões, sem refinamento. Já para vídeos mais longos, eles preferem a interface com possibilidade de refinamento.

4.3.2 Avaliação objetiva

Nos testes, ainda foi feito o *log*, com a marcação de cada segmento para medir a precisão. Para cada um dos 6 segmentos marcados — foram 3 segmentos marcados com cada interface — foi armazenado num banco de dados os pontos de início, fim e etiqueta aplicada a cada trecho.

Para medir a precisão, para cada um dos segmentos foi estabelecido previamente um instante considerado “correto” para início e fim daquele segmento. Então para cada segmento marcado pelo usuário, foi calculada a diferença do ponto marcado pelo usuário e o ponto considerado correto. Por exemplo, se for solicitado ao usuário marcar o trecho de “alimentação dos macacos”, e consideramos que esse trecho inicia aos 15 segundos do vídeo e encerra aos 20 segundos e o usuário marca que inicia aos 14 segundos e encerra aos 17 segundos, afirmamos que ele errou por 1 segundo o início e errou por 2 segundos o fim.

Nesse teste, aconteceu o aparecimento de *outliers*, os pontos fora da curva. Esses pontos foram descartados se o erro do usuário é superior a 15 segundos. Isso foi feito porque acreditamos que, se o usuário errou por tanto tempo, isso não ocorreu devido à interferência da interface em si, e sim ocorreu por algum erro de interpretação ou de comunicação.

A partir disso, calculamos a média e o desvio padrão dos pontos obtidos por cada interface, como podemos observar na tabela 4.1. Observamos que na interface em que o usuário deve utilizar a *storyboard* para marcar os segmentos, o erro médio foi de 5,88 segundos e o desvio padrão foi de 9,78 enquanto na interface que possui somente a *timeline*, em que o usuário marca os trechos com botões o erro e desvio foram muito menores, de 2,6 segundos e 3,43 segundos, respectivamente.

Com isso, podemos concluir que os usuários conseguiram utilizar de maneira mais adequada a segunda interface, que é mais simples e comunica-se melhor com o usuário. O uso de recursos mais avançados da primeira interface, como a possibilidade de ir refinando aos poucos não ajudou, na prática, a criar segmentos mais precisos.

	Média de erro	Desvio padrão do erro
Interface com <i>storyboard</i>	5,88s	9,78s
Interface somente com <i>timeline</i>	2,6s	3,43s

Tabela 4.1: Média e desvio padrão do erro para cada interface.

5 CONCLUSÃO

A tecnologia sempre esteve presente no contexto educacional, sempre oferecendo técnicas para facilitar o aprendizado. Diversos repositórios educacionais surgiram com intuito de armazenar o conteúdo e facilitar seu reuso, aumentando o alcance que cada objeto educacional possui. Os conteúdos, que antes se resumiam basicamente a textos, com a disponibilidade de maiores recursos computacionais, evoluíram para objetos multimídia, como textos e vídeos e deve-se evoluir as ferramentas que são utilizadas para manipulação de conteúdo.

Existem diversos padrões de metadados para conteúdo educacional que evoluíram ao longo do tempo: desde padrões mais rústicos e simples até os novos padrões, que dão suporte a novas funcionalidades, como o suporte à segmentação de vídeos, oferecido pelo padrão OBAA. Além do suporte ao novo tipo de conteúdo multimídia, também deve-se atentar para a forma com que esses conteúdos são manipulados, a fim de que o usuário possa utilizar esses recursos de maneira simples.

Foi proposta nesse trabalho uma ferramenta que possibilita que usuários comuns, de grande experiência com uso de computadores mas sem muita experiência com manipulação de vídeos pudessem aplicar etiquetas semânticas a vídeos educacionais. A ferramenta coleta esses dados e cria os metadados propostos pelo padrão OBAA. Foram propostas, para essa ferramenta, 2 interfaces — uma interface bastante simples, com botões e uma *timeline* e outra com recursos mais complexos, como *storyboard* e possibilidade de refinar segmentos iterativamente — e foram avaliadas através de experimentos com usuários. Observamos que precisamos evoluir também na forma de interação, com utilização desses recursos, mas que devemos manter os recursos mais simples e intuitivos para o usuário, como o uso de botões.

Para trabalhos futuros, sugere-se a realização de melhorias na implementação das interfaces, como a troca dos frames igualmente espaçados na *storyboard* por *keyframes* e a tentativa de fazer uma interface utilize os recursos avançados mas que mantenha o uso dos recursos mais como uso de botões. Além disso, sugere-se que o padrão OBAA avalie uma forma de adicionar suporte ao armazenamento de miniaturas para conteúdo multimídia.

REFERÊNCIAS

DIAS, Alessandro da Silveira **Aumentando a Acurácia de Predição de Sistemas de Recomendação de Vídeo com o Uso de Pontos de Interesse**. 2013

JAY, Michael **LRMI Background and State of the Spec**. 2013. Disponível em <<http://www.slideshare.net/edpublishers/jay-cic-lrmipreso130605>>

LTS Commitee. **Draft Standard for Learning Object Metadata**. Disponível em <http://ltsc.ieee.org/wg12/files/LOM_1484_12_1_v1_Final_Draft.pdf>. 2004. Acessado em 27/05/2013.

MACKAY, Wendy **EVA: An experimental video annotator for symbolic analysis video data**. 1998

McGREAL, Rory. **Learning Objects: A Practical Definition**. Disponível em <<http://auspace.athabascau.ca/bitstream/2149/227/1/Practical%20definition.doc>>. 2004. Acessado em 28/05/2013.

MILLS, Michael et al. Human Interface Group/Advanced Techonology **A magnifier Tool for Video Data**. California, 1992

RICHARDS, Griff. **The Evolution of Learning Object Repository Technologies: Portals for On-line Objects for Learning**. Disponível em <<http://www.jofde.ca/index.php/jde/article/download/297/193>>. 2002. Acessado em 28/05/2013.

VICARI, Rosa Maria et al. Relatório Técnico RT-OBAA-01 **Proposta de Padrão para Metadados de Objetos de Aprendizagem Multiplataforma**. In: ROESLER, Valter. Metadados de segmentação, p. 38, 2009.

VICARI, Rosa Maria et al. **The OBAA Proposal for Learning Objects Supported by Agents**. 2010. Disponível em <http://www.portalobaa.org/padrao-obaa/artigos-publicados/aamas2010_FINAL.pdf/at_download/file>