

Avaliação de software educacional centrada no diálogo¹

Flávia Peres²

flaviapr@ufpe.br - UFPE

Luciano Meira

lrlmeira@ufpe.br- UFPE

Resumo

Este artigo apresenta um modelo de avaliação de software educacional baseado na análise dos diálogos entre usuários durante o uso de um software de matemática. Quatro duplas de alunos de 6^a série do ensino fundamental foram videografadas enquanto resolviam questões de aritmética propostas pelo programa. Usando técnicas da Análise da Conversação, focalizamos as "quebras" nos tópicos emergentes nos diálogos das crianças em interação com o software. O modelo proposto considera a relação entre as características do design da interface e a aprendizagem decorrente da colaboração entre usuários.

Palavras-chave: Avaliação de software educacional. Interação humano-computador. Análise da Conversação.

A dialogue-based approach for evaluating educational software

Abstract

This paper presents a model of software evaluation based on the analysis of dialogues among users as they explore math software. Four pairs of 6th graders were videotaped while solving arithmetic puzzles and operations, and their talk and actions at the computer transcribed. Using techniques borrowed from Conversation Analysis, we then looked for "breaks" in the emergent dialogue among the children as well as mismatches between the children's actions and the software feedback. The suggested model integrates a user-centered approach based on talk and action and more traditional views of software evaluation based on features of the interface design.

Keywords: Educational software. Software evaluation. Interface design. Conversation Analysis.

¹ Original em Inglês: Meira, L.; Peres, F. (2004). A dialogue-based approach for evaluating educational software. *Interacting with Computers*, 16, 615–633.

² Departamento de Psicologia - Universidade Federal de Pernambuco – CEP 50670-901 - / [081] 32721599.

Introdução

Decorrente do discurso social de valorização da informática educativa na escola, o computador e o software educacional ganharam grande destaque como instrumento para a construção e desenvolvimento de conceitos científicos. Surgem, então, novas configurações no processo de ensino-aprendizagem. Mas, é claro, a adoção de computadores pelas escolas não garante, por si só, uma melhoria nos processos de ensino e aprendizagem. Essa melhoria dependerá, principalmente, das atividades de uso associadas ao software e dos recursos computacionais efetivamente utilizados em tais atividades.

Entre os problemas que circundam a temática da informática na educação, temos aqueles sobre como avaliar um software educacional, ou seja, como qualificar o adjetivo educacional de um software ou como estabelecer que uma interface é mais ou menos adequada para uso educacional.

No presente trabalho, lançaremos luz sobre tal problemática, ao propormos uma alternativa de avaliação de software educacional baseada na Análise da Conversação. Partimos de uma perspectiva dialógica que tanto está centrada nas ações dos usuários, como considera as estruturas do *design* da interface, e a inter-relação entre ambas.

Com nossas idéias, buscamos orientar o aperfeiçoamento de interfaces educacionais mais ricas para a aprendizagem, auxiliar implementações de ferramentas com representações conceituais mais consistentes e considerar o *design* para uma interação com fins de construções conceituais.

Neste artigo descrevemos nossos esforços para desenvolver tal modelo de avaliação centrada no diálogo. Na primeira seção especificamos peculiaridades referentes à temática de avaliação de software educacional e nossas linhas diretrizes:

- Estudos sobre *design* de interface, que nos permitiram pensar melhor sobre as seleções de padrões específicos pelo autor do software (princípios e *guidelines*);
- Estudos da Análise da Conversação, detalhando alguns aspectos e estratégias desse campo fundamentais para a compreensão da nossa proposta.

Prosseguiremos com dois exemplos retirados de nossas análises com o uso de um software específico de matemática; e finalmente apontamos para os passos da elaboração do modelo descrito.

Avaliação de software educacional

Avaliação de software educacional é um tema que tem atraído vários interesses. Esses interesses transitam entre diferentes áreas que, motivadas por tal idéia comum, criam um novo campo interdisciplinar cuja interseção ainda não possui características definidas claramente. Na verdade, o que tem acontecido diante desse quadro é que, ao avaliar um software educacional, ora a ênfase recai sobre os aspectos da programação propriamente dita, ora priorizam-se as questões conceituais e de aprendizagem, ou ora, ainda, enfatizam-se os aspectos do *design*.

Por se tratar de software com finalidades educacionais, todos esses campos têm relevância e afetam direta ou indiretamente a aprendizagem dos conceitos específicos pretendidos ou as situações de ensino. Ou seja, se o software apresenta problemas de programação, provocando falhas ou "bugs" durante o uso, isso resultará consequentemente em quebras e decontinuidades no raciocínio do usuário e interromperá o fluxo do pensamento no processo de aprendizagem. Além disso, aspectos da interface que contrariem alguns princípios de *design* poderão também dificultar processos de uso. Logo, a aprendizagem será afetada. Mais diretas ainda sobre a construção conceitual pelo aluno-usuário são as conseqüências decorrentes da forma como os conceitos estão implementados no programa, suas representações e as concepções de usuário que estão implícitas em sua estruturação e apresentação.

Avaliar um software educacional é, portanto, um trabalho que requer um olhar criterioso e um modelo que permita contemplar os aspectos mencionados acima de forma integrada, sem esquecer o objetivo decorrente dessa integração: a aprendizagem de conceitos específicos.

Diversos campos e áreas profissionais têm interesse no desenvolvimento e avaliação de softwares educacionais, das ciências da computação à psicologia, da educação ao *design*. Enquanto é necessário tal campo interdisciplinar, a diversidade de objetivos, *backgrounds* e conceitos envolvidos (freqüentemente de modo implícito) tem também gerado um cenário confuso no qual o foco de interesse varia, muitas vezes sem justificativa, de estruturas da programação a aspectos do processo de aprendizagem do usuário, e de objetivos didáticos em uma área particular aos aspectos de *design* da interface.

Como um caminho para essa complexidade, nós decidimos investigar não as estruturas do software diretamente, mas o diálogo que emerge entre os usuários enquanto engajados com um ambiente do software. Em particular, escolhemos focalizar as seqüências discursivas, seguindo as *quebras de tópico* em tais diálogos. De acordo com os analistas da conversação (SCHIFFRIN, 1997; POMERANTZ, 1997; SACKS, 1984;

LEVINSON, 1983) qualquer diálogo é estruturado por contribuições discursivas alternadas em torno de um tópico específico, e vários mecanismos conversacionais (e.g., pares adjacentes, marcadores de tomada de turno) são empregados pelos falantes para manter estabilidade tópica, coerência e inteligibilidade mútua. Focalizar os momentos em que um tópico conversacional muda, relacionando-os às ações do usuário e aos sucessivos estados do software é, assim, um importante recurso de informação sobre como as estruturas do software ajudam os usuários a manter a coerência conversacional sobre o tópico. Mais importante ainda, argumentamos que uma detalhada análise da qualidade das mudanças de tópico devido à quebras permitem-nos determinar a relativa contribuição dos usuários e do software para o fluxo da ação.

Para ilustrar nossa visão sobre o assunto, apresentaremos nesse artigo uma análise de diálogos entre estudantes de 6ª série enquanto trabalhavam com um software de matemática chamado *Calcule!*[©], cujo objetivo é desenvolver nos usuários o entendimento de conceitos em aritmética. Em nossa análise, o software funciona como um meio material no contexto do qual usuários podem acessar objetos e processos pretendidos por um terceiro participante: o autor do software.

O autor do software constrói diálogos com um usuário através de estrutura tópica projetada no software, como em uma narrativa (a qual no caso do software educacional fala sobre como um conceito específico pode ser organizado). Este diálogo é realizado através de vários caminhos, que podem variar no modo como ícones e mensagens aparecem na interface a como conceitos são significados e representados no comportamento do software. Então, nosso objeto de análise não é nem o software nem o usuário sozinho, mas a atividade comunicativa que toma lugar entre usuários e entre usuários e software, cuja interface e comportamentos representa as contribuições do autor ao diálogo. A interface do software pode ser vista como desempenhando um papel semiótico, e assumindo o especial *status* de um parceiro comunicativo na interação, dando suporte ao mesmo tempo em que limita a compreensão de cada usuário e dos conceitos implementados no software.³ Neste sentido, a interface age como se ela fosse ainda um outro interlocutor, que entra em um tipo particular de dinâmica comunicativa com os usuários.

Os caminhos que o autor do software desenvolve e implementa qualquer conceito (estruturando a interface e o comportamento do software de acordo com tal conceito)

³ Pesquisas no campo da engenharia semiótica (e.g., DE SOUZA, 1993) concebem também a interface como sendo constituída de mensagens estruturadas do designer para o usuário. Esta interpretação é próxima da nossa, ao atribuímos um “papel semiótico” à interface. Como vemos, contudo, a perspectiva da engenharia parece recriar uma antiga dicotomia emissor-receptor, que queremos evitar ao assumirmos as idéias de Bakhtin e seu círculo.

estará refletida na fala dos usuários quando eles se engajam na atividade com o software. Isto faz sentido, desde que, como já pontuamos, o entendimento das pessoas pode ser identificado através do diálogo (alguns diriam, é diálogo!). Além do mais, como temos argumentado, o software entra nesse diálogo através dos significados pretendidos, projetados pelo autor na forma de ícones e de objetos da interface, estruturas do conteúdo, *feedback* etc. Em suma, nossas observações teóricas e empíricas sobre a dinâmica do diálogo entre usuários, o autor e a interface como um tipo especial de interlocutor, conduziu-nos de tradicionais *guidelines* de avaliação de software para uma perspectiva baseada na Análise da Conversação.

Esta perspectiva não é inteiramente nova para a área de *design* e avaliação de software. Em particular, muitos estudos têm olhado para o papel das ações dos usuários e para a colaboração entre usuários e designers de sistema de informação, a partir da perspectiva da Teoria da Atividade de Leontiev (e.g., ENGESTRÖM; MIDDLETON, 1996; BODKER; GRONBÆK, 1996). Na área de Interação Humano-Computador (IHC), muitos estudos já encontraram na Teoria da Atividade uma base sólida para o desenvolvimento de suas pesquisas. Muitos desses estudos têm buscado, através desse prisma, olhar para o papel das ações realizadas durante o uso de determinado software, em um modelo de desenvolvimento que prevê a colaboração entre designers de sistemas de informação e usuários (BODKER; GROBAEK, 1996; ENGËSTROM; MIDDLETON, 1996). Esta teoria orienta os pesquisadores a buscarem uma unidade de análise baseada nos motivos e objetivos da ação, ainda que combinada com ferramentas analíticas mais específicas. Dessa forma, diferentes combinações têm ampliado o leque de aplicação da Teoria da Atividade: Etnometodologia (NARDI, 1996; PLOWMAN; ROGERS; RAMAGE, 1995; SUCHMAN, 1996), Interacionismo Simbólico (STAR, 1996), *Actor Network* (ENGESTROM; ESCALANTE, 1996), entre outras. Nossa própria pesquisa focaliza o olhar sobre as ações dos indivíduos em atividade na avaliação de software, mas a ferramenta analítica está mais estreitamente baseada na Análise da Conversação e Linguística Interacional, tendo como referencial teórico fundamental em particular os trabalhos de Bakhtin.

Como nós, Susan Bodker (1991) trabalha com análise de quebras em seqüências conversacionais, mas em um contexto diferente de atividade: prototipagem cooperativa envolvendo designers e usuários potenciais de software profissional. Seu objetivo é desenvolver softwares que dêem suporte a trabalhos cooperativos. A análise de quebras sugerida por Bodker visa investigar a natureza de alguns momentos em que indivíduos mudam seu foco de atenção da tarefa de trabalho para o software. Assim, busca o entendimento de como designers e usuários redirecionam seus diálogos nesses

momentos críticos. Nossa análise de quebras é diferente, pois nosso foco está na aprendizagem de conceitos científicos pelos indivíduos e, assim, estamos interessados particularmente no jogo entre dois tipos de seqüências conversacionais: subordinadas (no caso, aquelas relacionadas ao desenvolvimento conceitual) e formulativas (aquelas que versam sobre o comportamento do software).

Na próxima sessão trazemos contribuições dos estudos sobre *design* e Análise da Conversação para nossa perspectiva de avaliação de software, e explicaremos em detalhes nossa investigação empírica.

Estudos sobre design

A concepção e desenvolvimento de interfaces computacionais envolvem a busca por elevados índices ergonômicos para a interação humano-computador (NORMAN, 1988). Para o designer de software, isto implica no conhecimento acerca de quando e porque selecionar padrões específicos, regras e princípios para a realização de uma tarefa (MANDEL, 1997; SHNEIDERMAN, 1997; VANDERDONCKT, 1999). O espaço da realização da tarefa é ocupado pelos objetos através dos quais o usuário interage com o sistema. Esses objetos são os ícones da interface, as imagens, o som e a palavra que se combinam para favorecer o diálogo com o usuário. Tais agregados de formas compreendem mais do que uma combinação de tipos de expressão e representação. A interface torna-se um espaço de mediação simbólica envolvendo o usuário num processo conversacional sobre o mundo que está representado na face visível do sistema.

Assim, ao conceber um software, seu autor buscará estabelecer um diálogo com um suposto interlocutor, ou "destinatário superior", e implementar um determinado conceito na interface. Segundo Bakhtin (1997, p. 356): "o autor de um enunciado, de modo mais ou menos consciente, pressupõe um superdestinatário superior (o terceiro), cuja compreensão responsiva absolutamente exata é pressuposta seja num espaço metafísico, seja num tempo histórico afastado".

Num sentido restrito, podemos considerar a interface como o ponto de mediação entre as duas partes - usuário e computador - capaz de tornar uma sensível à outra. É, portanto, uma relação semântica, caracterizada pela possibilidade de fazer com que a máquina represente a si mesma para o usuário, numa linguagem compreensível. A interface "traduz" a linguagem da computação a fim de tornar possível a comunicação com o universo humano através de imagens estáticas e dinâmicas, palavras, sons e associações.

Os estudos sobre *design* muito têm contribuído para a elaboração de interfaces mais interativas e elevados índices ergonômicos que permitam uma interação mais fluida entre usuário e computador.

Como estamos tratando de interfaces educacionais e defendendo a particularidade desses tipos de software, uma vez que visam o desenvolvimento de conceitos ou aplicação em determinadas situações de ensino, a preocupação dos autores de ferramentas como estas deve ir além da criação de interfaces “amigáveis”, pois há a necessidade de entender as tarefas conceituais a serem implementadas para uma melhor aprendizagem.

Kaput (1997) discute várias possibilidades de utilização do software educacional como mídia singular no uso de múltiplas representações para o desenvolvimento de conceitos científicos em crianças.

Na seleção de convenções representacionais do conceito, o designer pode usar ícones ou representações abstratas. Cada tipo de representação tem suas vantagens e o software pode contemplar uma ligação entre ambos os tipos, oferecendo múltiplas representações do conceito: verbal, pictográfica, numérica e gráfica.

No entanto, a organização e *qualidade* da disposição de tais representações na tela dependerão de características de *design* e não, propriamente, da área a que se referem. E, então, são os avanços nos estudos gerais sobre *design* que têm orientado e fornecido diretrizes para o desenvolvimento de interfaces, mesmo as educativas.

Shneiderman (1997) aponta para algumas medidas centrais a serem levadas em conta e dizem respeito aos fatores humanos implicados no processo de uso de um ambiente informático. Os *princípios de design* de interface sugeridos por ele são aqueles conceitos, idéias e crenças que guiam e/ou orientam o *design* de software. Escolhe-se quais são os princípios mais adequados para aplicação em determinado sistema, com um objetivo específico, e também para o estabelecimento dos *guidelines*.

Os *guidelines* recomendam os cursos específicos da ação, baseados em um grupo de princípios. São mais específicos que os princípios e também menos abstratos, requerendo menos conhecimento em *design* e experiência para interpretá-los.

Grandes operadoras de sistema como a Apple, a Microsoft, a IBM, entre outras, têm seus *guidelines*, seus princípios e materiais de referência publicados, de modo a exemplificar e especificar sua perspectiva em *design* de interface, o que nos indica o quanto esses fatores são importantes para a área. Embora existam especificidades relacionadas a cada uma dessas operadoras - como pudemos observar através da leitura

dessa publicação da IBM (2001), seus princípios e *guidelines* encontram-se dentro das três grandes áreas de princípios descritas por Mandel (1997):

- Controle sobre o sistema pelo usuário;
- Redução da carga de memória do usuário;
- Consistência.

Complementar a isso, são os *guidelines* que permitem a criação de elementos, o estabelecimento de suas aparências e características adequadas a uma boa interação. São eles que definem o modelo da apresentação, comportamento e interação possíveis através dos elementos e objetos. Por exemplo, o botão *iniciar* tem uma ação esperada pelo usuário, de acordo com o próprio significado da palavra usada, que remete o usuário à idéia de início para disparar um determinado evento.

Estes pontos são de extrema relevância para a observação de ambientes computacionais, e ganham peso maior quando cuidadosamente ponderados em relação ao contexto de utilização.

Em relação à educação, segundo podemos perceber, a atenção a esses pontos é, então, bastante uma vez que é através da interface que acontecerá a mediação entre o sujeito e o conhecimento, e muitos dos signos no *design* do ambiente referem-se à representação do conceito.

No entanto, o papel da representação em teorias contemporâneas (NUNES, 1997), por exemplo, nos leva à idéia de que o conhecimento matemático não é somente uma questão do desenvolvimento dos indivíduos em interação com o mundo físico, mas também com o mundo social. O conhecimento matemático incorpora um sistema de signos e modos de falar característicos da matemática enquanto atividade cultural.

Os usuários engajados em uma atividade com um software estarão interagindo com os elementos da tela e entre si. Precisamos então olhar para o fenômeno como um todo e, nesse sentido, a análise da conversação foi particularmente útil.

Análise da Conversação

Estamos considerando que durante o uso de um determinado software por duplas de usuários, haverá convergência da atenção dos participantes para aspectos relevantes que devem ser *falados*, constituindo tópicos específicos na seqüência da fala compartilhada. Defendemos que através das estratégias metodológicas da Análise da Conversação pode-se, portanto, verificar os mecanismos orientadores do diálogo dos usuários e identificar os processos da atividade que estão orientando essa convergência.

Logo, poderemos verificar em que medida a interface está participando dessa interação com vistas à construção de significados específicos.

O que estamos chamando Análise da Conversação - AC é uma perspectiva teórico-metodológica que pode providenciar aspectos peculiares relacionados ao contexto e ao ambiente em que um sistema computacional está inserido. A AC visa principalmente a descrição dos processos pelos quais os indivíduos interagem e se engajam em situações específicas. Basicamente, portanto, visa examinar as interações sociais e especificar as organizações das ações, estabelecendo características das variadas práticas comunicativas humanas.

Vários projetos em IHC já têm incorporado a AC para modelar a conversação em sistemas interativos, simulando modelos de conversação humana e mesmo atentando para aplicar suas regras como analogias para sistemas interativos de outros tipos (NORMAN; THOMAS, 1991). Além disso, a AC também tem sido usada como método experimental para o desenvolvimento de interfaces e ambientes virtuais interativos.

Woodruff et al (2002) indicam e descrevem algumas estratégias para integrar a AC no processo de *design* de software. Eles acreditam que sua metodologia fornece possibilidades analíticas bastante úteis por alguns motivos como: permitir ao designer isolar fatores que contribuam para a ocorrência de um dado comportamento; ajudá-lo a fazer predições sobre como usuários se comportam com um novo design; permitir o entendimento estruturado sobre comportamentos do usuário, confirmando ou negando a adequação do design, entre outras.

Assim como Woodruff, também adotamos a perspectiva da AC por ser a que melhor se adequou à analogia conversacional que estabelecemos para a relação usuário-computador e por mostrar-se bastante útil aos nossos objetivos com interfaces educacionais.

Vemos a AC especialmente útil para avaliar interfaces, pois providencia uma rigorosa análise focalizada na interação e não em lados isolados. Além das orientações metodológicas para a verificação das minúcias do processo de desenvolvimento das ações turno-a-turno, de cada interlocutor e do papel do computador, ela forneceu-nos também algumas idéias-chave para a elaboração de um modelo de avaliação de software educacional.

Nosso modelo foi desenvolvido a partir de alguns conceitos centrais: seqüência conversacional, tópico e quebra de tópico (KOCH, 2000; MARCUSCHI, 1991; STECH, 1982).

Modelo de avaliação centrada no diálogo

Numa conversação, o falar é sempre sobre alguma coisa, a qual é delimitável no texto conversacional. A isso a AC denomina *tópico*. Mas a noção de tópico é mais complexa e abstrata do que simplesmente "aquilo sobre o que se fala". Segundo Koch (2000, p. 72),

(...) poderíamos dividir (segmentar) um texto conversacional em fragmentos recobertos por um mesmo tópico. Acontece, porém, que cada conjunto desses fragmentos irá constituir uma unidade de nível mais alto; várias dessas unidades, conjuntamente, formarão outra unidade de nível superior e assim por diante.

De acordo com Marcuschi (1991, p.80):

A regra básica para a organização tópica da conversação é: dois turnos contíguos que apresentam o desenvolvimento do mesmo conteúdo seqüenciam o mesmo tópico, e dois turnos que não seqüenciam o mesmo conteúdo constituem uma mudança de tópico. Mas entre a *continuidade* e a *mudança* temos a possibilidade de quebra de tópico.

Quando dois parceiros se engajam numa conversação sobre um interesse compartilhado, a negociação de significados tem que acontecer para que ambos façam sentido do tópico. Portanto o que é expresso em uma sentença (seu conteúdo proposicional) somente poderá ser compreendido relacionando-se ao que é conjuntamente pressuposto pelos interlocutores.

Então, numa colaboração com vistas à construção de significados, parece-nos crucial não apenas a diferença entre *Mudança de Tópico* e *Quebra de Tópico*, mas os momentos onde ocorreram e por quê ocorreram tais redirecionamentos. A *Mudança de Tópico* acontece quando o tópico chegou ao seu final, delimitando seu término. A *Quebra*, por outro lado, indica a interrupção do tópico, podendo o mesmo retornar em momentos seguintes.

Esse ponto é de fundamental importância para atingir nossos objetivos, pois as quebras e mudanças de tópico percebidas na conversação dos alunos em atividade com o software apontarão para as características da interação mediada, sendo possível indicar as causas e conseqüências da construção dos diálogos com fins educacionais.

Trata-se de um modelo centrado principalmente nos momentos de quebra de tópico, onde sugerimos que, a partir da verificação daquilo que mobiliza tais quebras e suas conseqüências na construção dos diálogos pelos alunos, conseguimos identificar as relações entre usuários e software e seus efeitos para a colaboração entre a dupla.

Os momentos de quebra nos permitem verificar quais as características do ambiente ou da atividade que levam à construção de uma nova seqüência tópica. Isto porque, a partir dessa verificação, é estabelecida a conexão entre a seqüência prévia e a seguinte, relacionando-as com a construção de significados e com o desenvolvimento dos conceitos específicos trabalhados pelo software.

Observamos que as crianças, enquanto usam um software educacional específico, estão motivadas por uma atividade altamente complexa que exige competências relacionadas tanto ao uso da informática quanto ao domínio de conhecimento tratado pelo software (no estudo de caso citado aqui, aritmética). Assim, nos diálogos entre os usuários, os tópicos conversacionais vinculam-se ora aos conteúdos e conceitos específicos representados no ambiente, ora às características do *design* da interface, permitindo avaliar o software sob um duplo olhar que focaliza tanto a colaboração com vistas à construção de conceitos, quanto às características da interface, e, mais ainda, cruza esses dois focos para capturar a relação entre eles.

Aquilo que desencadeia mudanças e quebras de tópicos, e as consequentes seqüências tópicas geradas a partir desses momentos, são pontos que norteiam a análise do processo colaborativo e a análise da construção conceitual durante o uso desta mídia específica. As observações centralizam-se nas trocas de turnos, nos pares adjacentes e nos processos locais estabelecidos entre um tópico e outro. Então, focaliza-se o momento de quebra ou mudança tanto de forma prospectiva quanto retrospectiva, priorizando o passado e o futuro deste momento, ou seja: o que conduz à quebra/mudança de tópico e qual a seqüência gerada a partir desse ponto.

Para um melhor entendimento, achamos adequado considerar as categorias das *Subseqüências Encaixadas* (STECH, 1982). Tratamos por este termo as subseqüências que ocorrem quando um tópico introduzido é decorrente de uma quebra do tópico anterior, podendo então ceder espaço ao retorno para a conclusão do tópico original.

Com tal definição, temos três modos de ocorrência de *Subseqüências Encaixadas*:

- 1) *Subseqüência Encaixada Subordinada*: O tópico inserido é parte ou está relacionado com o tópico em andamento;
- 2) *Subseqüência Encaixada Associativa*: O tópico encaixado está associado apenas acidentalmente, não contribuindo propriamente ao desenvolvimento do tema geral;
- 3) *Subseqüência Encaixada Formulativa*: Os tópicos são introduzidos para indicar como tratar o tema em pauta, especificando-se em como e sobre o que se deve falar em relação àquele ponto.

Para capturar a continuidade através das transformações e/ou progressivas diferenciações dos tópicos durante qualquer trecho coerente de análise, temos que examinar as contribuições das ligações dos componentes individuais tanto retrospectivamente quanto prospectivamente: como co-operam e como um tópico proposto é compartilhado e apoiado pelos interlocutores.

Propomos fazer o tratamento da interface dentro do espaço simbólico gerado na situação de uso, sendo possível avaliarmos em que medida tal espaço pode propiciar a convergência e a troca conceitual.

Portanto, fundamentados na Análise da Conversação e considerando ainda outros campos semióticos (GOODWIN, 2000) constitutivos da atividade humana (tais como gestos, registros materiais e artefatos), sustentamos que é possível avaliar a qualidade educacional de um software, a partir da análise da colaboração dos usuários agindo em dupla diante do mesmo.

O quadro analítico preenchido a cada quebra/mudança de tópico na conversação (Quadro 1) permitiu não apenas caracterizar a construção de significados no ambiente informatizado em tela, mas também identificar os mecanismos orientadores do diálogo nesse tipo de interação e propor um modelo geral de avaliação do software educacional.

Quadro 1 - Modelo de quadro analítico preenchido nos momentos de quebras na conversação

Ambiente em uso		Dupla analisada	
Quadro	Causa/	Subseqüência Encaixada	
Tópico	Quebra	Tipo	Subtópico
Tópico anterior à quebra	Fator mobilizador da quebra	Tipo de subseqüência gerada a partir da quebra	Tópico posterior à quebra

Com base nesta perspectiva de análise, demonstramos que a organização tópica da conversação, estruturada durante a atividade com o software, fornece algumas diretrizes para a avaliação do próprio software enquanto instrumento de mediação entre o sujeito e o conhecimento. Ressaltamos que os limites das unidades tópicas são depreendidos não apenas pelo conteúdo (tópico ou assunto), mas por um conjunto de marcas formais bem estudado pela Análise da Conversação.

Este estudo levou em conta que não existe sentença fora ou externa a alguma seqüência específica, bem como não existe um “contexto nulo” para a ocorrência dos atos de fala (SEARLE, 1995). Desse modo, as seqüências tópicas ocorridas durante a interação com o software estavam situadas no contexto de uso e remetiam tanto aos turnos de cada participante quanto ao papel do software na emergência destes turnos.

Assim, podemos já propor uma primeira afirmação que resulta de nossas investigações e é respaldada tanto por estudos em *design* quanto pela Análise da Conversação: *Quanto mais fluida for a conversação entre usuários de um software educacional durante sua utilização, mais amigável podemos considerar sua interface.*

No entanto, sabemos que uma interface com fins educacionais pode ser bastante amigável e ainda assim não possibilitar o desenvolvimento conceitual nos seus usuários. Este é um problema que verificamos nos critérios que fundamentam várias propostas de avaliação de interfaces voltadas à educação, ao enfocarem quase exclusivamente as características do *design*.

A fim de evitar este tipo de problema, complementamos a análise da organização tópica da conversação com as contribuições de estudos sobre colaboração e convergência conceitual (ROSCHELLE, 1995). Esses estudos sugerem que a troca conceitual entre estudantes é possível a partir do compartilhamento de significados, alcançado gradualmente através das ações colaborativas dos indivíduos em torno de um objetivo comum.

O software pode ser um componente fundamental para a convergência das ações e atenções dos participantes para um mesmo ponto focal. Em se tratando de software educacional, o ponto focal devem ser conceitos específicos representados na tela. Então, uma segunda afirmação complementa nossa proposta: *Quanto mais o software contribuir para o compartilhamento de significados e a troca conceitual entre seus usuários, abrindo um campo para interconexões entre conceitos, mais propício mostra-se à aprendizagem.*

Vejamos a seguir dois exemplos de análise para a avaliação de software, de acordo com o modelo apresentado.

Exemplos de análise do software *Calcule!*

O *Calcule!* é um software de matemática e possui três ambientes: Descalculadora, Cripto e Estima. Dado o escopo deste artigo, discutiremos a seguir episódios da atividade das crianças com apenas um dos ambientes, o Estima. Neste ambiente, algumas tarefas para desenvolver cálculos estimados são propostas. Três modalidades de estimação do valor de expressões aritméticas são apresentadas ao usuário ao longo

de três ambientes: *Grandeza, Intervalo e Aproximação*. Cada um desses módulos apresenta alterações de interface e de representação do conceito. Portanto, tratam-se de três cenários nos quais a capacidade de estimação e competências associadas a este conceito são trabalhadas. Para os três módulos, no entanto, existe um cronômetro que limita um tempo máximo no qual as respostas do usuário devem ser emitidas.

Exemplo 1

Este exemplo foi extraído da atividade dos alunos com o módulo *Intervalo*. Neste módulo, a estimativa sobre o resultado da expressão apresentada (gerada automaticamente pelo software) deverá ser indicada, marcando-se o intervalo em que se encontra. Para tanto, o usuário dispõe de uma reta numérica com intervalos marcados a cada 20 unidades, de -100 a + 100.

No segmento a seguir, as alunas F e S discutem acerca do intervalo no qual se encontraria o resultado da expressão “33- (67 - 20)” (P é a pesquisadora):

/.../

F - vai dar vinte, sessenta e sete

S - sessenta/ cinquenta e sete menos trinta e três

F - vai dar cinquenta e sete menos trinta e três

S - vai dar menos, aqui

F- menos, é cinquenta/ quarenta/ vinte/ vinte/ dez e/ vinte e quarenta, dá aqui ((aponta com as mãos para a reta)) Ah! só sei disso, vinte e quarenta, dá aqui

S - ((clica no intervalo que vai do 20 ao 40 positivos na reta numérica))

<som de clique no intervalo>

<som de erro>

F - não/ não/ aqui, aqui ((aponta para o intervalo do 20 ao 40 negativos)) ((QUEBRA))

<uma vez respondida, cada expressão é registrada em uma janela que mostra o histórico de acerto e erros do usuário; ao clicar em qualquer destas expressões, o usuário pode fazer nova tentativa de resposta; na seqüência a seguir, os alunos tentam fazer isso, mas selecionam uma expressão anterior àquela que eles vinham trabalhando: “11 - 47”>

< cronômetro dispara>

((S clica no intervalo sugerido -20 a -40))

<som de clic no intervalo>

<som de erro>

S - dá errado ((permanecem nove segundos em silêncio)) P- peraí, mas vejam lá a expressão que vocês repetiram, vocês clicaram em onze menos quarenta e sete.

((QUEBRA))

S - ah!

F - eita, foi!

S - ((ri))

F - é essa aqui, né?

S - era essa? ((apontando com o mouse para a expressão "33 - (67 - 20)" no quadro de acerto e erro))

F - não!

S - era, agora o tempo.

F- era?

P - era.

/.../

O intervalo situado entre 20 e 40 positivos na reta numérica não corresponde à resposta correta, fato reparado por F imediatamente após o aviso de erro oferecido pelo software: *não! não! aqui, aqui* (apontando para o intervalo -40 a -20). S já havia sugerido que o resultado seria negativo: *vai dar menos, aqui*. Mas a contribuição do turno seguinte, de F, não desenvolve esse tópico, não constituindo uma quebra ou uma nova seqüência. Então, após o erro indicado, o apontar de S para a reta representada na interface e a reparação da aluna, situa a interlocutora e ambas convergem as ações para o novo tópico: números negativos. No entanto ao clicarem no intervalo (entre -40 e -20) o programa acusa novo erro, pois a expressão que selecionaram para repetir não foi a expressão que tinham a intenção de repetir. As alunas não perceberam este equívoco e imediatamente clicaram num novo intervalo na reta, também recebendo como *feedback* o aviso de erro. Há, então, uma pausa de hesitação, com a provável motivação cognitiva de estabelecerem, agora, o porquê do erro.

A pesquisadora P, a partir de um marcador de reparação (*Peraí!*), sugere que a dupla observe a expressão selecionada para uma nova tentativa: *mas vejam lá a expressão que vocês repetiram*.

Após o reconhecimento do equívoco há, portanto, o desenvolvimento de uma seqüência tópica encaixada que não contribui com o tema anterior, e desenvolverá um tópico sobre a expressão que deveriam ter marcado. É significativo o fato dessas alunas

terem selecionado uma expressão diferente da que gostariam mas não perceberem o fato. No quadro do histórico de acertos e erros não há uma hierarquia entre as expressões registradas. Assim, quando muitas expressões já estão registradas fica fácil ocorrerem equívocos, como também encontramos na análise dos episódios de outras duplas. Abaixo (Quadro 2), vemos o passado e o futuro deste momento de quebra:

Quadro 2 - Análise de quebra neste exemplo 1

Estima/Intervalo		S e F	
Quadro Tópico	Causa/ Quebra	Subseq. Encaixada	
		Tipo	Subtópico
Resolução de expressão numérica	Falta de hierarqui- zação no quadro de acerto e erro	Formulativa	Qual expressão deveriam ter clicado

Se tivesse sido seguido um princípio de *design* da interface tratado por *hierarquização*, durante a organização dos elementos no quadro de registros de acertos e erros do Calcule! teria permitido uma melhor visualização para os usuários, o que possivelmente evitaria quebras deste tipo, que afastaram os usuários de tópicos conversacionais que possibilitavam o desenvolvimento conceitual.

Exemplo 2

Este segundo exemplo foi extraído de um segmento da atividade dos alunos com o módulo *Aproximação*. Neste módulo, o usuário tem na tela uma espécie de "máquina digitadora" com os dígitos de 0 a 9, um botão de vírgula (para números decimais), e um botão para indicar números negativos. Neste ambiente, o usuário deve digitar um resultado numérico aproximado para cada expressão apresentada.

O quadro de histórico das expressões resolvidas, desta vez, estabelece um valor percentual que corresponde à margem de erro para cada resposta aproximada pelo usuário. Ou seja, quanto mais a resposta do usuário aproximar-se da resposta exata, mas próximo de 0% será a margem de erro. Na versão do *Calcule!* utilizada nesta pesquisa, o visor no qual o usuário deveria inserir suas respostas iniciava com a marca de um 0 (zero) a cada expressão (quando, de fato, deveria estar vazio (sem qualquer número, como na versão final do software). No exemplo a seguir, analisamos que esta falha na interface gerou nos alunos um equívoco importante, ocasionando quebras que não conduziam à exploração do conceito:

/.../

<15 + 11 - 60>

H - quinze mais onze?

D - quinze mais onze?

H - dá vinte e seis, menos sessenta?

D - vinte e seis menos sessenta? Dá quarenta e quatro.

H - hum?

D - quarenta e quatro

H - tem certeza?

D - quarenta e quatro, bota aí, quarenta e quatro

H - negativo, não? vai dar um número negativo, vinte e seis menos sessenta, vai dar um número negativo ((QUEBRA))

D - vai dar um número negativo? Não, pô!

H - negativo!

D - quarenta e quatro?

H - é

((clica <4> <4> sendo que permanece um <0> à esquerda dos dígitos, no visor <044>))

H - menos

((clica no botão <negativo>))

D - quarenta e quatro, quarenta e quatro, oxe! ficou menos zer/

H - será que vai dar certo?

<som de erro>

D - vinte e nove ((referindo-se à margem de erro que aparece no quadro de histórico: <29%>)) ((QUEBRA))

H - ôxe, será que é o zero?

H - é por causa do zero.

/.../

Ao tentarem resolver a expressão, H responde que *dá vinte e seis* o resultado de 15 + 11. D reforça esta resposta ao continuar os cálculos a partir do 26 (*vinte e seis menos sessenta? Dá quarenta e quatro*). Pelos sinais seguintes emitidos por H (*hum? e tem certeza?*), vemos que não há uma concordância quanto ao resultado, já que tal indagação por parte do ouvinte é sinal de discordância. Assim, embora D --sem o controle do mouse-- insista em sua resposta através da diretiva “*bota aí, quarenta e*

quatro”, H realiza uma quebra na seqüência ao sugerir “*negativo, não? vai dar um número negativo*”. H, convicto quanto à sua resposta, indica “*menos*”, ao passo que D (com o controle do mouse) clica no botão negativo. A implicitude na conversação aqui pode ser verificada como decorrente dos conhecimentos prévios partilhados pelos interlocutores a respeito dos números negativos, e do *guideline* semântico <negativo> na interface, favorecendo os movimentos cooperativos. Percebemos que esta subsequência na conversação desenvolveu o tópico sobre números negativos, e está, por sua vez, subordinada ao tópico sobre expressão numérica que vinham tratando anteriormente. O raciocínio da dupla foi adequado do ponto de vista matemático no que diz respeito a números negativos. No entanto, o cálculo desenvolvido pela dupla para solucionar a expressão $26 - 60$ não corresponde à resposta correta (-34, com 0% de erro). De qualquer forma, o que nos interessa aqui é o fato de as crianças nem sequer verificarem o cálculo - procedimento comum nessa dupla após a emissão de cada margem de erro emitida pelo programa. Desta vez, entretanto, atribuíram a margem de 29% de erro ao zero presente no visor, do lado esquerdo do 44 digitado (ficando, portanto, -044).

Como podemos perceber, para a pergunta de H, marcada com um sinal de indagação e surpresa (*oxe? será que é o zero?*), D responde convicto: *é por causa do zero*, não dando espaço para a reparação do verdadeiro motivo do erro. Ora, apesar de sabermos que o zero à esquerda não altera o resultado, a sua permanência no visor após a digitação do número pode conduzir o usuário a equívocos como o ilustrado neste segmento. O que levou ao percentual 29% (margem de erro) foi o equívoco no cálculo da expressão (porque $26 - 60$ é igual a -34 e não a -44, como estabeleceram os alunos). No entanto H e D não retomaram estes cálculos uma vez que para eles a causa do erro foi o zero à esquerda.

Podemos, então, analisar neste episódio dois momentos de quebra (como visualizados para a análise nos Quadros 3 e 4). No primeiro momento de quebra, o tópico seguinte vinculou-se ao tópico anterior de forma subordinada; no outro momento, o tópico anterior vincula-se apenas acidentalmente com o seguinte, caracterizando uma associação apenas casual.

Quadro 3: Análise da primeira quebra neste exemplo 2

Estima/Aproximação		D e H	
Quadro	Causa/	Subseq. Encaixada	
Tópico	Quebra	Tipo	Subtópico

Resolução de expressão numérica	Intervenção do aluno sobre números negativos	Subordinada	Número negativo
---------------------------------	--	-------------	-----------------

Aqui, a representação para números negativos aparece na tela sob forma de registro escrito. O autor do software fez uso da palavra "negativo" em um botão na interface. Ao clicar neste botão, é acrescentado aos dígitos um sinal < - > característico dos números negativos. Esta representação do conceito contribuiu para um encadeamento de seqüências tópicas subordinadas à esta idéia conceitual, favorecendo o compartilhamento e o desenvolvimento de um conceito matemático.

Quadro 4 - Análise da segunda quebra neste exemplo

Estima/Aproximação		D e H	
Quadro	Causa/Quebra	Subseq. Encaixada	
Tópico	Quebra	Tipo	Subtópico
Número negativo	Acusação da margem de erro	Formulativa	Causa do erro: zero no visor

Neste momento, a partir da quebra, a discussão sobre conceitos matemáticos (números negativos) foi substituída para o tratamento de um outro tópico que não contribuía com o desenvolvimento de significados em aritmética. Este tipo de quebra, podemos avaliar, não se mostra adequado para a aprendizagem e foi decorrente de uma característica do programa.

Seguindo um passo-a-passo dos tópicos conversacionais desenvolvidos e analisando microgeneticamente as quebras que conectam tais tópicos, vemos que a qualidade da interação pode ser mapeada, permitindo a avaliação do software.

Elaboração de um modelo

Vejamos então as possibilidades de pensarmos um modelo de avaliação de softwares educacionais, com base nas idéias desenvolvidas acima:

1. *Quebras com desenvolvimento de Subseqüência Encaixada Formulativa (aborda como tratar o tema)*

As subseqüências encaixadas formulativas desviam do tópico em andamento, pois os participantes passam a abordar a forma de tratar o tema. Quando geradas a partir dos significados construídos pelos usuários para características do *design* da interface (princípios e/ou *guidelines*), ou a partir destes, não vemos estas subseqüências como positivas, uma vez que interrompem o fluxo da conversação e dificultam o desenvolvimento conceitual do ponto de vista dos usuários.

Por outro lado, esse mesmo tipo de subseqüência pode ter suas causas vinculadas a outros motivos, que dizem respeito muito mais à subjetividade do usuário do que à interface, como nos casos de dúvidas quanto à regra do jogo, distração, adaptação a um novo nível de dificuldade etc. Nestes casos, devem ser verificadas caso a caso, pois podem contribuir para a adaptação do usuário à atividade, e são mesmo necessárias em um primeiro momento para que haja um bom desenvolvimento da atividade, com seqüências cada vez mais voltadas para a construção dos significados visados pelo ambiente.

2. *Quebras com desenvolvimento de Subseqüência Encaixada Associativa (vincula-se acidentalmente ao tema)*

Quanto às subseqüências encaixadas associativas, podemos dizer que quando acontecem devido às construções de significados dos participantes em relação aos princípios e *guidelines* do *design* de interface, também não se mostram positivas para o desenvolvimento conceitual. Ora, se vinculam apenas acidentalmente ao tópico principal, durante a atividade com o software, a construção de subseqüências desse tipo tanto pode favorecer a aprendizagem em um dado processo de uso, como pode não favorecer em um outro. Então parece mais coerente com nossos pressupostos que devem ser evitadas. Por outro lado, essa associação pode, por exemplo, ser consequência de *insights*, de intervenções oportunas ou colaborações entre os usuários e nesses casos será positiva do ponto de vista conceitual.

Reforçamos com isso a idéia de que para valorarmos cada tipo de subseqüência, devemos olhar para a estruturação tópica como um todo e inseri-las numa estrutura maior na qual as seqüências precedentes e as futuras são fundamentais para o processo. As causas das quebras, portanto, e também àquilo que remetem, é que permitirão qualificar os diálogos durante o uso do software.

3. *Quebras com desenvolvimento de Subseqüência Encaixada Subordinada (vincula-se ao tópico em andamento)*

Quando a quebra gera uma Subseqüência Encaixada Subordinada, outros tópicos referentes ao conceito ganham lugar na atividade, mostrando que está havendo o desenvolvimento do supertópico e que os alunos (durante o uso do software), estão

construindo um espaço propício para o compartilhamento de significados, favorecendo a convergência conceitual. Nesse caso, se a Subseqüência Encaixada Subordinada é decorrente dos significados que os usuários construíram para uma característica da interface (seus princípios e/ou *guidelines*), verifica-se que essa característica é bem vinda, adequada e, mais que isso, teve uma participação na atividade colaborativa. Ou seja, tal característica não apenas torna o ambiente amigável ao usuário, como também permite ligações conceituais. As subseqüências deste tipo favorecem o compartilhamento de significados e contribuem para a troca conceitual.

4. *Quebras com desenvolvimento de Subseqüência Encaixada Alternadas (quebras constantes sem retorno ao tópico principal)*

A partir dos três exemplos de subseqüências anteriores, podemos concluir para as subseqüências alternadas o seguinte: caso haja quebras constantes sem retorno ao tópico (característica das seqüências alternadas), e sejam decorrentes de princípios/*guidelines* da interface e de como os usuários criaram significado para os mesmos, o processo de uso terá sido bastante interrompido e truncado, uma vez que a conversação entre os usuários não está acontecendo de forma contínua. Logo, a interação com o software não é amigável e não propicia um espaço para possíveis trocas de significados. Caso não possamos remeter essas causas às características da interface, deve-se especificar as outras possíveis causas: distrações, adaptações às regras da atividade, falta de interesse em participar da atividade e outras que requerem uma nova aplicação do modelo.

De um modo geral, podemos estabelecer que quanto menos seqüências alternadas e menos subseqüências formulativas forem construídas ao longo do uso de um software, mais fluida e sem truncamentos terá sido a atividade. Ou seja, mais amigável é a interface daquele ambiente informatizado. Também podemos dizer que as subseqüências encaixadas subordinadas são as mais importantes para a construção de sentidos, pela possibilidade de encadeamento de uma teia de conceitos que amplia o espaço simbólico para o compartilhamento de significados entre os usuários. Mas é a causa de cada subseqüência, como já enfatizamos, que nos permite valorá-las. É como se houvesse subcategorias para cada tipo de subseqüências, categorias estas especificadas pelas causas das quebras.

5. *Mudança de tópico (o tópico anterior chegou ao seu final)*

Quanto às mudanças de tópico, vemos que a cada novo ambiente existe uma abertura para que um novo quadro tópico seja construído pelos usuários, indicando que o anterior chegou ao seu final. É, portanto, quando há a mudança de ambiente no software que a mudança de tópico será bem vinda. Estamos considerando ambiente

enquanto um espaço informatizado, possibilitado por uma interface específica, para o desenvolvimento de algumas ações próprias àquele espaço na atividade. O *Calcule!*, por exemplo, possui três ambientes distintos -Descalculadora, Cripto e Estima-, sendo que o Estima ainda está dividido em outros três sub-ambientes - Grandeza, Intervalo e Aproximação. Subseqüências geradas da mudança de ambientes desse tipo estão vinculadas a conceitos e abrem espaço para que um novo quadro tópico seja desenvolvido. A cada mudança de ambiente o quadro tópico pode também ser alterado sem comprometer a qualidade educacional do software.

Mas cada um deles possui, ainda, ambientes específicos para configuração, avaliação, ajuda e outros cujos quadros tópicos objetivados pelo autor não são conceitos específicos ou situações locais para o ensino. Logo, aparecerão como quebras quando são acessados durante a atividade na interface principal do ambiente.

6. *Nível de problematização*

Para completar o ciclo de avaliação, é importante que os usuários sejam dispostos, para a avaliação, em diferentes faixas de idade de modo a abranger os níveis de dificuldade possibilitados pelo software. No exemplo de caso ilustrado aqui para avaliação do software *Calcule!*, mais do que a 6ª série - ponto médio de destinação - tanto alunos iniciantes (4ª série) quanto avançados (8ª série) deveriam ter sido analisados, para tornar o processo mais completo. Um programa como o *Calcule!*, por exemplo, oferece no ambiente Descalculadora a possibilidade de construção de expressões elaboradas pelo usuário, o que faz com que as operações matemáticas apareçam de modo padronizado e não aleatório. Essas expressões previamente especificadas podem ser elaboradas também pelo professor, ou, no caso aqui tratado, pelo avaliador. Isso permite fazer uma comparação entre as duplas de alunos através de um parâmetro específico controlado com fins de avaliação do software.

7. *Acerto e Erro*

A forma como o software lida com acerto e erro tem grande influência na emergência das seqüências. Como vimos, o software analisado permite *feedback* para acerto e erro, com o registro da resposta do usuário em uma janela específica. Após cada *feedback* era possível uma transição, podendo a atividade ser conduzida pelos participantes para a continuidade do tópico ou para a quebra.

No entanto, existem softwares que ao invés de apenas indicarem o erro, corrigem as respostas dos usuários (muitas vezes de maneira bastante enfática); outros não permitem a passagem para um próximo nível ou para o problema seguinte, antes que a resposta correta seja oferecida pelo usuário; outros ainda emitem dicas e sugestões para a resolução da atividade; enfim, as múltiplas possibilidades de *feedback* para acerto e

erro trarão contribuições específicas à construção dos diálogos, até porque existem softwares que não lidam com as respostas dos usuários à questões específicas e são bastante abertos, claramente trazendo particularidades para a conversação entre os usuários.

8. *Desenho da organização tópica*

A análise dos limites entre unidades tópicas é um dos momentos da avaliação, mas a mesma, por si só, não garante a qualidade educacional de um software. Para tanto, deve-se primar por um outro momento, que consiste em "desenhar" a organização tópica da conversação e a forma como estão conectados os tópicos ao longo da atividade do usuário nos ambientes, de modo a verificar os quadros tópicos abertos no decorrer da atividade, sendo então possível, a partir disso, estabelecer os possíveis ou prováveis caminhos de desenvolvimento conceitual possibilitados pelo software. É importante mencionar ainda que diferentes ambientes podem gerar um mesmo quadro tópico, como pudemos observar em relação aos ambientes Descalculadora e Estima do *Calcule!*. Muitos software dão suporte somente à interação usuário-sistema, oferecendo um mínimo de suporte para interação humano-humano. Essa última, porém, é necessária para favorecer a natureza colaborativa da aprendizagem. O mapeamento dos tópicos conversacionais abertos e desenvolvidos durante o uso pode levar ao aprimoramento de interfaces educacionais que conduzam ao compartilhamento de informação, coordenação das ações e construção de significados.

Conclusão

Os conceitos escolares são compartilháveis e debatíveis e o uso de tecnologias nesse cenário fornece um contexto particular no qual a natureza colaborativa da aprendizagem pode ser organizada (CROOK, 1996). O computador pode ter características fortes para a colaboração. Mas somente analisando esta colaboração podemos indicar quais são essas características e sua ligação com a representação na interface.

Embora muitas propostas de avaliação de software educacional existam, sentimos a necessidade de estruturar um modelo capaz de abarcar os diferentes lados da questão: interface, alunos e conceito científico pretendido pelo software.

Desse modo, para apreender o fenômeno que nos propomos a avaliar, interessa o processo desenvolvido durante o uso enquanto interação semiótica mediada. Logo, o olhar deve voltar-se para a atividade como um todo, e inclui os dois lados: usuários e software.

A metáfora do diálogo ou conversação entre humano-computador serviu-nos não apenas como vaga analogia, mas norteou toda a elaboração do modelo de avaliação.

Nosso fio-condutor é, portanto, a idéia de que o usuário está de algum modo presente nas construções do autor desde o início da elaboração do software educacional, antecipando e prevendo a compreensão do uso e das representações do conceito, num espaço metafísico. Cabe ao usuário - no momento da atividade - mobilizar seu universo de conhecimentos para dar sentido, resgatar a interdiscursividade presente na estruturação do conceito, recriar as omissões, preencher as lacunas e desvendar o que ficou oculto no "texto" do autor. Esse diálogo com o autor é mediado pelos elementos da interface.

Como almejamos capturar os detalhes das ações e das falas dos usuários em interação com o software, justifica-se a Análise da Conversação como perspectiva teórico-metodológica fundamental ao modelo. O diálogo e a seqüência das ações de uso foram analisados tanto retrospectivamente quanto prospectivamente através do rastreamento turno-a-turno e dos momentos de quebra nas seqüências, situando o software nessa seqüência de ações e permitindo pensarmos em *quase-turnos* atribuídos ao software.

A avaliação assim descrita permite configurar a gênese social de cada ação e as transformações no curso dos eventos. Isso pode levar à identificação das relações dinâmico-causais da atividade de uso.

Um usuário generalizado está desde o início envolvido num suposto diálogo com o autor do software, que antecipa as ações do usuário, os caminhos da aprendizagem, e responde àqueles aspectos da programação que correspondem ao comportamento do software. Ao mesmo tempo, os usuários são convidados a participar deste diálogo com o autor do software, na atividade de uso.

Por isso o diálogo serviu-nos mais que somente como uma analogia. Através da Análise da Conversação, os diálogos emergentes dos usuários foram examinados em busca de seqüências conversacionais e quebras que poderiam nos dizer sobre como o software se comportava. Em certo sentido, nós adaptamos *guidelines* e princípios de *design* às ações dos usuários, algumas vezes como causa da interrupção no fluxo da conversação. Então, ilustramos como investigar a natureza e implicações da quebra de tópico, e como avaliar o papel dos objetos da interface e dos comportamentos do software na aprendizagem de conceitos específicos por estudantes.

Em trabalhos futuros, estenderemos essas análises em duas direções. A primeira, e mais óbvia, é aplicar a perspectiva sugerida aqui para diferentes software educacionais (em matemática e outros assuntos escolares) e contextos de uso (por exemplo, em atividade de sala de aula de modo menos controlado possível). Um segundo

direcionamento de pesquisa, que já está sendo iniciado pelos autores, investigará o processo de autoria, e os caminhos em que os autores de software concebem seus usuários como interlocutores pressupostos. Em tais estudos, focalizaremos sobre como o discurso do autor sobre um domínio de conhecimento é atualizado e encapsulado em objetos e relações entre objetos na interface, visto como texto ou discurso.

Por tudo isso, temos claro que o uso de computadores em educação e o desenvolvimento de software educacional tem aberto um campo para trabalho interdisciplinar entre engenheiros de software, designers, especialistas em diversos campos didáticos, educadores e psicólogos. Gostaríamos de enfatizar que a proposta deste trabalho tenta situar-se precisamente no diálogo entre educação, ciência, informática e *design*.

Agradecimentos

Esta pesquisa contou com o apoio do CNPq. Agradecemos também a todos da escola Souza Veras (Recife-PE), alunos e professores, que participaram deste estudo.

Referências

- BAKHTIN, M. *Estética da criação verbal*. São Paulo: Martins Fontes, 1997.
- BODKER, S. *Through the interface – A human activity approach to user interface design*. Hillsdale, NJ: Lawrence Erlbaum Associates, 1991.
- BODKER, S.; GRONBÆK, K. Users and designers in mutual activity: an analysis of cooperative activities in system designs. In: ENGSTRÖM, Y.; MIDDLETON, D (eds.). *Cognition and communication at work*. Cambridge: Cambridge University Press, 1996.
- CROOK, C. *Computers and the collaborative experience of learning*. London: Routledge, 1996.
- DE SOUSA, C. S. The semiotic engineering of User-Interface Languages. In: *International Journal of Man-Machine Studies*, v. 39, p. 753-773, 1993.
- ENGSTRÖM, Y.; MIDDLETON, D. *Cognition and communication at work*. Cambridge: Cambridge University Press, 1996.
- GOODWIN, C. Action and embodiment within situated human interaction. *Journal of pragmatics*, 32, p. 1489-1522, 2000.

- ENGESTRÖM, Y.; ESCALANTE, D. Mundane tool of objects of affection: the rise and fall of the postal Buddy. In: NARDI, B. (Ed.). *Context and consciousness: activity theory and human computer interaction*. Cambridge, MA: MIT Press, 1996.
- IBM Corporation. *Object oriented interface design: IBM common user access*. Basic interface design guide. New York, 2001.
- KAPUT, J. J. Creating Cybernetic and Psychological ramps from the concrete to the abstract: examples from multiplicative structures In: PERKINS, D. N. et al. *Software goes to school: teaching for understanding with new technologies*. New York: Oxford University Press, 1997.
- KOCH, I. G. V. *A interação pela linguagem*. São Paulo: Contexto, 2000.
- MANDEL, T. *The elements of user interface design*. NY: John Wiley & sons, 1997.
- MARCUSCHI, L. A. *Análise da Conversação*. São Paulo: Ática, 1991.
- NARDI, B. *Context and consciousness: activity theory and human computer interaction*. Cambridge, MA: MIT Press, 1996.
- NORMAN, D. A. *The Psychology of every day things*. New York: Basic books, 1988.
- NORMAN, M. A. and Thomas. Informing HCI design through Conversational Analysis, *Int'l J. ManMachine Studies* 35, p. 235-250, 1991.
- NUNES, T. Systems of signs and mathematical reasoning In: NUNES, T.; BRYANT, P. *Learning and teaching Mathematics - an international perspective*. Psychology Press Ltd., Publishers, 1997.
- PLOWMAN, L., ROGERS, Y., RAMAGE, M. *What are workplaces studies for?* Proceedings of the European Conference on Computer-Supported Cooperative Work, 1995.
- ROSCHELLE, J. Learning by collaborating: convergent conceptual change In: KOSCHMANN, T. (Ed.). *cslc: Theory and practice of emerging paradigm*. Lawrence Elbaum associates, Inc 1995.
- SEARLE, J. *Expressão e significado: estudo dos atos de fala*. São Paulo: Martins Fontes, 1995.
- SHNEIDERMAN, B. *Designing the user interface: strategies for effective human-computer-interaction*. Addison Wesley Pub Co, 1997.
- SNIR, J., SMITH, C.; GROSSLIGHT, L. Conceptually enhanced simulations: a computer tool for science teaching In: PERKINS, D. N. et al. *Software goes to school: teaching for understanding with new technologies*. New York: Oxford University Press, 1997.

STAR, D. L. Working together: symbolic interacionism, activity theory and information systems. In: ENGSTROM, Y.; MIDDLETON, D (Ed.). *Cognition and communication at work*. Cambridge: Cambridge University Press, 1996.

STECH, E. L. The Analysis of conversational topic sequence structure. *Semiotic*, XVII: 75 – 95, 1982.

SUCHMAN, L. Constituting shared workplaces In: ENGSTROM, Y.; MIDDLETON, D. (Ed.). *Cognition and communication at work*. Cambridge: Cambridge University Press, 1996.

VANDERDONCKT. Development milestones towards a tool for working with guidelines In: *Interacting with computers*, Louvain-la-Neuve, Belgium, n.12, p. 81-118, 1999.

WOODRUFF, A.; SZYMANSKI, M. H.; GRINTER, R. E.; AOKI, P. M. Practical strategies for integrating a conversation analyst in an interactive design process, *Proceedings of conference on designing interactive system 2002*. p 255-264, ACM Press, 2002.

Apresentado ao Conselho Editorial em 30 julho de 2005