

Dissertação de Mestrado

Desenvolvimento de Software para Atividades Educacionais

Leo Burd

- Prof. Dr. José Armando Valente (Orientador)
Núcleo de Informática Aplicada à Educação - UNICAMP
- Profa. Dra. Beatriz Mascia Daltrini (Co-orientadora)
Faculdade de Engenharia Elétrica e de Computação - UNICAMP

Banca Examinadora:

- Prof. Dr. Ivan Luiz Marques Ricarte
Faculdade de Engenharia Elétrica e de Computação - UNICAMP
- Profa. Dra. Afira Vianna Ripper
Faculdade de Educação - UNICAMP
- Profa. Dra. Heloísa Veira da Rocha
Instituto de Computação - UNICAMP

Este exemplar corresponde a redação final da tese
defendida por _____
_____ e aprovada pela Comissão
Julgada em _____ / _____ / _____
_____ Orientador

UNIDADE	Be
N.º CHAMADA:	F/UNI CAMP
	B 894d
V.	Ex.
TOMBO BC/	40513
PROC.	278/00
C	<input type="checkbox"/>
D	<input checked="" type="checkbox"/>
PREÇO	\$ 99,00
DATA	15/03/00
N.º CPD	

CM-00139098-6

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA - BAE - UNICAMP

B894d	<p>Burd, Leo</p> <p>Desenvolvimento de software para atividades educacionais / Leo Burd.--Campinas, SP: [s.n.], 1999.</p> <p>Orientadores: José Armando Valente, Beatriz Mascia Daltrini.</p> <p>Dissertação (mestrado) - Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.</p> <p>1. Software - Desenvolvimento. 2. Análise de sistemas. 3. Projeto de sistemas. 4. Ensino auxiliado por computador. 5. LOGO (Linguagem de programação de computador). 6. Tecnologia educacional. I. Valente, José Armando. II. Daltrini, Beatriz Mascia. III. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. IV. Título.</p>
-------	---

Resumo

Este trabalho parte da idéia de que não existe software que, por si só, seja capaz de educar uma pessoa. Se acreditarmos que a educação é resultado de um conjunto de atividades que propiciam situações favoráveis para o ensino-aprendizado, então os sistemas computacionais só poderão ser considerados mais ou menos "educativos" dependendo do maior ou menor suporte que oferecerem a estas atividades.

Em outras palavras, os programas dito educativos não devem ser analisados ou desenvolvidos fora da atividade educacional para a qual são dirigidos. Um editor de textos usado por uma equipe de alunos na produção do jornal da escola provavelmente teria um papel educativo diferente do obtido pelos mesmos alunos caso utilizassem o mesmo editor para copiar trechos de livros.

O desafio é conseguir descrever as atividades educacionais de modo que, de um lado, os engenheiros de software possam se orientar e, de outro, que os educadores tenham suas idéias e necessidades contempladas. Este é o principal objetivo do presente trabalho.

Para tratar deste problema, propõe-se uma abordagem para o desenvolvimento de software baseada na Teoria da Atividade, um corpo teórico que estuda, dentre outras coisas, como a cultura, as relações sociais, as ferramentas e outros elementos influenciam as atividades humanas. A Teoria da Atividade vem, nos últimos anos, ganhando cada vez mais espaço na Engenharia de Software, principalmente nas áreas onde a consideração de fatores humanos é mais importante.

Como domínio de aplicação e elemento de referência para a abordagem proposta, buscou-se suporte no Construcionismo, uma teoria que estuda o uso da tecnologia na criação de ambientes educacionais. O resultado pragmático mais conhecido do Construcionismo é a linguagem de programação Logo que, nos últimos 20 anos, tem sido utilizada por milhões de estudantes como um solo fértil principalmente para o desenvolvimento do raciocínio crítico e da construção de conceitos matemáticos.

A partir de uma análise da educação construcionista, discute-se, detalhadamente, o desenvolvimento de uma atividade utilizando o software Cocoa. Os resultados indicam que a abordagem proposta modela características relevantes do contexto educacional e possibilita a identificação de vários aspectos a serem considerados no desenvolvimento de novos aplicativos para a educação.



Abstract

This work is centered on the idea that no given software or technology, by itself, is able to educate a person. If we believe that education is the result of a set activities that create favorable conditions for learning and teaching, then computational systems will be more or less “educational” depending on the kind of support that they offer to these activities.

In other words, educational software should not be analyzed or developed without considering the educational activity where it is going to be used. A text editor used by a group of students in the creation of the school newsletter will probably have a different educational result from the one achieved by the same students when they use the same editor to copy parts of books.

The challenge is how to describe educational activities in such a way that, on the one hand, would allow software engineers to understand what has to be done and, on the other hand, would consider the ideas and needs of educators. This is the major goal of the present work.

To deal with this problem, an approach to software development is proposed based on Activity Theory, a theory that studies how culture, social interactions, tools and other elements affect the development of human activities. Recently, Activity Theory has gained broader attention in the field of Software Engineering, mainly in the consideration of human factors.

As a domain of application and reference for the proposed approach, we focused on Constructionism, a theory that studies the use of technology in the creation of educational environments. The most well known pragmatic result of Constructionism is the Logo programming language which, for the past 20 years, has been used by millions of students as a fertile soil for the development of critical reasoning and the construction of mathematical concepts.

Based on the analysis of constructionist education, the development of an activity using the Cocoa software is discussed in full detail. Results indicate that the proposed approach models relevant characteristics of the educational context and allows the identification of several aspects to be considered in the analysis and development of new software for education.



Agradecimentos

Esta dissertação constitui um marco muito importante para mim, realizando um sonho que se iniciou com uma carta ao Alan Kay em 1993. Sem dúvida, nunca imaginaria que um simples desejo me conduzisse por um caminho tão rico e com que me identificasse tanto.

Segundo a linha teórica que adotei, a da Teoria da Atividade, o desenvolvimento dos indivíduos é muito influenciado pela dinâmica sociocultural do meio onde vivem. É claro que temos nossa própria fatia de responsabilidade pelo que fazemos, mas boa parte do que aprendemos se originou do que vimos e da qualidade da interação que tivemos com aqueles que nos cercam. De uma forma geral, eu acredito muito nisso e sinto enorme prazer em procurar fazer meus os gestos e atitudes das pessoas que admiro.

Assim foi com este trabalho. Em seu percurso, encontrei inúmeras idéias fortes e pessoas interessantes que me abriram a mente e fizeram notar coisas que nunca imaginara. Este lado humano ligado à tecnologia é realmente fantástico! Só espero ter feito um mínimo de jus ao que me foi oferecido. Também espero que estas pessoas possam reconhecer suas cores e formas neste mosaico que construí.

Antes de mais nada, gostaria de fazer um agradecimento especial a meus pais e a meus irmãos. Pó, Bu, Ôs, Rá, Dã e Vi: seu jeito de ser, sua atenção e curiosidade, sua acolhida, incentivo, confiança e paciência foram-me e são inestimáveis. Sem dúvida, se o tempo voltasse 28 anos e eu tivesse opção, pediria à cegonha que me deixasse novamente junto a esta família tão sensacional!

Também gostaria de prestar minha gratidão ao pessoal que orbita (ou orbitava) o Núcleo de Informática Aplicada à Educação (NIED, UNICAMP). Desde o telefonema atendido pela Nanda em 1994 até as revisões finais da dissertação, passando pelas inúmeras discussões em corredores e restaurantes, o carinho de sua receptividade para comigo, o respeito pelas escolas, professores e alunos, as preocupações com os desafios da área e a seriedade de sua pesquisa são referências que carrego comigo onde quer que vá.

Este trabalho também não teria voado muito se não fosse o estímulo da pequena mas ambiciosa comunidade internacional que defende o uso da Teoria da Atividade no desenvolvimento de software. Realmente, vocês provaram que a Internet também é grande fonte de calor humano! Victor Kaptelinin, Bonnie Nardi, Rachel Bellamy, Yrjo Engeström: muito obrigado pela orientação, pelos artigos enviados e pelos incansáveis diálogos eletrônicos!

Milhões de obrigados ao pessoal do Instituto do III Millennium por trazer realidade a este trabalho, ao Marcus Vinicius por estar sempre junto, à Stella pelas cutucadas e carinhos certos, à Vitória e à turma da PUC-SP pela mão

constante, à dupla Seabra e Fernando pelo espaço que me deram, à Sílvia e à moçada da Casa Dom Macário pela inspiração e pela força, à equipe do CDI pelo incentivo e ao Alex Repenning, ao Ken Kahn e ao Allen Cypher pela oportunidade de interagir com o desenvolvimento de suas tecnologias.

Por fim, gostaria ainda de agradecer à Bia Daltrini pela tranqüilidade e abertura que trouxe ao meu trabalho e ao Valente, cujas observações ajudaram a fortalecer ainda mais minhas convicções.

Índice dos capítulos

1	Introdução.....	1
1.1	Justificativa.....	1
1.2	Objetivos.....	3
1.3	Gênese do trabalho.....	5
1.4	Estrutura.....	7
2	O contexto educacional e o desenvolvimento de software.....	11
2.1	O desenvolvimento tradicional de software.....	11
2.2	O desenvolvimento de tecnologia para a educação.....	13
3	Introdução à Teoria da Atividade.....	19
3.1	Breve histórico.....	19
3.2	Conceitos fundamentais.....	23
3.2.1	A estrutura hierárquica da atividade.....	24
3.2.2	Orientação a objetos.....	30
3.2.3	Internalização e externalização.....	31
3.2.4	História e desenvolvimento.....	34
3.2.4.1	A análise histórica da atividade.....	34
3.2.4.2	A análise do desenvolvimento da atividade.....	34
3.2.5	Mediação.....	36
3.2.5.1	Mediação social.....	39
4	A Teoria da Atividade e o computador.....	43
4.1	A atividade mediada pelo artefato computacional.....	43
4.2	A atividade de desenvolvimento do artefato computacional.....	49
5	Introdução ao Construcionismo.....	53
5.1	Características gerais.....	53
5.2	O aprendizado construcionista.....	57
5.2.1	As pessoas constroem ativamente o seu conhecimento.....	57
5.2.2	A possibilidade de articular os processos do pensamento permite aprimorá-los.....	58

5.2.3	O aprendizado de um conceito está relacionado com a estrutura deste conceito	60
5.2.4	O aprendizado é influenciado pelo ambiente	61
5.2.4.1	Características dos ambientes educacionais construcionistas	64
5.2.4.2	A evolução dos aspectos socioculturais do Construcionismo	69
5.3	A educação construcionista	72
6	O desenvolvimento de software para atividades educacionais	75
6.1	Princípios para uma metodologia	75
6.2	A proposta de uma nova abordagem	77
6.3	A Análise da Atividade Educacional	81
6.3.1	Sobre a estrutura hierárquica da atividade	83
6.3.2	Sobre a orientação a objetos	86
6.3.3	Sobre a internalização/externalização	87
6.3.4	Sobre a história e o desenvolvimento	88
6.3.5	Sobre a mediação	88
6.4	A Análise do Artefato Computacional	89
6.4.1	Artefatos computacionais, artefatos digitais, objetos digitais e aplicativos..	90
6.4.2	Sobre a estrutura hierárquica da atividade	95
6.4.3	Sobre a orientação a objetos	96
6.4.4	Sobre a internalização/externalização	97
6.4.5	Sobre a história e o desenvolvimento	97
6.4.6	Sobre a mediação	98
6.5	Considerações sobre as demais fases do desenvolvimento	99
7	A análise da atividade educacional construcionista	103
7.1	A estrutura hierárquica da atividade construcionista	104
7.1.1	A atividade de aprendizado construcionista	104
7.1.1.1	Os componentes do aprendizado construcionista	105
7.1.2	As ações do aprendizado construcionista	109
7.1.2.1	A Idealização	113
7.1.2.2	Construção	114
7.1.2.3	A Avaliação	115
7.1.2.4	A Depuração	116

7.1.2.5	A Descrição.....	117
7.1.3	O nível das operações.....	120
7.1.4	Conceitualização e operacionalização.....	121
7.2	Orientação a objetos.....	124
7.3	Internalização e externalização.....	126
7.3.1	Internalização.....	126
7.3.2	Externalização.....	127
7.4	História e desenvolvimento.....	128
7.5	Mediação.....	131
8	A análise do artefato computacional na atividade construcionista.....	135
8.1	A estrutura hierárquica da atividade construcionista.....	135
8.1.1	Conceitualização e operacionalização.....	140
8.2	Orientação a objetos.....	141
8.3	Internalização e externalização.....	142
8.4	História e desenvolvimento.....	145
8.5	Mediação.....	146
9	Um caso prático: a Oficina de Jogos.....	149
9.1	Introdução.....	149
9.2	A atividade educacional da Oficina de Jogos.....	151
9.2.1	A estrutura hierárquica da Oficina de Jogos.....	151
9.2.1.1	A atividade.....	151
9.2.1.2	Os componentes da Oficina de Jogos.....	152
9.2.1.3	As ações.....	156
9.2.1.4	As operações.....	161
9.2.2	Orientação a objetos.....	161
9.2.3	Internalização e externalização.....	166
9.2.4	História e desenvolvimento.....	167
9.2.4.1	História.....	167
9.2.4.2	Desenvolvimento.....	169
9.2.5	Mediação.....	174
9.3	O artefato computacional frente à Oficina de Jogos.....	175
9.3.1	A infra-estrutura computacional básica.....	175

9.3.1.1	O software Cocoa	176
9.3.2	A estrutura hierárquica da atividade	180
9.3.2.1	A atividade em si	180
9.3.2.2	As ações	180
9.3.2.3	As operações	187
9.3.2.4	Conceitualização e operacionalização	188
9.3.3	Orientação a objetos	191
9.3.4	Internalização e externalização	193
9.3.4.1	Internalização	193
9.3.4.2	Externalização	194
9.3.5	História e desenvolvimento	194
9.3.5.1	História	194
9.3.5.2	Desenvolvimento	195
9.3.6	Mediação	196
9.4	Reflexões gerais sobre a Oficina	197
9.4.1	Sobre o planejamento e a execução da Oficina	198
9.4.2	Sobre o uso do computador na Oficina de Jogos	199
10	Conclusões	203
10.1	Sobre o trabalho em si	204
10.2	Sobre a Teoria da Atividade	206
10.3	Sobre o Construcionismo	208
10.4	Sobre a abordagem proposta	209
10.5	Sobre a análise da atividade construcionista	211
10.6	Sobre o papel do computador no Construcionismo	213
10.7	Sobre a Oficina de Jogos	215
10.8	Sobre a continuidade do trabalho	216
	Referências bibliográficas	219

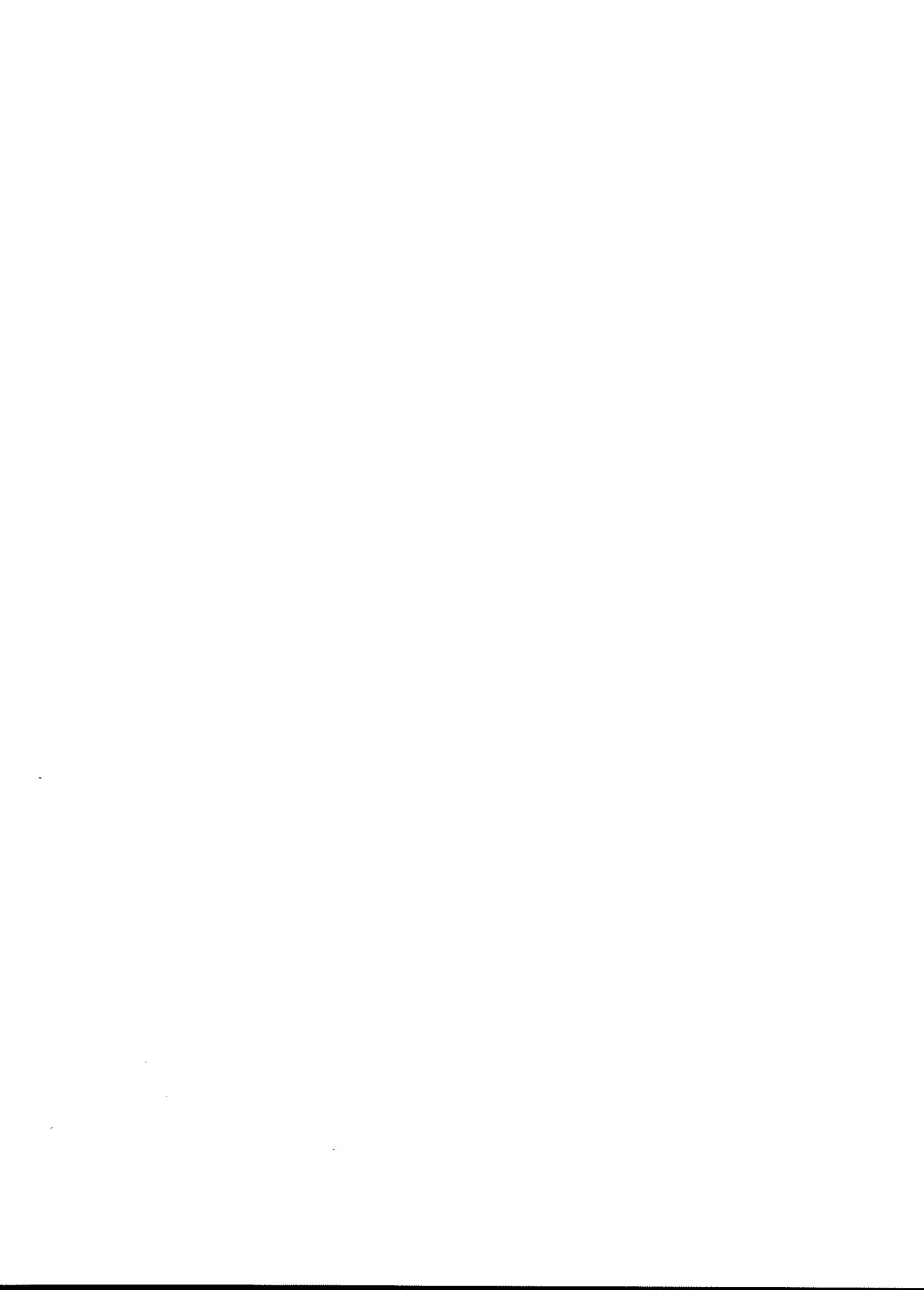
Índice das figuras

Figura 1 - A organização dos capítulos do trabalho.....	9
Figura 2- Diagrama do processador humano	12
Figura 3 - A hierarquia atividades-ações-operações.	26
Figura 4 - Conceitualização e operacionalização	28
Figura 5 - A mediação segundo Vygotsky.	36
Figura 6 - O diagrama de Engeström.	40
Figura 7 - O diagrama de Engeström para a educação fundamental e média	42
Figura 8 - Desenho feito a partir de comandos do Logo Gráfico.....	55
Figura 9 - O ciclo da descrição-execução-reflexão-depuração.	59
Figura 10 - Uma das telas do MUSIC.....	70
Figura 11 – Uma tela do MOOSE Crossing.....	71
Figura 12 - O ciclo de vida tradicional de um software	78
Figura 13 - O ciclo de vida baseado na análise da atividade educacional	81
Figura 14 - A descrição genérica e a descrição específica da atividade educacional	82
Figura 15 - O diagrama de Engeström adaptado para a atividade de aprendizado construcionista	105
Figura 16 - O diagrama de Engeström adaptado para a atividade de aprendizado com o Logo Gráfico.....	108
Figura 17 - O diagrama de Engeström adaptado para a atividade com o MUSIC.....	109
Figura 18 - Um exemplo de diário-de-bordo usado na Oficina de Jogos.	118
Figura 19 - A mensagem de erro "Ainda não aprendi parabaixo".	124
Figura 20 - O diagrama da atividade educacional construcionista da Oficina de Jogos	153
Figura 21 - A folha de proposta de projeto.	158
Figura 22 – Folha do diário-de-bordo com as idéias do dia	159
Figura 23 – A folha para depoimento dos participantes da Oficina de Jogos.....	160
Figura 24 – Telas dos jogos criados durante a Oficina	164
Figura 25 – Disposição dos participantes na sala.....	165
Figura 26 – Cartaz anunciando a Oficina de Jogos	168
Figura 27 - O computador usado na Oficina de Jogos	175

Figura 28 – A tela do software Cocoa.	177
Figura 29 – As propriedades e as regras de um personagem criado no Cocoa	178
Figura 30 – Passos da criação de uma regra no Cocoa.....	179
Figura 31 – A nova regra depois de pronta.	179
Figura 32 – Os botões que controlam a execução dos programas.....	183
Figura 33 – Visualizando as condições de disparo de uma regra.....	186
Figura 34 – Detalhe de um jogo onde o aprendiz inseriu um texto escrito à mão no tabuleiro do Cocoa	192
Figura 35 – Construcionismo, Engenharia de Software e Teoria da Atividade contribuíram mutuamente no desenvolvimento do trabalho.	205

Índice das tabelas

Tabela 1 - Atividades, ações e operações.	27
Tabela 2 - O enfoque da Oficina de Jogos ao longo do tempo.	170



1 Introdução

O objetivo deste capítulo é apresentar o trabalho, buscando, em primeiro lugar, explicitar ao leitor a necessidade de uma metodologia de desenvolvimento de software voltada para a educação e, na seqüência, introduzir os objetivos buscados ao longo do texto.

Depois, é apresentado um relato do caminho percorrido desde a concepção da idéia até se chegar ao formato atual do trabalho. Por fim, apresenta-se a estrutura dos capítulos, destacando-se as dependências lógicas existentes entre eles.

1.1 Justificativa

Este trabalho trata do desenvolvimento de tecnologia para a educação, um tema que relaciona dois assuntos atualmente bastante em voga – tecnologia e educação – e que cuja evolução parece cada vez mais entrelaçada e mutualmente dependente.

A evolução tecnológica e o desenvolvimento social. De modo simplificado, pode-se dizer que, a partir do século passado, o crescente desenvolvimento da tecnologia levou a um salto na produção industrial, a uma especialização e segmentação das profissões e a uma visão generalizada de que quanto mais se produzisse, melhor. Esta mentalidade, centrada na produção em série, também teve grande repercussão no sistema educacional que, por sua vez, acabou por refletir e sustentar o que se via nas indústrias. As salas organizadas em fileiras de carteiras umas do lado das outras, aulas centradas em um professor e a segmentação do currículo em matérias que pouco se misturam são exemplos evidentes disto.

Atualmente, no entanto, este mesmo desenvolvimento tecnológico parece estar atingindo um ápice e as empresas começam a voltar seus olhos para o seu diferencial humano como a única forma de sobrevivência (BRASIL EM EXAME 97, 1997). A indústria moderna, preocupada com a concorrência cada vez mais acirrada, está caminhando na direção de sistemas de produção mais enxutos, baseados em um número reduzido de empregados que, apoiados na tecnologia, acumulam uma diversidade maior de responsabilidades (MAZZONE, 1995, p.3). As novas funções, mesmo as mais básicas, exigem empregados que tenham uma visão mais abrangente dos meios de produção, que sejam fluentes em tecnologia, que saibam trabalhar em equipe e que exerçam suas funções de modo crítico e criativo.

O novo perfil empresarial, aliado à velocidade e quantidade cada vez mais crescentes das transformações, gerou uma necessidade de atualização constante por parte das pessoas e pôs em xeque o tipo de educação vigente que, além de se restringir a um período predeterminado de anos e de oferecer uma formação

estática, fragmentada e individualista, não estimula nem o senso crítico, nem a criatividade e nem a capacidade de aprender do indivíduo.

Apesar do aparente paradoxo, o desenvolvimento tecnológico fez com que, pela primeira vez na história, esteja se considerando o desenvolvimento das capacidades humanas como uma necessidade econômica real. É esta percepção que está levando a um deslocamento de esforços e investimentos para a construção de uma educação mais contextualizada, dinâmica e respeitadora das características de cada aprendiz.

A necessidade de uma reforma educacional. No entanto, muito mais complexa do que uma otimização administrativa ou a troca de um equipamento por outro mais sofisticado, a transformação educacional exige uma reforma cultural geral que envolve todos os setores da sociedade. É preciso formar educadores que trabalhem esses novos valores, conscientizar pais (e até mesmo alunos) de que nem sempre a educação é aquilo que se vê na escola ou se mede pelos métodos tradicionais, criar políticas educacionais que orientem e incentivem a integração dos esforços e é preciso desenvolver tecnologia para facilitar o processo da mudança.

Se fosse possível, hoje, tirar uma radiografia da transformação, perceberíamos que ela já possui um coração batendo, mas ainda não constitui um corpo sólido. Já se tem um sentimento generalizado, pelo menos nos grandes núcleos urbanos, da distância entre o que se aprende na escola e o que será necessário para se viver. Há diversos centros de pesquisa procurando compreender como deveriam ser os ambientes ideais de aprendizagem. As empresas, com seus programas de qualidade, aumentaram a quantidade de cursos ministrados aos seus funcionários. Os governos começam a incentivar alguns projetos de formação de professores e de informatização das escolas.

A importância de um desenvolvimento tecnológico contextualizado. No caso do desenvolvimento de software, o mesmo está acontecendo: há uma série de esforços independentes estudando o potencial da máquina e procurando descobrir soluções para o problema educacional. O que está se tornando cada vez mais aparente, tanto na informática quanto nas outras áreas, é a necessidade de uma maior integração de esforços no sentido de definir melhor o objetivo a ser seguido, trabalhar mais próximo da realidade educacional e viabilizar a implantação em larga escala (CUBAN, 1986; MCARTHUR et al., 1993; NORMAN e SPOHRER, 1996).

Acreditamos que a informática pode contribuir muito para a concretização da transformação educacional. O ideal seria aproveitar as facilidades de representação e comunicação do computador, bem como a penetração que ele já possui na educação, na indústria, nos lares e usá-lo como aliado para a implantação das idéias apresentadas acima (SOLOWAY et al., 1994). A questão é como fazer isso da forma mais eficaz, eficiente e viável.

Na nossa opinião, um problema como este exige, dentre outras coisas, uma abordagem de análise e desenvolvimento de software que esteja em sintonia com esse movimento de transformação.

Na realidade, embora recentes, já existem algumas abordagens de desenvolvimento de software com este enfoque – ver, por exemplo, a do Design Centrado no Aprendiz¹ (SOLOWAY et al., 1994; SOLOWAY e PRYOR, 1996), a do *ESSCOTS for Learning* (MCARTHUR et al., 1994) e a do *Educational Object Economy* (JOURNAL OF INTERACTIVE MEDIA IN EDUCATION, 1998; <http://www.eoe.org>).

A questão é que elas estão muito limitadas a apenas alguns dos aspectos levantados acima, como a relação aprendiz-computador, ou a viabilização técnica e econômica da tecnologia. Nenhuma delas dá grande ênfase aos demais elementos normalmente envolvidos na situação educacional, como a interação entre os aprendizes e seus colegas de turma, o papel dos professores, a duração das sessões, a necessidade dos sistemas de avaliação, etc. (NORMAN e SPOHRER, 1996).

Raramente o computador será o único recurso com o qual o aprendiz interagirá e dificilmente o seu uso não será influenciado pelo contexto maior onde estiver inserido. A duração das sessões, sua frequência, a quantidade de alunos por computador e o tipo de interação permitida em sala são fatores geralmente predeterminados nas instituições educacionais e não podem ser simplesmente ignorados. Nas empresas e nos lares também existem restrições semelhantes e, se este *modus operandi* não for considerado, é muito provável que o esforço gasto no desenvolvimento da tecnologia seja desperdiçado. Segundo Cuban (1986), a relativa baixa aceitação na educação de tecnologias que, teoricamente, teriam um grande potencial como o cinema, rádio e televisão se deu justamente por não considerar as necessidades dos professores e as características específicas da rotina escolar.

Na nossa opinião, é necessário uma abordagem para o desenvolvimento de software para a educação que lide com uma visão mais ampla e integrada do universo educacional. Uma abordagem que se fundamente nas teorias educacionais vigentes (para obter a eficácia educacional), que considere o potencial e as limitações do uso do computador na educação (eficiência) e que trabalhe em cima das condições técnicas, culturais e econômicas da realidade a que se dirija (viabilidade).

1.2 Objetivos

O objetivo principal deste trabalho é propor uma abordagem de desenvolvimento de software para a educação que considere o computador dentro do universo amplo delimitado entre os princípios educacionais e o que se pode fazer com os recursos e valores presentes na realidade aonde ele será desenvolvido e utilizado.

¹ Também conhecida como *Learner-Centered Design*.

Com este intuito, propomos uma abordagem baseada na análise do que chamamos de “atividade educacional”. Segundo esta idéia, não existe software que, por si só, seja capaz de educar uma pessoa ou gerar aprendizado. O que existe, de fato, são atividades que propiciam situações favoráveis para que uma pessoa memorize, reformule ou construa conhecimentos e sistemas computacionais que oferecem maior ou menor suporte a este tipo de atividade.

Sob esta ótica, nenhum software dito “educativo” pode ser analisado ou desenvolvido sem se levar em consideração a atividade educacional para a qual se dirige. Um editor de textos usado por uma equipe de alunos na produção do jornal da escola incentiva um resultado pedagógico diferente do obtido pelos mesmos alunos quando utilizam o mesmo editor para copiar trechos de livros.

De modo similar, um software que ajude um professor a gerenciar os trabalhos de sua turma também pode ser considerado educativo. Tanto ele quanto o editor de textos auxiliam na atividade educacional, ainda que dirigidos às ações do aprendiz e do professor, respectivamente.

Este tipo de abordagem, que analisa uma ferramenta frente à atividade em que é utilizada, é bastante nova em informática e vem sendo desenvolvida por um grupo pequeno de pesquisadores principalmente da Rússia, Escandinávia, Austrália e Estados Unidos. Basicamente, o que este grupo tem feito é adaptar e aplicar o corpo teórico de mais de 70 anos da Teoria da Atividade da escola russa de psicologia ao desenvolvimento de programas de computador mais adequados ao contexto sociocultural do qual farão parte.

Ainda não existem muitos estudos específicos relacionando a Teoria da Atividade ao desenvolvimento de software ou mesmo ao uso do computador na educação. A análise da aplicação da Teoria da Atividade nestas duas áreas constitui um dos objetivos do presente trabalho.

Como domínio de aplicação e elemento de referência para a abordagem sendo proposta, buscamos suporte no Construcionismo, uma teoria iniciada por Seymour Papert do MIT na década de 80 que se dedica ao estudo do uso da tecnologia na criação de ambientes educacionais cada vez mais efetivos.

Sem dúvida, o resultado pragmático mais conhecido do Construcionismo é a linguagem Logo, uma linguagem de programação bastante simples de ser utilizada e que propicia um solo fértil principalmente para o desenvolvimento do raciocínio crítico e da construção de conceitos matemáticos. Embora o Logo já tenha sido utilizado por milhões de pessoas, poucas são as que compreendem os princípios em cima dos quais este software foi criado. Como decorrência, em muitos casos, elas acabam por associar os bons (e maus) resultados obtidos à tecnologia em si. Não à cultura e dinâmica de aprendizado possibilitadas por ela.

A falta de divulgação do Construcionismo se deve, principalmente, ao fato de, assim como ocorre com a maior parte das pesquisas na área de tecnologia para a educação, seus princípios estarem limitados a alguns poucos centros de pesquisa e se encontrarem fragmentados ao longo de artigos e teses científicas que raramente chegam ao mercado e que dificilmente são acessados por outros educadores (como professores e pais).

Acreditamos que o tipo de abordagem aqui proposta pode contribuir muito para a formação de uma visão do Construcionismo mais integrada que sirva como referência para pesquisadores, educadores e desenvolvedores de tecnologia. A busca por esta visão constitui outro objetivo deste trabalho.

Por fim, um terceiro objetivo é descrever um caso prático que ilustre as dificuldades encontradas na criação de uma atividade educacional e como uma abordagem de desenvolvimento de software baseada nas idéias acima levantadas poderia contribuir para a facilitação de todo o processo.

1.3 Gênese do trabalho

O objetivo desta seção é descrever o caminho percorrido até chegar na atual proposta e formato de trabalho. Espera-se com isto, subsidiar o tipo de enfoque dado e refletir sobre o próprio processo de desenvolvimento. Em se tratando de uma dissertação que aborda o Construcionismo, uma teoria educacional que enfatiza a reflexão do aprendiz sobre seu processo de aprendizagem, uma seção como esta não poderia deixar de estar presente.

Vale notar que, por se tratar de um relato pessoal, esta é a única seção do trabalho escrita em primeira pessoa do singular.

Tudo começou no final de 1994 quando, recém-saído de uma experiência de 4 anos como desenvolvedor de software para telecomunicações, bati às portas do Núcleo de Informática Aplicada à Educação (NIED) da UNICAMP. Já havia tempos que me interessava pela área de tecnologia para a educação – ou da “tecnologia para o desenvolvimento humano, social e cultural”, que é como prefiro tratá-la hoje – e, depois de pedir demissão de meu emprego, estava buscando uma oportunidade de estágio que me ajudasse a concretizar este sonho.

A primeira idéia que me foi proposta no NIED era a de ajudar na criação de uma oficina para professores que mostrasse as diferentes variações da tão falada linguagem Logo de programação e discutisse suas principais tendências.

O trabalho me pareceu bem interessante e não muito complicado. Eu gostava muito de programar e já tinha uma boa experiência com várias das ferramentas de programação mais modernas do mundo. Talvez eu até pudesse apresentar algumas contribuições minhas.

No entanto, depois de meses estudando “Logo orientado a objetos”, “Logo para Windows”, “Logo para Macintosh”, “StarLogo”, “Slogo”, “Micromundos” e outros, ainda não conseguia entender por que as pessoas se preocupavam tanto com aquela linguagem aparentemente simples e sem grandes atrativos.

Minha análise era bem técnica (quantidade de tartarugas, nomes diferentes para os mesmos comandos, etc.) e, tecnicamente, eu não estava vendo nada de excepcional. Muito pelo contrário, o Logo em si não trazia nenhum efeito

extraordinário e seus comandos praticamente não exploravam os recursos de mídia e de telecomunicação do computador.

Um belo dia, explicaram-me que a tartaruga andava devagar na tela não por alguma deficiência de projeto ou de programação, mas para que as crianças pudessem visualizar o resultado de cada comando que elas tivessem dado e, com isso, refletir sobre as estratégias que estavam usando. Esta constatação, aparentemente óbvia, abriu um universo completamente novo em minha mente e foi assim que eu comecei a procurar entender o que mais havia ainda por trás daquela tartaruga e que não podia ser encontrado em outros software.

Foi aí que comecei a fazer meu mestrado.

Li diversos artigos do Papert, estudei um pouco de Piaget e, apesar de concordar com as idéias ali descritas, não conseguia juntá-las de um modo claro. Falava-se muito em “desenvolvimento de uma cultura educacional em sala de aula”, em “formação de professores” e em como o Logo era um excelente micromundo para o aprendizado etc., mas eu não conseguia visualizar como estes princípios poderiam constituir um todo.

Só depois de mais estudos é que fui percebendo que o que importava era a atividade de design e reflexão que os alunos faziam, não simplesmente o fato de programarem o computador. O computador auxiliava este processo e ainda se mostrava extremamente motivante para os alunos. No entanto, também não era qualquer software que trazia os mesmos benefícios.

Não sabia bem o que realmente deveria ser considerado nestes aplicativos e também permanecia com uma questão constante na cabeça: o que de fato é o computador? O que diferencia, sob o ponto de vista de um usuário, a passagem de um filme em vídeo, da passagem do mesmo filme digitalizado no computador? Se só se deixasse a tela e os comandos visíveis, o resultado não seria o mesmo? Que características desta máquina a tornam realmente importante para a educação?

No fim de 1995, tive a oportunidade de viajar para os EUA e lá conversei com diversos pesquisadores que me incentivaram a pesquisar metodologias de desenvolvimento de software para a educação. Também participei do “*Computer Support for Collaborative Learning’ 95*” (SCHNASE e CUNNIUS, 1995) e lá acabei assistindo uma palestra de Victor Kaptelinin sobre Teoria da Atividade. Entusiasmei-me pelos conceitos por ele apresentados. Talvez a Teoria da Atividade pudesse ajudar a organizar as idéias construcionistas de modo que eu pudesse compreendê-las e tirar critérios para o desenvolvimento de software.

Foi aí que comecei a estudar e a interagir com a pequena comunidade que está tentando interligar a Teoria da Atividade com a Engenharia de Software. A dificuldade de encontrar referências e de entender os conceitos gerais foi muito grande. Felizmente, o pessoal desta comunidade se mostrou muito solícito e me deu bastante suporte, respondendo a dúvidas e enviando artigos via Internet e correio. Neste ponto, o recém-lançado livro organizado por Nardi (1996b), com a relação dos nomes e emails dos autores dos diferentes capítulos, foi de extrema valia.

Em fevereiro de 1997, desenvolvi a “Oficina de Jogos” no Ibirapuera. Foi minha primeira tentativa de construção de uma atividade educacional construcionista. Apesar de, atualmente, eu ter consciência de que muitos procedimentos efetuados na Oficina não serem os mais adequados, ela mostrou uma série de pontos aos quais eu não estava dando muita atenção e que não estavam muito claros na minha pesquisa. Embora a influência da Oficina tenha se espalhado por todo o trabalho, o relato de sua execução e de suas conclusões mais diretas encontra-se no capítulo 9.

A partir da Oficina de Jogos, o trabalho tomou uma dimensão mais densa e aprofundada e resolvi embrenhar-me mais na literatura sobre a Teoria da Atividade. Consegui obter, via Internet, uma cópia do livro de Susane Bodker (1991) sobre design de software e Teoria da Atividade (o livro estava esgotado fazia tempo) que me trouxe uma série de conceitos e *insights* de grande importância. Enquanto os princípios mais gerais da Teoria da Atividade podem ser encontrados no capítulo 6, o capítulo 7 se concentra principalmente na relação da Teoria com o computador. Como será visto, este último capítulo deveu muito às idéias de Bodker.

Em fevereiro de 1998, com cerca de 150 páginas escritas, reuni vários amigos professores de diversas áreas e, além da qualificação obrigatória da Engenharia, fiz uma revisão mais específica para cada área do trabalho. Na época, o trabalho parecia um agrupamento de conceitos, principalmente de Construcionismo e Teoria da Atividade. Faltava uma linha mais contínua de pensamento que costurasse as coisas. Faltava também estabelecer a relação entre as idéias apresentadas e o desenvolvimento de software.

O resultado desta qualificação foi extremamente positivo. Diversas referências sobre design de software e sobre Teoria da Atividade e o computador foram indicadas e vislumbres de estruturação do texto também foram apresentados. De certa forma, toda a continuação do trabalho foi um aprimoramento do que foi discutido nesta qualificação.

Resumindo o percurso, parti de uma questão prática que me afligia. Segui para o universo do Construcionismo e da Teoria da Atividade. Depois fui à prática da Oficina de Jogos e voltei para esclarecer novos pontos teóricos que não havia compreendido bem. Se a história tende a se repetir, ao que tudo indica, o próximo passo será, brevemente, voltar para a prática e continuar me esforçando para que este ciclo se multiplique e se intensifique cada vez mais.

1.4 Estrutura

Esta seção descreve a estrutura com que o trabalho foi organizado. Vale notar que, para que possa ser utilizado de modo efetivo, este trabalho foi escrito para educadores e engenheiros de software interessados no uso e no desenvolvimento de ferramentas computacionais para a educação. Por ser um público heterogêneo e

de formação diversa, optou-se por uma distribuição dos capítulos não baseada na seqüência histórica na qual a pesquisa se desenvolveu (descrita na seção 1.3), mas sim no embasamento teórico que seria necessário para a compreensão das idéias.

Assim sendo, o trabalho foi estruturado em 5 seções principais: a de introdução, a de revisão da literatura, a da proposta de uma abordagem de desenvolvimento de software para a educação, a da aplicação da proposta e a das conclusões. Esta estrutura pode ser visualizada na Figura 1.

A primeira seção, de introdução, composta pelo presente capítulo, discorre sobre as principais justificativas por trás do “desenvolvimento de software para atividades educacionais”, descreve os objetivos almejados, explica a origem e o desenvolvimento do trabalho e apresenta a estrutura na qual o texto foi organizado.

A segunda seção compreende uma revisão da literatura. Como o trabalho aborda o desenvolvimento de software para a educação e se aprofunda nos aspectos levantados pela Teoria da Atividade e do Construcionismo, esta seção inclui 4 capítulos, trazendo os conceitos mais relevantes de cada uma destas áreas:

- O capítulo **02. O contexto educacional e o desenvolvimento de software**, introduz critérios gerais a serem considerados no desenvolvimento de tecnologias, entra nas especificidades do caso da educação e discute como as questões levantadas têm sido tratadas pelas principais abordagens de desenvolvimento de software para a educação.
- O capítulo **03. Introdução à Teoria da Atividade** inicia com um histórico da Teoria da Atividade, indo desde sua criação até sua crescente aplicação no desenvolvimento de software. A seguir, ele apresenta as principais dimensões da atividade humana tal como analisadas pela Teoria da Atividade. Ainda que este capítulo seja de fundamental importância para a compreensão dos demais, caso o leitor já esteja familiarizado com os conceitos da Teoria da Atividade, a simples leitura da introdução deste capítulo já será suficiente para que possa seguir adiante.
- O capítulo **04. A Teoria da Atividade e o computador**, realça detalhes da relação da Teoria da Atividade com o uso do computador e com o desenvolvimento de software. Embora não esteja diretamente voltado para a educação, é ao redor das idéias apresentadas neste capítulo que se desenvolveu a abordagem apresentada no capítulo 6.
- O capítulo **05. Introdução ao Construcionismo** encerra a revisão da literatura, apresentando o Construcionismo como uma teoria de aprendizado e de educação. Mesmo que este capítulo não se dedique explicitamente à questão tecnológica, os conceitos por ele abordados contribuem para a percepção (principalmente dos engenheiros de software e dos educadores não familiarizados com teorias socioconstrutivistas) da importância dos aspectos tratados no capítulo 6.

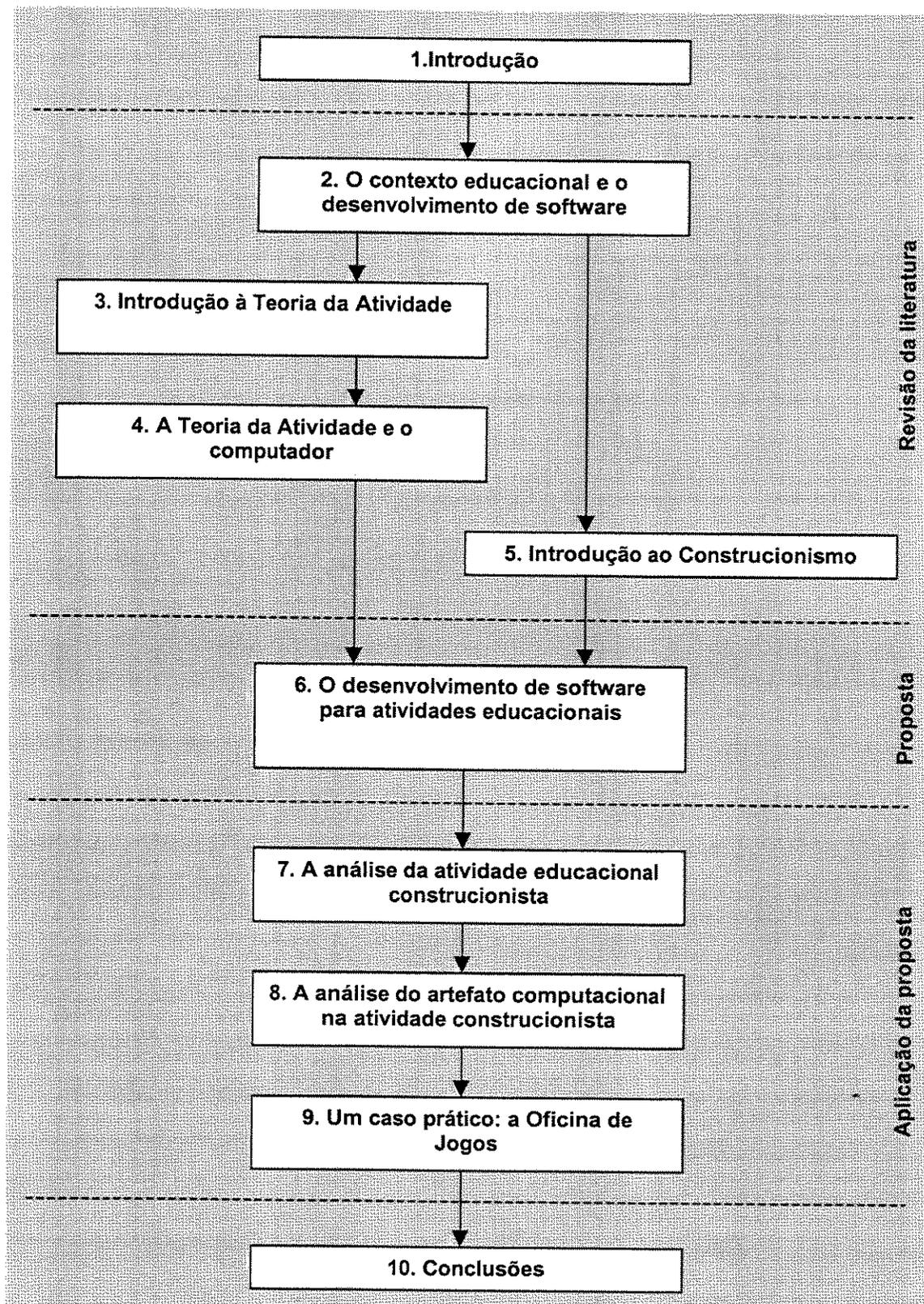


Figura 1 - A organização dos capítulos do trabalho.

A terceira seção do trabalho, composta pelo capítulo **06. O desenvolvimento de software para atividades educacionais**, reúne os conceitos discutidos nos capítulos acima e propõe uma nova abordagem para o desenvolvimento de software para a educação. É a partir desta seção que se encontram as principais contribuições deste trabalho.

Buscando exemplificar como seria esta abordagem na prática, a quarta seção do trabalho, de aplicação da abordagem proposta para o desenvolvimento de software para a educação, entra em detalhes do que seria uma atividade educacional segundo o Construcionismo.

Ela é composta pelos seguintes capítulos:

- **07. A análise da atividade educacional construcionista**, que analisa e descreve, segundo os parâmetros levantados no capítulo 6, os atributos de uma atividade educacional construcionista genérica.
- **08. A análise do artefato computacional na atividade construcionista**, que discute os potenciais e as limitações do computador frente a esta atividade genérica.
- **09. Um caso prático: a Oficina de Jogos**, que analisa uma atividade construcionista real, a Oficina de Jogos e o uso que nela foi dado ao computador.

Por fim, a quinta seção, composta pelo capítulo **10. Conclusões**, reúne os principais pontos levantados pelos demais capítulos e indica possíveis encaminhamentos para a continuidade do trabalho.

2 O contexto educacional e o desenvolvimento de software

O objetivo deste capítulo é apresentar, de uma forma geral, os principais elementos considerados no desenvolvimento de um software e identificar as características que diferenciam a construção e uso de um software para a educação, de um software dirigido a outras áreas.

Com base nos pontos levantados, é feita uma análise de algumas das principais abordagens de desenvolvimento de software atualmente utilizadas, destacando-se os aspectos que precisariam ser melhor considerados no caso da educação.

2.1 O desenvolvimento tradicional de software

Como será defendido ao longo deste trabalho, na nossa opinião, todo desenvolvimento tecnológico deve buscar o melhor equilíbrio entre eficácia, eficiência e viabilidade. Isto é, deve estar sempre voltado para o problema que se pretende resolver e a solução proposta deve ser a melhor dentre as que forem possíveis com os recursos disponíveis. Não adianta se pensar em uma solução que seja cara demais e nem tão pouco em uma que seja esteticamente atraente, mas que não atenda ao objetivo desejado.

A busca pela solução ideal não é nada fácil e, como será explorado para a educação, costuma envolver, além dos aspectos técnicos, uma série de outros elementos específicos do domínio e do contexto que precisam ser igualmente considerados. A questão é como identificar os fatores relevantes em cada situação e determinar que tipo de tecnologia seria a mais apropriada.

Normalmente, no caso da informática, a complexidade do desenvolvimento é tratada por metodologias que orientam desde a avaliação do problema até a implantação e manutenção do software construído. A fim de lidar o melhor possível com as restrições impostas pela máquina, cada metodologia tem um enfoque específico que determina que aspectos do problema deverão receber maior prioridade do que outros. Algumas metodologias, por exemplo, são dirigidas para o desenvolvimento de software mais rápidos e seguros, como os necessários em transações bancárias. Outras, para lidar com grandes quantidades de informações, como ocorre nos software de gerenciamento de estoque e de pessoal nas empresas.

À medida em que os computadores vão se tornando mais potentes e mais baratos, novos usos começam a mostrar-se viáveis e as metodologias de desenvolvimento de software vão diversificando-se e tornando-se mais sofisticadas. Nota-se, por exemplo, que com a popularização das interfaces gráficas e dos microcomputadores na década de 80, surgiram metodologias mais preocupadas com o usuário. Até

então, o poder de processamento do computador mal dava para efetuar os cálculos que lhe eram exigidos e, por isso, os aspectos relacionados à estética e à maior ou menor facilidade de uso de um software eram delegados a um segundo plano e raramente questionados.

Por facilitar a compreensão dos fatores relevantes na relação humano-computador e tratar da criação de software “mais amigável”, destacaram-se, nesta época, duas abordagens básicas: a da Ciência Cognitiva, de Card et al. (1983) e a do Design Centrado no Usuário, de Norman e Draper (1986).

A idéia central da Ciência Cognitiva é modelar o ser humano como um processador de informações (ver a Figura 2). Segundo este conceito, a relação do ser humano com o mundo pode ser descrita através de um ciclo em que o indivíduo recebe estímulos do meio a partir de componentes sensoriais (olhos, ouvidos, etc.), armazena estes estímulos em memórias de maior ou menor duração, processa-os e envia comandos para os órgãos motores (os músculos).

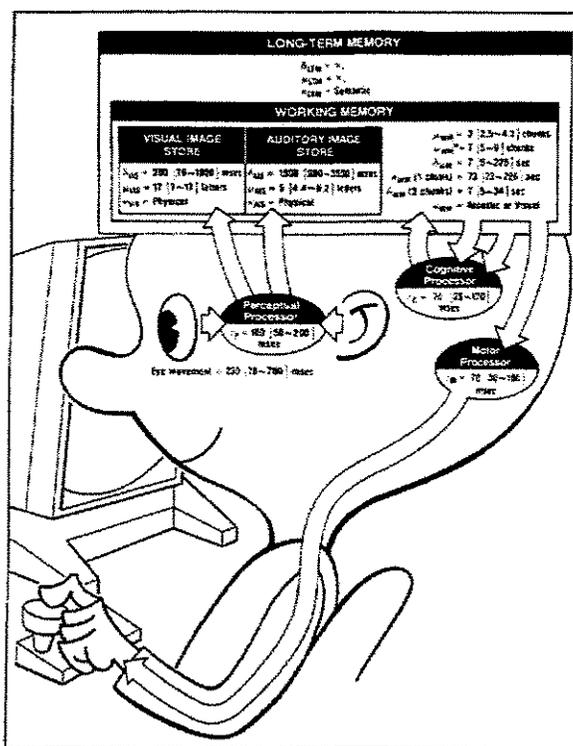


Figura 2- Diagrama do processador humano (CARD et al., 1983, pg. 26)

Esta visão, que trabalha o ser humano à imagem da máquina, é muito interessante para estudos de ergonomia e avaliação de desempenho – úteis, por exemplo, no design de teclados mais eficientes para a digitação de dados, seleção de cores, etc. – mas parece ser muito limitada no que se refere às situações em que o desenvolvimento humano e a motivação tenham que ser consideradas (KUUTI, 1996, p.21), como ocorre no caso da educação. Os próprios autores da teoria do processador humano de informações levantam esta questão: “Por outro lado, os aspectos de motivação e personalidade não estão incluídos [na teoria]. Novamente,

praticamente não há dúvidas da importância de incluí-los em uma psicologia aplicada, mas não está claro como integrar o conhecimento relevante existente destes tópicos” (CARD et al., 1983, p.14).

Seguindo por um caminho alternativo ao da Ciência Cognitiva, a abordagem do Design Centrado no Usuário analisa a relação humano-computador como algo assimétrico. Segundo ela, pessoas distintas possuem necessidades, níveis de experiência e visões de mundo diferentes umas das outras e isso interfere diretamente na forma com que interagem com o computador. Esta observação é extremamente relevante, uma vez que o maior ou menor respeito a estas características particulares tornará um software mais atraente ou fácil de usar do que outros.

Outro ponto importante levantado pelo Design Centrado no Usuário é de que existe uma distância bastante grande entre o tipo de interação usuário-computador idealizada pelos desenvolvedores e a que ocorre de fato na realidade. A percepção deste aspecto incentivou o aparecimento de metodologias de desenvolvimento de software baseadas na criação e avaliação de protótipos e valorizou a formação de equipes multidisciplinares que, além dos desenvolvedores, também integrassem especialistas das diversas áreas relacionadas e futuros usuários².

Hoje em dia já existe uma enorme variedade de metodologias, heurísticas e diretrizes descrevendo como se observar usuários em ação, como criar maquetes e cenários de uso, como avaliar o impacto causado por alterações em um produto, como testar novos sistemas, etc. (LAUREL, 1990; NIELSEN, 1993).

Por outro lado, o tipo de abordagem lançado pelo Design Centrado no Usuário ainda é muito recente e se encontra fragmentado em pedaços muito específicos. Os trabalhos disponíveis não chegam a constituir um corpo teórico que possa orientar o desenvolvimento e reduzir o esforço gasto na implementação das várias alternativas (KUUTI, 1996).

2.2 O desenvolvimento de tecnologia para a educação

No caso particular do desenvolvimento de tecnologia para a educação, os mesmos princípios de eficácia, eficiência e viabilidade descritos na seção anterior também devem ser respeitados.

² Diversas facilidades de uso encontradas nos computadores modernos resultaram de parcerias multidisciplinares e da utilização de protótipos em seu desenvolvimento. O projeto Star (SMITH et al., 1982), do Xerox PARC, por exemplo, considerado o primeiro sistema a utilizar, de forma integrada, os conceitos de janelas, ícones e mouse em uma metáfora do tipo “desktop”, foi desenvolvido por uma equipe formada por psicólogos e engenheiros que, ao lado de usuários, trabalharam extensivamente em cima de modelos do que seria um computador mais intuitivo e fácil de usar.

No entanto, deve-se ressaltar que a **eficácia educacional**, ou o grau com que um indivíduo desenvolve um determinado conhecimento ou capacidade, não é resultado direto da utilização de uma determinada tecnologia, mas sim da maior ou menor influência de todo um conjunto de condições (sociais, culturais, históricas, biológicas, políticas e econômicas) propícias para o ensino-aprendizado do conteúdo desejado. A eficácia da tecnologia em si dependerá de sua contribuição para a viabilização e implementação destas condições.

Infelizmente, devido a transformação educacional descrita no capítulo anterior, a proposta educacional moderna é muito diferente da tradicional deixando, com isso, de ser algo intuitivo onde as pessoas possam se basear em suas próprias experiências escolares.

Além disso, não existe um consenso teórico que possa orientar a prática e as condições educacionais a serem enfatizadas e a forma de se avaliar os resultados obtidos variam conforme a teoria educacional adotada.

O que se sabe, por outro lado, é que, hoje em dia, as chamadas teorias sócio-construtivistas estão sendo bem valorizadas. Estas teorias educacionais, baseadas principalmente nas idéias de Piaget e Vygotsky, enfatizam a motivação e a participação ativa do aprendiz na construção de seu conhecimento.

Na educação sócio-construtivista, os aprendizes são incentivados a identificar e resolver problemas que lhes sejam interessantes e úteis. Procura-se obter, com isso, um contexto significativo para a exploração dos tópicos curriculares. É a chamada "educação baseada em projetos", onde mesmo as aulas tradicionais encontram seu espaço, alternando-se com sessões mais práticas e participativas, nas quais os aprendizes propõem, analisam e discutem as diferentes soluções adotadas.

Convém notar que este tipo de filosofia educacional já existe há um bom tempo, como pode-se ver, por exemplo, nos escritos de Dewey (1938). Ainda assim, embora não seja um fator essencial para a sua realização, acredita-se que, com a utilização de novas tecnologias, ela ganha um novo impulso, tornando viáveis uma série de idéias até então difíceis de serem implementadas (NORMAN e SPOHRER, 1996, p.26).

Ainda assim, apesar de sua importância e do apoio que a tecnologia poderá lhes dar, estas abordagens educacionais não costumam estar organizadas de forma a facilitar o desenvolvimento de software. Engenheiros e educadores falam linguagens diferentes e raramente pensam na mesma direção. Na maior parte dos casos, os artigos e trabalhos se dirigem unicamente aos aspectos técnicos ou aos puramente educacionais, tornando difícil uma visão mais ampla e integradora das diferentes áreas.

Tal como no caso da eficácia, a **eficiência educacional** de uma tecnologia não pode ser medida independentemente do contexto e da dinâmica nos quais ela está inserida. Conforme dito anteriormente, no exemplo do editor de textos sendo usado para criar um jornal ou para copiar um texto, é o tipo de dinâmica criado e não a

simples presença de uma ferramenta que possibilitará resultados pedagógicos mais ou menos satisfatórios.

Não existe nenhuma tecnologia educacional que mereça o adjetivo “mais educacional” ou que possa ser considerada “melhor” do que qualquer outra, sem que se faça a consideração de um contexto maior. Cada caso é um caso e a tecnologia mais eficiente será aquela que melhor suprir as necessidades da situação educacional que está sendo criada. Neste sentido, a eficiência viria da facilidade de uso da tecnologia, das formas com que ajuda na implantação e acompanhamento da dinâmica educacional, na sua integração com o resto do ambiente, etc.

Apesar do modismo ao redor da informática, nem sempre o computador mais veloz e mais cheio de recursos é a solução mais adequada para um problema educacional. Às vezes, uma atividade criativa com um pedaço de papel ou com uma bola pode resolver o problema de uma forma mais eficaz, motivante, simples e barata.

No entanto, isso não significa que a informática não tenha contribuições a oferecer para a educação. Muito pelo contrário, como será discutido nos capítulos seguintes, ela pode contribuir muito. A questão é identificar os aspectos em que ela realmente se sobressai frente a outras alternativas e utilizá-la da melhor forma possível³.

No que se refere à **viabilidade**, a análise de uma tecnologia para a educação deve considerar os gastos de desenvolvimento e, além deles, dedicar uma atenção especial aos custos de implantação e manutenção.

Os custos de desenvolvimento são, em sua maioria, técnicos e incluem, basicamente, o material gasto e o trabalho investido na construção da tecnologia. Os custos de implantação envolvem principalmente a formação e treinamento do pessoal que usará a tecnologia e mais todo o esforço que será gasto para se transformar a situação que se tem correntemente na que se deseja (mudanças físicas e de funcionamento).

Os custos de manutenção estão relacionados às mudanças, previstas ou não, que deverão ser efetuadas na tecnologia ao longo de sua utilização (obsolescência de equipamentos, mudança de sistemas operacionais, etc.).

É importante frisar que a não consideração dos elementos sócio-histórico-culturais envolvidos na implantação e na manutenção pode condenar uma tecnologia, por mais eficaz ou eficiente que ela pareça. Do início do século até hoje, diversas tecnologias, como o rádio, o cinema e a televisão, já foram vistas como as soluções que faltavam para resolver os grandes problemas educacionais. Segundo Cuban (1986), apesar dos gigantescos investimentos efetuados neste sentido, os resultados não foram muito satisfatórios justamente pela não consideração, por

³ Complementando este ponto, o capítulo 6 introduz toda uma discussão sobre o diferencial do computador em relação a outras tecnologias e, a partir de idéias da Teoria da Atividade, define conceitos como artefato computacional, artefato digital, aplicativo e objeto digital.

exemplo, da rotina e da opinião dos professores que utilizariam diretamente estas novas ferramentas.

Além da avaliação destes custos, a viabilidade da tecnologia dependerá de uma boa avaliação do mercado e de outros fatores, como o montante de investimento e os incentivos governamentais, que indicarão se o projeto poderá ter continuidade ou não.

Percebe-se que, atualmente, a demanda por tecnologia para a educação tem se ampliado e se diversificado bastante, ultrapassando o tradicional suporte a escolas e universidades, penetrando cada vez mais na formação continuada de funcionários e na indústria de entretenimento nos lares (PERKINS e NUÑEZ, 1994).

As características específicas de cada um destes mercados têm que ser bem avaliadas e respeitadas para que a nova tecnologia seja aceita e possa reverter lucros em benefício a novos investimentos na área. De nada adianta um produto teoricamente perfeito que não seja reconhecido por seus usuários em potencial ou que não possa ser desenvolvido em uma escala mínima aceitável.

Por outro lado, na busca pela maximização do lucro imediato, a tendência dos desenvolvedores é priorizar as soluções mais baratas de criar e que possam ser vendidas para a maior quantidade de pessoas possível. O resultado disto é que, na maior parte dos casos, embora as empresas, os lares e as instituições educacionais tenham dinâmicas e necessidades diferentes – o tipo de atividade exercida por uma criança na escola ou em sua casa é completamente diverso do de um profissional trabalhando – existe uma pressão muito grande para que todos utilizem o mesmo tipo de máquinas e software, fazendo com que a questão da eficácia educacional acabe sendo delegada a um segundo plano.

Analisando-se as metodologias tradicionais de desenvolvimento de software frente a estas idéias de eficácia, eficiência e viabilidade educacional, percebe-se que muitos pontos acabam ficando em aberto. Embora a Ciência Cognitiva e o Design Centrado no Usuário façam considerações com relação a melhorias de performance ou aos diferentes níveis de experiência do usuário, os problemas educacionais envolvem todo um contexto sociocultural mais amplo do que o possibilitado pela análise isolada da relação entre o usuário e a máquina⁴.

Mesmo que a crítica a estas abordagens não seja totalmente justa, uma vez que nenhuma delas foi criada com objetivos educacionais, ela acaba por ressaltar o fato de que, apesar de sua complexidade específica, ainda resta muito a ser feito no que se refere ao desenvolvimento de tecnologia para a educação.

A maior parte do software continua sendo criada em centros de pesquisa, onde são efetuados principalmente estudos em modelos ideais e distantes da realidade

⁴ Convém notar que, no livro fundamental do Design Centrado no Usuário, "*User centered system design*" (NORMAN e DRAPER, 1986), apareciam alguns artigos, como os de Brown (1986) e Bannon (1986a; 1986b), que já tratavam da questão sociocultural. No entanto, por razões históricas, o maior desenvolvimento da pesquisa acabou se dando principalmente na relação humano-computador.

cotidiana, ou em empresas comerciais, que tendem a perpetuar o modelo educacional vigente, usando recursos gráficos e sonoros como forma de motivação. Não existe um objetivo compartilhado e, para completar, a falta de um vocabulário comum entre pesquisadores, educadores, educandos e desenvolvedores de tecnologia ajuda a manter esta situação estagnada.

Apesar disso, já existem abordagens que se propõem a tratar de alguns problemas específicos da educação, mas nenhuma delas consegue tratar o problema de uma forma mais abrangente.

A do Design Centrado no Aprendiz (ou *Learner-Centered Design*), por exemplo, procura estender os conceitos do Design Centrado no Usuário no sentido de criar ferramentas que, além de fáceis de usar, também ajudem o usuário a adquirir mais conhecimento sobre o que estiver fazendo (SOLOWAY et al., 1994; SOLOWAY e PRYOR, 1996; BORGES, 1997).

O Design Centrado no Aprendiz parte de uma abordagem educacional sócio-construtivista (SPITULNIK et al., 1995) e realça muito o papel diferenciado do computador, devido à sua crescente utilização em todos os setores da sociedade, na implantação de uma educação mais contextualizada e próxima da realidade.

Segundo esta abordagem, de uma forma geral, aprendizes também são usuários, o que significa que as necessidades comuns de facilidade de uso também se aplicam. O que o software para a educação traz de diferente é uma preocupação maior com a motivação (o aprendizado envolve lidar com situações desconhecidas para o aprendiz e isto é bastante desgastante), com o desenvolvimento (as necessidades de um novato são diferentes das de um especialista) e com a diversidade (aprendizes têm conhecimentos e estilos de aprendizagem diferenciados) dos usuários.

Apesar dessa sua grande contribuição, o Design Centrado no Aprendiz concentra-se quase que exclusivamente no tipo de suporte que o software em si poderá oferecer ao aprendiz, as chamadas técnicas de *scaffolding*. Ele não considera as necessidades dos demais envolvidos no processo educacional, como é o caso de professores, administradores, familiares, colegas, etc. e nem as diferenças de tempo e recursos existentes nos ambientes onde o software será implantado⁵.

Tal como frisado anteriormente, estes fatores podem levar ao sucesso ou fracasso da empreitada. Isso porque, *“para a revolução [educacional] dar certo, as necessidades de todos os envolvidos devem ser consideradas, ou eles poderão bem se tornar inimigos das mudanças”* (NORMAN e SPOHRER, 1996, p.27). Assim, na nossa opinião, ao invés de uma abordagem centrada exclusivamente no aprendiz, talvez o ideal fosse uma centrada na educação como um todo.

Outra abordagem de desenvolvimento de software para a educação é a do *ESSCOTS for Learning* (MCARTHUR et al., 1994). Esta abordagem é muito similar

⁵ Parece claro, por exemplo, que o tipo de suporte a ser oferecido pela tecnologia em uma sala de aula com professores é muito diferente do que deveria existir no caso de um ensino individual à distância...

à do Design Centrado no Aprendiz e sofre, inclusive, da mesma deficiência. No entanto, ela tem uma preocupação maior com a questão da viabilidade e propõem que o aproveitamento de software comercial já existente poderia facilitar a criação de software para a educação e permitir uma maior disponibilidade de recursos para as questões mais específicas da área.

Por fim, outra abordagem que vem ganhando impulso é a do *Educational Object Economy* (ROSCHELLE et al., 1998; <http://www.eoe.org/>). Ela discute a falta de integração entre os diferentes tipos de software para educação existentes no mercado e propõe a criação de padrões e a utilização da Internet para a formação de uma comunidade de desenvolvedores que discuta os pontos comuns e maximize a reutilização dos esforços já efetuados.

Tanto o *ESSCOTS for Learning* quanto o *Educational Object Economy* ressaltam que, apesar das dificuldades técnicas, os maiores problemas do desenvolvimento da tecnologia educacional se dão no aspecto social e cultural. É preciso aproximar mais a pesquisa e o desenvolvimento da prática e criar formas de operação em que cada um possa contribuir na sua área de especialização.

Felizmente, com a proliferação das redes e dos computadores móveis nos últimos cinco anos, os aspectos socioculturais relacionados ao uso da tecnologia passaram a se tornar mais evidentes e uma série de estudos neste sentido começaram a ser desenvolvidos dentro e fora da área educacional (NARDI, 1993; SCHNASE e CUNNIUS, 1995).

Dentre as diversas tendências que estão ganhando força, a abordagem baseada na Teoria da Atividade merece uma atenção especial, principalmente pelo tipo de estrutura teórica que oferece para a análise de casos reais e pelo potencial de integração com as diversas áreas relacionadas ao desenvolvimento de tecnologia para a educação.

3 Introdução à Teoria da Atividade

Este capítulo descreve, em primeiro lugar, um breve histórico da Teoria da Atividade, explicitando a sua expansão no ocidente e o seu crescimento junto à comunidade de Engenharia de Software.

Depois disto, são apresentados os principais conceitos da Teoria da Atividade relacionados com a utilização da tecnologia e a educação. No capítulo 4, esses conceitos serão melhor detalhados para o caso do computador e, no capítulo 6, servirão de base para a abordagem de desenvolvimento de software para a educação nele proposta.

3.1 Breve histórico

A Teoria da Atividade é considerada por muitos uma das abordagens conceituais mais importantes da psicologia soviética, podendo ser vista quase que como um pilar central sobre a qual as diversas linhas de pesquisa foram desenvolvendo-se e especializando-se.

Apesar disso e de suas quase oito décadas de existência, ela é ainda pouco conhecida no ocidente. Segundo Wertsch, isso se deve principalmente a problemas de tradução e à falta de interesse dos pesquisadores ocidentais que a julgavam desatualizada e irrelevante (WERTSCH, 1981, p.6).

Historicamente, assim como a maioria das correntes da escola russa, a Teoria da Atividade foi fortemente influenciada pelo trabalho de Marx e Engels (WERTSCH, 1981, p.9), que tinham o conceito de atividade como central em suas teorias de integração entre mundo físico e mundo mental. Na década de 20, a Teoria da Atividade foi apropriada pelo grupo de Vygotsky, sendo depois consolidada e integrada ao formato que conhecemos hoje por um de seus alunos, A. N. Leontiev⁶.

No que se refere às abordagens cognitivas mais tradicionais do ocidente, as maiores diferenças do enfoque da Teoria da Atividade residem principalmente na integração entre a atividade mental e a atividade externa do sujeito (LEONTIEV, 1974); em um direcionamento voltado para o estudo dos sistemas funcionais que se formam na interação do ser humano com o mundo – ao invés de se concentrar na

⁶ Convém enfatizar, no entanto, que a Teoria da Atividade e as idéias de Vygotsky são coisas distintas. A Teoria da Atividade teve suas raízes no pensamento de Vygotsky e dele incorporou principalmente os aspectos relacionados à influência do social no desenvolvimento do indivíduo. Os conceitos de atividade, motivação e de percepção subjetiva do mundo são mais específicos da própria Teoria da Atividade.

busca por estruturas genéricas que operem independente de objetivos e contextos, como ocorre, por exemplo, na teoria de Piaget (WERTSCH, 1981, p.21); e na influência do contexto no comportamento do indivíduo, o que a leva a considerar os experimentos simplificados feitos em laboratório como de pouca valia.

A Teoria da Atividade e o desenvolvimento de software. Com relação ao uso da Teoria da Atividade no desenvolvimento de software, sabe-se que, tal como discutido no capítulo 2, no fim da década de 80, com o rápido desenvolvimento e conseqüente expansão da informática para os mais diversos domínios (computação pessoal, multimídia, telecomunicações, etc.), iniciou-se uma busca por metodologias de desenvolvimento de software que dessem suporte à resolução dos novos problemas.

Nesta época, na Escandinávia, talvez por uma questão cultural ou de proximidade geográfica com a Rússia, já havia uma série de trabalhos de diversas áreas sendo desenvolvidos usando a Teoria da Atividade. Isso acabou por criar condições favoráveis para que a teoria também fosse utilizada no campo da computação. Depois, o intercâmbio de pesquisadores entre universidades, os congressos internacionais (como os *East-West International Conference on Human-Computer Interaction* de 1993, 1994 e 1995, o *Computer Support for Collaborative Learning* de 1995 e 1997, etc.) e a publicação de livros ajudaram a divulgar as novas idéias para o resto do mundo.

Talvez a primeira grande obra sobre a aplicação da Teoria da Atividade no desenvolvimento de software seja "*Through the interface: a human activity approach to user interface design*" (BODKER, 1991), em que a autora, por meio de uma coleção de exemplos práticos, atenta para uma série de pontos interessantes como, por exemplo, a questão da multi-disciplinaridade nas equipes envolvidas no design das interfaces e a da importância do contexto sociocultural na criação de software bem sucedidos.

Em 1996, o casamento entre Engenharia de Software e Teoria da Atividade recebeu um novo impulso com a publicação de "*Context and consciousness: activity theory and human-computer interaction*" (NARDI, 1996b). Este livro traz uma coletânea de artigos que cobrem a apresentação de conceitos básicos, a discussão dos principais problemas da área, exemplos práticos e também um pouco do desenvolvimento teórico.

No início de 1997, a Teoria da Atividade ocupou lugar de destaque no CHI'97 (KAPTELININ e NARDI, 1997), um dos mais importantes congressos do mundo na área da informática que estuda a Interação Humano-Computador (IHC). Desde então, temas relacionados com a Teoria da Atividade têm aparecido mais frequentemente em artigos da área.

Expectativas com relação à Teoria da Atividade. De um modo geral, esses trabalhos têm mostrado que, apesar de a Teoria da Atividade ainda estar ensaiando seus primeiros passos na computação, há uma grande expectativa e muitos defendem suas potenciais contribuições para as diversas frentes do desenvolvimento de software. Dentre elas, destacam-se:

- **Incorporação de fatores humanos.** Diferente de algumas abordagens tradicionais como a da Ciência Cognitiva (ver capítulo 1), a Teoria da Atividade aborda, de forma integrada, a questão da consciência, da motivação e da intencionalidade, insistindo em diferenciar pessoas e coisas. *“Na Teoria da Atividade, artefatos são mediadores das atividades humanas; eles não ocupam o mesmo espaço ontológico”* (NARDI, 1996b, p.13). Mesmo assim, a Teoria da Atividade não deve ser vista como uma rejeição da Ciência Cognitiva, mas como uma expansão radical dela. A IHC precisa estudar os artefatos e suas influências na vida das pessoas. A Ciência Cognitiva se concentrou demais na representação de modelos mentais e ignorou o estudo dos artefatos (NARDI, 1996b, p.14).
- **Mudança no enfoque.** Os estudos baseados na Teoria da Atividade levam ao questionamento inclusive do próprio nome “Interação Humano-Computador”. Partindo da concepção do computador como um artefato, a Teoria da Atividade propõe que as pesquisas abordem não somente a interação humano-máquina, mas também todas as interações que o indivíduo, com a máquina, tem com os demais objetos e pessoas do seu contexto de trabalho (KAPTELININ, 1996b, p.50). Neste caso, conforme será discutido no capítulo 4, o nome mais adequado para IHC talvez fosse “Interação Mediada pelo Computador”.
- **Integração interna.** Acredita-se que o uso da Teoria da Atividade possibilitará a formação de um vocabulário comum que poderá ser usado para descrever a atividade humana, com ou sem computador, de tal forma que facilite a integração entre as diversas especialidades encontradas dentro da própria IHC. Atualmente, cada uma tem sua metodologia e terminologia, o que impede a troca de idéias e limita o desenvolvimento da área como um todo (NARDI, 1996b, p.10; KUUTI, 1996, p.37).
- **Integração com outras áreas.** Os problemas mencionados acima também inibem a integração da Engenharia de Software com suas áreas correlatas, como ocorre com a educação. A Teoria da Atividade pode abrir o escopo da pesquisa para outras áreas e já traz uma ampla bagagem com trabalhos de filósofos, psicólogos, antropólogos, lingüistas, educadores e outros cujos pensamentos influenciaram o seu desenvolvimento (NARDI, 1996b, p.10).
- **Exploração de novos domínios.** Além de contribuir para unificação e a integração, a Teoria da Atividade também pode contribuir para a exploração de novas áreas como, por exemplo, a do desenvolvimento de software que suporte o trabalho em equipe, de software que se reconfigure de acordo com o desenvolvimento do nível de competência de seus usuários, de software que trabalhe a aprendizagem de novos conceitos, etc. (KUUTI, 1996, p.38; KAPTELININ, 1996, p.48). Pouco se sabe destas áreas e a Teoria da Atividade já oferece pelo menos alguns conceitos básicos por onde iniciar a pesquisa.
- **Análise de situações de uso.** Como o desenvolvimento da atividade é visto como intimamente ligado à sua prática, a Teoria da Atividade privilegia os estudos feitos fora de laboratório, principalmente aqueles que observam os sujeitos em seu contexto normal de trabalho. Nesse sentido, ela já traz todo um

ferramental conceitual que poderá ser aproveitado no desenvolvimento de software. Ainda assim, a Teoria da Atividade não se prende a nenhum método específico de estudo. Ela só prescreve que os métodos devem ser escolhidos a partir do problema que se tiver em mãos (KAPTELININ e NARDI, 1997, p.77).

Na realidade, apesar dessas fortes expectativas, convém notar que os pesquisadores da Teoria da Atividade não a vêem como uma solução milagrosa capaz de resolver todos os problemas da IHC. Ao invés disto, acredita-se que as melhores soluções virão de uma expansão da Teoria da Atividade e de uma combinação com as outras abordagens já existentes (SHUKLA, 1997).

De fato, até o momento, pouco parece ter-se avançado, além do trabalho de Bodker, no desenvolvimento de uma metodologia mais prática que pudesse orientar, de uma forma geral, a atividade de desenvolvimento de software – quanto mais a atividade de desenvolvimento de software para a educação.

Os conceitos parecem claros e a Teoria parece trazer respostas (ou abordagens) para uma série de problemas tradicionais da interação humano-computador. No entanto, um analista de sistemas motivado pelas idéias da Teoria da Atividade teria, provavelmente, grande dificuldade em saber por onde começar o seu trabalho, ou mesmo avaliar o impacto que uma abordagem orientada pela atividade humana traria para a sua prática corrente.

Pensando-se por este lado, tem-se a impressão de que a relação entre a Teoria da Atividade e a Interação Humano-Computador esteja agora iniciando uma nova fase. Da fase inicial, que procurava mostrar a viabilidade e levantar possíveis contribuições, começa a surgir um amadurecimento e a busca por uma abordagem mais sólida e aceita que pudesse servir de referência para a prática.

Na nossa opinião, a *Activity Checklist* de Kaptelinin e Nardi (1997; KAPTELININ et al., 1999) pode ser vista como um trabalho dentro desta segunda fase. A *Activity Checklist* é, basicamente, uma lista dos pontos principais do contexto que devem ser considerados na análise e no projeto de artefatos computacionais. Nas palavras dos próprios autores, “é um guia para as áreas específicas que um pesquisador ou praticante deverá prestar atenção quando estiver tentando compreender o contexto no qual a ferramenta é ou será usada” (KAPTELININ et al., 1999, p.28).

A *Checklist* cobre uma área bem ampla. Ela deve ser usada para se ter uma noção geral do contexto e identificar pontos críticos que precisariam de um melhor aprofundamento. Apesar de servir como um bom ponto de partida para o desenvolvimento, os próprios autores a definem como uma ferramenta criada para uma análise “rápida e rasteira” (“*quick and dirty*”). Ela apresenta uma série de questões que deveriam ser consideradas na análise e no desenvolvimento, mas não traz explicações mais detalhadas com relação ao uso e às situações em que as questões se tornam mais pertinentes.

Assim, pelo menos a partir das referências consultadas, tem-se a impressão de que ainda faltam trabalhos mais detalhados e completos, mostrando como, de fato, seria uma abordagem de desenvolvimento de software baseada nas idéias da Teoria da Atividade.

No caso específico do software para a educação, a situação ainda é pior. Há uma certa quantidade de material relacionado com a análise de situações de uso do computador em atividades educacionais (KAPTELININ e COLE, 1997; NEWMAN, 1997), mas muito pouco no que se refere a uma metodologia de desenvolvimento propriamente dita (BELLAMY, 1996). O objetivo deste trabalho, como será melhor visto nos próximos capítulos, é oferecer mais uma contribuição neste sentido.

3.2 Conceitos fundamentais

A Teoria da Atividade, como diz o próprio nome, é um conjunto de idéias que procura compreender e explicar a atividade humana. “Por que as pessoas fazem as coisas do jeito que fazem?”, “O que as motiva?”, “Como elas se desenvolvem?”, “Que fatores influenciam suas ações?”, “O que varia de uma situação para a outra?” são exemplos de questões que ela procura responder.

Devido ao seu amplo escopo, a Teoria da Atividade está longe de ser um corpo fechado de idéias. Mais do que uma teoria altamente preditiva, ela deve ser compreendida e usada como uma abordagem conceitual geral que ajuda a levantar questões significativas para as mais diferentes áreas (KAPTELININ et al., 1999, p.32).

De acordo com Kuuti, a Teoria da Atividade pode ser descrita como “*uma abordagem filosófica e transdisciplinar para o estudo das diferentes formas das práticas humanas e dos processos de desenvolvimento, com ambos os níveis individual e social interligados ao mesmo tempo*” (KUUTI, 1996, p.25). Neste sentido, ela se mostra bem interessante para o estudo das dinâmicas educacionais.

Embora não haja nenhuma estrutura rígida que oriente a descrição de seus conceitos fundamentais, resolveu-se apresentar a Teoria da Atividade a partir dos cinco princípios básicos levantados por Kaptelinin e Nardi⁷ (1997; KAPTELININ et al., 1999):

- **Estrutura hierárquica da atividade**, onde se apresenta o conceito de atividade, seus diferentes níveis de análise e a relação dinâmica existente entre eles.
- **Orientação a objetos**, que trata de motivação e de alguns aspectos culturais que influenciam a atividade humana.
- **Internalização e externalização**, mostrando algumas relações entre as práticas sociais e o desenvolvimento do indivíduo.

⁷ Cada autor parece ter sua própria estrutura de apresentação dos conceitos fundamentais da Teoria da Atividade. Wertsch, por exemplo, tem o seu (WERTSCH, 1981) e o próprio Kaptelinin também trabalha com outras divisões (KAPTELININ, 1996).

- **História e desenvolvimento**, mostrando como as atividades se desenrolam no tempo e como elas são afetadas pelos resíduos históricos incorporados durante sua evolução.
- **Mediação**, onde é explorada a importância dos artefatos e seu papel de potencialização e limitação dos demais conceitos levantados acima.

Estes conceitos serão melhor abordados a seguir⁸.

3.2.1 A estrutura hierárquica da atividade

Atividades implicam um contexto. Teoria da Atividade define como unidade de análise o que ela chama de atividade humana, isto é, um conjunto de ações acoplado a um contexto mínimo que permita a sua compreensão.

Para um observador externo, a simples visão de uma pessoa fazendo algo (trabalhar no computador, por exemplo) não é suficiente para compreender suas razões ou o que ela está pensando. Dependendo do grau da análise desejada, será necessário um foco de observação mais amplo que englobe, dentre outras coisas, algumas características psicológicas, sociais, culturais, econômicas e históricas.

Toda atividade tem um motivo. Simplificadamente, pode-se visualizar a atividade humana como uma transformação de objetos em um outro objeto, material ou não, que satisfaça uma determinada necessidade ou desejo da pessoa que a executa. Sem a compreensão desta necessidade ou desejo, torna-se praticamente impossível identificar quais dos fatores sendo analisados na atividade têm maior ou menor relevância.

Segundo Leontiev (1974, p.22), o conhecimento do objeto almejado é fundamental para a boa compreensão da atividade. É ele quem a orienta, motiva e a distingue das demais atividades.⁹

Diferentes atividades podem se intercalar. Um mesmo indivíduo pode estar engajado em diferentes atividades ao mesmo tempo – é até relativamente comum dividirmos nosso tempo entre atividades de trabalho, lazer, etc. – e diferentes atividades podem ser combinadas em atividades coletivas maiores, como aconteceria, por exemplo, no caso de um trabalho em grupo.

As atividades são coletivas. Na realidade, as atividades só podem ser compreendidas quando analisadas em seu contexto coletivo. Isso ocorre, inclusive, quando uma pessoa trabalha “sozinha” em alguma coisa. Neste caso, sua atividade está condicionada a diversos artefatos – o computador em que está mexendo, o livro que está lendo, a forma com que aborda seus problemas, ... – e outras variáveis criadas por outras pessoas. *“Nós bem podemos falar da atividade do*

⁸ Os princípios de internalização e externalização, história e desenvolvimento e mediação são comuns à teoria de Vygotsky.

⁹ O conceito de objeto na Teoria da Atividade, tal como utilizado neste trabalho, será melhor discutido na seção 3.2.2.

indivíduo, mas nunca de uma atividade individual; apenas ações são individuais" (ENGESTRÖM, 1987, p.66).

Neste caso, os autores e criadores destes artefatos (como desenvolvedores de software) acabam também por participar, ainda que não tão diretamente, da comunidade envolvida na atividade que está sendo executada.

Além deste tipo de envolvimento mais passivo na atividade, Kaptelinin ressalta que, para a Teoria da Atividade, a noção geral de um sujeito interagindo com um ambiente em um contexto social "*é aplicável não só a indivíduos, mas também a grupos e a organizações*" (KAPTELININ, 1996b, p.57) e pode ser utilizada para se compreender atividades de grupo mediadas por computador¹⁰.

Atividades são compostas por ações. Outro ponto importante levantado pela Teoria da Atividade é que a transformação de um objeto em produto não é direta. As atividades são realizadas por meio de uma série de passos conscientes, as ações, que têm metas definidas e imediatas.

Ainda que interligadas, atividades e ações possuem uma relativa independência. Uma atividade pode ser realizada pela combinação de diferentes ações e, por outro lado, uma mesma ação pode pertencer a diferentes atividades.

No entanto, convém lembrar que o que energiza as ações não são suas metas, mas o motivo da atividade. Por esta razão, as ações não podem ser compreendidas fora da atividade a que pertencem (LEONTIEV, 1974, p.24).

A formação de metas não é direta. A Teoria da Atividade também defende a idéia de que a determinação das metas que devem ser alcançadas para se atingir um determinado objetivo não é algo instantâneo que pode ser feito diretamente pela vontade do sujeito. Muito pelo contrário: é um processo longo, no qual várias metas candidatas são testadas e descartadas por meio das ações do sujeito (LEONTIEV, 1974, p.25). É a experiência individual que dará condições para uma escolha mais ou menos adequada de metas a serem atingidas.

Este conceito traz uma série de implicações para a análise e planejamento de atividades. Para a análise, isso implica que os participantes da atividade devem ser observados sob uma perspectiva de desenvolvimento. Para o planejamento, que a nova atividade deixe espaço para que os participantes possam utilizar as ações que lhes forem mais adequadas para cada momento¹¹.

¹⁰ Como será explorado mais adiante, é interessante notar que, no Construcionismo, apesar da interação social ser incentivada e ser apresentada como uma das características mais importantes da atividade de aprendizado, pouco estudos foram feitos onde os aprendizes trabalham juntos, desenvolvendo um mesmo projeto. O próprio Logo não traz facilidades para intercâmbio de partes ou para a incorporação de projetos em grupo.

¹¹ É o que Papert diz a respeito do planejamento nas atividades em que o aprendiz está construindo algo. É preciso dar condições para que o aprendiz desenvolva suas próprias teorias, perceba suas próprias inconsistências e acabe por chegar aonde queria.

Ações são compostas por operações. As ações, por sua vez, podem ser constituídas por outras ações e estas por outras ações e assim por diante; cada nível dirigido a metas cada vez mais específicas. Em seu nível mais inferior, as ações são compostas por cadeias de operações não conscientes que adaptam a ação às condições específicas do contexto¹² (ver Figura 3).

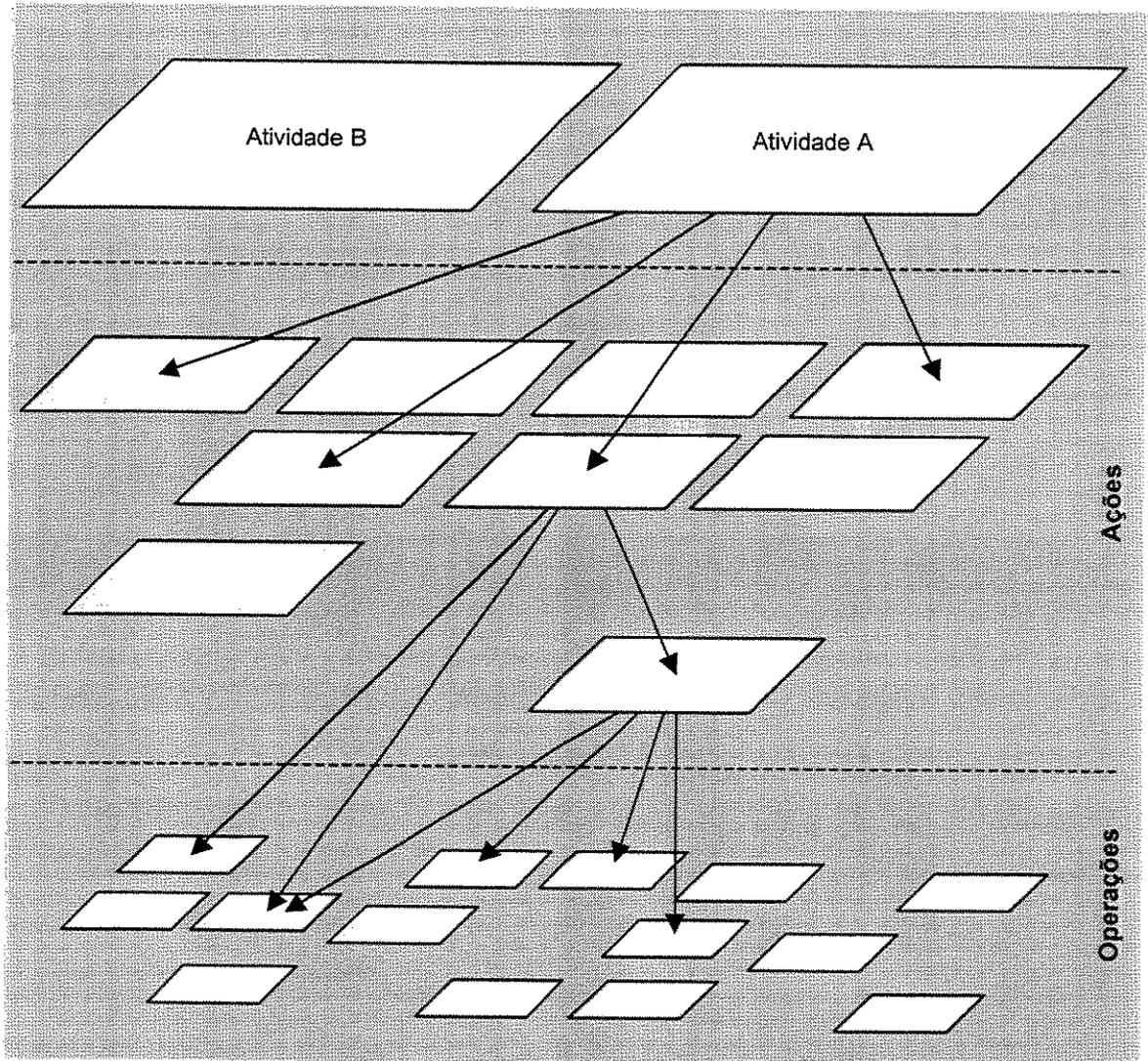


Figura 3 - A hierarquia atividades-ações-operações.

Quando uma pessoa escreve, por exemplo, raramente está consciente dos pequenos movimentos coordenados efetuados por suas mãos e braços. O mesmo

¹² Nota para educadores: o conceito de operações da Teoria da Atividade nada tem a ver com o conceito de operações no construtivismo piagetiano. Enquanto que, para a Teoria da Atividade, operações são atos físicos ou mentais, rotineiros e não conscientes, para o construtivismo elas estão associadas à abstrações mentais dos atos efetuados do sujeito. Diversos autores fazem comparações mais detalhadas. Werstch (1981) é um deles.

ocorre quando ela anda, dirige ou resolve problemas rotineiros. Simplificadamente, as operações são atos que, de tão rotineiros, deixam de ser conscientes para ela.

As operações também podem ser vistas como modos de se efetuar as ações. Dependendo das condições físicas ou socioculturais, uma ação será efetuada por um conjunto ou outro de operações. No caso da escrita, isso seria equivalente a usar uma caneta para escrever no papel ou um giz para escrever em uma lousa.

Vale notar que, por serem não conscientes e por dependerem das condições específicas de cada situação, é impossível descrever, *a priori*, que operações farão parte das ações que o sujeito planeja efetuar. Ainda assim, a partir de uma reflexão *a posteriori* e da observação de atividades reais torna-se possível fazer um levantamento das operações mais comuns utilizadas pelo sujeito no correr da atividade. Este aspecto da atividade é extremamente importante para o desenvolvimento de artefatos específicos, sejam eles computacionais ou não.

A Tabela 1 sumariza as principais idéias abordadas até agora:

Tabela 1 - Atividades, ações e operações.

Atividades	<ul style="list-style-type: none">• Dirigidas para objetos que satisfazem uma necessidade ou desejo (os motivos)• São compostas por ações
Ações	<ul style="list-style-type: none">• Dirigidas para metas conscientes• São compostas por outras ações ou operações• Só podem ser compreendidas no contexto da atividade a que pertencem
Operações	<ul style="list-style-type: none">• Disparadas por situações materiais bem específicas do ambiente• Só podem ser descritas depois de efetuadas

Outro conceito da Teoria da Atividade interessante para o desenvolvimento de novos artefatos é a visão da **tecnologia como uma extensão do homem**. Leontiev enfatiza que, geralmente, o destino das operações é tornar-se, mais cedo ou mais tarde, função das máquinas. Ele próprio cita o exemplo da transmissão automática dos carros, que pode ser vista como a automatização da mudança manual de marchas¹³.

E ele prossegue dizendo que, mesmo quando são executadas por máquinas, elas não devem ser consideradas como dissociadas do sujeito. As operações efetuadas

¹³ Como será discutido no capítulo 6, na nossa opinião, computadores são máquinas especiais que, além de automatizarem determinadas operações predefinidas, também permitem ao sujeito recombinar estas operações na criação de novas mais adequadas a sua necessidade.

pelas máquinas não interrompem a ação; são simplesmente uma opção para a sua execução (LEONTIEV, 1974, p.27).

De fato, por trás das transformações que o uso dos instrumentos acaba por infringir no sujeito, existe um mecanismo de formação de **órgãos funcionais**.

Segundo Kaptelinin, órgãos funcionais podem ser definidos como a “*combinação das habilidades naturais humanas com as capacidades de componentes externos – instrumentos – para se desenvolver uma nova função ou executar uma função já existente de modo mais eficiente*” (KAPTELININ, 1996a, p.109). O exemplo que ele fornece é o da combinação dos olhos com os óculos na formação de um órgão funcional que resulta em uma visão melhor. Outro exemplo seria a combinação mouse-mão, que forma um órgão funcional capaz de manipular os objetos disponíveis na tela do computador.

Operacionalização e conceitualização. Além do caráter estático que as une, atividades, ações e operações também têm uma relação dinâmica que trabalha por meio de um mecanismo de operacionalização e conceitualização (BODKER, 1991, p.27). É este mecanismo que delimita a fronteira entre os processos conscientes e os automáticos do indivíduo (ver Figura 4).

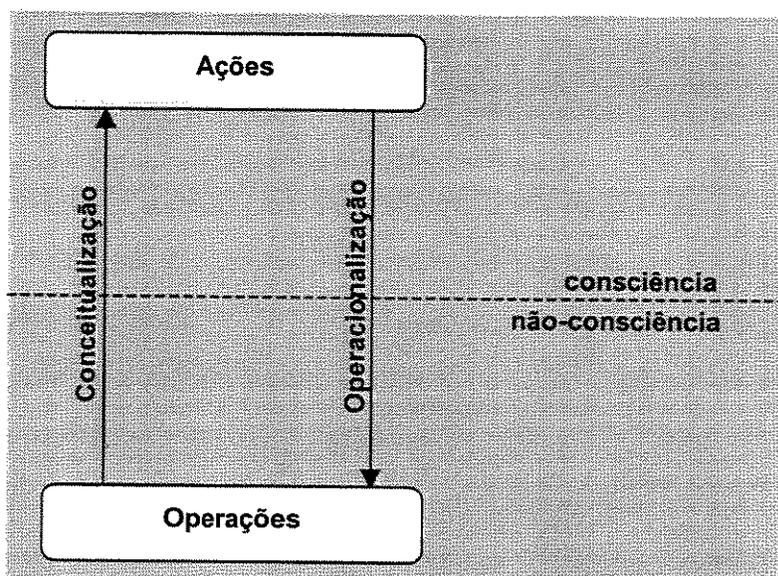


Figura 4 - Conceitualização e operacionalização

A **operacionalização** é a transformação de ações em operações. Inicialmente, cada ação possui um componente planejador, consciente, baseado em um modelo mental que o sujeito constrói da realidade e um componente executor, não consciente, composto pelas operações. Quando uma ação é praticada durante muito tempo em condições estáveis, o seu componente de planejamento desaparece e ela se transforma em uma operação que, embora não seja consciente, é muito mais fluente.

Por exemplo, assumindo-se o ato de dirigir um carro como uma atividade, para um motorista novato possíveis ações seriam apertar o freio, trocar de marchas, ligar o

pisca-pisca, etc.. Inicialmente, ele tem que prestar atenção em cada uma destas ações mas, à medida em que vai adquirindo prática, elas vão se tornando cada vez mais não-conscientes e mais fáceis de serem executadas em diferentes situações.

Com isso, as unidades de percepção do indivíduo vão se modificando. Ao mesmo tempo em que a ação rotineira se transforma em operação, uma nova ação é criada com um escopo mais amplo e tem a nova operação recém-formada como uma subparte (KUUTI, 1996, p.31).

Em outras palavras, à medida em que o sujeito vai se tornando fluente naquilo que está fazendo, o foco de sua atenção vai se ampliando e ele pode se concentrar mais no objetivo da atividade. No caso do carro, as ações básicas do motorista vão se operacionalizando e, ao invés de se preocupar com a mudança das marchas e a ordem dos pedais, o sujeito volta sua atenção para ações de mais alto nível, como fazer ultrapassagens, prestar atenção à sinalização, etc.¹⁴

A **conceitualização**, ou tomada de consciência, é o movimento oposto ao da operacionalização, isto é, é a transformação de operações em ações. Quando as condições físicas ou sociais mudam e impedem a execução de ações por meio das operações já existentes, a ação pode se particionar em uma seqüência de ações com metas intermediárias (LEONTIEV, 1974). Neste caso, o sujeito volta a tomar consciência das operações que estava executando na ação original.

Da mesma forma que as operações se formam a partir de uma rotina de ações sob condições constantes, elas voltam a tornar-se ações quando esta rotina se quebra. No caso do carro, isso aconteceria, por exemplo, se o motorista fosse dirigir um carro com um tipo de marcha completamente diferente, ou caso se, de repente, estivesse dirigindo e tivesse que fazer uma manobra imprevista.

Como será visto mais adiante, o Construcionismo trabalha com este mecanismo de operacionalização e conceitualização para tornar o aprendiz consciente do domínio que está sendo estudado. A própria linguagem Logo pode ser vista como um software de desenho que propositadamente leva à conscientização de noções de geometria.

As atividades estão em constante transformação em todos os níveis. Uma atividade pode perder o seu motivo e se transformar em uma ação de outra atividade. Esta seria a diferença, por exemplo, entre dirigir pelo prazer de dirigir e dirigir como uma forma de se chegar a algum lugar. O dirigir por prazer constitui uma atividade, ou seja, é motivante por si só. O dirigir para se chegar a algum lugar é uma ação. O motivo está em se chegar ao lugar, não em dirigir.

Por outro lado, uma ação pode tornar-se independentemente motivante e transformar-se em uma atividade por si só (adquirir o prazer de escrever ou de tocar um instrumento). A mesma ação também poderia, caso fosse repetida em condições estáveis, transformar-se em uma operação.

¹⁴ Esta parte sobre o desenvolvimento das competências do sujeito durante a atividade será aprofundada, para o caso do computador, no capítulo 4.

Essa dinâmica e flexibilidade dos componentes básicos da atividade são características bastante úteis para a descrição de processos de desenvolvimento, como o aprendizado. Elas podem ser vistas como uma importante contribuição da Teoria da Atividade com relação a abordagens de desenvolvimento de software como a da Ciência Cognitiva ou do Design Centrado no Usuário, que descrevem a atividade observada em uma estrutura puramente estática e rígida.

3.2.2 Orientação a objetos

A noção de objetos é central na Teoria da Atividade¹⁵. Tanto é que, segundo Kuuti (1996), as atividades podem ser vistas como uma transformação de objetos em outros objetos.

Objetos, objetivos e produtos. Na realidade, na nossa opinião, e partindo da própria definição apresentada acima, o termo “objeto”, tal como usado pela Teoria da Atividade, deve ser compreendido de diferentes maneiras que, dependendo do contexto, podem dar margem à interpretações errôneas.

A primeira interpretação, mais geral, é a de que um objeto pode ser entendido como qualquer coisa, material ou mental, que seja manipulada, acessada ou transformada durante a atividade: *“um objeto pode ser uma coisa material, mas ele também pode ser algo menos tangível (como um plano), ou totalmente intangível (como uma idéia comum), mas pelo menos ele tem que poder ser compartilhável para manipulação e transformação pelos participantes da atividade”* (KUUTI, 1996, p.27).

A segunda interpretação, mais difícil de se perceber nos textos, é a que entende o objeto da atividade como o objetivo desta atividade¹⁶ (NARDI, 1996a, p.73). Neste sentido, conforme descrito na seção anterior, as atividades são orientadas a objetos (produtos) que cristalizam algum desejo ou necessidade do sujeito. Sob este enfoque, são os objetos que motivam a atividade e dão sentido às ações efetuadas: *“uma atividade é uma ‘forma de fazer’ dirigida para um objeto e as atividades se distinguem umas das outras de acordo com os seus objetos. A transformação de um objeto em um produto motiva a existência de uma atividade”* (KUUTI, 1996, p.27).

Para efeitos deste trabalho, procurou-se utilizar o termo “objeto” para representar a forma mais genérica, tal como descrita na primeira interpretação acima. O termo

¹⁵ **Nota para os engenheiros de software:** o conceito de orientação a objetos da Teoria da Atividade (*object-orientedness*) não deve ser confundido com o conceito de orientação a objetos da Engenharia de Software (como em *object-oriented analysis, design e programming*). O primeiro está mais relacionado com toda a parte de significados que é dado aos objetos na atividade humana. O segundo, com uma metodologia de desenvolvimento e estruturação de software baseada em componentes que procuram representar a funcionalidade dos objetos conceituais e reais manipulados pelo software.

Na nossa opinião, há um paralelo conceitual entre estas duas concepções que merece ser melhor aprofundado. No entanto, tal empreitada ultrapassa o escopo do presente trabalho.

¹⁶ A utilização do termo “objeto” (*object*) como “objetivo” aparece inclusive em dicionários de psicologia, tal como “3. n. *A goal or an end state; here the term is really a shortened form of an objective.*” (REBER, 1985, p.483)

“objetivo” é utilizado para representar o objeto almejado na atividade, isto é, aquele que incorpora o motivo (necessidade ou desejo) responsável pela existência da atividade. Por fim, o termo “produto” é utilizado para representar o objeto que, de fato, foi construído ou gerado pela atividade. Em alguns casos, é possível que o produto alcançado seja diferente do objetivo inicial desejado, gerando um conflito a ser resolvido pela atividade.

Propriedades dos objetos. Diferente de outras abordagens como a piagetiana e a da ciência cognitiva, que só consideram as propriedades físicas, químicas e biológicas das coisas, a Teoria da Atividade leva em consideração, além destas propriedades, aquelas definidas socioculturalmente (KAPTELININ et al., 1999, p.29; KAPTELININ e NARDI, 1997, p.76; KAPTELININ, 1996a, p.107). Sob esta ótica, a propriedade de um objeto ser um livro – algo que tem um significado cultural – é tão objetiva quanto a de ele ter uma determinada cor, forma ou peso – características percebidas pelos órgãos dos sentidos.

Para a Teoria da Atividade, tanto as propriedades físicas quanto as culturais influenciam o modo como o sujeito interage com seu ambiente. Isso pode servir de base para explicar, por exemplo, por que preferimos usar a caneta que ganhamos de um amigo ao invés de uma outra, ou por que algumas crianças preferem desenhar no computador ao invés de no papel¹⁷.

Neste sentido, além da quantidade e variedade de objetos que estarão disponíveis e do tempo que é dado para interação – fatores determinados pela divisão do trabalho e pelas regras implícitas e explícitas na atividade – é muito importante se considerar, tanto na análise quanto na criação de atividades, quais os significados que cada um destes objetos traz para cada participante e como todas estas características somadas contribuem para o desenvolvimento da atividade.

3.2.3 Internalização e externalização

A Teoria da Atividade distingue entre atividades internas e externas. As atividades internas são aquelas que ocorrem na mente do indivíduo e são normalmente associadas à noção de processos intelectuais (por exemplo, todo o raciocínio que uma criança executa enquanto desenha). As atividades externas são aquelas que ocorrem sobre objetos materiais e correspondem ao que costuma-se entender por comportamento externo (os gestos que ela efetua desenhando).

Para a Teoria da Atividade, atividades externas e internas não podem ser estudadas isoladamente, uma vez que há uma série de transformações mútuas entre as duas. De um modo geral, estas transformações são reguladas por um mecanismo de internalização e de externalização.

¹⁷ De certa forma, o Construcionismo também leva em consideração as propriedades socioculturais dos objetos quando defende que alguns objetos são mais pessoalmente significativos do que outros e que isso influencia no aprendizado.

Internalização¹⁸. A internalização é o processo pelo qual atividades externas se transformam em atividades internas, isto é, passam a ser mentais (LEONTIEV, 1974, p.18; VIGOTSKI, 1994, p.74; KAPTELININ e NARDI, 1997, p.76).

De acordo com Vygotsky¹⁹ (KAPTELININ e NARDI, 1997, p.77; VIGOTSKI, 1994, p.75), as habilidades mentais do indivíduo aparecem primeiro na sua relação com outras pessoas (inter-subjetivo) e só depois é que são internalizadas (intra-subjetivo). Somente após praticar, socialmente, a ação é que o sujeito se torna capaz de imaginar, generalizar e levantar hipóteses sobre essa ação.

No entanto, a transformação das atividades externas em internas não é instantânea. Ela é resultado de uma longa série de eventos, inclusive biológicos, que o sujeito passa ao longo de sua vida. Além disso, as atividades internas não são meras imagens das externas. Na internalização são abstraídas as propriedades e as operações dos objetos e instrumentos envolvidos nas atividades externas, fazendo com que os mecanismos mais gerais se perpetuem mesmo em condições adversas. São estes mecanismos mais gerais que permitem que o sujeito possa transportar experiências de uma situação para outra como, por exemplo, desenhar com materiais diversos ou contar objetos de diferentes tipos.

Segundo Leontiev, quando são internalizados, os processos externos "*são generalizados, verbalizados, abreviados e, mais importante, tornam-se susceptíveis a um desenvolvimento posterior que excede as possibilidades da atividade externa*" (LEONTIEV, 1974, p.18).

Vygotsky exemplifica este processo na transformação do movimento de pegar no gesto de apontar. Inicialmente, a criança tenta apanhar um objeto que está fora de seu alcance e, por não conseguir, acaba por ficar com as mãozinhas balançando no ar. A situação muda completamente de figura quando a mãe vem em seu auxílio e percebe que ela está interessada no objeto. Com o tempo e a repetição de vários episódios similares, a criança começa a estabelecer relações entre estes seus gestos e as reações das pessoas e, ao invés de dirigir o movimento para o objeto, começa a dirigi-lo para elas, indicando o que deseja. Quando isso acontece, o

¹⁸ O conceito de internalização, ou da aquisição/desenvolvimento de conteúdos e procedimentos, varia de uma teoria cognitiva para outra e isto costuma levar a uma série de conflitos e discussões. Por exemplo, "*enquanto Piaget, especialmente em seu trabalho mais recente, se preocupou primeiramente em determinar como a criança abstrai e internaliza certos aspectos lógicos de sua relação com o mundo físico, Vygotsky estava mais preocupado em saber como a criança internaliza certos aspectos das atividades que são sociais e culturais por natureza*" (WERTSCH, 1981, p.31; VIGOTSKI, 1994).

Devido a estas diferenças de enfoque, é preciso tomar muito cuidado quando se utiliza de uma abordagem para se analisar uma outra. Não raro, termos parecidos são usados para descrever conceitos completamente diferentes, o que leva a uma série de enganos nas interpretações.

¹⁹ O nome de Vygotsky aparece escrito de formas diferentes conforme a referência bibliográfica. Neste trabalho, optou-se pela grafia "Vygotsky" (com dois "y"). Para evitar problemas na hora de fazer buscas nas bases de dados das bibliotecas, a sintaxe original foi mantida nas referências, como em "(VIGOTSKI, 1994)".

próprio movimento físico é simplificado e se reduz ao gesto de apontar que todos reconhecemos (VIGOTSKI, 1994, p.74).

É o mecanismo de internalização que permite que o sujeito teste situações hipotéticas sem ter de fato que manipular os objetos da realidade (KAPTELININ e NARDI, 1997, p.76).

Externalização. Indo em sentido oposto ao da internalização, a externalização é a transformação de atividades internas em externas. É ela que permite às pessoas envolvidas na atividade terem alguma percepção do que se passa internamente com os demais participantes – como no caso da mãe percebendo as necessidades de seu bebê no exemplo acima. Este tipo de percepção é fundamental em atividades em grupo de uma forma geral, como acontece também na relação de acompanhamento e orientação existente entre professores e alunos.

Além de facilitar a compreensão das atividades internas, a externalização social de ações acaba por provocar uma série de reações nas pessoas (o bebê percebeu que, ao esticar o braço, a mãe apanhava o objeto) que, por meio do mecanismo de internalização, acabarão por ser incorporadas aos indivíduos, alterando o seu comportamento. Deste modo, internalização e externalização se complementam e ajudam a perpetuar a relação mútua entre o desenvolvimento individual e coletivo (KAPTELININ e COLE, 1997).

Partindo-se da idéia de que as práticas sociais influenciam no desenvolvimento dos indivíduos, seria possível identificar que tipo de situações sociais seriam capazes de criar melhores condições para a educação de uma pessoa e/ou de um grupo. Atividades educacionais específicas deveriam enfatizar determinados tipos de práticas sociais e restringir outras, algo que pode ser facilitado pelo uso de artefatos adequados para este fim²⁰.

É muito importante, por exemplo, que seja incentivada a externalização dos conceitos que estiverem diretamente dirigidos ao objetivo comum e às sub-atividades dependentes entre si e isto de forma que não distancie o sujeito do que estiver fazendo. De acordo com Kaptelinin e Nardi (1997, p.76), a externalização se faz “naturalmente” necessária (i.e. intrinsecamente motivante) nas seguintes condições:

- Quando uma atividade interna precisa ser “consertada”, isto é, quando o indivíduo acha, por exemplo, que efetuou algum cálculo errado e precisa rever o processo com mais atenção.
- Quando não é possível efetuar a atividade mentalmente. Isso pode ocorrer quando o cálculo é grande ou complicado demais para ser feito só de cabeça.
- Quando é necessária coordenação da atividades entre pessoas.

²⁰ Como será visto mais adiante, ainda que não tenha sua origem na Teoria da Atividade, a atividade educacional construcionista com a linguagem Logo de programação incentiva determinados tipos de ação que levam os aprendizes a se concentrarem em determinados conceitos e a raciocinarem de uma determinada maneira.

3.2.4 História e desenvolvimento

Como pôde-se perceber pelos mecanismos de internalização e externalização, operacionalização e conceitualização, etc., as atividades não são entidades estáticas; são sistemas dinâmicos e em constante transformação. O sujeito trabalha com artefatos, desenvolvendo novos objetos e artefatos. Neste processo, ele transforma a si mesmo e o ambiente que o cerca. Esta transformação acaba por gerar novas necessidades ou desejos, fazendo com que a atividade fique em evolução constante.

3.2.4.1 A análise histórica da atividade

Como cada atividade tem uma história individual, resultante da combinação de eventos de fontes variadas e, em boa parte das vezes, imprevisíveis, é preciso que sua análise seja feita em situações tão próximas do real quanto possível – para não deformar a atividade – e em cima de um contexto de desenvolvimento.

É preciso compreender a origem da atividade, que fatores influenciaram cada um de seus elementos (pessoas, práticas comunitárias, instrumentos, etc.) e que fatores continuam sendo relevantes. Só assim torna-se possível entender a situação atual (KUUTI, 1996, p.26).

No caso das atividades educacionais, é preciso determinar, por exemplo, até que ponto os seus componentes atuais estão de acordo com os princípios educacionais originais e até que ponto eles foram influenciados por outros fatores históricos que, por vezes, nem possuem mais razão de ser.

Em se tratando do desenvolvimento de tecnologia para a educação, embora, por uma questão histórica e econômica, a maior parte do equipamento usado seja aquele consagrado pelo setor profissional, é muito importante que se avalie freqüentemente as características específicas da educação e se procure criar equipamentos mais apropriados.

3.2.4.2 A análise do desenvolvimento da atividade

Além desta visão histórica do desenvolvimento da atividade como um todo, a Teoria da Atividade também permite focar no desenvolvimento que ocorre no nível do indivíduo e de cada componente ao longo da atividade. Este é um ponto que, segundo Kaptelinin (1996b, p.55), as abordagens tradicionais (como a cognitiva) não parecem trabalhar de forma tão adequada.

Aprendizado. Dependendo do nível de competência inicial do sujeito e de sua evolução, atividades similares poderão se desenvolver de formas completamente diferentes. Por isso, aprendizado humano deve ocupar uma posição de destaque na análise e na criação de atividades, sejam elas educacionais ou não.

Bodker apresenta uma série de pontos sobre o aprendizado levantados pela Teoria da Atividade que deveriam ser considerados no desenvolvimento de atividades e artefatos (BODKER, 1991, p.32):

a) O aprendizado com objetos físicos é diferente do aprendizado com representações. Segundo ela, atividades dirigidas a objetos materiais, como dirigir um carro ou construir uma mesa, não podem ser aprendidas sem experiência

prática. Por mais que se visualize, se explique ou se leia sobre estes objetos, há uma série de fatores sensoriais ou não conscientes pertinentes às operações desta atividade que não podem ser descritos.

Mesmo as atividades que tenham metas abstratas, como a resolução de problemas matemáticos ou aprendizado de uma prática social, são mais facilmente aprendidas e encaminhadas em conexão com objetos físicos do que com representações dos mesmos. Nestas atividades, o aprendizado com representações é mais fácil do que o aprendizado feito em conexão com a linguagem (escrita ou falada) que, por sua vez, é mais fácil do que nas atividades totalmente baseadas na reflexão mental. Por exemplo, a adição é primeiro efetuada pelas crianças por meio da contagem de objetos físicos, então elas passam a dominar esta operação baseada em figuras, daí para um estado em que a adição funciona melhor se elas puderem falar e assim por diante.

Na nossa opinião, estes pontos levantados por Bodker levam a um questionamento da eficiência do computador – um artefato essencialmente manipulador de representações – no aprendizado em comparação ao uso de materiais manipulativos mais concretos. No nível das experiências sensório-motoras, as atuais interfaces computacionais ainda estão muito longe de poderem ser comparadas com a realidade que representam. No entanto, para os casos em que as situações reais não sejam viáveis (devido ao seu tamanho, custo ou segurança), os modelos computacionais podem mostrar diversas vantagens com relação aos materiais tradicionais normalmente usados.

b) A qualidade da interação do sujeito com os artefatos e objetos se transforma ao longo da atividade. A variedade de situações abrangidas pelas operações do sujeito se amplia e suas ações tendem a tornar-se cada vez mais genéricas. Além disso, a quantidade de ações dirigidas ao artefato tende a diminuir, tornando o artefato gradualmente mais transparente para o sujeito.

Como será discutido no capítulo 4, idealmente, o tipo de suporte oferecido pelo artefato deveria se adaptar a estas mudanças de enfoque que ocorrem durante a atividade. Quando isso não for possível, novos artefatos deveriam ser disponibilizados para atender a nova demanda.

Em geral, no caso da informática, o que tem sido feito é desenvolver software que possa ser configurável pelos usuários para atender as suas necessidades específicas, como é o caso, por exemplo, do Design Centrado no Aprendiz. No entanto, alguns estudos mostram que as facilidades de configuração raramente são utilizadas e, quando são, isso se dá por meio do auxílio de pessoas mais experientes, algo que exige toda uma reestruturação do ambiente de trabalho (NARDI, 1993).

c) O tipo de educação influencia a atividade. Bodker também indica que o tipo de educação oferecida (treinamentos, orientações) influencia diretamente na maior ou menor facilidade com que o artefato será operacionalizado e, nos casos de conceitualização (devido a algum conflito cognitivo ou questionamento pedagógico), na maior ou menor facilidade com que o sujeito retornará a assumir o controle da

atividade. Sob este aspecto, na nossa opinião, as sessões educacionais, as apostilas e os materiais explicativos sobre o artefato também devem ser considerados como parte integrante do seu desenvolvimento.

3.2.5 Mediação

Segundo a Teoria da Atividade, a grande maioria das interações do sujeito com o ambiente (pessoas e objetos) é mediada por artefatos. Artefatos podem ser materiais ou não, indo desde martelos, livros, computadores e chaves-de-fenda até sistemas de valores e estratégias de resolução de problemas.

Essa mediação não apenas permite a melhoria das operações já existentes como também permite que os seres humanos, com o auxílio de estímulos extrínsecos, possam controlar o seu próprio comportamento (VIGOTSKI, 1994, p.54; ENGESTRÖM, 1987, p.59), que é o que acontece, por exemplo, quando utilizamos agendas ou nos orientamos por meio de placas nas ruas.

De um modo simplificado, Vygotsky representava a mediação usando o esquema da Figura 5:

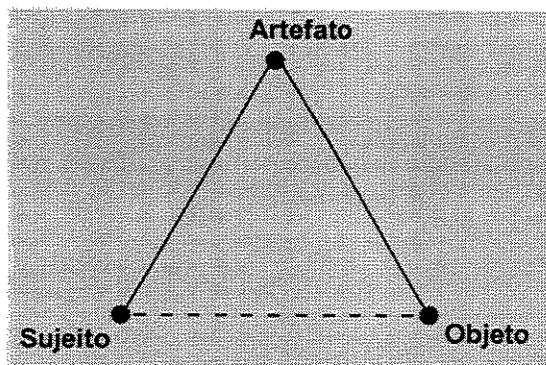


Figura 5 - A mediação segundo Vygotsky.

De acordo com este esquema, os sujeitos (S) atuam sobre os objetos do ambiente (O) de duas maneiras: uma “natural” ou direta, que é representada pelo segmento S-O; e outra “cultural”, ou mediada pelo artefato (A), representada pela ligação S-A-O. Embora alguns autores passem a impressão de que, com o desenvolvimento, o caminho “cultural” acabe por substituir completamente o “natural”, a Teoria da Atividade defende a idéia de que ambos os caminhos existem simultaneamente. Os seres humanos não deixam de ter suas relações naturais com o mundo pelo fato de criarem, transmitirem e adquirirem cultura (COLE e ENGESTRÖM, 1993, p.5).

Algumas considerações devem ser feitas com relação ao caráter mediador dos artefatos na atividade humana:

a) Os artefatos só podem ser compreendidos no contexto da atividade (LEONTIEV, 1974, p.28; KAPTELININ, 1996b, p.46), isto é, identificando-se as formas com que eles são utilizados, as necessidades que eles atendem e a história de seu desenvolvimento. Fora de seu contexto de uso, os artefatos tornam-se entidades abstratas sobre as quais poucas conclusões podem ser tiradas. Para quê, por exemplo, serve um computador sem eletricidade, ou uma chave-de-fenda em

algum lugar que não tenha parafusos? Em ambos os casos estes artefatos deixam de ter sua utilidade específica e passam a ter o mesmo valor que pedaços de plástico e papel.

O presente trabalho centra-se principalmente nesta relação artefato-atividade quando se refere ao desenvolvimento de tecnologia para a educação. O software só poderá ser considerado mais ou menos educacional na medida em que der maior ou menor suporte à atividade educacional a que se destina. Conforme discutido no capítulo 1, sem a especificação desta atividade, não existe o conceito de software para a educação.

b) Os artefatos não são neutros. Um dos principais pontos levantados pelo conceito de mediação é que o uso dos artefatos acaba por transformar não apenas o objeto da ação, mas também o seu sujeito²¹, algo que merece ser cuidadosamente considerado no uso de tecnologias na educação.

Em primeiro lugar, os artefatos determinam como o sujeito percebe e interage com o seu ambiente. *"O artefato [tool] é, ao mesmo tempo, capacitador e limitante: ele potencializa o sujeito no processo de transformação com a experiência histórica e as habilidades 'cristalizadas' em si, mas ele também restringe a interação para somente aquela específica da ferramenta [tool] ou do instrumento [instrument]; as outras características potenciais do objeto permanecem 'invisíveis' ao sujeito"* (KUUTI, 1996, p.27).

Em ambos os casos, por meio do mecanismo de internalização, estas características dos artefatos acabam influenciando tanto a forma com que o sujeito age, como a forma com que ele pensa sobre o mundo. O resultado final acaba por ser uma mistura entre o estado anterior do sujeito e as novas propriedades trazidas pelo artefato (KAPTELININ, 1996b, p.53).

Em segundo lugar, além de influenciarem diretamente as pessoas que os utilizam, os artefatos também contribuem para a propagação de valores culturais. *"Os artefatos [tools] costumam refletir a experiência de outras pessoas que tentaram resolver problemas similares anteriormente e criaram/modificaram o artefato para torná-lo mais eficiente. Esta experiência é acumulada nas propriedades estruturais do artefato (material, forma, ...) e também no conhecimento sobre como ele deveria ser utilizado. Os artefatos são criados e transformados durante a própria atividade e carregam consigo uma determinada cultura – restos históricos de seu desenvolvimento"* (KAPTELININ e NARDI, 1997, p.77)²².

²¹ Este conceito é bastante explorado por Sherry Turkle em seu livro *The Second Self*, publicado em 1984 pela editora Simon and Schuster, Nova Iorque, EUA. Nele, partindo da idéia de que artefatos como o telescópio e o trem mudaram toda nossa concepção de universo e distância, a autora se propõe a analisar que tipo de influências o computador traz para o modo de ser das pessoas.

²² Na Teoria da Atividade existe um grave problema no que se refere à tradução de termos. Às vezes, uma mesma palavra recebe traduções que, embora em linguagem coloquial tenham conotações semelhantes, trazem significados diferentes para os especialistas da área. Por exemplo, o que

Embora hoje em dia exista uma certa predisposição contra a influência da tecnologia no lado humano das pessoas (isto é, que os joguinhos eletrônicos farão com que as crianças deixem de brincar com os amiguinhos, que a informatização da indústria irá transformar todos os funcionários em robôs, etc.), convém notar que esta influência também pode ser usada em proveito do desenvolvimento humano.

Os artefatos nunca devem ser tratados como elementos estáticos, sobre os quais o sujeito não tem nenhum poder. A própria definição de atividade implica na transformação do meio pelo sujeito e isto inclui a própria transformação dos artefatos (que, dentre outras coisas, também são objetos). Sob este aspecto, a Teoria da Atividade tem uma abordagem mais humana e otimista do que a das teorias que buscam explicar o desenvolvimento humano como uma função única de mecanismos biológicos ou do meio (KUUTI, 1996, p. 26).

Como será visto mais adiante, no caso do Construcionismo, é possível desenvolver artefatos que explicitem determinados aspectos educacionais dos objetos ou que incentivem determinadas práticas sociais.

Artefatos técnicos e artefatos psicológicos. No que se refere à mediação, Vygotsky distinguia entre dois tipos inter-relacionados de artefatos: os técnicos (*technical tools*) e os psicológicos (*psychological tools*). Os primeiros servem para transformar objetos, como é o caso dos martelos, chaves-de-fenda, máquinas, editores de texto, etc.

Os demais, que incluem a noção de signos, servem principalmente para visualizar, comunicar e representar conceitos. Os artefatos psicológicos “são dirigidos ao domínio ou controle dos processos comportamentais – de outra pessoa ou de si próprio – tal como os técnicos podem ser usados para controlar os processos da natureza” (VYGOTSKY, 1981, p. 137). Como exemplos de artefatos psicológicos pode-se citar: escrita, sistemas algébricos, mapas, diagramas, trabalhos de arte, todos os tipos de signos convencionais, estratégias para resolução de problemas, etc.

Vygotsky (1994, p.71) chama de “artefato” (*artifact*, em inglês), Engeström (1987, p.59) chama de *instrument* e Kaptelinin (1996a, p.109) como *tool*. Engeström parece intercambiar *instrument* e *artifact* e associa *tool* a algo mais específico.

No Brasil, o termo mais usual para *tool* é instrumento. No entanto, o termo “instrumento” também traz para a comunidade que estuda Vygotsky a conotação de um tipo de artefato que é usado na transformação de objetos – que é o que Engeström (1987) chama de “ferramenta técnica” (*technical tool*). Para completar, ainda não parece haver uma padronização real dos termos e mesmo alguns autores da Teoria da Atividade acabam usando estes nomes de forma indiscriminada.

Apesar disto, pelo menos um ponto parece haver em comum: o termo “artefato” (*artifact*) é o mais amplo e genérico de todos. Por isso, para efeito deste trabalho, optou-se por usá-lo em detrimento de outros termos como “ferramenta” ou “instrumento”. De qualquer forma, para minimizar as dúvidas, procurou-se colocar o termo original, entre parênteses, ao lado da tradução.

Engeström (1987, p.60) aponta que “a essência dos artefatos psicológicos é que eles são originalmente artefatos para dar forma e controlar cooperativamente, comunicativamente e conscientemente os procedimentos de uso e construção de artefatos técnicos”. Em outras palavras, enquanto os artefatos técnicos são usados na transformação da realidade, os psicológicos são usados para uma melhor compreensão, reflexão e comunicação destas transformações e seus impactos. Em se pensando em informática, pode-se adicionar ao rol dos exemplos de artefatos psicológicos os simuladores (que facilitam a visualização de processos), os sistemas de correio eletrônico e redes em geral (que facilitam a comunicação), as ferramentas de consulta a bases de dados (que facilitam a visualização dos dados) e outros.

Outras classificações dos artefatos. Artefatos também podem ser classificados em coletivos ou individuais, pelo nível de competência necessário para operá-los, etc.

Bodker divide os artefatos entre aqueles que existem como objetos independentemente das ações que o sujeito estiver executando (como a linguagem escrita, um martelo, ou uma televisão, ...) e aqueles que só existem durante estas ações ou operações (como a linguagem falada ou técnicas de se fazer as coisas).

Ela também divide os artefatos da primeira categoria em 2: os artefatos externalizados passivos (*passive externalized artifacts*), que facilitam na execução de determinadas operações e os artefatos externalizados ativos (*active externalized artifacts*) que, além de facilitarem, também substituem (ou automatizam) algumas operações (BODKER, 1991, p.29). No caso da reprodução de imagens, pode-se considerar um lápis como artefato externalizado passivo (o sujeito desenha com ele) e uma impressora como um artefato externalizado ativo (ela desenha para o sujeito).

Computadores são artefatos externalizados que, dependendo do software, podem ser ativos ou passivos conforme a situação. O interessante dos artefatos ativos é que eles liberam o sujeito para desenvolver mais operações em paralelo, sem ter que ficar preocupando-se com tudo. Como será discutido a partir do capítulo 6, a passividade e a atividade de um software podem ser usadas para ajudar cada sujeito da atividade a se concentrar mais naquilo que lhe for pertinente. Por exemplo, o software pode gerar relatórios automaticamente para professores sem que os alunos percebam. Pode também atuar como um professor dando aula, no caso de sistemas tutores, pode efetuar cálculos complexos que no momento não devem chamar a atenção do aprendiz, etc.

3.2.5.1 Mediação social

Engeström atenta para o fato de que, na Teoria da Atividade, muitos autores enfatizam demais a questão da mediação na relação sujeito-objeto e acabam por deixar de lado os aspectos social e comunicativo (relação sujeito-sujeito). Ainda assim, ele mostra que, no trabalho original de Leontiev, esses pontos eram considerados como parte fundamental da atividade. “O indivíduo, a criança, não é simplesmente jogada dentro do mundo humano; ela é apresentada a este mundo

pelas pessoas a sua volta e são elas que a guiam” (LEONTYEV, 1981, p. 135 sendo citado em ENGeström, 1987, p. 69)²³.

Porém, segundo Engeström (1987, p.70), Leontiev acabou por não fazer um modelo unificado que integrasse também estes aspectos comunicativo e social da atividade e acabasse por suceder ao modelo gráfico proposto por Vygotsky (ver Figura 5).

Visando resolver esta questão, Engeström propõe um diagrama que estende o de Vygotsky (apresentado na Figura 6). Neste diagrama, explicitando a diferença entre pessoas e objetos, além da relação sujeito-objeto, foi traçada a relação sujeito-comunidade. Todas estas relações são mediadas por artefatos dos mais variados tipos. No entanto, Engeström dá uma ênfase especial às regras e à divisão do trabalho, que regulam a interação entre as pessoas e o espaço que cabe a cada um.

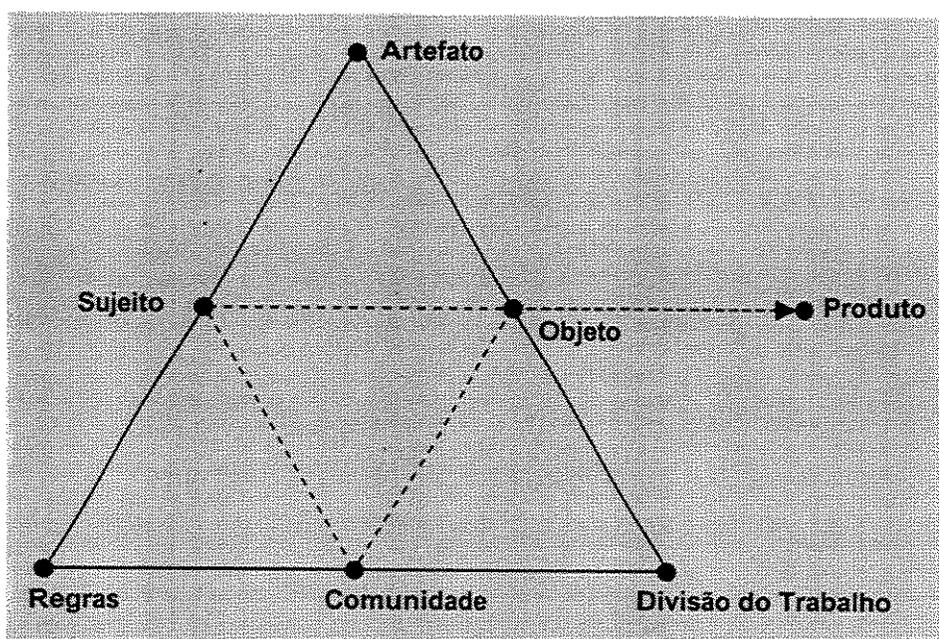


Figura 6 - O diagrama de Engeström (COLE e ENGeström, 1993, p.8).

O próprio Engeström explica o diagrama: “Primeiro, o fato de que os indivíduos (‘sujeito’) estão constituídos em comunidades está marcado pelo ponto chamado ‘comunidade’. Como indicado na figura, as relações entre sujeito e comunidade são mediadas, de um lado, pela coleção de todos os ‘artefatos mediadores’ do grupo e,

²³ Nota para educadores: Engeström (1987, p.62) também discute que o conceito de signos foi super enfatizado por Vygotsky e que a noção mais geral de artefatos psicológicos (*psychological tools*) e sua relação com os artefatos técnicos foi deixada praticamente de lado. Engeström aponta ainda para o fato de que o movimento da Teoria da Atividade pós-Vygotsky também tentou ignorar os artefatos psicológicos e deixar a questão dos signos de lado, uma vez que ela levantava muita polêmica. Hoje, apesar de terem dado uma reanimada na questão dos signos, pouco tem-se falado dos artefatos psicológicos.

de outro, pelas 'regras' (as normas e as sanções que especificam e regulam os procedimentos corretos esperados e as interações aceitáveis entre os participantes). As comunidades, por sua vez, implicam em uma 'divisão do trabalho', a continuamente negociada distribuição de tarefas, poderes e responsabilidades entre os participantes do sistema da atividade." (COLE e ENGSTRÖM, 1993, p.9)

Em se pensando no caráter coletivo das atividades, a divisão do trabalho traz uma série de implicações, acabando por gerar uma distância entre as ações dos indivíduos e o motivo coletivo da atividade (LEONTIEV, 1974, p.23). A simples análise de um papel nem sempre é suficiente para se determinar a razão de ser de uma atividade. Nas linhas de montagem isso é bem evidente. Cada operário tem a sua função (alguns têm mais do que uma e algumas funções podem ter mais do que um único operário) e todos contribuem para o produto final. Nas salas de aula, algo semelhante também acontece. Nas escolas tradicionais tem-se, por exemplo, professores que assumem papéis de difusores de informação, de motivadores e de avaliadores e alunos que assumem o papel de receptores de informação.

Convém notar que, tanto no caso da linha de montagem quanto da sala de aula, torna-se impossível tirar conclusões a respeito de algum dos papéis sem se considerar os demais papéis e o restante do ambiente. Como seria possível, por exemplo, analisar a atividade de um operário mecânico sem levar em consideração a linha de montagem na qual ele está inserido?

De certa forma, na nossa opinião, os papéis podem ser vistos como descrições de sub-atividades, cada uma com seus próprios objetivos, mas que só poderiam ser totalmente compreendidas dentro da atividade maior da qual são componentes. Sob esta ótica, a atividade principal seria formada por sub-atividades específicas integradas por um conjunto de regras que, dentre outras coisas, seria responsável pelo estabelecimento de prioridades e sincronização.

O diagrama de Engeström, genérico, levantou diversos pontos que, até então, não podiam ser visualizados no diagrama de Vygotsky (Figura 5). Por isso, tem sido usado como base em uma série de trabalhos, inclusive relacionados com o desenvolvimento de software (NARDI, 1996b). Tal como pode ser visto na Figura 7, Bellamy (1996, p.126), por exemplo, descreve como seria uma aplicação deste diagrama para o caso de uma escola:

A questão da mediação social está tornando-se especialmente importante nos dias de hoje. A integração da informática com as telecomunicações tem possibilitado uma série de transformações nos meios de produção, principalmente no que se refere à formação de equipes remotas de trabalhadores e à própria distribuição geográfica das empresas. O mesmo também está ocorrendo no setor educacional – a rápida proliferação de cursos à distância talvez seja o maior reflexo disto.

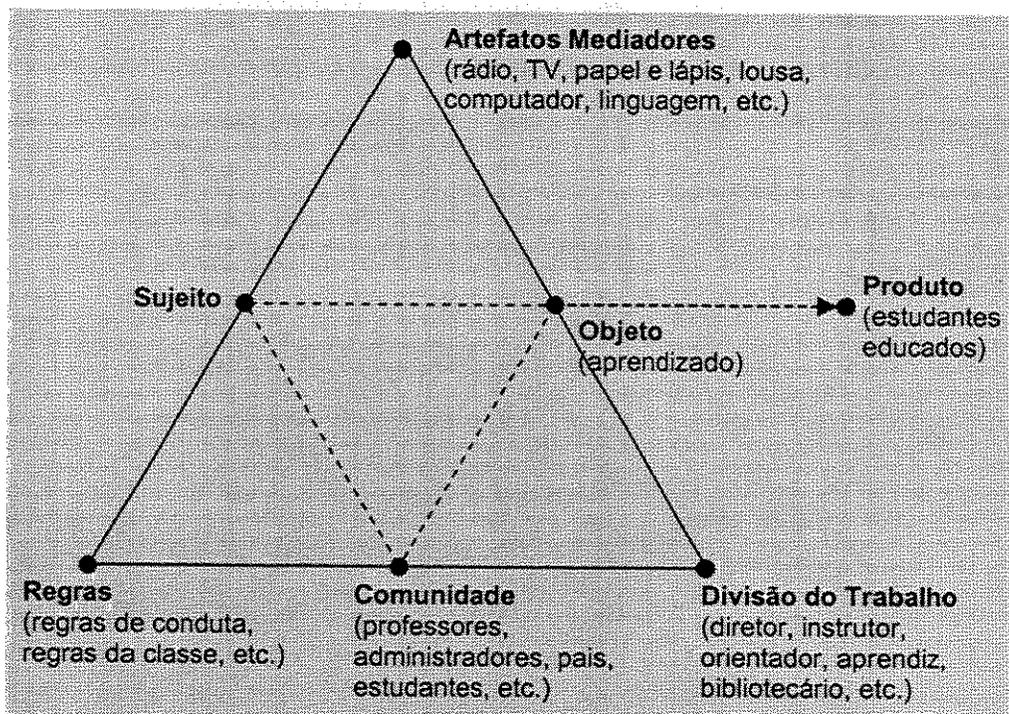


Figura 7 - O diagrama de Engeström para a educação fundamental e média (BELLAMY, 1996, p.126)

Como consequência, a problemática sociocultural tem se tornado cada vez mais central e tem levantado questões que, praticamente, não apareciam anteriormente no desenvolvimento de novas tecnologias. Que fatores são importantes para manter uma equipe unida? Como motivar ou acompanhar o desenvolvimento de um indivíduo sem ter contato direto com ele?

Neste sentido, ainda que não traga respostas prontas, a integração dos aspectos instrumentais e comunicacionais da Teoria da Atividade parece ser bastante interessante e diversos pesquisadores a têm utilizado como base em suas pesquisas.

4 A Teoria da Atividade e o computador

Como mencionado no capítulo anterior, a aplicação da Teoria da Atividade no desenvolvimento de software é bastante recente. Ainda assim, apesar das poucas referências bibliográficas que se encontram disponíveis, é uma área que vem recebendo grande aceitação por parte dos pesquisadores e analistas de sistemas, principalmente daqueles que tratam dos problemas relacionados à interação humano-computador.

Diferente do que ocorre na Ciência Cognitiva (ver capítulo 1), sob o enfoque da Teoria da Atividade, o computador não é analisado a partir das similaridades que possui com os seres humanos. Ele é visto tanto como um objeto quanto como um artefato mediador da atividade humana.

Com base nisso, para efeitos deste trabalho, optou-se principalmente pela linha proposta por Bodker (1989, 1991), que defende um desenvolvimento de software a partir do estudo de duas atividades intimamente relacionadas: a chamada atividade mediada pelo artefato computacional e a atividade de desenvolvimento propriamente dita, que tem a primeira como objetivo.

Este capítulo descreve ambas as atividades, procurando destacar os principais aspectos relacionados com o desenvolvimento e uso de software para a educação. Tal como acontece com os conceitos levantados no capítulo anterior, estes aspectos servirão de base para a abordagem de desenvolvimento de software para a educação proposta no capítulo 6.

4.1 A atividade mediada pelo artefato computacional

De uma forma geral, Bodker explica a relação de uso do computador como uma atividade na qual o sujeito executa pelo menos parte de suas ações por meio de um artefato computacional. Nestas atividades, o sujeito também pode interagir e contar com o auxílio de pessoas e outros artefatos. O artefato computacional (*computer-based artifact*), no caso, seria aquela combinação hardware-software que normalmente entende-se por "computador".

É interessante notar que a simples consideração do computador como um mediador já traz uma série de implicações importantes para o desenvolvimento de software. Uma delas é a de que, dada sua condição de artefato, espera-se que ele não seja o objeto das ações do sujeito, mas o seu meio. Idealmente, o sujeito não deve tomar consciência de estar manipulando um computador, a não ser que isso seja desejado pelo designer do artefato.

Outra implicação direta do conceito de mediação é a de que, ao invés de considerar, exclusivamente, a interação humano-computador, a análise do software deveria

incorporar também a interação entre o órgão funcional humano-computador com o mundo à sua volta. Em outras palavras, o designer do software passa a questionar, além do tradicional “o que o usuário poderá fazer no software”, “o que a combinação usuário mais software poderá fazer no ambiente que os cerca”.

Esta idéia, aparentemente simples, provoca um impacto tão profundo, que alguns pesquisadores sugerem que a própria noção de interação humano-computador, que dá nome a esta área de pesquisa, deveria ser modificada para “atividade mediada pelo computador” (BODKER, 1991; KAPTELININ, 1996b).

Características específicas do artefato computacional. Tentando ir além destas considerações genéricas, diversos autores procuram explicar que características distinguem um artefato computacional de outros tipos de artefatos. A identificação destas características é muito importante para que se possa analisar o artefato computacional frente a outros que também poderiam ser utilizados em uma mesma atividade (ver discussão sobre eficiência no capítulo 2).

Ao que parece, como será apresentado a seguir, cada autor descreve o computador à sua maneira. Embora estas definições sejam, por vezes, bastante semelhantes, falta ainda um consenso sobre o que é o computador para a Teoria da Atividade. Visando minimizar esta questão, no capítulo 6 é proposta uma definição que oferece subsídios para os principais conceitos aqui apresentados²⁴.

Kuuti (1996, p.34), por exemplo, exalta a qualidade que o computador possui de dar suporte e penetrar em todos os níveis da atividade humana:

- No nível das operações, seu maior uso é na automação de cálculos e da manipulação de dados administrativos. É esta característica que levou ao grande desenvolvimento da informática.
- No nível das ações, a informática pode servir como um instrumento para a transformação e manipulação de objetos, isto é, como um artefato técnico. Exemplos disso são os editores de texto e outras ferramentas de manipulação de símbolos (planilhas, etc.). Também pode ser utilizada como um artefato psicológico, ajudando nas ações voltadas à compreensão de coisas, como na geração de relatórios, visualização de processos, etc. e nas ações que visam à comunicação entre os participantes da atividade.
- No nível das atividades, a informática pode fazer com que a atividade se torne possível e viável (interligando participantes, acelerando as operações, ...). Em certos casos, a informática também pode criar objetos que seriam impossíveis de se obter em outras situações, como é o caso do grande acervo de páginas interligadas na WWW, modelos fictícios, etc.

²⁴ O Construcionismo tem uma visão do computador mais dirigida para a educação que, na nossa opinião, complementa as visões defendidas pela Teoria da Atividade. Esta visão será melhor discutida nos capítulos 5, 7 e, principalmente, 8 do presente trabalho.

Por outro lado, segundo Tikhomirov (1981, p.276), mais do que simplesmente automatizar parte das operações humanas ou suplementá-las, o computador deve ser visto como um artefato que altera, qualitativamente, a atividade humana. O computador propicia novas formas de organização da memória, de relação entre as pessoas e de aquisição de conhecimento, abrindo uma série de novos meios para a atividade mental e ajudando o ser humano a se concentrar mais no lado intuitivo do pensamento.

Já para Kaptelinin (1996b), um dos aspectos mais importantes dos computadores é a grande quantidade e variedade de órgãos funcionais que eles podem gerar. De fato, na nossa opinião, como será mais profundamente discutido no capítulo 5, um computador é como um canivete cheio de utilidades. Ele pode tanto ser visto como um artefato único (um editor de textos) ou como uma série de artefatos (as funções de formatação de parágrafos, de imprimir, etc. do editor) que trabalham de forma integrada, estendendo (e focando) nossa capacidade de agir, compreender e comunicar²⁵.

Kaptelinin ressalta que uma das mais importantes funções do computador é servir como extensão do IPA (Plano Interno de Ações, *Internal Plane of Actions*), um conceito da Teoria da Atividade que se refere à capacidade que as pessoas têm que permite efetuar manipulações de representações mentais do mundo antes de ter que efetua-las nos objetos reais²⁶.

Os computadores não são os únicos artefatos que podem ser usados como uma extensão do IPA e também esta não é a única função deles (podem ser usados para comunicação, ...). No entanto, para Kaptelinin (1996b, p.53), é a sua facilidade para efetuar cálculos, criar e manipular modelos dos objetos-alvo que os diferencia e os torna tão atraentes. Ainda assim, se estas características fossem mais valorizadas, o software existente poderia ser aprimorado pela adição de facilidades para comparação de objetos, mecanismos que permitissem a visualização e retorno dos caminhos de busca percorridos, suporte na coordenação de tarefas, etc.

Por fim, para Bodker (1991, p.35 e 36), o que chama a atenção nos artefatos computacionais não é nem a possibilidade de suportar a atividade como um todo, de alterá-la qualitativamente ou de gerar inúmeros órgãos funcionais. Para ela, o

²⁵ É interessante notar que, embora não apareça associada à Teoria da Atividade, há toda uma corrente de pesquisa que vê a combinação humano-tecnologia como um potencializador de situações educacionais. Segundo ela, "*a música não está no piano, mas sim no pianista, que se expressa através de seu instrumento*". O computador pode ser o maior "piano" já inventado, mas seu sucesso na educação dependerá muito da abordagem com que for trabalhado e de uma série de fatores a serem considerados (KAY, 1991).

²⁶ O IPA é algo semelhante ao que na Ciência Cognitiva seria descrito como a habilidade de se lidar com modelos mentais e a memória de trabalho. Infelizmente, apesar de sua importância e suas possíveis relações com simulações feitas por computador, durante a execução deste trabalho não foi possível encontrar mais referências relacionadas com o IPA e optou-se por limitar a sua explicação a este capítulo.

verdadeiro diferencial dos artefatos computacionais reside na amplitude e variedade de objetos e sujeitos que podem ser acessados e manipulados por meio de um mesmo artefato.

Por exemplo, um mesmo aplicativo pode acessar objetos e sujeitos que não estejam presentes no alcance físico ou temporal do sujeito, pode manipular objetos que não existam fora do artefato (como personagens de joguinhos, mensagens de correio eletrônico e outros), etc.. A autora chega inclusive a identificar os diferentes tipos de situações que podem ser encontradas na relação do sujeito com objetos e outros sujeitos mediadas pelo artefato computacional (BODKER, 1991, p.38 e 180):

- O objeto está presente apenas no artefato (não existe na realidade), como é o caso das células das planilhas eletrônicas, efeitos especiais e personagens de joguinhos.
- O objeto também existe como um objeto físico, mas só está presente na atividade como uma representação no artefato computacional (por exemplo, uma carta sendo escrita em um editor de textos).
- O objeto está presente fisicamente na atividade e também pode ser acessado via uma representação no artefato computacional (por exemplo, painéis que controlam robôs).
- O outro sujeito não está presente fisicamente na atividade, como ocorre nos sistemas de correio eletrônico ou de teleconferências.
- Os sujeitos estão fisicamente presentes, mas se comunicam parcialmente por meio do artefato (redes locais, etc.).

Cada uma destas situações pode trazer implicações na motivação (engajamento), compreensão e articulação do sujeito que deveriam ser analisadas caso a caso (BODKER, 1989, p.181). Na nossa opinião, seria interessante pesquisar como cada uma destas situações afeta e poderia ser usada na construção de ambientes educacionais. Uma pesquisa nesta linha deveria abordar questões do tipo: em que situações a maior ou menor presença de objetos manipulados via computador é recomendada na aprendizagem, quais as principais diferenças entre cursos à distância e cursos presenciais, etc.

Embora a maior parte da funcionalidade dos computadores possa ser obtida por artefatos não-computacionais isolados, isto é, telefone, correio, lápis e borracha, ... a grande flexibilidade e facilidade de integração dos diferentes meios é uma característica que se torna evidente nos artefatos computacionais.

Sob este enfoque, assume vital importância, no design de artefatos computacionais, o maior ou menor suporte que será dado para que o sujeito possa focar e operar os diversos objetos/sujeitos desejados. Isso de forma com que ele consiga atingir seus objetivos sem se distrair e da maneira que melhor lhe convier. É esse tipo de suporte que Bodker define como **interface com o usuário** ou, nos termos propostos, interface entre o sujeito e o artefato computacional.

Em todas as atividades, parte das ações sempre acaba por se dirigir ao artefato sendo utilizado ao invés de aos objetos. Por outro lado, o ideal seria que todas as

ações dirigidas ao artefato fossem completamente operacionalizadas e que o sujeito conseguisse concentrar suas ações no domínio de seu interesse. Por exemplo, em um software de matemática, o ideal seria que todos os comandos fossem intuitivos para o aprendiz e, se possível, que ele não tivesse que se preocupar com problemas inerentes ao computador como abrir e salvar arquivos, etc.

No entanto, o trabalho de Bodker indica que a transparência do artefato computacional é algo dinâmico e resultado de todo um desenvolvimento. O sujeito tem que passar algum tempo trabalhando diretamente com o artefato até que possa dominá-lo e aí conseguir concentrar-se no objeto de sua atividade. Apenas com a prática é que as ações dirigidas ao artefato vão sendo operacionalizadas (BODKER, 1991, p.29). Ainda assim, em situações que fogem da rotina (*breakdown*), o sujeito volta a tomar consciência dos diversos aspectos (aspecto no sentido de formas de se enxergar) do artefato.

Bodker identificou 3 diferentes tipos de aspectos para os quais o sujeito pode dirigir suas operações no uso do artefato computacional (BODKER, 1991, p.40):

- Os **aspectos físicos** (*physical aspects*), que suportam as operações dirigidas ao aplicativo como se este fosse um objeto físico.
- Os **aspectos de manuseio** (*handling aspects*), que suportam as operações dirigidas ao aplicativo em si.
- Os **aspectos dirigidos ao sujeito/objeto** (*subject/object-directed aspects*), que suportam as operações dirigidas aos sujeitos ou objetos com que lidamos com o artefato ou por meio dele.

No caso de uma pessoa escrevendo em um editor de textos, os aspectos físicos seriam, por exemplo, as formas e aparências do mouse, do teclado, do monitor; a intensidade da imagem, etc. – todas as características físicas que pudessem afetar o manuseio do artefato.

Os aspectos de manuseio compreenderiam tudo aquilo que, tirando os aspectos físicos, poderiam desviar a atenção do sujeito do domínio em que ele estivesse trabalhando: os botões que aparecem na tela, opções de menu, etc. com suas aparências e comportamentos básicos (clicar um botão e ele mudar de cor, por exemplo) e também comandos como abrir, fechar, salvar documentos que, no caso, não são inerentes ao ato de escrever.

Por fim, os aspectos dirigidos ao sujeito/objeto oferecem suporte às operações específicas do domínio, como apagar uma palavra, acentuar, corrigir a sintaxe, etc.

A partir destas definições, entende-se que, em “*uma boa interface com o usuário, nem os aspectos físicos nem os de manuseio originam ações em situações ordinárias de uso. Todas as ações devem ser dirigidas aos objetos e sujeitos apropriados ao invés de em direção ao artefato*” (BODKER, 1991, p.146).

Além disso, o ideal de uma interface é que ela seja flexível, consistente e simples. “*Flexibilidade tem a ver com as possibilidades de se mudar de foco entre objetos e sujeitos, mas também com a possibilidade de se atingir a mesma meta por diferentes caminhos (diferentes ações e operações). Consistência significa que o*

computador responde às ações do sujeito do mesmo modo em situações que pareçam as mesmas para o sujeito. Simplicidade se refere à quantidade de ações que leva para um usuário atingir uma certa meta” (BODKER, 1991, p.143, ênfase nossa).

Nível de competência. Dependendo do nível de competência do usuário, diferentes aspectos do manuseio do artefato chamarão mais a atenção do que outros. Usuários novatos tendem a seguir regras e a executar uma maior quantidade de ações dirigidas ao artefato em si. Usuários competentes já abreviam comandos, agrupando ações e se concentram mais no domínio do problema sendo resolvido. Experts já manipulam o artefato de uma forma bem mais operacional e têm uma visão mais ampla da atividade como um todo (BODKER, 1991, p.83).

No desenvolvimento de um artefato computacional deve-se estar sempre atento aos níveis de competência corrente e esperado do futuro usuário e facilitar ao máximo a passagem de um nível para outro. Isso implica que, para além da construção do artefato, deve-se planejar toda a educação que será dada aos usuários atuais para que eles atinjam, rapidamente, o grau de competência esperado na nova atividade.

A percepção da variação do uso de acordo com o nível de competência implica que um artefato não deve oferecer uma mesma interface para todos os tipos de usuários. Os novatos exigem uma interface que lhes auxilie passo-a-passo e que lhes oriente nos diversos detalhes da execução da ação. Usuários mais experientes exigem uma interface mais enxuta e que permita a criação e execução de grupos de comandos da forma mais simplificada possível. Às vezes, como mostra Bodker, a interface que é mais adequada para um pode atrapalhar a atividade do outro e vice-versa (BODKER, 1991, p.86).

Outro ponto importante a ser considerado é que, mesmo para um especialista, os três aspectos da manipulação do artefato computacional sempre continuam presentes e podem voltar à consciência quando algo deixar de se ser familiar. Por exemplo, um mouse com botões diferentes ou um travamento do computador pode levar o sujeito a se concentrar nos aspectos físicos; uma reorganização dos menus do aplicativo podem atrair a atenção para os aspectos de manuseio.

Por mais que se planeje a interação do sujeito com o artefato computacional, sempre haverá situações imprevistas (como falhas de hardware e software, ou limitações específicas do computador, como falta de espaço ou tamanho limitado dos nomes dos arquivos) que desviarão o sujeito do curso previsto da atividade. Nestas situações, a interface deve levar o sujeito de volta ao controle da situação sem forçá-lo a explorar áreas que fujam de sua competência. Para que isto seja possível, o designer pode contar tanto com a prática do usuário como com o desenvolvimento de atividades educacionais que indiquem o procedimento para casos específicos.

Por outro lado, na nossa opinião, embora o ideal fosse que o sujeito sempre estivesse concentrado no domínio de seu interesse, às vezes estas variações de foco podem ser planejadas, propositadamente, para levar o sujeito a prestar atenção em aspectos específicos do que estiver fazendo (como no caso de

apagamento de arquivos e outras operações que podem ser irreparáveis). O bom planejamento das variações de foco do sujeito é especialmente importante para o desenvolvimento de software para a educação, onde espera-se que os usuários percebam os novos conceitos sendo explorados.

Sumarizando o que foi visto até agora, a interface com o usuário, segundo Bodker tem os seguintes papéis:

- Facilitar a interação com os objetos/sujeitos desejados (e as propriedades desejadas destes objetos/sujeitos) e operacionalizar todos os demais aspectos;
- Facilitar a mudança de foco intencional e evitar a mudança de foco não intencional.
- Dar condições, em situações inesperadas, para que o usuário retorne à sua atividade original sem se distanciar de seu domínio de atuação.
- Suportar os diferentes níveis de competência do usuário frente ao artefato e facilitar as mudanças entre níveis.

4.2 A atividade de desenvolvimento do artefato computacional

A percepção de que o uso de artefatos, ao mesmo tempo que afeta a atividade, é afetado por ela, levanta uma série de aspectos que precisam ser considerados no desenvolvimento de novos artefatos.

Em primeiro lugar, é muito importante que a **atividade de desenvolvimento de um novo artefato extrapole o artefato em si e considere a atividade de uso como um todo sob seus diferentes aspectos** (BODKER, 1991, p.141; BELLAMY, 1996, p.135). Por exemplo, no caso da educação, muito se fala a respeito do computador e de tudo o que ele poderia vir a fazer de positivo ou negativo em cima do sistema educacional vigente. No entanto, da mesma forma que a introdução de um artefato altera os processos socioculturais e individuais que estiverem se desenrolando, estes mesmos processos também afetam o modo com que o artefato será utilizado, atuando como uma força de perpetuação do estado corrente.

Fatores como motivação de professores e alunos, suporte técnico para a manutenção dos equipamentos, quantidade de máquinas disponíveis e apoio institucional influenciam diretamente no resultado que será obtido. Por este motivo, o projeto da tecnologia não pode ser feito sem as considerações com relação à comunidade, às regras, à divisão do trabalho e à história do ambiente aonde será implantada. O ideal seria que tanto o artefato como a atividade educacional na qual ele será usado pudessem ser projetados em conjunto (BELLAMY, 1996).

Além disso, convém ressaltar que o desenvolvimento de artefatos para a educação não deve se limitar apenas à atividade dos aprendizes. Os novos artefatos também

devem oferecer suporte aos pais, professores, administradores e outros envolvidos. Os estudantes representam apenas uma fração dos participantes da atividade educacional (BELLAMY, 1996, p.144).

Ainda assim, é muito raro o software desenvolvido a partir de uma visão mais abrangente como a descrita acima. A maioria do software para a educação normalmente está centrada na relação aprendiz-computador e não considera outros fatores contextuais. Mesmo o Logo, como será melhor descrito mais adiante, não traz, praticamente, nenhuma facilidade que ajude os educadores a acompanharem o trabalho dos aprendizes.

Em segundo lugar, como cada atividade tem uma história e um desenvolvimento próprios, **não é possível assegurar que metodologias ou sugestões de desenvolvimento de artefatos sejam aplicáveis em todas as situações** (BODKER, 1991, p.141; BODKER, 1989, p.173). As metodologias de desenvolvimento não podem ser generalizadas de qualquer maneira, nem tão pouco devem ser seguidas como receitas passo-a-passo. Elas devem ser entendidas como artefatos criados com determinados fins e que, conforme a necessidade e a experiência do desenvolvedor, poderão ser utilizadas em combinação com outras metodologias.

Conforme mencionado anteriormente, os defensores da Teoria da Atividade não acreditam que ela seja uma panacéia capaz de resolver todos os problemas. A Teoria levanta diversos pontos a serem observados e traz consigo uma bagagem teórica bastante extensa que pode ajudar em várias situações, mas não exclui a experiência que já foi adquirida em outras abordagens que, às vezes, tratam problemas específicos de forma mais apropriada. Por exemplo, para tratar de sistemas onde a velocidade de processamento é crítica, talvez uma combinação com metodologias dirigidas a sistemas de tempo-real, fosse mais adequada. Se o ponto fosse a quantidade de dados, talvez regras de modelagem de dados fossem as mais indicadas. Em ambos os casos, no entanto, pelo menos sob o ponto de vista teórico, a Teoria da Atividade poderia ser utilizada para se determinar o papel do computador e suas relações com o contexto.

Em terceiro lugar, também devido às particularidades de cada atividade, por mais que se descreva e projete, a interface, ou a **adequabilidade do artefato, só se revela completamente nas situações de uso** (BODKER, 1991, p.141). É a vida diária que indica as necessidades a serem atendidas pelo novo artefato (BODKER, 1989, p.49). Por isso, na medida do possível, o desenvolvimento do artefato deve trabalhar bastante próximo de situações reais. O desenvolvimento de artefatos computacionais deve ser visto, então, como resultado de um processo iterativo que intercala projeto e prática num ciclo contínuo de refinamentos, tanto do artefato quanto da atividade em si. As metodologias devem ter como objetivo o aperfeiçoamento deste processo buscando sempre manter o usuário focado no que é relevante para a atividade e operacionalizando ou eliminando o que não interessar.

A fim de assegurar um desenvolvimento orientado para e pela realidade, a Teoria da Atividade incentiva a criação de protótipos do que se deseja. Esta ênfase em

protótipos e cenários de uso também é recomendada por uma série de outras metodologias de desenvolvimento de software.

Em quarto lugar, deve-se lembrar que **a atividade de desenvolvimento é multidisciplinar por natureza**, com profissionais de informática buscando implementar soluções em um campo que normalmente não é o seu. *“No design, nós devemos lidar com diferentes práticas, pelo menos com a dos usuários e com a dos designers”* (BODKER, 1991, p.143).

Para minimizar os possíveis problemas, cuidados especiais devem ser tomados com a comunicação das idéias, com o aprendizado mútuo e a integração do pessoal entre si. Um deles é compor a equipe de desenvolvedores com elementos das diversas áreas envolvidas. Bellamy (1996), por exemplo, sugere que, para o caso de software para educação, a equipe seja composta por educadores, analistas e alunos.

Outra idéia é seguir um ciclo de desenvolvimento que se fundamente na criação de “visões materializadas” (cenários, protótipos, ...) intermediárias do artefato computacional (BODKER, 1991, p.144). Estas visões, sejam elas informatizadas ou não, além de assegurarem que o processo de desenvolvimento se oriente pela e para a realidade (ajudando na identificação das situações mais ou menos freqüentes de uso e nos pontos em que o artefato leva à conceitualização), também servem como um meio de comunicação bastante efetivo entre engenheiros de software, usuários e demais envolvidos²⁷.

Recomenda-se que, nas fases iniciais do desenvolvimento, sejam aplicados os métodos mais genéricos possíveis – como fazer modelos de papelão ou plástico do artefato – a fim de não limitar o produto resultante ao que costuma ser normalmente usado. Mais tarde, técnicas e artefatos mais específicos podem ser utilizados (BODKER, 1991, p.149).

Essa idéia de se construir protótipos, usando materiais alternativos, é bastante interessante para o caso de artefatos computacionais dirigidos à educação. Conforme mencionado anteriormente, a tecnologia educacional corrente ainda segue os modelos lançados para o mercado de trabalho, com “computadores retangulares manipulados através de mouse e teclado”. Será que estes computadores têm o melhor formato para o trabalho com crianças, ou para o tipo de manuseio que se espera encontrar em uma escola? O uso de protótipos não computadorizados poderia trazer uma luz para o esclarecimento deste tipo de questões.

Outro conceito importante levantado a partir da Teoria da Atividade é que a **comunicação entre desenvolvedores e usuários** não deve terminar quando o

²⁷ Embora o uso de protótipos seja muito recomendado por diversas metodologias de software, em especial pelas derivadas do Design Centrado no Usuário (NIELSEN, 1993; LAUREL, 1990), a sua utilidade como meio de comunicação dentro da equipe é algo enfatizado principalmente pela Teoria da Atividade.

produto for entregue. Considerando-se que a atividade de desenvolvimento tem por objetivo resolver necessidades do uso e que a atividade de uso identifica novas necessidades a serem resolvidas pelo desenvolvimento, a comunicação deve se perpetuar (e ser incentivada) enquanto este ciclo continuar existindo.

Um artifício para assegurar este canal contínuo de comunicação é considerar, no projeto da atividade, dois papéis genéricos que poderiam ser aplicados em qualquer situação: o do designer da atividade (ou do software) e o do ator (GUOHUA, 1992, p.9).

O designer da atividade é o sujeito que inicia tecnicamente a criação da atividade e que, mais tarde, monitora o seu funcionamento e implementa as alterações necessárias ou sugeridas na prática de cada um dos demais papéis, inclusive do seu. O ator (social) é um sujeito que, de certa forma, age como um inimigo do sistema, questionando a atividade como um todo e oferecendo sugestões para o seu aprimoramento. É ele quem dá indícios, conscientemente ou não, do que o designer deverá alterar na atividade.

Na prática, o papel de ator pode ser exercido por qualquer sujeito da atividade que queira expressar alguma opinião com relação a atividade em si (ou, mais especificamente, com relação ao artefato computacional). Às vezes, a simples observação da atividade por parte do designer já é suficiente para se recolher dados que levem ao aprimoramento da mesma.

No caso do sistema Universe desenvolvido pela *Umeå University*, foi criado um "centro de comunicações" no software onde os estudantes podiam deixar seus comentários a respeito do sistema e onde os desenvolvedores podiam recolher e analisar as informações dadas (GUOHUA, 1992). O mesmo tipo de solução poderia ser usado nos software para a educação a fim de facilitar o intercâmbio entre os alunos, professores, etc. com os desenvolvedores da atividade e/ou do artefato computacional.

5 Introdução ao Construcionismo

Este capítulo apresenta os princípios gerais do aprendizado e da educação construcionista. Ele se concentra nos fundamentos teóricos, ressaltando sua interdependência com o desenvolvimento da tecnologia e, mais especificamente, do computador.

De fato, embora o Construcionismo tenha crescido ao redor do Logo e apareça normalmente vinculado a essa linguagem de programação, a teoria tem um corpo próprio com um forte componente sociocultural que também precisa ser considerado. Focando nos princípios, espera-se realçar este outro lado e, com isso, ilustrar e oferecer subsídios para justificar a importância do tipo de abordagem de desenvolvimento de software para a educação proposta no capítulo 6.

5.1 Características gerais

De uma forma geral, o Construcionismo pode ser entendido como um conjunto de idéias (ou uma teoria) que estuda o desenvolvimento e o uso da tecnologia, em especial, do computador, na criação de ambientes educacionais.

Concebido por Seymour Papert no final da década de 60, o Construcionismo é uma síntese da teoria de Piaget e das oportunidades oferecidas pela tecnologia para o desenvolvimento de uma educação contextualizada, onde os estudantes trabalhem na construção de elementos que lhes sejam significativos e através da qual determinados conhecimentos e fatos possam ser aplicados e compreendidos (PAPERT, 1986, p.8).

Um ponto importante do Construcionismo é que ele vai além do aspecto cognitivo, incluindo também as facetas social e afetiva da educação. Assim, ele abre espaço para o estudo das questões de tecnologia, gênero, cultura, personalidade, motivação, etc. que normalmente não são tratadas em abordagens educacionais mais tradicionais.

Influenciado diretamente pelos anos que Papert trabalhou ao lado de Piaget em Genebra e pelos conceitos da Inteligência Artificial que floresciam no MIT (*Massachusetts Institute of Technology*), o Construcionismo propôs uma série de idéias inovadoras a respeito do uso da tecnologia na educação. Diferente da abordagem predominante da época, onde o computador era visto como uma máquina de ensinar que ia passando informações de acordo com as respostas do aluno, Papert propôs uma visão mais centrada na construção do conhecimento em si, onde a tecnologia deveria ser utilizada não como um elemento regente, mas sim como algo que possibilitasse a criação de situações mais propícias e específicas para o aprendizado.

Atualmente, o resultado prático mais conhecido do Construcionismo é o Logo Gráfico, usado por milhões de pessoas em diversos países. Popularizado na década de 80 com o advento dos microcomputadores, o Logo Gráfico é, realmente, o produto modelo do Construcionismo, um exemplo concreto de como as idéias de Papert podem ser aplicadas no ensino principalmente de matemática e de como o computador pode ser visto como um poderoso auxiliar para o desenvolvimento humano.

No Logo, a criança interage com uma tartaruga desenhada na tela do computador. A tartaruga entende um conjunto de comandos gráficos bem simples (como “parafrente 10”, que move a tartaruga 10 passos para frente, ou “paradireita 30”, que gira a tartaruga 30 graus para a direita), todos eles relacionados com conceitos de geometria e facilmente compreensíveis pela criança.

Por exemplo, o desenho na Figura 8 foi criado a partir dos seguintes comandos: parafrente 100 paradireita 30 parafrente 80 paradireita 50 paradireita 10 parafrente 70.

Através de uma linguagem de programação de fácil compreensão (a linguagem Logo), o aprendiz pode ir ensinando comandos cada vez mais complexos para a tartaruga (como “desenhe_quadrado”, que combina diversos “parafrente” e “paradireita”). Assim, além de obter resultados cada vez mais sofisticados e motivantes, a criança vai aprendendo os conceitos básicos de geometria espacial (como distância, ângulo, posição no espaço, etc.) de uma forma significativa, prática e gradual.

Na utilização do Logo Gráfico, o aprendiz assume uma postura ativa frente ao seu aprendizado e ao computador e vai, através do desenvolvimento de projetos pessoais, explorando novos conceitos e progredindo em seu próprio ritmo. Além disso, como todos os comandos “ensinados” para a tartaruga ficam registrados e podem ser manipulados por meio do computador, o aprendiz tem à sua disposição um recurso bastante concreto que lhe permite visualizar o que foi feito e aprimorar seus projetos. Este tipo de potencial propiciado pela tecnologia é um ponto-chave enfatizado pelo Construcionismo.

Por outro lado, é importante frisar que, ainda que a tecnologia seja realmente importante e constitua um dos focos centrais da pesquisa construcionista, para o Construcionismo um ambiente educacional efetivo exige muito mais do que um aprendiz e um computador carregado com o Logo. É preciso todo um ambiente acolhedor que motive o aprendiz a continuar aprendendo, um ambiente que seja rico em materiais de referência, que incentive a discussão e a descoberta e que respeite as características específicas de cada um.

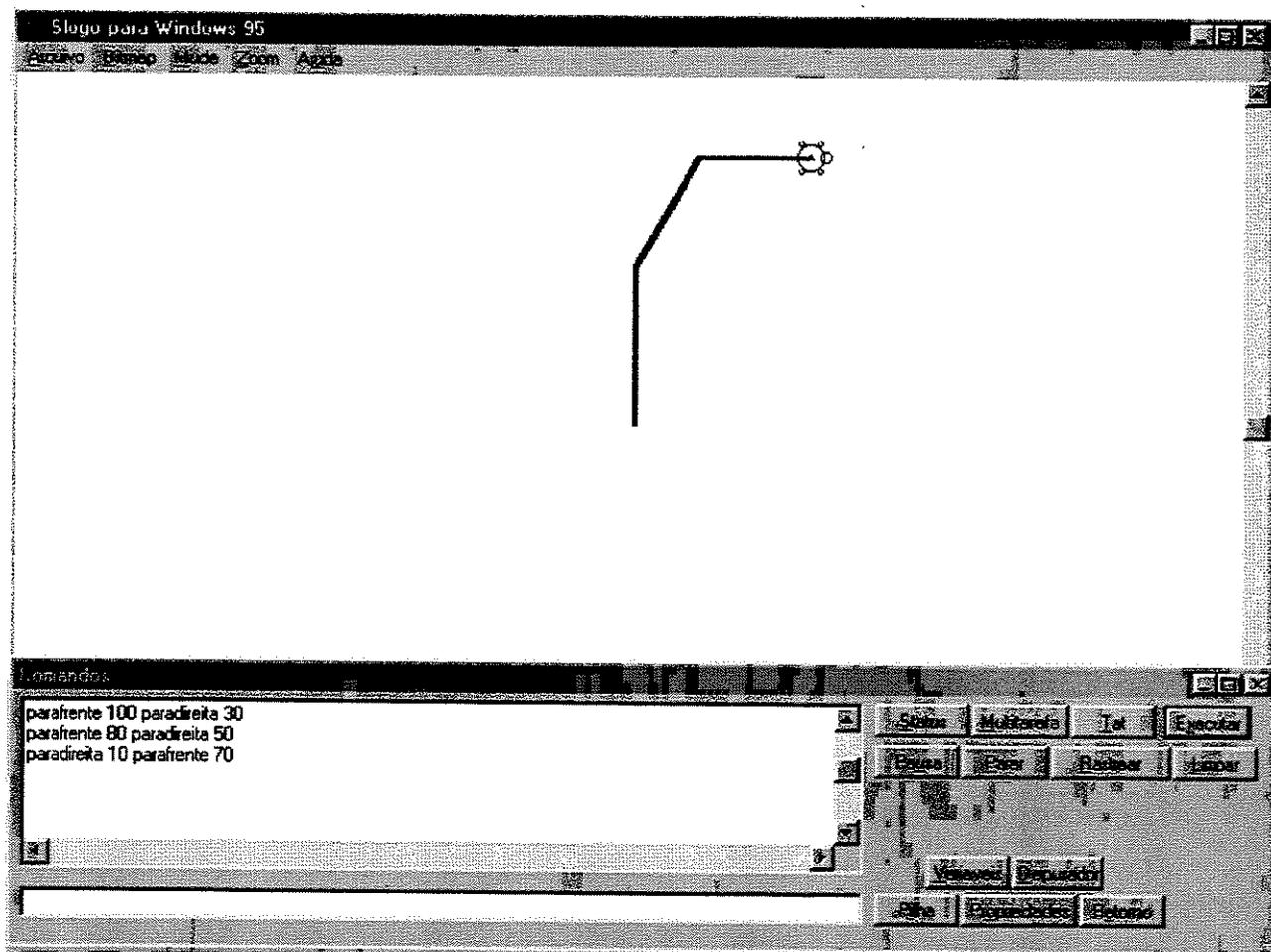


Figura 8 - Desenho feito a partir de comandos do Logo Gráfico.

O Logo Gráfico é apenas um dos materiais utilizados na construção de ambientes construcionistas. Mesmo esta ferramenta, dependendo da forma como for utilizada, pode levar a resultados completamente diferentes dos esperados.

Dificuldades relacionadas ao desenvolvimento de software construcionista. Ainda que o Construcionismo seja um corpo teórico revolucionário na consideração de fatores afetivos, sociais e culturais no aprendizado e na abordagem dada ao papel da tecnologia na construção de ambientes educacionais mais efetivos, é interessante notar que pouca gente o conhece e que pouco se sabe a respeito da tecnologia construcionista.

Há uma série de fatores que poderiam explicar esse fato. O primeiro deles é que o Construcionismo é uma teoria muito recente e ainda muito restrita a alguns poucos centros acadêmicos, como o Grupo de Epistemologia e Aprendizado do MIT Media Lab, o Laboratório de Estudos Cognitivos (LEC) da UFRGS e o Núcleo de Informática Aplicada à Educação (NIED) da UNICAMP. Só agora é que começam a aparecer mais textos dirigidos ao público em geral, como é o caso de *The connected family: bridging the digital generation gap* (PAPERT, 1996).

Além disso, mesmo no meio educacional não parece haver um consenso claro do que venha a ser o Construcionismo, algo que seria essencial ao desenvolvimento da

teoria em si e de suas tecnologias associadas. *“Enquanto nós não tivermos o substrato para fundamentar qualquer tipo de inovação, a tendência é deixar as coisas do jeito que estão. O que acaba acontecendo é um pouco do que eu chamo de ‘deslumbramento tecnológico’”*²⁸. Não existe nenhum documento ou coisa parecida que defina o que é o Construcionismo e quais as suas características essenciais.

É óbvio que não se pode esperar uma definição fechada, uma vez que o Construcionismo é um conjunto de idéias que está em constante desenvolvimento, principalmente nas universidades. O problema é que os artigos e teses construcionistas são raramente traduzidos (mais um fator para dificultar o seu intercâmbio) e a maior parte das pesquisas aborda pontos específicos da teoria, tornando difícil obter uma visão do todo e de como os trabalhos se relacionam entre si.

Embora o Logo Gráfico tenha servido como um foco concentrador de forças para o próprio desenvolvimento e propagação do Construcionismo, a centralização neste software foi tão grande que gerou uma série de efeitos colaterais sentidos ainda hoje.

Um dos mais marcantes foi a visão tecnocentrista com que as pessoas passaram a associar os bons resultados das atividades construcionistas ao software em si, deixando de lado todos os demais fatores socioculturais como liberdade de expressão, incentivo à troca de idéias e à criatividade, enfatizados pela teoria (PAPERT, 1990).

Ao redor do Logo Gráfico também se criou uma cultura quase que religiosa, defensora da utilização deste software em todas as situações de aprendizado, quaisquer que sejam elas. Não que o Logo não possa ser utilizado para o estudo de Gramática, História, Química ou outro assunto qualquer, dado o caráter relativamente aberto do software. O ponto é que ele foi criado como um ambiente para a exploração do raciocínio lógico-matemático e de alguns conceitos de Geometria, não oferecendo muitas facilidades que tornem explícitas as características específicas de outras áreas do conhecimento.

Por exemplo, se o objetivo de uma aula de Geografia fosse analisar dados demográficos de um país, talvez o Logo não fosse a ferramenta mais adequada. Os alunos provavelmente concentrariam seus esforços mais na programação da tartaruga do que nos conceitos sendo explorados. Neste caso, talvez fosse mais interessante utilizar uma planilha eletrônica ou um software mais específico como o ARC/INFO (MCARTHUR, 1994), onde os aprendizes pudessem fazer e testar hipóteses sobre as variáveis que achassem pertinentes. De qualquer forma, problemas como esse poderiam ser minimizados caso houvesse uma conscientização maior dos pontos essenciais do Construcionismo.

²⁸ Heloísa Vieira da Rocha sendo citada em (VALENTE, 1996b, p.418).

Uma exposição mais objetiva das idéias, além de facilitar o diálogo entre pesquisadores, educadores, desenvolvedores e demais interessados, também serviria como um elemento norteador para o desenvolvimento de ambientes construcionistas voltados para outras áreas (além da matemática) e que pudessem usufruir dos novos recursos (como a multimídia e as redes) desenvolvidos pela tecnologia.

Seguindo nesta linha, procurou-se reunir, neste capítulo, aqueles que acreditamos ser os princípios mais importantes do Construcionismo. Convém notar que, na nossa opinião, o Construcionismo abrange tanto uma teoria do aprendizado – um conjunto de idéias que descrevem como as pessoas aprendem – como também uma teoria educacional – princípios sobre como criar situações onde as pessoas aprendam determinados conteúdos. É a teoria do aprendizado construcionista que dá embasamento para a criação de ambientes educacionais construcionistas.

5.2 O aprendizado construcionista

Na literatura construcionista se destacam os seguintes princípios de aprendizado:

- As pessoas constroem ativamente o seu conhecimento, isto é, conhecimento não é transmitido.
- A possibilidade de articular os processos do pensamento permite aprimorá-los, isto é, a visualização e a manipulação das estratégias permite otimizá-las.
- O aprendizado de um conceito está relacionado com a sua estrutura, isto é, o aprendizado de alguns conceitos e a possibilidade de combiná-los facilita o aprendizado de outros conceitos.
- O aprendizado é influenciado pelo ambiente. Algumas dinâmicas e contextos facilitam a percepção e a construção de determinados conhecimentos, atitudes e procedimentos.

Cada um destes princípios será melhor discutido nas seções restantes deste capítulo.

5.2.1 As pessoas constroem ativamente o seu conhecimento

As situações com que nos deparamos em nosso dia-a-dia são, constantemente, contrastadas com nossos conhecimentos anteriores e, através de um mecanismo de assimilação do novo e acomodação das estruturas antigas, vão sendo incorporadas em nossas mentes.

A visão de que o indivíduo constrói seu próprio conhecimento é central no construtivismo piagetiano e segue uma linha contrária ao de outras abordagens que defendem, por exemplo, que o conhecimento pode ser transmitido, unidirecionalmente, em pequenos “pacotes”, de um professor (transmissor ativo) para seus alunos (receptores passivos).

O conceito de construção de conhecimento pelo aprendiz é tão fundamental para o Construcionismo que até mesmo o nome da teoria reflete esta idéia. Segundo Papert, o termo Construcionismo (Constructionism) foi escolhido por representar os dois temas principais de sua teoria: o Construtivismo, de Piaget e a idéia dos kits de montagem para crianças (os *construction sets*).

A importância do erro. É interessante notar que, neste processo de construção, o sujeito não passa diretamente de um conhecimento para outro mais avançado. O aprendizado se dá através da construção de uma série de teorias transitórias. Esse processo ocorre via tentativas e erros, no qual o indivíduo parte dos aspectos já conhecidos do problema e segue construindo suas próprias teorias. As teorias que não forem adequadas vão sendo descartadas ou alteradas até se tornarem cada vez mais estáveis.

Sob este enfoque, os erros dos indivíduos são tão importantes quanto seus acertos para o processo de aprendizagem. Enquanto os acertos representam situações de relativa adequação do conhecimento do indivíduo com relação às coisas do mundo, são os “erros” que questionam esta estabilidade e agem como a força motriz do processo de aprendizagem.

Por exemplo, uma criança tentando desenhar um quadrado com o Logo raramente irá conseguir na primeira vez. A figura “errada” ou diferente que surgir na tela é que servirá de base, junto com os comandos dados, para que ela reflita sobre o que foi feito e assim possa corrigir seu desenho.

Ainda assim, não se deve super-valorizar os erros. Eles são momentos transitórios que levam o indivíduo de um estado de segurança para outro. Os erros são os “maus necessários” que temos que passar para vivermos cada vez melhor no ambiente que nos cerca.

5.2.2 A possibilidade de articular os processos do pensamento permite aprimorá-los

Quando Papert mudou para o MIT, em 1964, deparou-se com uma cultura que via no computador um grande auxiliar para a compreensão da inteligência humana. A resolução de problemas por computador e, principalmente, sua capacidade de simulação em muito contribuíram para a compreensão da mente humana. Foi esta visão do computador, como uma ferramenta que permite pensar de forma mais concreta sobre as questões abstratas do pensamento e do funcionamento da mente, que determinou o seu papel como central dentro do Construcionismo.

A partir de então, Papert passou a "*visualizar as idéias da ciência da computação não apenas como instrumentos que pudessem explicar como o aprendizado e o pensamento de fato funcionam, mas também como instrumentos de mudança que poderiam alterar e possivelmente melhorar, a forma com que as pessoas aprendem e pensam*" (PAPERT, 1993a, p.208).

Do mesmo modo que um mapa ajuda na orientação e a encontrar o melhor caminho entre dois pontos, para o Construcionismo a possibilidade de visualizar e manipular de uma forma clara as estratégias e tentativas percorridas na resolução de uma

tarefa traz uma série de facilidades para que se identifique o que ainda não foi feito, possíveis deficiências, etc. e, com isso, se obtenha uma solução melhorada. A questão é como registrar este meta-processo de uma modo compreensível e útil.

É justamente por este motivo que a programação é um dos aspectos que os construcionistas mais enfatizam no Logo. Conforme aponta Valente (1995), um programa é um registro preciso da organização e da seleção dos comandos que o aprendiz usou na resolução de um problema (como um desenho em Logo), podendo ser tratado como uma representação objetiva dos procedimentos adotados, facilitando o aprimoramento dos mesmos.

O ciclo da descrição-execução-reflexão-depuração-descrição. Valente (1993, p.34) descreve, através de um ciclo composto pelas fases de descrição, execução, reflexão e depuração, a atividade cognitiva de um aprendiz utilizando o Logo (ver Figura 9). Este ciclo começa quando o aprendiz tem uma idéia, talvez de um desenho, que gostaria de implementar no computador. Ele busca então descrever esta idéia através de comandos e procedimentos que a tartaruga geométrica possa executar.

A tartaruga (ou o computador) executa os comandos dados e produz um resultado na tela do computador. O aprendiz então olha para a figura sendo desenhada e para o resultado final e reflete comparando-os com o que havia imaginado.

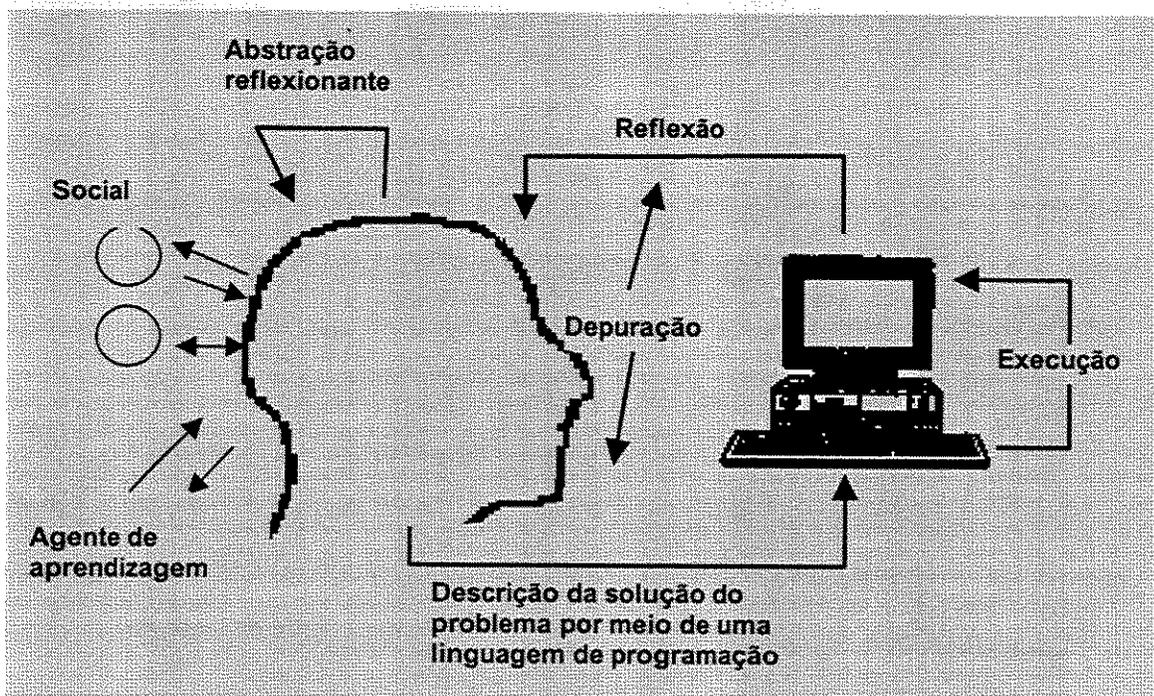


Figura 9 - O ciclo da descrição-execução-reflexão-depuração.

Neste momento, ele pode optar por deixar o programa do jeito que está, concluindo o ciclo; ou optar por depurar o programa, buscando resolver as diferenças identificadas,

A fase de depuração é onde o programa se mostra mais importante. Como ele guarda os comandos dados, as decisões tomadas pelo aprendiz ficam

documentadas. Revendo o programa, o aprendiz pode repensar o que fez e, à medida em que procura consertar a figura, pode desenvolver novas idéias, buscar novos conceitos e, com isto, construir novos conhecimentos.

Valente também ressalta que, além do suporte dado pelo computador, a efetividade do ciclo também depende do acompanhamento do aprendiz por parte de um facilitador e de uma boa integração entre a atividade e a realidade sociocultural do aprendiz. Este aspecto será melhor abordado na seção sobre educação construcionista.

5.2.3 O aprendizado de um conceito está relacionado com a estrutura deste conceito

Piaget acreditava que, para se compreender o aprendizado era primeiro necessário compreender as chamadas “estruturas-mãe” do conhecimento a ser aprendido. O aprendizado seria resultado da aquisição e a manipulação destas estruturas fundamentais.

Mas que características possuem estas estruturas-mãe que as tornam “aprendíveis”? Os estudos de Piaget mostraram que (PAPERT, 1993a, p.160)

- As estruturas-mãe são conceitos que fazem sentido para a criança e que, em princípio, podem ser aprendidos e compreendidos isoladamente das outras estruturas.
- Podem ser agrupadas entre si, possibilitando a formação de novos conceitos que também podem ser compreendidos pela criança.
- Se suportam mutualmente, isso é, ainda que independentes, o aprendizado de uma facilita o aprendizado das outras²⁹.

As estruturas-mãe levam a entender o que Papert chamou de “princípio da continuidade” (*continuity principle*) que diz que os novos conceitos devem continuar o conhecimento bem estabelecido que o indivíduo já tem. Isso traz um senso de segurança e valor e também de competência cognitiva que constitui uma base sólida que incentiva o aprendiz a se arriscar por novos domínios (PAPERT, 1993a, p.54).

O Logo Gráfico foi criado a partir de uma série de conceitos básicos que podem ser analisados como estruturas-mãe da geometria espacial. Este é o caso das noções de posição no espaço, ângulo, deslocamento relativo, etc., todos eles representados na tela de uma forma compreensível pelo aprendiz.

A linguagem de programação do Logo Gráfico permite combinar os comandos básicos da *tartaruga* em estruturas mais complexas possibilitando, através de um mecanismo de descrição-execução-reflexão-depuração-descrição (ver seção 5.2.2), a compreensão de conceitos mais avançados como o de polígonos, áreas,

²⁹ Foi a partir da identificação de algumas estruturas-mãe e da aplicação destes conceitos que Piaget elaborou sua teoria sobre o desenvolvimento do conhecimento para uma variedade de domínios.

somatória de ângulos internos, circunferências, etc.. O entendimento de um conceito básico ajuda na compreensão de outro, como perceber que uma reta é uma combinação de vários pontos alinhados ou que uma curva pode ser vista como uma seqüência de pequenos deslocamentos não alinhados.

Tal como visto na seção anterior, o próprio aprendizado da informática é visto por Papert como um tipo de conhecimento que suporta o aprendizado de outros conhecimentos (PAPERT, 1986, p.2; HAREL e PAPERT, 1991, p.75). Por exemplo, a utilização de ferramentas de simulação, por mais que seja virtual, pode ajudar a compreender melhor o fenômeno sendo estudado e, inclusive, as restrições por trás do modelo.

Para o Construcionismo, o aprendizado de informática, dependendo de como for feito, ao invés de exigir tempo extra no horário escolar, poderia ser integrado às demais matérias, resultando em um melhor aproveitamento global. E esta sinergia cognitiva não estaria relacionada exclusivamente ao computador. Do mesmo modo, o aprendizado de física poderia ser integrado com o de química, o de geografia com o de história e o de línguas e assim por diante.

5.2.4 O aprendizado é influenciado pelo ambiente

Diferente da abordagem de Piaget, a teoria construcionista dá uma importância especial à influência dos aspectos culturais e sociais no desenvolvimento cognitivo. Enquanto Piaget se preocupava com o estudo das estruturas cognitivas que se desenvolvem em todos os seres humanos, independente da cultura, Papert se preocupou com aquelas que poderiam se desenvolver em situações socioculturais específicas e que, portanto, acabavam diferenciando umas pessoas das outras (ACKERMANN, 1990).

Segundo Papert, o conhecimento não pode ser construído do nada. É o meio sociocultural que fornece o material a ser usado, influenciando os tipos de construções e a forma com que elas são construídas. Papert vai inclusive mais fundo neste aspecto, sugerindo que até mesmo a seqüência dos estágios do desenvolvimento cognitivo proposta pela teoria piagetiana poderia ser mais ou menos acelerada, dependendo da cultura e dos materiais presentes no ambiente.

Mais especificamente, Papert acha que a cultura contemporânea oferece relativamente pouca oportunidade para que se possa pensar e falar sobre as hipóteses levantadas e procedimentos adotados na resolução de problemas e que este é o principal fator por detrás do desenvolvimento tardio deste tipo de raciocínio (que, por Piaget, era conhecido como "pensamento formal") nas crianças. Neste caso, o computador e, em especial, a programação, pelas razões discutidas anteriormente, poderiam levar a uma relação mais concreta com o pensamento, possibilitando uma alteração nesta situação (PAPERT, 1993a, p.174).

Também é interessante notar que, apesar de ter um desenvolvimento pouco incentivado pela cultura contemporânea, o pensamento lógico-abstrato é tradicionalmente visto não como um estilo de pensamento, mas sim como um

modelo a ser seguido. Isso acabou trazendo uma série de implicações para a educação que devem ser consideradas³⁰.

Em termos de estilos de pensamento, conforme mostram os estudos construcionistas, nas atividades educacionais encontram-se, basicamente, dois tipos de sujeitos: os planejadores (*planner* ou *hard thinker*) e os escultores (*bricoleur*, *tinker* ou *soft thinker*). Os planejadores são aqueles sujeitos que, antes de partir para a construção, gastam bastante tempo imaginando e descrevendo, em seus mínimos detalhes, tudo o que pretendem fazer. Já os escultores são aqueles que, a partir da manipulação dos materiais, vão definindo o que pretendem construir.

Tradicionalmente, da mesma forma que ocorre com o pensamento abstrato, o estilo planejador é mais incentivado. No entanto, “*a diferença entre planejadores e escultores não está na qualidade do produto, mas no processo de sua criação*” (TURKLE, 1991, p.172). Assim como os planejadores, os escultores também têm objetivos. A diferença é que eles se permitem negociar estes objetivos à medida em que o projeto vai se desenvolvendo. Enquanto os planejadores têm um pensamento predominantemente analítico, abstrato e genérico, que leva à otimização dos procedimentos e ao cumprimento dos prazos estipulados, os escultores têm uma forma de pensar considerada culturalmente mais feminina, privilegiando o pensamento mais narrativo, concreto e específico, que leva à descoberta de novas propriedades e combinações dos objetos.

Outra diferença apontada é o tipo de relação que os diferentes sujeitos têm com os objetos da atividade. Os planejadores tendem a ter uma visão de fora e mais sistêmica do conjunto. Os escultores se imaginam como parte do problema sendo resolvido e procuram se identificar mais com seus componentes.

Papert ressalta que tanto as características dos planejadores quanto as dos escultores são fundamentais para a resolução de problemas. Em algumas situações, pode ser mais importante aprimorar as técnicas e otimizar o trabalho. Outras situações podem exigir abordagens mais criativas que extrapolem as tradicionais.

Hoje em dia, no entanto, parece que o tipo de pensamento escultor tem se mostrado cada vez mais reconhecido. Dada a velocidade com que as coisas mudam, os problemas enfrentados raramente se repetem e exigem um tipo de indivíduo que saiba lidar com dados imprecisos e incompletos que se alterem constantemente. Já

³⁰ Papert é bastante incisivo nesta questão da valorização da lógica na cultura moderna e isto se reflete também diretamente em suas críticas à visão de que o pensamento formal proposto por Piaget seja visto como o mais avançado. Segundo Papert (1993b, p.167), “*O tipo básico de pensamento é intuitivo; o pensamento lógico formal é uma construção que, embora freqüentemente seja extremamente útil, é artificial*”.

Em outras palavras, a lógica deveria ser vista como uma ferramenta e não como a melhor forma de pensar. “*Logic is on tap, not on top*” (TURKLE e PAPERT, 1991, p.168).

se foi o tempo em que empresas e pessoas faziam planejamentos de longa duração e conseguiam se ater a eles por vários anos.

Convém notar que a divisão apresentada dos estilos de aprendizagem é meramente figurativa. Ninguém é totalmente planejador ou escultor e uma mesma pessoa, dependendo da situação, pode tender mais para um estilo do que para outro. A predominância dos estilos de aprendizagem não é genética; ela está ligada à cultura e à formação recebida.

Assim sendo, os ambientes construcionistas procuram privilegiar ambos os estilos da mesma forma, deixando amplo espaço para explorações planejadas e não planejadas, favorecendo o intercâmbio de idéias e oferecendo um equilíbrio de condições para o desenvolvimento de descrições e representações formais, favorecendo os planejadores e a manipulação de materiais concretos com os quais o aprendiz se identifique, favorecendo os escultores .

O ideal seria que os aprendizes reconhecessem seus próprios estilos de aprendizagem, valorizassem os estilos dos colegas e percebessem os momentos mais apropriados para usar um estilo ou outro.

Novamente, o Logo Gráfico é um bom exemplo de ferramenta que permite trabalhar com ambos os estilos. Ao mesmo tempo em que a tartaruga gráfica representa um objeto antropomorfizado, no qual o aprendiz reconhece características do próprio corpo, os novos comandos ensinados e as anotações feitas são convites para a abstração e ao formalismo. Além disso, a possibilidade de ir montando o programa aos poucos, de ir experimentando e refazendo sem muito trabalho o que vai sendo construído permitem que o aprendiz explore seu lado escultor sem inibir o lado planejador da atividade.

Apesar desta abertura, de nada adianta a ferramenta oferecer as possibilidades se ela não for utilizada de modo adequado. Mesmo o Logo pode ser usado de forma totalmente diferente da defendida pelo Construcionismo. É o que aconteceria, por exemplo, se os alunos fossem incentivados a copiar programas já feitos ao invés de construírem os seus próprios.

De fato, para Papert, a maioria das críticas levantadas contra o Logo (ou contra o uso do computador na educação) baseia-se em critérios tecnocentristas, onde o computador é colocado em primeiro plano e os aspectos socioculturais são considerados como meros facilitadores do uso da tecnologia. "*O contexto para o desenvolvimento humano é sempre uma cultura, nunca uma tecnologia isolada*" (PAPERT, 1990, p.3).

Da mesma forma que não se pode garantir que "madeira produz boas casas", é preciso que se comece a perguntar "*não o que o Logo pode fazer às pessoas, mas o que as pessoas podem fazer com o Logo*" (PAPERT, 1990, p.12). O Logo foi criado e avaliado no MIT como "*um material de construção para ambientes de aprendizado*" (PAPERT, 1986, p.16), não como algo que causa efeitos específicos em quem usa. "*Ele pode ser usado de diversas maneiras e, dependendo de como for usado, poderá ter diferentes tipos de efeitos*". O Logo não é um agente ativo no

aprendizado, "ele é um elemento que pode se tornar parte de uma cultura e será moldado pela cultura tanto quanto ele a moldará "(PAPERT, 1986, p.16).

É comum pensar que os "tecnologistas" seriam os que mais caem nesta armadilha tecnocentrista. Porém, os "humanistas", talvez pela falta de um conhecimento mais profundo sobre a tecnologia, tendem a dogmatizá-la e acabem por centralizar seu pensamento sobre ela.

O problema é que mudar esta mentalidade tecnocêntrica não é nada trivial e exige não apenas um reexame do que é assumido com relação à tecnologia, como também da área em que esta está sendo utilizada. No caso da informática para a educação, isso exigiria o repensar do que se entende por educação, algo que, por si só, já seria uma fantástica contribuição da informática.

Por outro lado, na nossa opinião, se o problema do uso do computador na educação é mais cultural do que tecnológico, uma ênfase maior deveria ser dada à propagação de uma cultura construcionista. Se "a dificuldade do Logo é que ele não vem acompanhado de um Papert", é preciso que se faça todo um trabalho de conscientização dos educadores (através de materiais de apoio ou de cursos) sobre o uso adequado da tecnologia.

Nesse sentido, vários esforços estão em andamento. A formação de professores para um trabalho construcionista com os computadores tem sido, há vários anos, o principal foco da pesquisa do NIED - Núcleo de Informática Aplicada à Educação da UNICAMP (VALENTE, 1996b). Os livros mais recentes escritos por Papert (1993b; 1996) também parecem dirigidos para um público mais amplo, como professores e pais em geral.

5.2.4.1 Características dos ambientes educacionais construcionistas

A partir de diversos estudos baseados no Logo, foram identificadas uma série de dimensões encorajadas em ambientes construcionistas de sucesso (PAPERT, 1986, p.14):

a) Dimensão pragmática, enfatizada nas atividades em que os aprendizes tenham a sensação de estarem aprendendo algo de uso imediato, não algo que só será utilizado no futuro. Desta forma, permitindo com que o aprendiz consiga desenvolver projetos pessoais significativos que anteriormente não seriam possíveis, os novos conceitos trazem uma sensação de praticidade e poder, incentivando cada vez mais a busca pelo saber. Esta dimensão também é conhecida como "princípio do poder" (PAPERT, 1993a, p.54).

Além da construção de algo imediatamente significativo, costuma-se descrever o aprendizado construcionista como uma atividade de construção no mundo. O "no mundo" deve ser entendido como um produto real, concreto. Algo que seja compartilhável com a comunidade, como um texto, um desenho, uma escultura, um carrinho ou um programa de computador.

Segundo Aaron Falbel, quando o aprendiz constrói algo no mundo, ele constrói, simultaneamente, conhecimento em sua cabeça. Este novo conhecimento permite ao aprendiz construir coisas mais complexas no mundo, o que levará a novos

conhecimentos e assim por diante, em um ciclo que se auto-reforça (FALBEL, 1993).

Além disso, é a construção no mundo que permite à comunidade reconhecer o trabalho que o aprendiz está desenvolvendo e, assim, torná-lo genuinamente significativo;

b) Dimensão sintônica, que é ressaltada quando o computador é usado na obtenção de um aprendizado sintônico, isto é, um aprendizado que, diferente do aprendizado dissociado encontrado nas salas de aula tradicionais, está diretamente relacionado com o senso que o aprendiz tem das coisas, em sintonia com aquilo que ele acredita ser importante. No aprendizado sintônico, o aprendiz se identifica com os elementos com que está lidando e com a atividade que está executando. O aprendizado está bem contextualizado e integrado com o seu dia-a-dia.

Por exemplo, o trabalho com a tartaruga tem (PAPERT, 1993a):

- **Sintonicidade com o corpo** (é *body syntonic*), por estar fortemente relacionado com o senso e conhecimento que a criança tem de seu corpo. Ela se identifica com a tartaruga, um animal que tem cabeça, se move e vira de um lado para o outro;
- **Sintonicidade com o ego** (é *ego syntonic*), no sentido de estar de acordo com o senso que as crianças têm de si, respeitando suas intenções, metas, desejos, afinidades e estilos. No Logo Gráfico, os comandos usam a própria tartaruga como referência – virar tantos graus em relação à posição corrente, caminhar tantos passos a partir de onde se está, etc. – não um sistema externo. Isso facilita com que a criança se coloque na posição da tartaruga, assimile os novos conceitos e, posteriormente, aprenda mais facilmente como utilizar sistemas externos de referência como o de coordenadas cartesianas e outros.

Além disso, o fazer algo que seja "pessoal" é muito importante. É ele que permite ao aprendiz reconhecer a sua marca no que estiver fazendo e, assim, fortalecer sua identidade.

- **Sintonicidade cultural** (é *cultural syntonic*), podendo ser relacionada com a cultura extra-escolar da criança, isto é, com aquilo que ela vê no seu dia-a-dia. No caso, o domínio do computador, algo extremamente valorizado em nossa sociedade e o trabalho com desenhos contribui para que ela se identifique com os materiais e a proposta.

Para os construcionistas, a sintonicidade é uma questão de possibilidade e liberdade de opção: "*Quanto mais escolha um estudante tiver para construir ou criar, maior a possibilidade de engajamento pessoal e investimento na tarefa. E quanto mais um estudante puder se relacionar ou se conectar com a tarefa em mãos, maiores as chances de que o novo conhecimento se conectará com seu conhecimento preexistente*" (FALBEL, 1993).

Convém notar que a escolha está diretamente relacionada com a variedade de materiais que o aprendiz tem à sua disposição, algo que depende, além da

disponibilidade física do material em si, da liberdade dada no ambiente para que ele possa ser usado.

c) Dimensão sintática. Para que possam ser usados, não basta que os materiais estejam disponíveis e que o aprendiz se identifique com eles. É necessário que estes materiais possam ser manipulados e combinados de acordo com as capacidades físicas e cognitivas dos aprendizes.

No caso do Logo Geométrico, por exemplo, os comandos que controlam a tartaruga foram projetados para serem facilmente aprendidos por iniciantes e a estrutura da linguagem permite que eles sejam combinados em procedimentos que executem funções mais complexas. O mesmo pode ser feito combinando-se tijolinhos de LEGO na construção de uma parede e na combinação de paredes para se fazer uma casinha de brinquedo.

O ideal seria que os materiais usados pudessem ser acessados sem nenhum pré-requisito e que também oferecessem um escopo de desenvolvimento ilimitado – o *"no threshold and no ceiling"* (PAPERT, 1986, p.15). Na prática isso acaba por se tornar inviável, mas é um ideal que deve ser perseguido o máximo possível. Diversas versões do Logo vêm sucedendo-se na tentativa de reduzir os requisitos de operação do sistema e aumentar as possibilidades do que pode ser feito. A incorporação de interfaces gráficas e comandos de multimídia são exemplos de algumas soluções que foram adotadas com esse objetivo em algumas versões mais recentes do Logo Geométrico.

d) Dimensão semântica. Para que, através da manipulação e construção, os aprendizes possam ir descobrindo novos conceitos, é necessário que os materiais usados carreguem significados múltiplos. Além de serem psicologicamente evocativos para o aprendiz, eles também devem trazer dentro de si conceitos e idéias que sejam representativas do assunto que está sendo estudado. Papert dá um exemplo disso quando fala da importância que as engrenagens tinham em sua infância. Elas eram objetos bastante familiares – ele conseguia imaginar-se rodando como elas – e também traziam dentro de si importantes conceitos de física e matemática. Essa multiplicidade de significados, segundo ele, serviu como uma porta de entrada bastante suave do seu mundo de brincadeira para o universo dos números (PAPERT, 1993a, p.xx).

O uso destes objetos transicionais (PAPERT, 1993a, p.xx) é uma característica que também se torna aparente no Logo Gráfico. A tartaruga gráfica é um objeto que, de um lado, as crianças se identificam (elas se imaginam como a tartaruga, brincam de tartaruga, etc.) e, de outro, é utilizada para transmitir noções de geometria. Devido a estas suas propriedades, a tartaruga acaba servindo como um objeto que ajuda as crianças a pensar sobre os conceitos da geometria. Nos termos de Papert, ela é considerada um *"objeto-para-se-pensar-com"* (*object to think with*).

Convém notar que, no Logo Gráfico, o que torna a tartaruga potencializadora do desenvolvimento de conhecimentos geométricos são as ferramentas que a criança têm à sua disposição para brincar com ela. A tartaruga é um objeto como qualquer outro que as crianças vêem por aí. Ela, em si, não tem nada demais (nenhuma

característica específica), que a torne melhor do que outros objetos como elefantes, formigas ou carros. Sob o ponto de vista da Geometria, ela é um objeto que possui uma determinada posição, que tem noções de ângulo e distância relativa e que, ao se deslocar no plano, pode deixar um rastro de seu caminho.

Os comandos do Logo foram cuidadosamente selecionados para explorar este aspecto. Não é à toa que, para girar a tartaruga, o comando "paradireita" receba um ângulo como parâmetro. Se o interesse dos criadores do Logo fosse explorar conceitos de Biologia, a tartaruga teria, provavelmente, comandos como "nasça", "morra", "coma", "gaste energia", "acasale", etc.³¹.

No caso das engrenagens, por outro lado, os comandos biológicos da tartaruga não fariam sentido nenhum. Alguns objetos se prestam a uma maior variedade de significados do que outros. Tal como defendido pela Teoria da Atividade, quando um mesmo objeto se presta a diversos significados, a ferramenta pode ajudar a centrar a atenção em apenas alguns deles. No Logo Gráfico, foram as ferramentas (os comandos) que, restringindo o acesso que o aprendiz tinha, ajudaram a focar a atividade com a tartaruga no lado da Geometria e não no da Biologia.

É interessante notar como as dimensões sintática e semântica se complementam nesta questão do foco da atividade. Diferentes objetos possuem diferentes significados que lhe são atribuídos pela cultura. Estes significados representam as possibilidades do que um aprendiz poderia explorar. O que de fato será explorado depende das ferramentas disponíveis. Enquanto a dimensão semântica se preocupa com os múltiplos significados dos objetos, a sintática é que delimita o que poderá ser acessado e como.

Outro elemento que ajuda a focar em um ou outro aspecto a ser aprendido é o tema. Se a tartaruga gráfica tivesse comandos "geométricos" e "biológicos", um tema como "faça um ecossistema" exploraria conceitos muito diferentes do que se fosse pedido algo como "faça um desenho".

³¹ A Dynaturtle (PAPERT, 1993a, p.124) é um exemplo de tartaruga cujos comandos "acelere", "mude_direção" foram criados para focar a exploração dos alunos conceitos de Dinâmica. Da mesma forma e saindo um pouco do mundo dos software tradicionais construcionistas, os objetos do SimCity, um jogo no qual se constrói e simula cidades, permitem quase que exclusivamente a exploração de conceitos de urbanismo.

De fato, seguindo nesta linha de raciocínio, deveria haver uma definição mais explícita para termos como "Logo Gráfico" e "Logo Geométrico", que normalmente aparecem na literatura. Na nossa opinião, o Logo Geométrico poderia ser definido como um tipo de linguagem Logo que enfatiza os conceitos da geometria e o Logo Gráfico poderia ser associado a um tipo de Logo onde a tartaruga aparece na tela gráfica do computador. Estes termos não são mutualmente exclusivos, uma vez que se pode ter um Logo Geométrico Gráfico ou um Logo Geométrico Mecânico (com uma tartaruga mecânica que se arraste no chão ou na mesa). Do mesmo modo, também poderia existir um Logo Biológico Gráfico, onde a tartaruga gráfica ressaltasse os conceitos de Biologia; ou um Logo Mecânico Cinemático, onde os comandos dados a uma tartaruga robô mecânica ressaltassem conceitos de Cinemática.

e) Dimensão social, que aborda a integração do Logo às relações pessoais e à cultura do ambiente em que se encontra.

Embora defenda a idéia de que o meio possa influenciar o processo de aprendizado, a mera presença do computador, ou de outros materiais culturais não é capaz de fazer isso. Segundo Papert, "*O que pode acontecer é uma questão técnica. Mas o que irá acontecer é uma questão política, baseada em escolhas sociais*" (PAPERT, 1993a, p.29).

De fato, são os gostos e as decisões tomadas pela sociedade que vão fazer com que determinadas práticas sejam incentivadas ou não. De nada adianta, por exemplo, oferecer uma série de recursos para a criação de música se isto não for algo valorizado pela sociedade e pela cultura do ambiente.

Para que o aprendizado tenha maior probabilidade de sucesso, "*o educador tem que ser um antropologista*" (PAPERT, 1993a, p.32). Ele deve descobrir que materiais valorizados culturalmente são relevantes ao desenvolvimento intelectual e criar a atividade de aprendizado ao redor deles. Nesse sentido, a programação de computadores e o domínio da tecnologia em geral representam bons materiais a serem aproveitados, uma vez que são bem valorizados na sociedade atual. A questão é aproveitá-los de modo educacionalmente produtivo.

Além de um estímulo inicial para o aprendizado, a valorização e o reconhecimento social do trabalho do aprendiz adiciona um novo impulso para sua atividade (ajuda a tornar a atividade mais *ego syntonic* ou *cultural syntonic*). O aprendizado é algo que envolve situações inovadoras e às vezes frustrantes. Nos momentos em que o aprendiz já cansou de tentar, este apoio pode ser crucial.

Por outro lado, é muito importante que o reconhecimento seja verdadeiro. Se tudo o que o aprendiz fizer for elogiado pela comunidade, além de ele estar perdendo a oportunidade de se aperfeiçoar ainda mais, é provável que perca a confiança em todo o processo a que está se dedicando. A única maneira de resolver este problema de forma natural e não forçada é trabalhar com atividades autênticas, realmente integradas nas relações sociais e culturais da comunidade e oferecer *feedback* genuínos ao aprendiz. Em seu último livro, "*The Connected Family*" (PAPERT, 1996), Papert discute bastante esta questão da honestidade e do desenvolvimento dos valores morais na educação.

Outra situação em que a dimensão social se faz valer é no suporte intelectual. Através da troca de idéias e da discussão com amigos, professores e familiares, o aprendiz pode vislumbrar saídas para questões até então consideradas insolúveis. Nestes casos, "*o aprendiz pode usar todos estes elementos sociais como fonte de idéias, conhecimento ou problemas*" (VALENTE, 1993, p.35).

Entrando um pouco mais a fundo neste tópico, assim como a diversidade de materiais aumenta as chances de que o aprendiz se identifique com a atividade, a diversidade de experiência entre os membros da comunidade também influencia a atividade de aprendizado. Os que têm menos experiência aprendem com os que tem mais e os que têm mais refinam seus conhecimentos e habilidades, ajudando os que têm menos (FALBEL, 1993).

Novamente, esta é uma questão principalmente política. Para que este fluxo de experiência possa ocorrer em ambos os sentidos, a comunicação e a troca de idéias tem que ser permitida e incentivada tanto pelas pessoas como pelas ferramentas que forem utilizadas.

5.2.4.2 A evolução dos aspectos socioculturais do Construcionismo

Convém notar que, apesar de não termos encontrado nenhuma referência direta nos escritos de Papert sobre isso, os trabalhos mais recentes têm apresentado o Construcionismo como um conjunto teórico que reúne além das idéias de Piaget, influências do pensamento de Vygotsky e de Paulo Freire (VALENTE, 1992) onde a questão sociocultural é mais evidenciada.

De fato, diversos conceitos defendidos pelo Construcionismo, como a criação de atividades autênticas, a manipulação de objetos significativos, a visão do computador como uma ferramenta, a preocupação em se criar uma cultura de aprendizado, a integração de membros com diferentes níveis de experiência na comunidade, etc. parecem ser comuns a idéias destes autores.

Em especial, o aspecto social tem evoluído bastante nos últimos anos dentro do Construcionismo. Os primeiros trabalhos, das décadas de 70 e 80, realçam o social como fonte de inspiração e de motivação para o desenvolvimento do indivíduo. O contexto sociocultural era importante, mas o foco da pesquisa se concentrava mais em como ele influenciava os indivíduos do que na situação contrária. Nessa fase se enfatizava, principalmente, o potencial da programação do computador como um espelho da mente do aprendiz. O papel do computador como uma ferramenta de comunicação era praticamente ignorado.

Nos últimos anos, no entanto, provavelmente devido à crescente integração das telecomunicações com a informática, percebe-se que o foco das pesquisas construcionistas está mudando do individual para o comunitário. Começa-se a levantar as características do trabalho em equipe, a utilização de redes locais e da Internet e a formação de comunidades de aprendizado.

Em especial, Alan Shaw, em sua tese de doutorado (SHAW, 1995), introduz o conceito de "Construcionismo Social" (*Social Constructionism*). Enquanto o Construcionismo mais tradicional se preocupa com a criação de ambientes que favoreçam a aquisição de conhecimentos específicos para a solução de problemas individuais, o objeto de estudo do Construcionismo Social são as relações entre os indivíduos e os papéis que eles exercem dentro de suas comunidades.

No Construcionismo Social, ao invés do indivíduo, é a comunidade que se constrói. O contexto social não é visto como algo estático, dado a priori. Pelo contrário, ele é constantemente transformado pelas atividades de seus membros. A idéia central é criar, com auxílio da tecnologia, condições que favoreçam o intercâmbio de idéias e a tomada de consciência dos problemas sociais, facilitando assim o desenvolvimento da comunidade como um todo.

Shaw construiu um sistema de redes de computador muito fácil de usar, o MUSIC (SHAW, 1995), para incentivar a troca de mensagens e o compartilhamento de

informações entre os membros de um bairro pobre dos arredores de Boston. A interface do MUSIC (ver Figura 10) é essencialmente gráfica e sonora, abrindo espaço para que até os analfabetos pudessem usufruir de suas potencialidades. Outra característica interessante do MUSIC é que ele funciona em computadores bastante simples, que estão de acordo com o poder aquisitivo de pelo menos alguns dos membros da comunidade. Os membros que não tinham computador reuniam-se na casa dos que tinham para poder acessar o sistema.

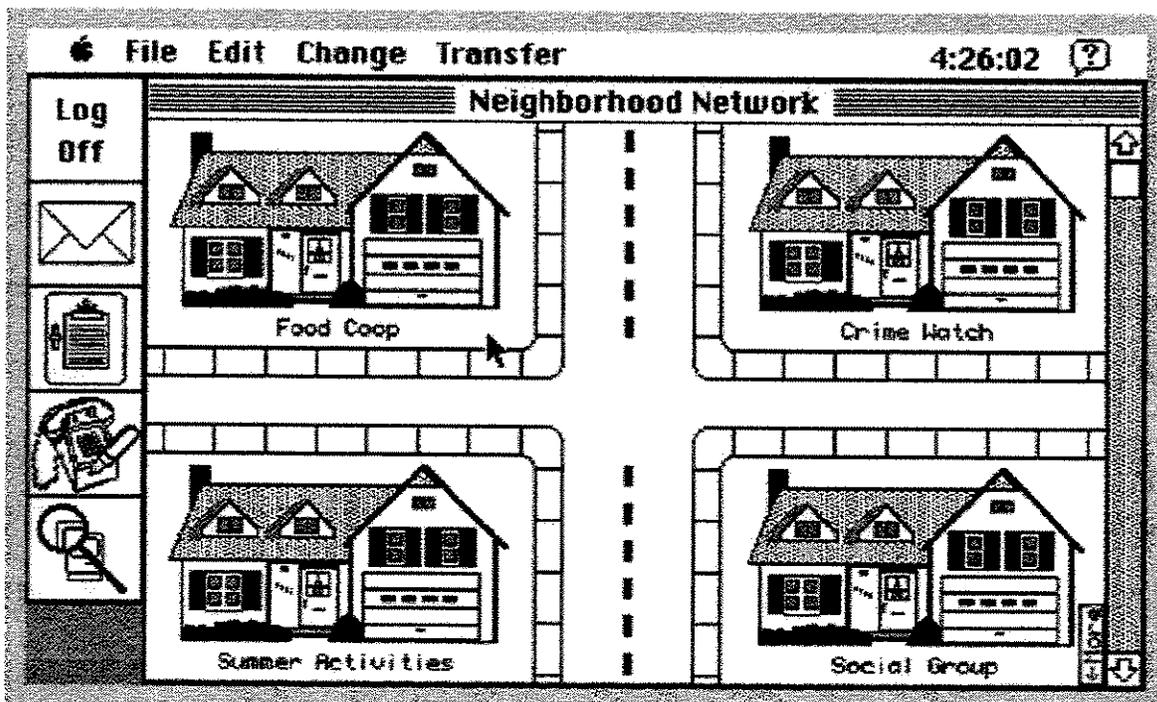


Figura 10 - Uma das telas do MUSIC

Por fim, ainda visando integrar a comunidade, o MUSIC foi desenvolvido para ser gerenciado pelos próprios membros da comunidade. A própria vontade de manter o sistema funcionando servia de pretexto para a ação social dos moradores do bairro.

Shaw também defende, a partir do conceito de "síntese cultural" de Paulo Freire, que as ferramentas usadas não deveriam trazer dentro de si nenhuma ideologia que fosse externa à comunidade. Pelo contrário, a ferramenta deveria permitir com que seus usuários desenvolvessem os temas que lhes fossem interessantes com um mínimo de intervenção.

Já a tese de Amy Bruckman (BRUCKMAN, 1997) deixa de lado a questão das comunidades locais tratadas por Alan Shaw e aborda o tema das comunidades virtuais. Bruckman desenvolveu o MOOSE Crossing (ver Figura 11), um software em que, através da Internet, crianças e adultos podem construir espaços virtuais para que outros possam visitar. Estes espaços podem ser quartos com móveis e animais, navios, ilhas desertas, estações espaciais e o que mais for desejado. Estes espaços também podem ser 'decorados' com qualquer objeto que a pessoa construir e o interessante é justamente isso. São os próprios usuários que constroem tudo e definem o que pode ou não ser feito com seus objetos.

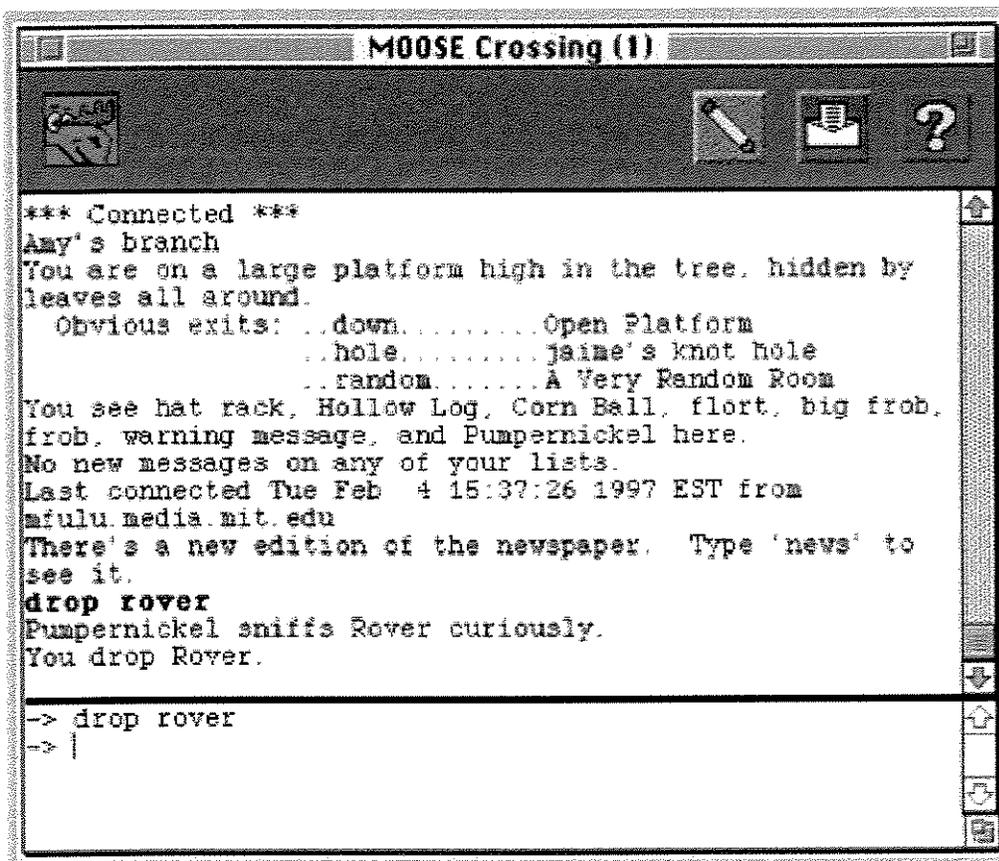


Figura 11 – Uma tela do MOOSE Crossing

Por exemplo, um usuário poderia criar uma sala com uma televisão e um sofá e definir operações como “ligar”, “desligar” e “mudar de canal” para a televisão e “sentar” e “levantar” para o sofá. Também poderia criar uma porta que levasse a um outro espaço criado por algum colega. Talvez uma piscina onde se pudesse ‘nadar’ ou ‘tomar sol’.

O MOOSE Crossing funciona ao redor de uma versão da linguagem Logo alterada especificamente para este sistema. O software tem sido utilizado nas mais diversas áreas, desde em escolas primárias para o desenvolvimento de projetos em grupo, até pela comunidade universitária para a discussão de novas idéias.

Na nossa opinião, é interessante notar que, tanto no MUSIC quanto no MOOSE Crossing, a ênfase da ferramenta (e das atividades que são descritas com ela) se dá, principalmente, em colocar os usuários como construtores de coisas que lhes sejam significativas (respectivamente, a comunidade onde vivem ou a comunidade virtual) e como membros de um grupo maior com o qual se identifiquem e se sintam reconhecidos.

Este último é um aspecto importante que, até então, não era tão evidenciado pela teoria. Por outro lado, outras características defendidas pelo Construcionismo, como a visualização das estratégias adotadas na resolução de problemas, acabaram sendo relegadas a um segundo plano, principalmente no MUSIC. Neste caso, o tipo

de reflexão dependerá basicamente do tipo de dinâmica feita fora do computador com os participantes da atividade.

5.3 A educação construcionista

Enquanto os princípios do aprendizado construcionista descrevem características gerais do aprendizado humano, a educação construcionista baseia-se na aplicação destes princípios para construir situações concretas que propiciem o aprendizado de conteúdos específicos.

Sintetizando o que foi visto na seção anterior, o Construcionismo é contrário ao modelo tradicional escolar que considera o professor como fonte ativa de conhecimento e o aluno como receptor passivo. Na educação construcionista, o educador não ensina; ele constrói micromundos, ou ambientes propícios para o aprendizado onde

- os novos conceitos sejam aprendidos através da construção de produtos significativos e de utilidade imediata para os aprendizes;
- os aprendizes possam explicitar idéias, executá-las e refletir sobre os conceitos e as estratégias adotadas;
- sejam usados materiais que, além de familiares e atraentes para o aprendiz, sejam representativos dos conceitos fundamentais do tópico sendo estudado;
- estes fundamentos sejam enfatizados por um tema e estejam ao alcance de serem manipulados direta ou indiretamente pelo aprendiz;
- o aprendiz possa seguir seu próprio ritmo e estilo; e
- onde o trabalho seja compartilhado, reconhecido e incentivado, genuinamente, pela comunidade.

De uma forma simplificada, o ideal construcionista de um ambiente de aprendizagem é, segundo Papert (1977, p.273; 1993a), muito semelhante a uma escola de samba: as pessoas se reúnem, espontaneamente, em torno de um objetivo comum (o desfile do ano seguinte); o tema sendo desenvolvido está diretamente relacionado com a vida dos participantes (folclore, história ou fatos marcantes); especialistas e novatos se misturam amigavelmente em um mesmo ambiente, trocando experiências de acordo com a necessidade de cada um.

As atividades construcionistas em torno do Logo Geométrico são um bom exemplo disso. Normalmente, pede-se que os aprendizes desenvolvam projetos sobre um tema comum e interessante para eles, como fazer uma animação ou um desenho na tela do computador.

Ao longo dos projetos eles são convidados a escrever suas idéias e seus problemas diários nos chamados "**Diários de Bordo**" – uma prática que incentiva a reflexão e que será descrita com mais detalhes nos capítulos 7, 8 e 9.

Nos ambientes construcionistas, os aprendizes são livres para conversar entre si, trocar idéias com os amigos e seguir seu próprio estilo de aprendizado. O professor assume a postura de um facilitador, isto é, alguém que dá as orientações gerais da atividade e ajuda a resolver os problemas específicos que cada aprendiz for encontrando no correr de seu projeto.

Embora não seja essencial em todas as situações, os construcionistas acreditam que a tecnologia tem um papel fundamental na viabilização e na implementação de ambientes deste tipo, principalmente quando o objetivo é abordar conceitos difíceis de serem visualizados nas situações normais. Nesse sentido, além do Logo Gráfico, foram criadas ferramentas como o LEGO-Logo, que explora conceitos de cibernética, mecânica e robótica; o StarLogo, que trata de sistemas descentralizados; o MOOSE-Crossing; e outros (RESNICK et al., 1996).

Mais especificamente, conforme discutido nos princípios do aprendizado construcionista, acredita-se que o computador tem a característica especial de poder ajudar bastante na visualização e no conseqüente aprimoramento das estratégias mentais utilizados pelos aprendizes no correr de suas construções.

O estudo das ferramentas e suas aplicações levanta uma série de novos pontos e conclusões que irão sendo incorporados ao Construcionismo na medida em que forem comprovados. Alternando-se entre teoria e prática, é assim que o Construcionismo vai se desenvolvendo.



6 O desenvolvimento de software para atividades educacionais

Este capítulo propõe uma nova abordagem para o desenvolvimento de software para educação. Ele parte de critérios levantados da discussão sobre o contexto educacional e tecnologia efetuada no capítulo 2 e utiliza os conceitos da Teoria da Atividade apresentados nos capítulos 3 e 4 para identificar os aspectos a serem considerados.

A seção 6.1 discute os princípios gerais a serem seguidos por uma abordagem de desenvolvimento de software para educação e mostra os benefícios de se buscar uma abordagem baseada na Teoria da Atividade. A seção 6.2 apresenta as fases da nova abordagem e faz uma breve comparação com as fases das abordagens mais tradicionais. As seções 6.3, 6.4 e 6.5 discutem cada uma das fases propostas com maiores detalhes.

Em especial, a seção 6.4.1 (“Artefatos computacionais, artefatos digitais, objetos digitais e aplicativos”) revê os conceitos de computador propostos pela Teoria da Atividade (ver capítulo 4) e propõe novas definições mais adequadas para a educação.

6.1 Princípios para uma metodologia

A discussão sobre eficácia, eficiência e viabilidade efetuada no capítulo 2 permite levantar uma série de pontos a serem observados por uma abordagem de desenvolvimento de tecnologia para a educação.

Do aspecto **eficácia**, por exemplo, torna-se claro que o desenvolvimento da nova tecnologia deve:

- ser guiado por uma teoria educacional que indique os pontos a serem enfatizados e sustente as escolhas feitas durante a implementação;
- envolver, no processo, educadores, estudantes e outros cuja experiência possa ajudar a complementar as idéias teóricas;
- facilitar, na medida do possível, o intercâmbio de idéias entre os desenvolvedores e o público atendido pela tecnologia;
- efetuar testes práticos em situações próximas das idealizadas para assegurar que as expectativas estejam, de fato, sendo atingidas.

O aspecto **eficiência** exige que se explicitem as contribuições e limitações específicas que a tecnologia, no caso, o software, trará para a situação educacional almejada. É

a identificação destes elementos que permitirá analisar a tecnologia escolhida frente as outras opções.

Por fim, o aspecto **viabilidade** ressalta que, além das questões levantadas acima, devem ser considerados os custos técnicos de desenvolvimento e, principalmente, os de implantação e manutenção da nova tecnologia. Estes custos envolvem, dentre outras coisas:

- uma análise detalhada da situação sociocultural vigente e do esforço necessário para a utilização efetiva e eficiente da nova tecnologia no ambiente para o qual ela se destina;
- uma análise do tipo de mercado do qual a tecnologia fará parte;
- um investimento em recursos técnicos, isto é, pessoal especializado, equipamentos, infra-estrutura, etc.;
- um investimento em implantação, incluindo a preparação de material de suporte, a formação e treinamento dos usuários e a implementação das demais mudanças que se mostrarem necessárias ao longo do tempo;
- um investimento em manutenção, considerando material de reposição, equipe de suporte, implementação de atualizações e correções, etc.

Também, conforme apresentado nos capítulos 2 e 3, há uma corrente crescente dentro da Engenharia de Software que acredita que uma abordagem baseada na Teoria da Atividade pode trazer uma série de contribuições para o desenvolvimento de software.

Por que a Teoria da Atividade. De fato, na nossa opinião, em se tratando de educação, acreditamos que o uso de algumas idéias da Teoria da Atividade pode ajudar no tratamento de vários dos pontos levantados acima. Mais especificamente, acreditamos que, sob o aspecto de eficácia:

- o tipo de análise proposto pela Teoria da Atividade permitirá incluir, no desenvolvimento da tecnologia, os elementos socioculturais propostos pelas teorias educacionais em voga atualmente;
- o dinamismo da estrutura e as observações relativas à motivação, desenvolvimento e história da atividade também parecem adequados para descrever processos educacionais;
- a Teoria da Atividade reúne, dentro de uma mesma estrutura, um corpo de pesquisa multidisciplinar bastante abrangente. Espera-se que, descrevendo-se a problemática educacional e tecnológica nos termos da Teoria da Atividade, as áreas envolvidas tenham maior possibilidade de interagir e de aproveitar os conhecimentos e a experiência de outras áreas correlatas.

No que se refere à eficiência, ainda que não muitos, a Teoria da Atividade possui diversos trabalhos sobre o uso do computador na educação. Embora estes trabalhos não tragam comparações explícitas do computador com outros artefatos, as características gerais levantadas pela Teoria da Atividade, principalmente no que

se refere à mediação e à interdependência artefato-atividade, parecem ser de grande utilidade para o tipo de análise que se deseja.

As considerações contextuais descritas acima também contribuem para o aspecto da viabilidade uma vez que, como foi visto, boa parte do investimento no desenvolvimento da tecnologia recairá sobre as transformações socioculturais que ela exigirá para obter bons resultados.

Por outro lado, apesar do caráter otimista demonstrado acima, não acreditamos que a Teoria da Atividade seja uma panacéia para o desenvolvimento de software para a educação. Ela parece ser muito indicada para o lado "mais humano" do desenvolvimento, isto é, para a parte que deverá lidar com motivação e relações humanas. Com relação ao lado mais técnico, que envolve lidar com estruturas de dados, testes de qualidade, otimização de processamento, etc., provavelmente a solução mais indicada virá de uma combinação com outras metodologias de engenharia já existentes.

De uma forma geral, não há discussões na Teoria da Atividade sobre como deve ser feito o desenvolvimento de software para a educação. A seção a seguir apresenta nossas idéias com relação a este assunto.

6.2 A proposta de uma nova abordagem

Tradicionalmente, de uma forma genérica, o desenvolvimento de software transcorre ao longo de um ciclo de vida composto pelas fases de análise do sistema e do software, projeto do software, implementação do software, teste do software e manutenção do software (ver Figura 12):

- A fase de **análise do sistema** descreve como será o ambiente no qual o software estará inserido.
- A fase de **análise do software** tem por objetivo especificar o que o software deverá fazer.
- A fase de **projeto do software** se preocupa em determinar como ele deverá cumprir com seus objetivos dadas as restrições impostas pelo contexto de desenvolvimento e uso.
- A **implementação do software** converte o projeto em um programa executável pelo computador.
- O **teste do software** verifica se o que foi implementado funciona de acordo com o especificado na análise.
- A **manutenção** cuida da instalação do software e de todas as correções e pequenas modificações que ele sofrerá até o dia em que for substituído ou desativado.

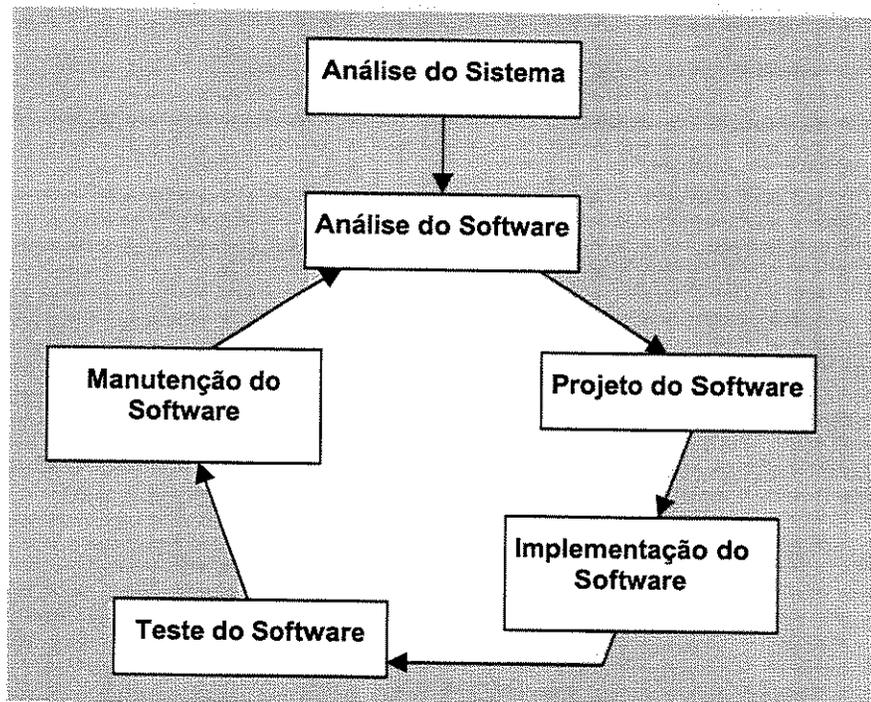


Figura 12 - O ciclo de vida tradicional de um software

Tradicionalmente, a análise do software parte de uma análise anterior, mais ampla e estratégica de todo o contexto no qual o software será utilizado. Em muitos casos, esta análise do sistema maior é tida, pelos desenvolvedores, como algo já definido e preparado de antemão. Tanto é assim que, grande quantidade de cursos e metodologias de Engenharia de Software se iniciam diretamente na análise do software. A análise do sistema, que envolve, além do aspecto tecnológico, a organização do pessoal e dos materiais, é tida como um tópico tratado principalmente por administradores de empresas.

Uma vez identificado o papel do software dentro do sistema, parte-se para o seu projeto, implementação, teste e manutenção, que é quando ele será implantado e manipulado por seus usuários.

Apesar do ciclo, as fases do desenvolvimento de um software não são, necessariamente, seqüenciais. Às vezes, elas podem ocorrer em paralelo ou em tempos diferentes, em partes distintas do software. Por exemplo, ao mesmo tempo que parte do programa pode estar sendo testada, outras partes podem estar sendo implementadas e novas idéias podem estar sendo incorporadas ao projeto.

Como é praticamente impossível prever-se, na fase de análise, todos os detalhes do futuro sistema – muitos pontos vão sendo esclarecidos durante o desenvolvimento – o ciclo de fases pode repetir-se diversas vezes entre análise e teste até que o software seja considerado aceitável. Em especial, isto é bastante comum nas metodologias baseadas no Design Centrado no Usuário (ver capítulo 2). Nestes casos, o próprio planejamento do desenvolvimento do software prevê versões cada vez mais sofisticadas de protótipos a serem testadas junto a futuros usuários e em situações próximas das esperadas.

Ainda assim, como o custo de desenvolvimento cresce consideravelmente na medida em que os problemas se propagam de uma fase para a outra, é aconselhável que boa parte do investimento recaia sobre as fases de análise e na resolução das possíveis dificuldades, tão rápido quanto possível. Nielsen (NIELSEN, 1993) recomenda, inclusive, que sejam feitos, antes da análise do software propriamente dita, estudos de caso em cima de produtos similares aos que se deseja construir.

No caso da abordagem aqui proposta, mais do que uma metodologia completa que guie todos os passos, procurou-se levantar apenas os principais conceitos e idéias a serem considerados no desenvolvimento de software para a educação. Em linhas gerais, a nova abordagem é uma compilação dos conceitos da Teoria da Atividade apresentados nos capítulos anteriores e enfatiza, principalmente, a contemplação do lado sociocultural da atividade educacional e o tratamento do artefato computacional (hardware e software) como uma unidade ao longo de todo o ciclo de vida.

Na nossa opinião, é muito importante que o desenvolvimento da tecnologia esteja intimamente amarrado ao desenvolvimento do contexto no qual ela será utilizada. Nos termos da Teoria da Atividade, as novas ferramentas surgem das necessidades percebidas no andamento de uma atividade. Quando esta ferramenta passa a ser utilizada, a atividade como um todo é afetada e novas necessidades tendem a surgir, incentivando o aprimoramento e desenvolvimento de mais ferramentas, formando um ciclo que se repete continuamente.

Nas metodologias tradicionais de desenvolvimento de software, embora esta interatividade também acabe ocorrendo, a relação bidirecional entre o sistema e o software não é tão enfatizada. Não se vê claramente como a utilização da nova ferramenta acaba interferindo no sistema maior do qual ela fará parte. É por isso que a Figura 12 apresenta a análise do sistema como uma parte mais isolada e externa ao ciclo do software.

No caso da educação, a consideração do todo é fundamental. Conforme demonstram inúmeros estudos (ver capítulo 1), o impacto da tecnologia causa uma série de receios e abre uma diversidade de possibilidades nas práticas tradicionais que merecem ser analisados constantemente. Por esta razão, como será visto adiante, a abordagem proposta considera a análise da atividade educacional como uma fase integrante do ciclo de desenvolvimento.

Outro aspecto enfatizado na nova abordagem é o de que, embora as metodologias mais tradicionais costumem separar hardware e software logo no início do desenvolvimento, acreditamos que a tecnologia moderna já tenha avançado o suficiente para viabilizar a criação de hardware específico e mais adequado às atividades desejadas. De fato, além das considerações feitas no capítulo 5 de que os usuários vêem o computador como uma unidade, a evolução da computação móvel e distribuída, que se baseia em *notebooks*, *palmtops* e outros pequenos artefatos (NORMAN, 1998), inclusive com formato de brinquedos (UMASHI, 1997), tem aberto uma série de novas possibilidades educacionais que merecem ser melhor exploradas (SOLOWAY et al., 1999).

Dado o seu carácter essencialmente sociocultural, as idéias aqui apresentadas devem ser vistas não como uma negação, mas sim como um complemento de outras abordagens mais técnicas de desenvolvimento de software. Assim sendo, recomenda-se um ciclo de desenvolvimento composto pelas seguintes fases (ver Figura 13):

- **Análise da Atividade Educacional**, que descreve as dimensões estruturais, funcionais, sociais, históricas e culturais da atividade educacional na qual o software estará inserido.
- **Análise do Artefato Computacional**, um refinamento da Análise da Atividade que se preocupa em determinar o papel que o computador terá dentro da atividade educacional. Normalmente, nesta fase também são melhor identificadas as restrições de hardware e software que poderão afetar o desenvolvimento do artefato computacional.
- **Projeto do Artefato Computacional**, onde se propõe uma solução para a implementação do artefato frente às restrições impostas pelo ambiente real no qual ele será utilizado. Similar ao que acontece no ciclo de vida tradicional, enquanto a Análise do Artefato Computacional indica o que a combinação hardware-software deverá fazer, a fase de Projeto preocupa-se em especificar o como isso deverá ser feito.
- **Implementação do Artefato Computacional**, é onde o software é codificado e o hardware, quando necessário, montado.
- **Teste do Artefato Computacional**, onde o artefato computacional é testado frente à atividade em que ele será usado.
- **Manutenção do Artefato Computacional**, onde são feitas as devidas atualizações e correções no artefato e nos demais elementos da atividade educacional em si.

Como este ciclo é iterativo, a fase de manutenção pode ser entendida quase que como uma fase perpétua que engloba todo o ciclo seguinte de desenvolvimento do artefato. É nela que os novos atributos vão sendo analisados, projetados, implementados e testados.

A idéia de uma atividade de desenvolvimento em constante evolução implica que todos os produtos gerados, como as descrições da atividade e do artefato computacional resultantes da análise, devam ser vistos como dinâmicos e passíveis de modificações. Este é um aspecto importante a ser considerado no modo com que as documentações e mesmo o código do software poderá ser acessado. Por outro lado, a forma específica com que isto deverá ser feito – se as descrições deverão estar em linguagem natural, esquemas ou algum outro tipo de formalismo; se deverão ser manipuladas por ferramentas especiais, etc. – fogem do escopo deste trabalho.

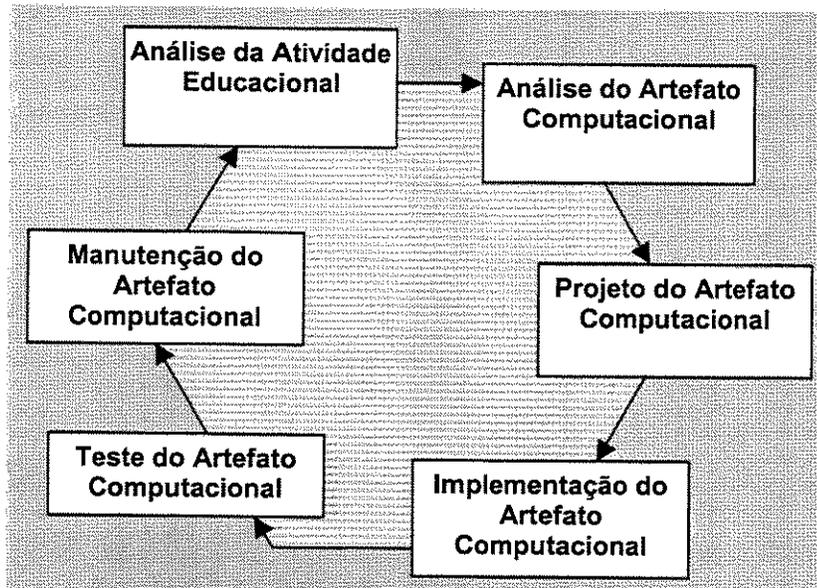


Figura 13 - O ciclo de vida baseado na análise da atividade educacional

Convém notar que, embora no ciclo proposto acima apenas a relação entre a análise da atividade educacional e a análise do artefato computacional esteja sendo delineada, cada uma das demais fases também deve ser vista como pertencendo a uma fase maior, que trata da atividade como um todo. Deste modo, a fase de projeto do artefato computacional deve ser considerada como parte integrante do projeto da atividade educacional da qual faz parte; a fase de implementação do artefato deve ser vista como parte da implementação da atividade e assim por diante.

Para efeitos deste trabalho, concentrou-se esforços nas fases de Análise da Atividade Educacional e na fase de Análise do Artefato Computacional frente a esta atividade. Conforme mencionado anteriormente, é nestas fases que se dá a principal contribuição da Teoria da Atividade com relação as outras abordagens de desenvolvimento de software. Apesar disso, ao final do capítulo reuniu-se diversos pontos relacionados as outras fases do ciclo.

6.3 A Análise da Atividade Educacional

A função da Análise da Atividade Educacional é descrever, da forma mais completa possível, a atividade educacional desejada em todas as dimensões propostas pela Teoria da Atividade, realçando os aspectos sócio-histórico-culturais que deverão ser alcançados e obedecidos.

Além destes aspectos humanos (como motivação, desenvolvimento e interação social), que raramente são considerados nas metodologias mais tradicionais da Engenharia de Software, a abordagem proposta recomenda que a Análise da

Atividade também descreva (ou referencie) os princípios teóricos que orientam e dão embasamento à atividade educacional sendo criada.

A atividade educacional genérica e a específica. Como uma mesma teoria educacional pode servir de base para a implementação de uma série de atividades que se diferenciam umas das outras apenas por atributos específicos (como o tema a ser abordado, a quantidade de participantes, a faixa etária, disponibilidade de recursos, quantidade e duração das sessões, etc.), recomenda-se que a Análise da Atividade seja dividida em duas partes.

A primeira, genérica, baseada nos princípios da teoria educacional, descreveria uma atividade ideal, livre de qualquer restrição e indicaria como possíveis restrições deveriam ser tratadas. A segunda parte da análise, mais específica, baseada na genérica, identificaria todas as restrições específicas impostas pelo ambiente em que a atividade educacional deverá ser implementada (casa, escola, museu, empresa, etc.) ou pelas características do domínio educacional que ela deverá explorar (ver Figura 14).

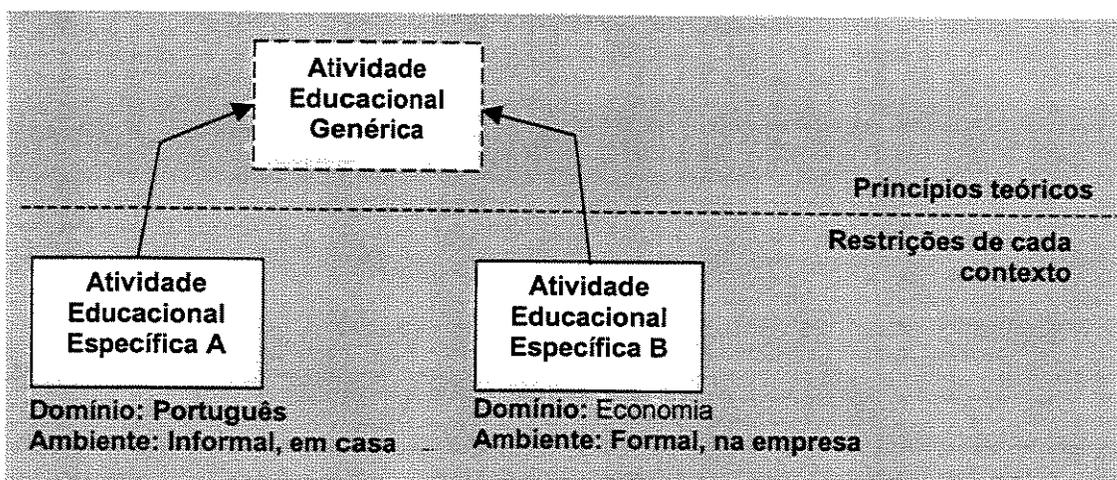


Figura 14 - A descrição genérica e a descrição específica da atividade educacional

A distinção entre a parte genérica da atividade e a parte específica é importante para a compreensão histórica da atividade. Para quem olha de fora, é muito difícil diferenciar o que foi de fato recomendado pela teoria educacional e o que foi imposto por restrições momentâneas. Por exemplo, quando se assume o tempo escolar baseado em aulas de 50 minutos, ou quando se divide uma classe de 30 alunos, em 3 turmas de 10, para ir mexer nos computadores da escola, até que ponto isto é recomendável sob a ótica pedagógica? Não seriam estas restrições impostas por fatores históricos?

É importante que ambas as partes da análise da atividade sejam descritas de acordo com a terminologia e a estruturação propostas pela Teoria da Atividade – para assegurar que elas possam ser aproveitadas pelas diferentes áreas do conhecimento abrangidas pela Teoria – e que elas sejam específicas o suficiente a

fim de sustentarem as decisões tomadas ao longo do desenvolvimento da atividade educacional e poderem ser comprovadas ao seu término.

Idealmente, a parte genérica da análise da atividade educacional poderia constituir um documento independente que pudesse ser simplesmente referenciado pelas análises específicas de atividades educacionais que nele se baseassem. No caso do Construcionismo, por exemplo, a descrição da atividade genérica deveria apresentar os fundamentos da educação construcionista. Diferentes atividades construcionistas, que usassem ou não o Logo, que ocorressem dentro ou fora do ambiente escolar, que trabalhassem diferentes domínios educacionais, etc., fariam menção à descrição genérica e explicitariam os pontos que não poderiam ser obedecidos ou que deveriam receber atenção por parte dos desenvolvedores.

Abaixo foram listadas algumas perguntas que, na nossa opinião, devem ser consideradas na Análise da Atividade Educacional. Tal como ocorre na *Activity Checklist* descrita no capítulo 3 (KAPTELININ e NARDI, 1997; KAPTELININ et al., 1999), a intenção destas perguntas é orientar o processo de desenvolvimento servindo de base, inicialmente, para traçar um panorama geral dos pontos a serem considerados e, posteriormente, para um aprofundamento daqueles que forem considerados mais importantes.

Diferente do que ocorre com a *Activity Checklist*, no entanto, a organização proposta neste trabalho assume que, antes de se analisar o uso da tecnologia frente à atividade, é preciso descrever a atividade de uma forma geral que dê, inclusive, subsídios para se avaliar até que ponto a tecnologia escolhida seria a mais apropriada para a atividade em questão (eficiência). Na *Activity Checklist*, parte-se do princípio que o computador já tem um lugar determinado na atividade.

Além disso, convém notar que a *Activity Checklist* é genérica, não tendo sido criada para o caso específico da educação. Assim como ocorre em outros artigos da Teoria da Atividade, ela também não parte de uma definição mais explícita do que seria o computador, tal como será discutido mais adiante na seção sobre a análise do artefato computacional.

6.3.1 Sobre a estrutura hierárquica da atividade

- Qual o motivo da atividade educacional?
- Que necessidade ou desejo a motiva? Qual o domínio educacional a ser abordado? Por quê?
- Quem são os sujeitos da atividade?

Na resposta a esta última pergunta espera-se uma descrição geral do tipo de pessoas participarão da atividade. Deverão ser descritas apenas as características relevantes à atividade educacional em questão como, por exemplo, o nível cognitivo, o grau de conhecimento que os participantes já possuem do domínio educacional e dos artefatos que poderão ser utilizados, a familiaridade que possuem com relação ao ambiente, que tipo de expectativas eles têm, etc.

- Qual a divisão do trabalho?

Aqui devem ser identificados os diferentes papéis que os sujeitos podem exercer na atividade. Dependendo da teoria ou do contexto, cada atividade implicará em uma série de papéis diferentes, cada um com suas funções e limitações. Em uma atividade educacional instrucionista, por exemplo, poderia ser descrito o papel do instrutor, que é aquele que transmite os conhecimentos e o papel do aprendiz, que é quem recebe estes conhecimentos³². Na atividade educacional construcionista, como será aprofundado mais adiante, há, dentre outros, o papel do aprendiz, que é visto como o construtor de seu próprio conhecimento e o do facilitador, que procura facilitar este processo de construção do aprendiz.

De qualquer forma, independente da abordagem educacional, recomenda-se, tal como descrito no capítulo 4, que toda análise de atividade considere, além dos seus papéis específicos, os papéis genéricos de “designer da atividade”, que é responsável pelas atualizações e correções da atividade e de “ator”, que apresenta sugestões e reporta problemas.

De uma forma geral, sob a ótica do designer, todos os demais papéis da atividade podem ser considerados como atores, cada um com suas características e demandas específicas. A idéia de facilitar a comunicação entre atores e designers e considerá-la como algo constante ao longo de todo o ciclo de vida da atividade é algo muito enfatizado pela abordagem aqui proposta. É este mecanismo que assegura a re-análise contínua da atividade e do artefato, caracterizando o desenvolvimento em ciclos.

Dependendo da complexidade da atividade, pode ser interessante listar os grupos de sujeitos da atividade e relacioná-los aos diferentes papéis que poderão assumir. Por exemplo, em alguns casos, um mesmo aluno pode atuar como aprendiz ou como fonte de referência para outros.

É interessante notar que, embora diversas metodologias de desenvolvimento trabalhem com a idéia de que a interação entre usuários e software possa ser modelada pela descrição de papéis, o conceito de que um mesmo usuário pode ficar se alternando de um papel para outro na mesma atividade parece ser algo levantado pela Teoria da Atividade. Este dado é importante, uma vez que, ao invés de simplesmente ajudar na execução dos diferentes papéis, o software também deverá auxiliar o usuário a mudar de papel, e isso sem que ele perca o foco do que deseja fazer.

- Quais as principais ações em cada papel? Quais suas metas?

Na realidade, cada papel da atividade define uma sub-atividade com seus objetivos próprios, ações e operações. O ideal, na análise da atividade educacional, seria que cada uma destas sub-atividades fosse detalhada. Por exemplo, no caso do Instrucionismo, um instrutor tem como objetivo transmitir conhecimento para os aprendizes e, para isso, utiliza uma série de abordagens e recursos didáticos. Como

³² No Instrucionismo, diferente do Construcionismo, o conhecimento é visto como algo que pode ser transmitido de uma pessoa (um professor) para outra (um aluno).

ações, ele deve preparar suas aulas, transmitir os conteúdos e avaliar o grau com que estes conteúdos foram absorvidos pelos aprendizes.

Já coordenadores pedagógicos e aprendizes, outros papéis da mesma atividade educacional, teriam outros objetivos e ações. O importante é mostrar a integração e dependência entre estas sub-atividades na formação da atividade educacional.

- Como comprovar se as metas foram atingidas?

Idealmente, cada item especificado na análise deveria ser passível de uma avaliação que ajudasse a comprovar o quanto ele foi ou não implementado na atividade.

Vale notar que este tipo de avaliação aqui descrita indica, simplesmente, até que ponto a atividade educacional criada está de acordo com a especificada, independente de sua qualidade. Esta avaliação não mede a efetividade educacional da atividade, algo que deveria indicar o grau de aprendizado gerado, a adequação da tecnologia ao ambiente, etc.

- Quais as maneiras mais comuns ou quais as maneiras permitidas para se atingir estas metas?

Esta questão está diretamente relacionada com o grau de flexibilidade que a atividade dará aos seus participantes. No caso do Construcionismo, ela se refere, dentre outras coisas, aos estilos de aprendizado.

- Quais os principais tipos de conflitos que podem surgir entre os papéis ou metas da atividade?

Em alguns casos, a execução de um papel precisa de informações sobre um outro papel relacionado e isso pode gerar em conflito. Por exemplo, professores precisam aplicar provas para avaliar como andam os seus alunos. Os alunos têm que parar o fluxo de suas ações diárias para fazer esta prova e não veem como isto pode contribuir para seu aprendizado. Este é um tipo de conflito típico em uma atividade educacional.

- Há situações em que os sujeitos competem pelo mesmo objeto? Há situações em que os objetivos de uns vão de encontro aos objetivos de outros?

Estes casos são mais comuns em atividades educacionais de caráter competitivo, como campeonatos esportivos, gincanas ou exames de seleção, onde um indivíduo ou equipe é colocado contra o outro.

- Há situações em que um mesmo sujeito deve executar mais do que um papel simultaneamente?

Normalmente, isso não deve ocorrer. Quando acontece, gera um conflito que precisa ser resolvido.

- Há papéis que podem ou devem ser efetuados por mais de um sujeito ao mesmo tempo?

Esta situação ocorre, principalmente, em atividades onde os sujeitos são organizados em equipes, como trabalhos em grupo.

- Há papéis que dependem de outros papéis para serem executados? Quais?

Como será visto no capítulo 7, a atividade principal da educação construcionista é a do aprendizado como construção do conhecimento e todos os papéis foram de certa forma criados com relação ao do aprendiz. A atividade do facilitador é totalmente dirigida para a do aprendiz.

- Quais as operações mais comuns encontradas em cada ação?

Como discutido no capítulo 3, uma vez que as operações só podem ser descritas à posteriori, esta é, tipicamente, uma pergunta que só poderá ser respondida pela experiência dos desenvolvedores e pela da análise de situações práticas similares à desejada.

- Que tipo de ações devem ser operacionalizadas? Que tipo de operações devem ser conceitualizadas?

Quais ações devem ser conscientes para os diferentes papéis? Tal como discutido no capítulo 4, todas as ações voltadas para os aspectos físicos e de manuseio do artefato devem ser operacionalizadas, assim como todos os aspectos do domínio educacional que não forem relevantes para a atividade em questão. Os demais aspectos deverão ser conceitualizados e, mesmo assim, de acordo com o nível de capacitação em que o sujeito se encontrar.

Por exemplo, se o objetivo de uma atividade educacional for estudar o movimento planetário, idealmente os aprendizes não devem ser distraídos por cores chamativas, nomes ou outros dados que não forem relevantes sobre os planetas.

6.3.2 Sobre a orientação a objetos

- Qual o objetivo (objeto por trás do motivo) da atividade?

Por exemplo, em uma educação baseada em projetos, a construção de um elevador pode ser usada para explorar conceitos de física. Neste caso, o aprendizado dos conceitos de física seria o motivo da atividade e o elevador, o seu objetivo.

- Quais os objetos manipulados por cada papel? Quais os artefatos usados na manipulação destes objetos?
- Que propriedades destes objetos são relevantes para cada ação? Por quê?
- Com quem (ou com que outros papéis) o sujeito poderá comunicar-se em cada papel? Através de que artefatos?
- Quais as características relevantes destas comunicações?

Estes aspectos realçam que tipos de propriedades dos objetos (ou da forma de manipulação destes objetos) e dos demais sujeitos da atividade são educacionalmente relevantes para cada papel. Como discutido no capítulo 5, no Logo Gráfico, por exemplo, apenas as características relacionadas com a

matemática eram mais importantes. Os demais atributos da tartaruga foram mascarados.

Nas comunicações com outros participantes da atividade, é importante especificar, por exemplo, se elas ocorrem de forma síncrona ou assíncrona, se elas exigem algum tipo de mídia específico, etc.. Por exemplo, sob o olhar da Teoria da Atividade, o autor de um livro pode ser considerado como um membro da comunidade da atividade. O livro, no caso, é um meio de comunicação assíncrono e unidirecional, isto é, o momento da leitura é relativamente independente do momento em que o livro foi escrito e os leitores não têm como opinar, via livro, para o autor. Do mesmo modo, o telefone seria um meio síncrono e bidirecional, o email seria um meio assíncrono e bidirecional.

Como é o ambiente em que se dará a atividade? Como estarão dispostos física e temporalmente os sujeitos e os objetos? Em que quantidade e por quanto tempo eles estarão disponíveis uns para os outros?

Vale notar que, para os engenheiros de software, freqüentemente a palavra “ambiente” é associada única e exclusivamente ao conjunto de ferramentas (compiladores, editores, navegadores, depuradores, etc.) disponibilizado no computador para o desenvolvimento de uma determinada tarefa. Esta idéia difere radicalmente do conceito de ambiente aqui defendido, que também inclui todos os elementos socioculturais relacionados com esta tarefa.

- Que objetos deverão ou poderão ser compartilhados? Por que papéis? Em que situações?
- Quais as principais mudanças de foco em cada papel?

Em escolas públicas, normalmente a quantidade de computadores é menor do a de alunos, o que força uma dinâmica de compartilhamento. Em alguns casos, o compartilhamento ou o isolamento dos sujeitos é até desejável, mas isto deve ser descrito explicitamente. No Construcionismo tradicional, por exemplo, os aprendizes são incentivados a desenvolver seus próprios programas. No Construcionismo Social, por outro lado, o foco está na disponibilização de informações para a comunidade.

6.3.3 Sobre a internalização/externalização

- Que tipos de conceitos e procedimentos internos do sujeito devem ser passíveis de visualização? Por que papéis? Em que situações?

Neste caso, a idéia é determinar que tipo de informações devem ser tornadas públicas por cada um para que os demais sujeitos da atividade possam continuar desenvolvendo seus papéis.

O modo de disponibilizar estas informações também deve ser descrito. Por exemplo, um aluno precisa externalizar seu conhecimento de alguma forma para que um professor possa acompanhar como anda o seu aprendizado. No entanto, como será discutido no capítulo 7, às vezes o tipo de informações que um professor precisa não são relevantes para o aprendiz. Idealmente, o aprendiz não deveria

preocupar-se com informações que não fossem importantes ao seu trabalho. Informações de interesse exclusivo do facilitador devem ser produzidas de forma automática e transparente para o aprendiz.

- Quais práticas sociais deverão ser incentivadas?

Exemplos de práticas seriam: discussão de idéias, aprendizado independente, trabalho em equipe, apresentação de trabalhos, ajudar na organização da sala, etc.

6.3.4 Sobre a história e o desenvolvimento

- Qual a história da atividade?
- Qual a sua origem? É a primeira vez que ela está sendo efetuada? Ela foi criada a partir de outras? Quais as principais modificações que ela sofreu ao longo de sua história?
- Quais elementos da atividade tendem a desaparecer?
- O que deve ser mantido?

Conforme discutido no capítulo 3, a análise histórica da atividade fornece pistas sobre o que é de fato essencial na atividade e o que foi por ela assimilado por questões circunstanciais ao longo de sua evolução. Por exemplo, em uma escola, há uma série de elementos como horário e duração das aulas, quantidade de alunos e professores por sala, etc., que, embora talvez não sejam os mais indicados para uma nova prática pedagógica, devem ser respeitados devido à própria organização da escola, que tem resistência às transformações. A atividade final será uma combinação destes elementos históricos imutáveis e os novos propostos.

- Qual o nível de competência esperado e o desejado dos sujeitos em cada papel? O que favorece ou atrapalha o seu desenvolvimento?
- Como lidar com situações imprevistas? Quais os componentes da atividade mais susceptíveis à mudança?
- Que tipo de impacto a nova atividade trará para o ambiente onde ela será implantada? Quais as maiores dificuldades esperadas?

Na educação, por exemplo, poderiam ser levantadas as habilidades iniciais e esperadas dos alunos, se há tendências de mudanças na turma, se é esperado algum tipo de resistência por parte dos professores, se o material usado está prestes a ser trocado, etc.

6.3.5 Sobre a mediação

- Quais as regras da atividade educacional?
- Quais as normas e convenções, explícitas ou implícitas, que deverão ser consideradas no desenvolvimento da atividade? Que tipo de relacionamentos são permitidos e proibidos entre os sujeitos de cada papel? Que objetos podem ser manipulados, por quem, quando e por quanto tempo?

- De que modo estas regras serão aplicadas?
- Elas estarão embutidas nos artefatos usados? Elas serão explicitadas de alguma forma (descritas em papel, explicadas no início do trabalho, etc.)?
- Que artefatos estarão disponíveis? Que papéis e níveis de competência deverão suportar? Com que objetivos?

Na resposta a esta questão, deverão ser considerados tanto artefatos materiais como os não-materiais. Pode ser que, para o encaminhamento de determinadas ações, os participantes tenham que aprender determinados procedimentos e estratégias. Neste caso, estes procedimentos e estratégias também deverão ser considerados como artefatos a serem disponibilizados pela atividade.

No caso dos artefatos materiais, é importante que o tipo e a quantidade deles também seja listado.

- Como os diferentes artefatos deverão ser integrados?

Determinadas ações, ou seqüências de ações, requerem combinações de artefatos que precisam funcionar de modo integrado. No caso do computador, por exemplo, esta situação é bastante comum. Às vezes, é preciso integrar um desenho feito em um editor gráfico em um texto feito em outro editor, ou é preciso mandar tudo através da rede, etc.

6.4 A Análise do Artefato Computacional

As perguntas levantadas acima ajudam a traçar um perfil do que se espera da atividade educacional sendo criada. Elas descrevem os objetos e os sujeitos que estarão presentes, como eles deverão se relacionar e como evoluirão ao longo do tempo. É função dos artefatos ajudar a viabilizar esta descrição e a delimitar o escopo do que poderá ser feito em cada situação. São artefatos como instrumentos materiais, regras e convenções sociais que tornarão possível um maior ou menor acesso a determinados objetos e sujeitos, a representação e comunicação dos conceitos desejados, a manutenção da divisão das tarefas, etc.

Para se garantir o sucesso da atividade educacional, é muito importante determinar quais são os artefatos mais adequados para cada situação e o que deve ser feito por cada um deles. O objetivo da Análise do Artefato Computacional é delinear o que deverá ser feito pelo artefato computacional.

Conforme enfatizado anteriormente, ao se fazer a análise de um artefato frente a uma atividade, é preciso levar em conta uma série de fatores. É preciso verificar até que ponto o artefato atende às necessidades da atividade (eficácia), como ele se compara com os outros artefatos candidatos (eficiência) e se o seu custo está dentro do aceitável (viabilidade).

Em especial, a análise da eficiência do artefato frente à atividade exige que se tenha uma definição clara do que é este artefato e quais características o diferenciam de outros artefatos. Sem esta definição, torna-se impossível fazer uma comparação fundamentada do artefato escolhido frente aos demais candidatos à atividade.

Ainda que, conforme discutido no capítulo 4, diversos autores da Teoria da Atividade realcem características específicas do computador, na nossa opinião, os atributos levantados não são suficientes para o tipo de análise que se deseja.

Visando contribuir para a resolução deste problema, a seção 6.4.1 propõe definições para conceitos como artefato computacional, artefato digital, objeto digital e aplicativo. Com base nestes novos conceitos e nas idéias apresentadas nos capítulos anteriores, as seções 6.4.2, 6.4.3, 6.4.4, 6.4.5 e 6.4.6 levantam questões a serem consideradas na análise de um artefato computacional frente a uma atividade educacional.

6.4.1 Artefatos computacionais, artefatos digitais, objetos digitais e aplicativos

Na nossa opinião, o conceito "artefato" pode ser entendido como um adjetivo, indicando que um objeto possui determinadas características que o tornam útil para determinadas funções. Ao dizer-se, por exemplo, que um certo objeto é uma chave-de-fenda, compreende-se que este objeto é feito de um material rijo e que tem uma ponta que se encaixa na cabeça de um parafuso e permite rodá-lo. O adjetivo "rádio", por outro lado, caracteriza objetos que recebem ondas de certas frequências e as convertem em som.

Nada impede que um mesmo objeto esteja presente em mais de uma categoria e, assim, seja considerado como diferentes artefatos ao mesmo tempo. Alguns, como é o caso dos canivetes e dos aparelhos de som combinados (como os *microsystems*, 3 em 1, etc.), até são conhecidos pela multiplicidade e pela integração dos artefatos que trazem dentro de si.

O conceito tradicional de computador. Sob este aspecto de atender a diferentes categorias, o artefato computacional é bastante atraente e é, normalmente, considerado como uma verdadeira caixa de ferramentas. Conforme foi discutido no capítulo 4, o computador é visto pela Teoria da Atividade como um artefato extremamente flexível e de possibilidades quase que ilimitadas. É um artefato que pode ser utilizado nos diversos níveis da atividade, que pode atuar sobre uma grande variedade de objetos e que pode ajudar na formação de comunidades, na implantação de procedimentos específicos, na visualização de conceitos e muito mais.

No entanto, acreditamos que estas descrições dos diversos usos do computador pecam por não explicitarem o que realmente este artefato traz de diferente. Embora a combinação de telefone, fax, email, editoração de textos e imagens, planilhas de cálculo e outras funcionalidades seja atraente e útil, na nossa opinião ela reflete apenas algumas aplicações do computador, mas não algo que só ele consiga fazer ou que realmente o diferencie de outros artefatos.

Por exemplo, quando passamos um filme no computador, ele funciona tal e qual um videocassete; quando enviamos um fax, funciona como um fax e assim por diante, só o formato é diferente. O mesmo ocorre com um editor de textos: como diferenciá-lo de uma máquina de escrever sofisticada?

Artefatos digitais. Para nós, aquilo que Bodker define como "artefatos computacionais" é, na realidade, algo que deveria ser definido como "artefatos digitais". Isto é, são artefatos eletrônicos, cujas operações manipulam bits em suas mais diversas representações. Exemplos de artefatos digitais são os editores de textos, CD-players, navegadores de WWW, gerenciadores de bancos de dados, telefones via Internet e outros muitos que fazem parte do que as pessoas, normalmente, entendem por computador.

De fato, a confusão tradicional entre computadores e artefatos digitais tem uma série de justificativas. Boa parte dos artefatos digitais de hoje, por razões históricas e econômicas, realmente foram desenvolvidos e funcionam com auxílio de um computador. A diferença é que eles não dependem, obrigatoriamente, de um computador para existir. Há diversos artefatos digitais que não são computadores, como as calculadoras eletrônicas, relógios, gravadores digitais e os chamados *information appliances* (como assistentes pessoais e agendas eletrônicas, aparelhos que ligam uma TV a cabo na Internet, sistemas de localização via satélite e outros). Do mesmo modo, também há diversos computadores que não são eletrônicos.

Artefatos computacionais. Na nossa opinião, o conceito "computador" deve ser usado para classificar determinados tipos de artefatos. Assumindo que o conceito "máquina" possa ser associado pela Teoria da Atividade a artefatos materiais que automatizam operações humanas, consideramos o computador como um espécie de máquina que, além de automatizar um conjunto predefinido de operações, traz uma série de facilidades³³ para que seu usuário recombine as operações automatizadas, criando operações mais específicas. Estas novas operações, dependendo do caso, poderão inclusive passar a fazer parte do arsenal de operações oferecido pela máquina.

A esta capacidade de construção de novas operações a partir das já existentes chamamos de programação. Colocando em outras palavras o que foi dito acima, o computador é considerado por nós uma máquina programável.

Sob esta ótica, enquadram-se no conceito de computador teares mecânicos que permitem a programação de novos padrões, pianolas de fita que tocam sozinhas, máquinas processadoras de cartões perfurados, telefones com memória e, obviamente, *mainframes*, estações de trabalho e os *personal computers* (os PCs).

O que varia de um computador, ou artefato computacional, para outro, além de suas características físicas (tamanho, forma, etc.) e do tipo de operações que trazem dentro de si, é a flexibilidade que oferecem para a criação e utilização das novas operações. Por exemplo, diferente dos computadores digitais, as máquinas

³³ No caso, operações e/ou procedimentos dirigidas para este fim.

processadoras de cartões perfurados exigiam uma boa dose de trabalho na preparação dos cartões com o programa e, dependendo do tipo de memória que possuíssem, poderiam exigir um esforço semelhante para se repetir ou se alterar algo que já tenham executado.

O potencial e as facilidades de se combinar operações são os principais responsáveis pela flexibilidade associada aos computadores. Com o advento da eletrônica digital, esta flexibilidade ganhou poder. Os computadores e os artefatos digitais como um todo, adquiriram mais velocidade, uma certa independência de formato físico e, acima de tudo, operações que lhes permitiam acessar aquilo que chamamos "objetos digitais".

Objetos digitais. Sob o ponto de vista lógico, um objeto digital é uma seqüência de 0s e 1s a qual se associa um determinado conceito ou significado (NEGROPONTE, 1995, p.14). Por exemplo, arbitrariamente, 1001 poderia significar carro, 10011 poderia significar carro amarelo e 10010 carro verde. Neste caso, foram associados os quatro primeiros dígitos do número ao conceito carro e o quinto dígito à sua cor. Por esta convenção, bastaria mudar este último dígito para transformar a cor do carro.

Sob o ponto de vista estrutural, no caso da eletrônica digital, cada dígito, ou bit, assume um valor elétrico real, que permite que ele possa ser manipulado materialmente por nós através dos artefatos digitais. De fato, por meio dos artefatos digitais, seqüências inteiras de bits podem ser manipuladas e visualizadas das mais diferentes formas, inclusive de modo que eles nos sejam mais representativos para cada situação. Por exemplo, voltando ao caso anterior, um artefato digital poderia converter a seqüência de bits 10011 na figura de um carro amarelo que fosse mostrada em um monitor de vídeo. Apertando um botão, o artefato poderia alterar o quinto bit para 0 e, simultaneamente, modificar a cor do carro para verde.

Graças à sua estrutura elétrica e ao crescente desenvolvimento da tecnologia, da padronização da eletrônica digital e de sua integração com as telecomunicações, os objetos digitais já podem, hoje em dia, ser conectados a uma gama bastante ampla de equipamentos locais ou remotos (como sensores, motores, teclados, monitores, impressoras, etc.) que lhes permitem interagir mais diretamente com a realidade e nos permitem acessá-los como se fossem objetos verdadeiramente materiais³⁴.

Por outro lado, como representações, os objetos digitais não precisam se ater às limitações do mundo concreto e, sob o ponto de vista de quem os manipula, apesar de parecerem materiais, não necessariamente precisam existir na realidade fora da máquina. Este é o caso de objetos puramente digitais, tais como personagens de jogos, células de planilhas de cálculo, o acervo de páginas da WWW e outros.

³⁴ Ao longo deste trabalho, objetos digitais, formados por bits, são usados em contraposição a objetos materiais, formados por átomos. Essa distinção é diferente da que é feita normalmente na engenharia, onde o digital, discreto, é contraposto ao analógico, contínuo.

Na nossa opinião, a possibilidade de manipular objetos tão diversos e, ainda, de combinar as operações efetuadas nestes objetos na criação de artefatos digitais mais específicos é o que torna os computadores tão poderosos e difundidos na sociedade atual³⁵.

Integração dos conceitos. Embora as definições acima pareçam diferentes das apresentadas nos capítulos anteriores, todos os demais conceitos levantados por Bodker parecem aplicar-se ao conceito de artefato digital e artefato digital computacional. Todo artefato digital possui uma parte hardware e uma parte software que, sob o olhar do usuário, manifestam-se nos aspectos físicos, de manuseio e os dirigidos ao sujeito/objeto; a definição de interface parece ser adequada; o suporte aos níveis de competência é pertinente, etc. (ver capítulo 5).

Vale ainda notar que o conceito de artefato computacional aqui apresentado também está de acordo com o proposto por Kaptelinin e Tikhomirov (ver capítulo 4). A facilidade de combinar operações, propiciada pelo artefato computacional, permite que o usuário dose o grau de automação da máquina para aquele que for mais adequado ao seu nível de compreensão e interesse em uma determinada atividade. Com isso, sob o controle da mente do usuário, o computador de fato estende a capacidade humana de transformar ações em operações, complementando a funcionalidade do Plano Interno de Ações e permitindo que o indivíduo se concentre mais ou menos nos componentes intuitivos (não formais ou mecânicos) do pensamento, conforme lhe for mais conveniente.

Aplicativos. Voltando para o desenvolvimento de software, a partir dos conceitos acima, pode-se entender aplicativos (software comercial ou não) como conjuntos de artefatos digitais que, normalmente, foram agrupados em torno de um propósito comum. Um editor de textos, por exemplo, pode ser visto como um conjunto de ferramentas para o tratamento de caracteres, palavras, parágrafos, textos, impressoras, páginas e muito mais. Um aplicativo administrativo traz um conjunto de artefatos para manipular registros de empregados e de materiais; um aplicativo de produção multimídia inclui artefatos para lidar com sons, imagens e animações e assim por diante³⁶.

Porém, além de meros conjuntos de artefatos, os aplicativos também podem incluir objetos digitais que poderão ou não ser acessados por outros aplicativos. Agendas eletrônicas, por exemplo, costumam armazenar dentro de si os nomes e telefones anotados. Navegadores de Internet, por sua vez, não armazenam todas as páginas acessadas, só circulam de uma para a outra.

³⁵ Em seu livro "*Being digital*" (NEGROPONTE, 1995), Nicholas Negroponte discorre sobre a relação bits e átomos e levanta uma série de implicações que o mundo dos bits têm trazido para a sociedade atual.

³⁶ Nota para engenheiros de software: esta definição proposta de "aplicativos" em muito se assemelha à idéia de software tal como defendido pelas metodologias de análise e design de software orientadas a objetos. Mais especificamente, o conceito de "método" e "atributo" destas metodologias parecem muito semelhantes aos aqui discutidos de "artefatos digitais" e propriedades de objetos.

É o tipo de análise desejada que determinará que artefatos deverão constituir o aplicativo, que objetos ele deverá acessar, onde estes objetos ficarão armazenados, quem poderá acessá-los e tudo o mais. Justamente por delinear a relação entre o que ficará no computador e o que ficará no mundo material, esta especificação de quais artefatos e objetos serão mapeados para dentro de que aplicativos é uma parte muito importante a ser considerada no projeto de um software. É ela quem determinará o que deverá ser implementado.

Implicações para o desenvolvimento de software para a educação. No caso do desenvolvimento de software para a educação, os conceitos de artefatos digitais, artefatos computacionais, objetos digitais e aplicativos ajudam a levantar uma série de pontos que merecem reflexão. Por exemplo, que tipo de liberdade em nível de programação deverá ser dada aos participantes da atividade?

Diferentes aplicativos podem permitir maior ou menor facilidade de acesso aos recursos de programação de um computador. Em determinados casos, dependendo da atividade educacional, pode ser que se deseje um software específico não-programável – um artefato digital mais tradicional – como um leitor de livros eletrônicos ou como um simulador de situações científicas. Em outros, pode ser que se deseje disponibilizar o potencial de programação da máquina – tornando-a um artefato computacional – para explorar a criatividade dos alunos, para que o professor possa criar seus próprios artefatos digitais ou para qualquer outro objetivo.

O próprio fato de, na programação, o sujeito ter que tomar consciência das ações que pretende automatizar pode aumentar a sua compreensão dos processos mentais que executa, incentivando a reflexão e inclusive o seu aprimoramento. Dependendo da qualidade das operações automatizadas pelo computador e das facilidades de combiná-las, tanto melhor será a imagem que o sujeito terá do que ocorre em sua mente³⁷.

Além disso, de uma forma geral, como os objetos digitais não precisam se limitar às propriedades dos objetos materiais e podem ser acessados dos mais diferentes locais e sob as mais diferentes formas, torna-se possível, com o auxílio do computador e dos artefatos digitais criar situações educacionais, fantásticas ou não, que extrapolem as restrições de tempo, espaço e recursos de uma sala de aula. Por exemplo, os aprendizes poderiam fazer intercâmbios de objetos, visualizar explosões químicas, manipular objetos distantes (como tirar fotos do espaço utilizando telescópios em outros países), participar interativamente de uma fábula, experimentar simulações micro ou macroscópicas (de modelos atômicos, moleculares, orbitais, etc.) e, inclusive, procurar expressar novas teorias.

³⁷ É interessante notar que, embora o Construcionismo parta de um pressuposto epistemológico diferente do da Teoria da Atividade (o Construcionismo tem origem no pensamento piagetiano), esta característica da programação representar uma imagem externalizada, ainda que simplificada, da mente é um dos pontos mais enfatizados por ele (ver capítulo 5) que só vem a ser reforçada pelo conceito de artefato computacional aqui defendido.

Por outro lado, é importante notar que, apesar de suas aparentes vantagens, nem sempre estas características dos objetos digitais descritas acima são positivas para a educação. Segundo a Teoria da Atividade, alguns tipos de aprendizado exigem a presença do objeto físico, enquanto outros conseguem desenvolver-se com a presença de representações destes objetos (BODKER, 1991). Por exemplo, mesmo que os objetos digitais possam se assemelhar aos reais em termos de imagem, som e comportamento, nas atividades em que as propriedades olfativas, tácteis e gustativas fossem importantes – como ocorre, por exemplo, na educação infantil – a utilização de objetos materiais deveria ser incentivada.

Ainda que discussão destas questões mereça um estudo específico, que ultrapassa o escopo do presente trabalho, fica a idéia de que, de qualquer modo, o computador e os artefatos digitais ajudam a abrir as portas das escolas para um sem número de possibilidades que, usadas com critério, podem contribuir para o desenvolvimento de atividades educacionais mais eficazes, eficientes e viáveis .

Por fim, tal como na seção anterior, abaixo são apresentadas algumas perguntas que deverão ser consideradas na análise do artefato computacional. Vale notar que, como a maioria das questões serve para análise, tanto de artefatos digitais quanto computacionais, deste ponto até o final do trabalho só serão diferenciados os aspectos específicos do artefato computacional quando se mostrar necessário. Do mesmo modo, o termo computador também passará de agora em diante a ser usado de forma ampla, mais no sentido de “artefato digital”, que é a função com que ele é mais normalmente usado pelas pessoas.

6.4.2 Sobre a estrutura hierárquica da atividade

- Há alguma funcionalidade específica do artefato que contribui para a motivação da atividade? Qual? De que forma?
- Que ações de que papéis deverão ser suportadas pelo artefato? De que modo? O artefato ajudará na definição das metas?
- Que transições entre papéis deverão ser facilitadas pelo artefato?
- Que operações deverão ser automatizadas/suplementadas pelo artefato?
- Que tipo de suporte o artefato poderá oferecer para as transformações mútuas entre ações e operações?
- Que aspectos físicos e de manuseio tendem a ser conceitualizados no uso do artefato? Que tipo de suporte o artefato oferece para a minimização destas conceitualizações?

No que se refere à conceitualização e operacionalização do artefato computacional, deve-se lembrar que:

- Embora seja virtualmente impossível eliminar todas as conceitualizações relacionadas aos aspectos físico e de manuseio do artefato computacional, elas podem ser minimizadas através de mecanismos de *feedback* (como mensagens e outros indicadores) que ajudem o sujeito a orientar-se e que tragam indicações sobre como proceder.

- A operacionalização dos aspectos físico e de manuseio pode ser facilitada pela utilização de padrões de interface (como os que definem a posição dos botões e as cores do sistema Windows), de conceitos que sejam familiares ao sujeito e de um treinamento apropriado. Por exemplo, se todos os tipos de software de um ambiente como o MS-Windows costumam salvar os arquivos através de um mesmo procedimento (no caso, a opção Salvar do menu Arquivo), o software educacional também deve buscar seguir este padrão para evitar que o sujeito se esforce em coisas que não estão relacionadas com o objetivo da atividade. Por outro lado, o ideal mesmo, neste caso, seria que o aprendiz não tivesse nem que ficar se preocupando em ter que salvar o seu trabalho. O artefato poderia fazer isso automaticamente e de modo transparente.
- O projeto do artefato computacional também pode ir de encontro a estes padrões de interface, buscando forçar uma conceitualização que atraia o sujeito para algum ponto específico da interface que ele tenha que dar maior atenção.

6.4.3 Sobre a orientação a objetos

- O artefato em si está diretamente relacionado à motivação da atividade? Até que ponto? De que forma?

Em alguns casos, dado o enorme prestígio que tem na sociedade moderna, o computador em si pode ser usado como objeto motivador de uma atividade educacional. Por exemplo, até certo tempo atrás e às vezes até hoje, a simples presença de computadores, independente da análise de que seria feito com ele, era usada como meio de atrair alunos para as escolas. Hoje, apesar de esta situação permanecer, já se percebe uma preocupação maior dos pais e educadores com o tipo de uso que será dado a esta ferramenta.

- Que objetos serão manipulados através do artefato? Eles também estarão presentes fisicamente na atividade? Quais de suas propriedades deverão ser acessadas ou manipuladas através do artefato?
- Quantos e que tipos de sujeitos poderão ser acessados com auxílio do artefato? Eles também estarão presentes fisicamente e/ou temporalmente na atividade? Que atributos da comunicação (sincronização, capacidade de visualização, audição, etc.) deverão ser suportados pelo artefato?
- Que mudanças de foco entre sujeitos/objetos deverão ser suportadas para cada papel/nível de competência?
- Como será a integração do artefato computacional com os demais artefatos da atividade? Os objetos manipulados por ele também poderão ser acessados por outros artefatos? Quais? Como?

Estas questões procuram retratar como o artefato fará a mediação com os objetos, digitais ou não e que tipo de restrições e contribuições trará à comunicação com os demais participantes da atividade.

Além das perguntas acima, que valem para a análise de qualquer artefato digital, abaixo são apresentadas algumas questões específicas para o caso dos artefatos computacionais.

- Que operações do artefato poderão ser combinadas entre si?
- Que papéis poderão combinar, eliminar e/ou acessar estas operações?
- Em que situações estas operações poderão ser combinadas e/ou acessadas?

A principal diferença nestas questões advém do fato de, nos artefatos computacionais, devido à programação, as próprias operações do artefato poderem ser manipuladas como objetos pelo sujeito.

6.4.4 Sobre a internalização/externalização

- Que facilidades o artefato deverá oferecer para a externalização? Ajudar na visualização de que processos? Possibilitar a comunicação com que papéis? Ajudar na execução de que cálculos?
- Que práticas sociais deverão ser estimuladas pelo artefato?
- Que tipo de facilidades o artefato oferecerá para a coordenação das ações de um mesmo sujeito ou entre diferentes sujeitos?

As simulações e as representações efetuadas com o auxílio do artefato podem contribuir bastante para a compreensão e comunicação das idéias do sujeito. Determinados tipos de software também favorecem a interação com grandes quantidades de pessoas (como listas de email e outras) ou incentivam certas práticas, exigindo que o usuário cumpra determinados passos antes de chegar aonde deseja. As questões acima ajudam a delimitar o escopo do que o artefato computacional deverá fazer em cada uma destas possibilidades.

6.4.5 Sobre a história e o desenvolvimento

- Há algum fator histórico que influencie na escolha ou no desenvolvimento do artefato? Qual? Que tipo de influência ele exerce?

Às vezes, fatores externos já existentes influenciam na escolha dos materiais e na aparência do artefato que está sendo construído. Em especial, padrões da indústria tendem a facilitar o desenvolvimento do artefato e a sua integração com outros que o complementem. Por sua vez, líderes de mercado tendem a apontar preferências já consagradas que podem ajudar na viabilização do produto.

- Quais os níveis de competência esperados para cada papel suportado pelo artefato digital/computacional?
- Que tipo de suporte o artefato deverá oferecer aos diferentes níveis de competência?

- Que tipo de suporte o artefato deverá oferecer para facilitar a passagem de um nível de competência para outro?
- Que tipo de suporte será oferecido para a operacionalização do artefato em si?

Normalmente, antes que o sujeito consiga focar no domínio da atividade, ele terá que operacionalizar as ações específicas do artefato (capítulo 3) e isso exige todo um planejamento especial. No caso da programação, por exemplo, o sujeito terá que aprender os procedimentos de como combinar as operações automatizadas, como alterar as combinações, etc., para que estas operações não atrapalhem o andamento da atividade.

- Que tipo de suporte será oferecido para ajudar o sujeito a se recuperar das situações onde as operações do artefato sejam conceitualizadas? E das situações em que o domínio educacional seja conceitualizado?

Dentre os tipos de suporte oferecidos, pode-se contar desde mecanismos de ajuda online até cartazes e sessões de formação que abordem as dificuldades mais comuns encontradas no uso do artefato dentro da atividade.

- Que tipos de modificações são mais susceptíveis de ocorrer no uso do artefato?

Às vezes, como tem acontecido cada vez mais freqüentemente, pode ser que uma nova geração de artefatos já esteja planejada para ser lançada antes mesmo da atual ter sido concluída. Nestes casos, é importante considerar as alternativas que facilitem as modificações previstas.

- Há algum tipo de padrão que deverá ser seguido na construção da interface do artefato? Qual?

Independente das razões de funcionalidade, custo e estética, pode ser que, por razões históricas o artefato sendo desenvolvido tenha que se ater a algum padrão já definido. Isso pode acontecer quando ele tem que dar continuidade a alguma linha de produtos já existentes, quando o padrão assegura uma melhor vendagem, etc.

6.4.6 Sobre a mediação

- Que tipo de potencializações/restrições o artefato trará para a atividade? Para quem? Como?
- Qual a relação entre o artefato em questão e os demais artefatos disponíveis para a atividade?

De certa forma estas questões já foram respondidas ao longo das anteriores. No entanto, estas perguntas ajudam a trazer uma visão mais geral da importância do artefato para a atividade.

Questões específicas dos artefatos computacionais estão descritas a seguir:

- Que facilidades o artefato trará para a visualização, manipulação e reaproveitamento das suas próprias operações?

- Como será o procedimento de combinação, eliminação e acesso às novas operações?

Como as operações do próprio artefato computacional podem ser manipuladas, diferente do que acontece em outros artefatos, o artefato computacional nunca deverá se tornar totalmente transparente para o sujeito. No entanto, é importante que ele ajude o sujeito a não se distrair do que deseja fazer.

6.5 Considerações sobre as demais fases do desenvolvimento

Enquanto a análise do artefato se preocupa em especificar tudo o que o artefato computacional deve fazer e em que condições, a fase de **projeto** busca uma solução que atenda a tudo o que se deseja dentro das restrições impostas.

Na nossa opinião, nesta fase devem ser enfatizados os seguintes pontos levantados pela Teoria da Atividade:

- É importante que o projeto leve em consideração a possibilidade de se desenvolver um hardware específico para a atividade sendo criada. Conforme mencionado acima, o desenvolvimento da tecnologia tem aberto uma série de portas neste sentido, tanto em termos de custos, quanto de possibilidades. Os projetistas devem começar a pensar mais seriamente nos aspectos físicos dos artefatos educacionais que forem desenvolver – não apenas usar o que já existe pronto no mercado – principalmente para o caso de projetos educacionais de grande porte (como aqueles que envolvem estados ou países inteiros), onde o investimento tende a diluir-se.
- O projeto do artefato computacional deve basear-se no desenvolvimento de protótipos. Estes protótipos, além de facilitarem a comunicação entre os diferentes participantes da atividade de desenvolvimento, também permitem que as idéias sejam testadas em situações próximas das reais, levantando pontos que, dificilmente, poderiam ser imaginados sem a vivência prática da atividade que estiver sendo criada;
- Dentre as diversas soluções possíveis para o projeto, deve-se buscar aquela que maximize a flexibilidade, a consistência e a simplicidade de uso do artefato. Conforme discutido no capítulo 4, a flexibilidade refere-se à possibilidade de o sujeito mudar de foco entre objetos e sujeitos e de atingir suas metas de diferentes maneiras, escolhendo aquela que melhor lhe convier em cada situação. A consistência passa ao sujeito a sensação de que o artefato se comporta de forma semelhante em situações que pareçam semelhantes para o próprio sujeito, mesmo que, no nível de processamento interno dentro do artefato, possam ser coisas completamente diferentes. E a simplicidade está relacionada com o número de ações que o sujeito tem que efetuar para atingir

seus objetivos. Quanto menor a quantidade de ações exigidas, mais simples será o uso do artefato.

Já na fase de **implementação**, que é onde o artefato computacional é realmente construído, por ser uma fase essencialmente técnica, não há, praticamente, comentários da Teoria da Atividade a serem apresentados.

Na fase de **teste**, deve ser verificado até que ponto o artefato computacional atende ao que foi especificado na Análise. Conforme mencionado anteriormente, o artefato computacional (ou seu protótipo) deve ser testado em situações mais próximas da atividade real quanto possível, inclusive no que se refere aos aspectos sociais e culturais do ambiente e com os diferentes papéis e níveis de capacitação esperados. Os experimentos desenvolvidos em laboratório são de pouca valia para a Teoria da Atividade.

Por fim, sobre a fase de **manutenção** do artefato computacional, vale notar que ela é tida como a mais longa e custosa do desenvolvimento. É na manutenção que o artefato vai sendo alterado e corrigido para adaptar-se às eventuais transformações que a atividade inevitavelmente irá sofrer e tornar-se cada vez mais adequado a ela.

Para que estas modificações possam ser feitas com um mínimo de esforço e impacto na atividade, é muito importante que a manutenção seja levada em consideração desde as primeiras fases do ciclo. É por isso que, na análise tanto da atividade quanto do artefato computacional, deve-se levar em consideração os papéis do designer da atividade e do ator (ver capítulo 4).

Para todos os efeitos, o designer da atividade representa os responsáveis por toda a concepção da atividade, o que inclui desde o seu projeto até a implementação, teste e manutenção do artefato computacional. É para ele que as críticas e sugestões relacionadas com a atividade (e com o artefato computacional) devem ser dirigidas.

O ator é o papel complementar do designer da atividade. É ele quem participa da atividade e tem as sugestões e críticas para oferecer.

Nas fases de **análise** da atividade e análise do artefato computacional é muito importante identificar que tipo de informações deverão ser passadas para o designer da atividade e em que situações.

No **projeto**, deve ser especificado como estas informações serão comunicadas. Será implementado algum sistema de correio eletrônico entre atores e designers? Haverá algum mecanismo que armazene um histórico de todas as operações efetuadas no artefato? Existirá alguma espécie de grupo de discussão relacionado com a atividade?

A idéia de manter um canal direto e ininterrupto, no próprio artefato, ligando os usuários aos desenvolvedores ajuda a reportar os problemas e sugestões no momento em que surgirem e ajuda a estreitar a distância que, normalmente, existe entre estes dois papéis (e entre a atividade de desenvolvimento e a atividade de uso).

Na realidade, se os usuários fizerem parte da equipe de desenvolvimento, este canal já começa a existir desde o início. Neste sentido, embora não tenha sido explorada durante este trabalho, a idéia de se ter uma equipe de desenvolvimento de software para a educação incluindo professores e alunos parece bem promissora.



7 A análise da atividade educacional construcionista

No capítulo anterior foi proposta uma abordagem de desenvolvimento de software baseada em uma análise da atividade educacional. Também foi dito que, como as atividades educacionais oriundas de uma mesma filosofia têm uma série de pontos em comum, estes pontos poderiam ser reunidos em uma descrição mais teórica e abstrata que servisse de elo entre todas elas e que norteasse a criação e avaliação das novas atividades.

No caso do Construcionismo, conforme discutido no capítulo 5, não há referências que agrupem os princípios da teoria de forma a incentivar sua discussão e disseminação. O próprio capítulo 5 ajuda a minimizar um pouco esse problema, reunindo as idéias centrais da educação e do aprendizado construcionista. Ainda assim, no entanto, esta descrição de princípios não é suficientemente detalhada para orientar o desenvolvimento de software.

Para trabalhar esta questão, o presente capítulo aplica a abordagem proposta no capítulo 6 para fazer uma análise aprofundada do que seria uma atividade educacional construcionista genérica. As questões levantadas pela abordagem orientam a discussão, ajudando a dar estrutura aos conceitos e a certificar que aspectos importantes não sejam desconsiderados. Em alguns casos, os conceitos da Teoria da Atividade foram usados para reforçar idéias ou para propor alternativas a aspectos da atividade não tratados pelo Construcionismo.

Vale notar que estas inserções foram efetuadas para ilustrar possíveis aspectos comuns ou complementares entre a Teoria da Atividade e o Construcionismo. No entanto, por partirem de pressupostos epistemológicos distintos e, às vezes contraditórios, as identidades de cada teoria foram respeitadas e seus limites, demarcados no texto.

A seção 7.1 define, com base no diagrama de Engeström (ver capítulo 3), a estrutura da atividade construcionista, explicitando os diferentes papéis que os sujeitos podem assumir e suas respectivas ações. Em especial, nesta seção é feita uma discussão em cima do ciclo da descrição-execução-reflexão-depuração-descrição (ver capítulo 5) e, a partir dela, é apresentada uma proposta do que seriam as principais ações do aprendizado construcionista.

Complementando a análise, a seção 7.2 discute a importância dos objetos e do tema explorado na motivação da educação construcionista, a seção 7.3 debate a influência das dinâmicas sociais no desenvolvimento do aprendiz, a seção 7.4 discute os aspectos históricos e temporais que podem afetar a atividade e a seção 7.5 discorre sobre a importância dos diferentes tipos de mediação encontrados no Construcionismo.

Por fim, vale notar que a análise aqui apresentada não se prende a nenhum tipo de tecnologia ou domínio pedagógico. Ainda assim, é ela que fornece embasamento

para a análise específica do papel do computador no Construcionismo descrita no capítulo 7.

7.1 A estrutura hierárquica da atividade construcionista

Conforme mencionado no capítulo 4, em um de seus trabalhos, Bellamy (1996), descreveu, a partir de um diagrama de Engeström, a atividade educacional do ensino fundamental e médio (K-12) americano. Neste diagrama, a educação é representada como uma atividade que tem por objetivo gerar aprendizes educados e nele é ressaltada a importância dos diversos artefatos, da comunidade, da divisão do trabalho e das regras no alcance deste objetivo. O diagrama também mostrou-se bastante útil no levantamento de critérios para o desenvolvimento de software para a educação.

Um dos pontos que se tornou aparente em seu artigo é que a atividade educacional é composta por uma série de atividades interligadas – como as atividades administrativas, as atividades de ensino e as atividades de aprendizado – e que, na medida do possível, todo este contexto deveria ser considerado no desenvolvimento da tecnologia para a educação.

No entanto, na nossa opinião, uma análise mais detalhada indica que, embora as atividades estejam intimamente relacionadas, há sempre algumas que parecem mais centrais do que outras. No caso do Construcionismo, a atividade central é a do aprendizado e todas as demais podem ser vistas como atividades de suporte que servem, principalmente, para viabilizar o aprendizado e adequá-lo às condições específicas de cada ambiente. É por isso que o Construcionismo pode ser considerado como uma teoria centrada no aprendizado.

Ainda que impossível na prática, dada a quantidade de interdependências e informações, uma atividade educacional só estaria de fato descrita se englobasse as descrições de todas as suas sub-atividades. No entanto, como será visto na seção seguinte, neste trabalho, devido às limitações de tempo, optou-se por um aprofundamento maior na atividade de aprendizado em detrimento das demais.

7.1.1 A atividade de aprendizado construcionista

Na nossa opinião, usando o diagrama de Engeström como referência, a atividade de aprendizado construcionista pode ser descrita como uma atividade de desenvolvimento de projetos onde o aprendiz é incentivado a trabalhar com objetos transicionais no desenvolvimento de um produto compartilhável que lhe seja significativo (ver Figura 15). A atividade é mediada por uma série de artefatos, computacionais ou não, que ajudam na manipulação dos objetos, ressaltando suas características específicas e propiciando condições para que o aprendiz reflita sobre o processo todo.

Conforme discutido no capítulo 5, diferente do que acontece nas atividades profissionais de design, onde o objetivo é o produto em si, na atividade de aprendizado construcionista a ênfase recai, principalmente, sobre o processo e sobre as reflexões efetuadas pelo aprendiz em cima deste processo. No design construcionista, além de um fim em si, o produto é visto como um meio para o desenvolvimento do aprendiz. O produto serve quase que como um pretexto, de preferência útil, para que o aprendiz possa passar por um processo específico, refletir sobre ele e, assim, desenvolver-se.

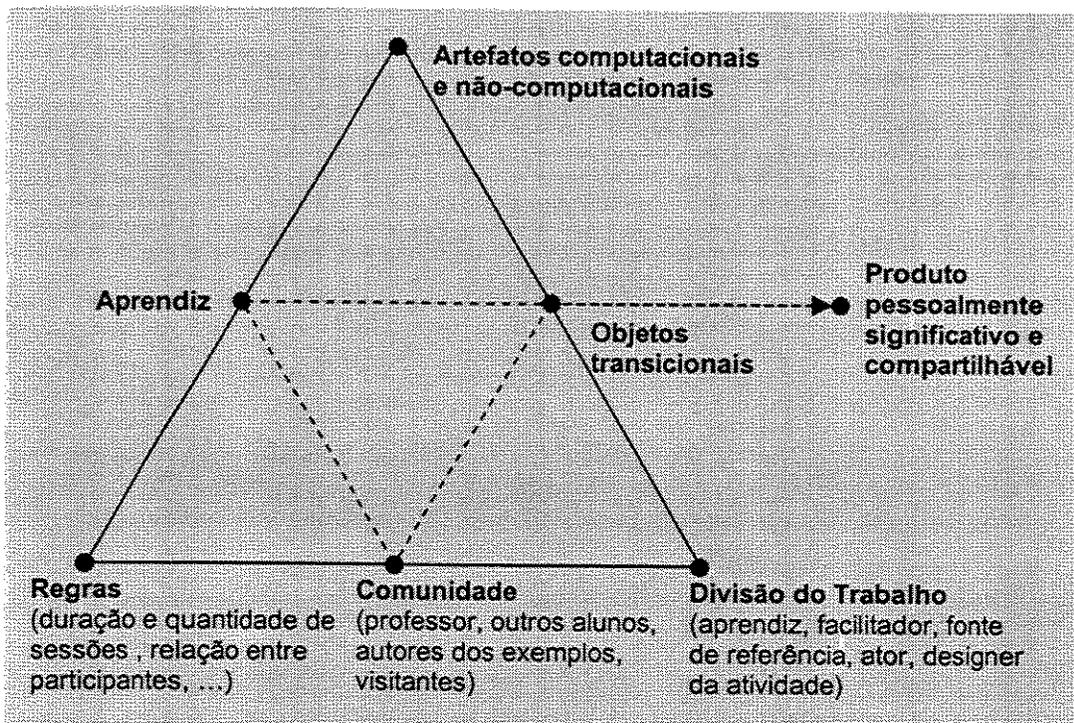


Figura 15 - O diagrama de Engeström adaptado para a atividade de aprendizado construcionista

Neste sentido, a atividade construcionista é uma atividade de **design reflexivo** onde, para cada ação efetuada no sentido de construção do produto, existe um esforço para se descrever (externalizar) o que foi feito e o por quê (descrição e meta-descrição do processo). Nas atividades construcionistas, além de aprenderem a desenvolver um produto, espera-se que os aprendizes adquiram uma visão crítica do processo de desenvolvimento.

7.1.1.1 Os componentes do aprendizado construcionista

Abaixo é apresentada uma descrição sucinta dos componentes da atividade de aprendizado construcionista (ver Figura 15):

O **aprendiz** é o sujeito central do aprendizado construcionista. Ele é visto pelo Construcionismo como uma pessoa que se desenvolve ativamente a partir de sua interação com o mundo, tem um determinado estágio cognitivo (semelhantes aos descritos por Piaget), alguns gostos e conhecimentos específicos que orientarão a

temática da atividade e um estilo próprio de aprendizado que poderá variar de situação para situação.

Na atividade construcionista, ele acessa e transforma uma série de **objetos transicionais** que, além de lhe serem familiares e poderem ser combinados entre si, são representativos dos novos conhecimentos sendo aprendidos. Estes objetos podem estar presentes local ou remotamente e em maior ou menor quantidade. Podem existir dentro e/ou fora do artefato computacional e podem ser acessados diretamente pelo sujeito e/ou por meio dos artefatos disponíveis.

É claro que, além dos objetos transicionais e dos demais elementos planejados especialmente para a atividade, sempre haverá alguns componentes que estarão presentes por razões históricas ou culturais. O ideal é que haja um mínimo de elementos que distraiam o aprendiz da atividade de aprendizado.

Os **artefatos** usados na atividade, para efeito deste estudo, serão, simplificada e genericamente, classificados em artefatos computacionais, que envolvem o computador tradicional e artefatos não-computacionais, que incluem, além de livros, cadernos, lápis, borracha, câmera de vídeo, TV, CDs, etc. todos os artefatos psicológicos como métodos, estratégias e outros. Estes artefatos, além de viabilizarem o desenvolvimento da atividade, também permitem que o aprendiz se concentre nos aspectos específicos do aprendizado em questão.

A idéia é que, através da manipulação e combinação dos objetos disponíveis no ambiente, o aprendiz consiga construir um **produto** compartilhável e que lhe seja significativo. Além dos artefatos, das particularidades dos aprendizes e dos objetos disponíveis, a atividade também é influenciada pelas regras, pela divisão do trabalho e pela comunidade.

Conforme discutido anteriormente, outro componente importante das atividades são as **regras**. Implícitas ou explícitas, são elas que definem o que é permitido ou não em termos de interação entre os participantes, regularizam o uso dos artefatos e ajudam na organização da atividade (número de sessões, duração das sessões, quantidade de participantes, formas de avaliação, etc.). Em alguns casos, nas atividades construcionistas, às vezes torna-se necessário usar regras, por exemplo, para que os aprendizes descrevam melhor o que estão fazendo, não tumultuem demais as aulas, ou não fujam do tema proposto.

A **divisão do trabalho** pode ser vista como as diferentes funções ou papéis que podem ser efetuados pelos participantes da atividade. No presente estudo foram identificados os seguintes papéis para a atividade de aprendizado construcionista:

a) Aprendiz. É o sujeito (ou sujeitos) que está construindo seu projeto e está sendo educado.

b) Facilitador. É o sujeito que procura adequar a atividade às condições específicas de cada aprendiz, procurando resolver os problemas tanto técnicos quanto pedagógicos que poderiam dificultar ou mesmo bloquear o desenvolvimento do aprendiz. É o facilitador quem incentiva as relações sociais, motiva o aprendiz nas horas difíceis e cria condições para que o mesmo encontre, por conta própria, as soluções para as suas dificuldades (VALENTE, 1996a).

De início, acreditava-se que era possível, através do desenvolvimento de tecnologias adequadas, eliminar o papel do facilitador. No entanto, os estudos construcionistas mostraram que o elemento humano é fundamental e que é necessário ter alguém com mais experiência que possa acompanhar de perto o trabalho do aprendiz e oferecer orientação e apoio. Não basta criar os componentes do ambiente, é preciso ter alguém que incentive o seu uso.

O facilitador pode ser visto como o sujeito de uma sub-atividade específica dentro da educação construcionista que é a atividade de facilitação do aprendizado. Ele trabalha, principalmente, com o aprendiz e todos os produtos intermediários e finais do aprendizado, utiliza de métodos pedagógicos e artefatos específicos para acompanhamento e intervenção e tem como objetivo facilitar o aprendizado.

c) Fonte de referência. É onde o aprendiz obtém informações que talvez possam lhe interessar e contribuir com o seu trabalho. As fontes de referência podem aparecer de maneiras mais ou menos diretas na atividade do aprendiz. As mais diretas (que, em sua maioria, são síncronas) seriam as pessoas (amigos, professores, parentes) para as quais ele faz uma pergunta ou de quem ele escuta algo que tenha a ver com a atividade.

As menos diretas (assíncronas) seriam, por exemplo, livros, CDs, filmes ou trabalhos de amigos. Todos estes materiais trazem informações produzidas por pessoas (os autores) que, embora não estejam, necessariamente, presentes no mesmo espaço físico-temporal da atividade, também acabam por contribuir com o aprendizado.

d) Ator. Conforme discutido no capítulo 6, é o sujeito que critica a atividade em todos os seus aspectos e apresenta sugestões para o seu aperfeiçoamento.

e) Designer da atividade. Conforme discutido no capítulo 6, é quem concebeu a atividade em si e quem tem a preocupação de torná-la cada vez mais adequada. Sua experiência com os fundamentos teóricos e com a prática influenciam, diretamente, no desenrolar da atividade e é a ele que devem ser enviadas todas as sugestões e críticas que forem aparecendo.

O designer da atividade é o sujeito que tem por objetivo conceber e aprimorar a dinâmica e a estrutura da atividade educacional. Seus objetos de trabalho são, principalmente, os conceitos teóricos e as recomendações tiradas da prática, isso para todas as áreas relacionadas com a atividade educacional, como a tecnologia, pedagogia, domínios específicos, etc.

Um mesmo sujeito pode assumir diversos papéis. Dependendo da organização da atividade e das condições locais, diferentes membros da comunidade poderão assumir diferentes papéis. Por exemplo, em uma atividade em sala de aula com o Logo, os professores costumam assumir os papéis de facilitador, fonte de referência e de designer da atividade. Os alunos assumem, na maioria dos casos, o papel de aprendiz mas, dependendo da liberdade dada, também podem servir de facilitadores e fontes de referência para o trabalho de seus colegas, de atores quando opinam e de designers da atividade quando procuram alterar as condições

do ambiente para que o aprendizado seja otimizado. Os pais, parentes e amigos também podem contribuir de diversas formas, dependendo da situação.

Representando outras atividades de aprendizado construcionista. Na nossa opinião, a idéia de representar atividades construcionistas com o diagrama de Engeström é adequada para diferentes situações e ajuda a identificar onde estão e como se relacionam os diversos componentes envolvidos em cada uma delas.

Por exemplo, o aprendizado construcionista com o Logo Gráfico pode ser descrito como uma atividade onde o aprendiz interage com o objeto “tartaruga gráfica” através dos comandos da linguagem de programação (ver Figura 16). Também interage com o objeto “programa” (ou com os objetos “procedimento”) através dos comandos do editor de programas que estiver sendo utilizado.

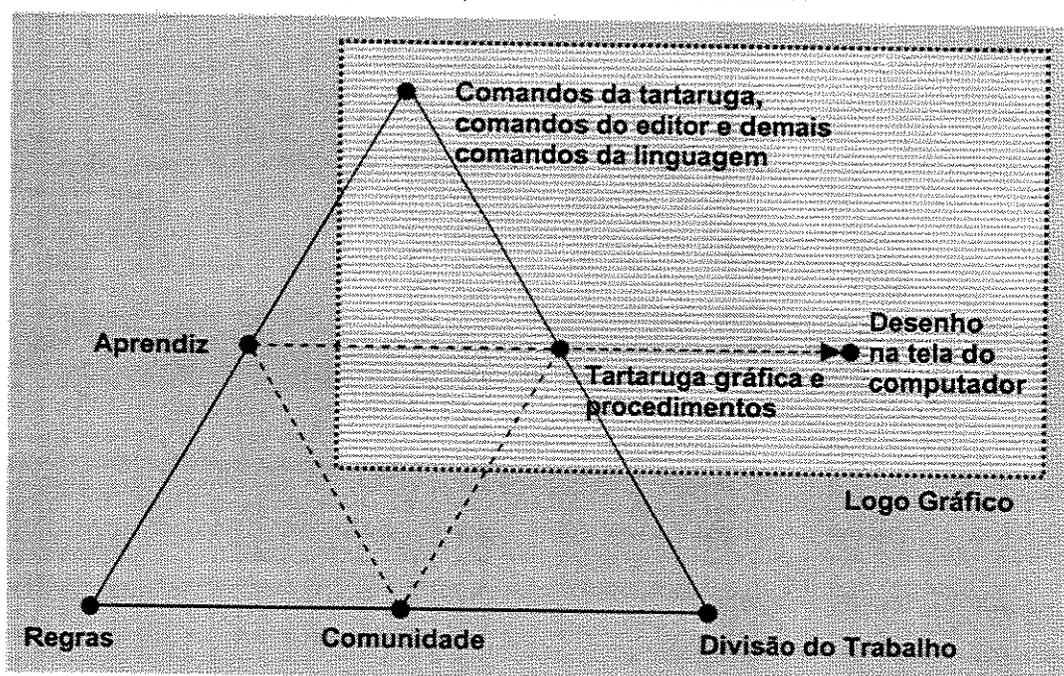


Figura 16 - O diagrama de Engeström adaptado para a atividade de aprendizado com o Logo Gráfico

O aprendiz tem como objetivo fazer um desenho na tela do computador. Este desenho e o programa associado servem de base para que ele reflita sobre o que está fazendo e também auxiliam o facilitador a determinar conceitos que precisariam ser esclarecidos.

Embora o Logo Gráfico em si não tenha recursos específicos para o intercâmbio de idéias (como envio de procedimentos por email, etc.) entre os participantes da atividade – daí a comunidade estar do lado de fora do retângulo marcado na figura – as regras costumam permitir que os aprendizes se sentem lado a lado, mostrem o que estão fazendo e discutam os avanços de seus projetos.

Além do Logo, embora não tenha sido representado no diagrama, os aprendizes trabalham com outros materiais como cadernos de anotações, tabelas de comandos, livros com desenhos, etc. que também fazem parte da atividade.

construcionistas (PRADO, 1996; PRADO e BARELLA, 1994; FREIRE e PRADO, 1995).

O ciclo também tem se mostrado uma ferramenta muito importante a ser utilizada na análise e desenvolvimento de software para atividades construcionistas. Por exemplo, analisando-se o Logo Gráfico a partir do ciclo, percebe-se que ele se diferencia de outras linguagens de programação pela facilidade com que os comandos são assimilados pela criança (devido à sua familiaridade), a relação direta entre os comandos e as ações executadas pela tartaruga e a possibilidade de visualização do andar da tartaruga pela tela.

No entanto, embora estas características sejam extremamente importantes para o bom andamento do aprendizado construcionista, elas têm sido deixadas de lado nos aplicativos mais modernos para a educação. A ênfase tem recaído mais em ferramentas que ajudem os aprendizes a construir produtos sofisticados do que em ferramentas que ajudem os aprendizes a refletirem sobre este processo de construção (VALENTE, 1995).

Considerações sobre o ciclo da descrição-execução-reflexão-depuração-descrição. Na nossa opinião, o ciclo oferece uma boa visão do tipo de ações efetuadas pelos aprendizes em uma atividade construcionista que use o computador, ainda mais no que se refere ao aspecto cognitivo desta atividade. No entanto, para o caso do desenvolvimento de novas atividades construcionistas que utilizem novos tipos de artefatos e explorem domínios diferentes dos possibilitados pelo Logo, acreditamos que há uma série de considerações que devem ser feitas:

a) O ciclo se baseia em uma atividade construcionista bem específica – a programação – onde, conforme discutido no capítulo 6, o computador realmente oferece um diferencial com relação a outros artefatos. No entanto, na nossa opinião, **não se deve confundir atividade construcionista com programação**. Como será explicado mais adiante, nem toda atividade construcionista envolve programação e nem toda atividade de programação é construcionista. A atividade educacional construcionista é uma atividade de design, que pode ser desenvolvida com diversos materiais diferentes, cada um com suas particularidades.

No caso do Logo, a atividade de aprendizado envolve a manipulação de comandos e objetos computacionais através da programação. No entanto, a atividade em si vai além disso, exigindo todo um processo de idealização, reflexão e interação social que ultrapassa o computador e a programação. A programação é apenas parte de um tipo específico de atividade construcionista.

Além disso, apesar de a programação ser algo que tradicionalmente diferencia o computador de outras máquinas, atualmente, com a integração das telecomunicações e dos recursos multimídia, a informática, por meio dos diferentes artefatos digitais, também vem destacando-se como instrumento de representação e comunicação de idéias. Conforme mostrado nos trabalhos de Shaw (1995) e Bruckman (1997), estes recursos representam outros aspectos em que o computador pode colaborar para a criação e a viabilização de atividades construcionistas.

b) Sustentando a idéia do aprendizado construcionista como resultado de uma atividade de design reflexivo, na nossa opinião, o ciclo dá **pouca proeminência à parte de idealização do produto e à distinção entre construção e descrição.**

Embora a idealização (ou externalização do que se pretende construir) seja algo que costume ser feito logo no início das atividades práticas construcionistas, em nenhuma fase do ciclo aparece um espaço para o aprendiz descrever o que pretende construir na atividade. Esta externalização dos objetivos é fundamental para que o aprendiz possa refletir mais concretamente sobre o que está fazendo e para que um facilitador possa determinar a melhor forma de agir.

Outro ponto a ser considerado é que, como será discutido mais adiante, a fase de descrição proposta no ciclo é muito genérica, misturando a construção do produto com a sua descrição. Apesar de que, na maioria dos casos, a construção e a descrição da construção sejam ações distintas, na atividade com o Logo Gráfico, elas tendem a misturar-se. Como a linguagem Logo é composta por termos de fácil compreensão, o programa é praticamente auto-descritível, o que o torna tanto produto como descrição da atividade.

No entanto, além de ser um elemento específico da programação, o programa representa uma descrição apenas parcial do processo de design em que o aprendiz está envolvido, não indicando por exemplo, as suas interações com os demais participantes da atividade, os problemas enfrentados ao longo da construção e as razões que levaram à tomada das decisões adotadas no programa. É por esta razão que construção e descrição deveriam ser diferenciadas.

c) Analisando sob a ótica da Teoria da Atividade, ainda que valorize muito a descrição como base para a reflexão sobre o processo, as ações do ciclo, assim como a literatura construcionista em geral, não ressaltam a **importância da meta-descrição.** Enquanto a descrição externaliza os passos tomados, a meta-descrição explica os motivos por detrás destes passos.

Na nossa opinião, a meta-descrição contribuiria para a atividade de duas maneiras: além de incentivar o aprendiz a refletir sobre as razões que o levaram a tomar determinadas ações, ela também ajudaria outros interessados, como os facilitadores, a compreender a importância da atividade para o aprendiz.

d) O ciclo também enfatiza muito o fato de o aprendiz interagir com o objeto "computador". De acordo com a visão proposta pela Teoria da Atividade, trabalhando com o Logo, o aprendiz não está centrado no computador em si, mas nos objetos que estão "por detrás da tela". O computador representa um ambiente no qual o aprendiz interage com o objeto tartaruga através de uma série de artefatos, isto é, os comandos da linguagem e os procedimentos criados. Os procedimentos e comandos também podem ser vistos como outros objetos disponíveis para manipulação via as facilidades de edição de texto disponíveis no ambiente. A **distinção entre artefatos e objetos** no Logo Geométrico é fundamental para que se possa tirar critérios para o desenvolvimento de novos artefatos para atividades construcionistas.

e) O ciclo enfatiza, principalmente, a relação do aprendiz com a máquina e não se aprofunda nas **relações dele com os demais participantes da atividade**. Assim como a maioria das pesquisas feitas com o Logo Geométrico, o ciclo retrata o aprendiz desenvolvendo atividades individuais, ainda que contando com o auxílio da comunidade. A comunidade, no ciclo, aparece de uma forma muito abstrata, sem explicitar claramente, por exemplo, quais as diferenças entre um facilitador, um pai ou um colega do aprendiz.

f) Por fim, outro ponto que talvez merecesse um melhor aprofundamento seria o desenvolvimento de **atividades construcionistas coletivas** onde diversos aprendizes compartilhassem um objetivo único que exigisse a participação de todos. Este tipo de situação é cada vez mais comum nos dias de hoje e cada vez mais viável em termos de tecnologia.

A proposta de um novo ciclo de ações construcionistas. A partir das idéias discutidas acima com relação ao ciclo da descrição-execução-reflexão-depuração-descrição, procurou-se descrever a dinâmica do aprendizado construcionista de uma forma que fosse mais genérica e abrangente do que a programação do Logo Gráfico e que servisse de base para uma análise posterior do papel do artefato computacional.

De uma forma geral, todas as atividades de design (ou de desenvolvimento de projetos) são compostas pelas seguintes ações:

- **Idealização**, onde o sujeito, imagina o que pretende construir.
- **Construção**, em que ele constrói na realidade, usando o material disponível no meio, uma versão do que havia idealizado.
- **Avaliação**, onde ele interpreta o que foi feito e compara com sua idéia original. A atividade termina quando o sujeito se der por satisfeito com o que produziu. Caso detecte coisas que ainda não estejam de acordo com o imaginado, ele passa para a depuração.
- **Depuração**, onde procura entender e resolver, a partir de reflexões sobre as ações anteriores, o que deu errado.

No Construcionismo, em se falando de uma atividade de design reflexivo, além das ações básicas descritas acima, há todo um esforço paralelo de **descrição** do que está sendo feito e do porquê (meta-descrição), isso com o objetivo de levar o sujeito a refletir sobre o processo em si e também ajudar na resolução dos possíveis problemas.

Como será abordado na seção 7.3, a descrição e a meta-descrição também facilitam o intercâmbio de idéias e ajudam o facilitador a intervir na atividade do aprendiz.

Deve ser notado que a seqüência com que as ações do aprendizado construcionista são apresentadas neste trabalho (idealização->construção->avaliação->depuração->descrição) é meramente didática e, como será abordado na seção sobre história e desenvolvimento, pode variar de caso para caso. Por exemplo, a descrição é uma ação que deve ocorrer em paralelo com as demais ações. Também é possível que,

em determinado momento, o aprendiz esteja construindo parte de seu projeto, idealizando outra, avaliando uma terceira e ainda depurando a parte restante.

Também deve ser lembrado que as ações levantadas acima estão associadas à sub-atividade de aprendizagem. As ações dos facilitadores e dos demais papéis da atividade educacional construcionista também deveriam ser consideradas em uma análise mais completa.

7.1.2.1 A Idealização

A Idealização é uma ação que representa um momento em que o aprendiz está concentrado em determinar que projeto ou idéia procurará desenvolver ao longo da atividade. Este projeto ou idéia pode surgir de alguma coisa que ele tenha visto em sua casa, algo que algum amigo tenha sugerido ou a partir de qualquer criação de sua fantasia.

A meta da Idealização, então, é determinar o produto esperado da atividade. É este produto que guiará todo o processo e dará condições para que o aprendiz e o facilitador consigam avaliar o andamento da construção. Sem imaginar aonde o aprendiz quer chegar, fica difícil saber se ele está caminhando para frente ou para trás.

No entanto, para que o facilitador tenha condições de intervir (ou mesmo para que o aprendiz não se esqueça de algum detalhe), é necessário que o objetivo da atividade seja descrito (externalizado) de alguma forma (um texto, um desenho, uma conversa, etc.). Normalmente, nas atividades com o Logo Gráfico, a descrição dos objetivos dos projetos é uma das primeiras coisas a ser pedida para os aprendizes e elas costumam ser feitas em folhas de papel especialmente preparadas para este fim.

Ainda assim, nada impede que o aprendiz mude de objetivo ao longo da atividade. Dependendo da situação, alguns aprendizes preferem ir identificando seus objetivos à medida em que vão conhecendo melhor os objetos do ambiente. Nestes casos, é importante que eles possam atualizar a descrição do objetivo sempre que necessário.

Embora não apareça explicitamente na literatura construcionista, na nossa opinião também é importante que o aprendiz externalize o motivo por trás do objeto que ele deseja construir. Conforme enfatizado pela Teoria da Atividade, toda atividade é dirigida a um objeto (no caso, o projeto do aprendiz) que concretiza uma necessidade ou desejo – o motivo (LEONTIEV, 1974). Enquanto a **descrição do objeto** permite o acompanhamento do desenvolvimento da atividade, a **descrição do motivo** permite saber até que ponto a atividade está sendo significativa e adequada para o aprendiz.

A coletânea das diversas descrições permitiria traçar o desenvolvimento do aprendiz através do tempo. Isso o ajudaria a refletir sobre os diversos caminhos percorridos e disponibilizaria maiores informações para o trabalho do facilitador. Esta questão voltará a ser discutida com mais detalhe no próximo capítulo.

7.1.2.2 Construção

Na Construção³⁸, a meta do aprendiz é construir, tornar real, o produto idealizado. Para isso, ele conta, no nível físico, com as ferramentas e objetos disponíveis no meio e, no nível mental, com os conceitos e estratégias que já possui sobre assuntos e problemas similares ao que está resolvendo.

A versão construída do produto só será de fato considerada como final depois que for submetida à avaliação pelo próprio aprendiz e for considerada satisfatória com relação ao produto idealizado.

Em paralelo à Construção, nas atividades construcionistas incentiva-se que o aprendiz descreva o que está fazendo, algo que nem sempre é óbvio de se descobrir pela simples observação do produto. Esta **descrição da construção**, explicando quais os elementos e os métodos usados, além de incentivá-lo a refletir sobre o que está fazendo, também serve de referência para uma futura depuração do projeto e ajuda os demais membros da comunidade a compreender o que está sendo feito.

Por outro lado, deve-se dosar bem a forma e o grau de descrição exigidos para não fazer com que o aprendiz perca sua motivação pelo processo de construção. Como será abordado mais adiante, à primeira vista, a descrição não parece contribuir para que ele atinja seus objetivos e, por isso, pode ser considerada como um peso para a atividade.

Tal como a descrição do motivo na Idealização, na nossa opinião uma **meta-descrição da construção** pode ser muito útil a um facilitador, indicando as razões que levaram o aprendiz a escolher estes elementos sendo usados e a combiná-los da forma em que se encontram.

Nas atividades com o Logo Gráfico, o produto é o programa desenvolvido pelo aprendiz. A descrição da construção costuma ser feita em um caderno à parte, o chamado “diário-de-bordo”, onde o mesmo registra os objetivos de cada sessão, o que foi feito e as principais dificuldades.

O programa Logo em si também é visto como uma forma de descrição do objeto e de como ele foi construído, na medida em que sua estruturação e a escolha dos comandos são representativos da abordagem escolhida pelo aprendiz. A meta-descrição raramente é escrita em algum lugar. Ela costuma aparecer nas discussões entre os aprendizes ou nas explicações que o aprendiz dá a um facilitador.

³⁸ A Construção aqui proposta equivale, basicamente, à fase de descrição no ciclo da descrição-execução-reflexão-depuração-descrição. Optamos pelo novo nome pelos seguintes motivos: a) a descrição é uma ação que também aparece em outras fases; b) no ciclo original a descrição aparecia vinculada à construção que o aprendiz estava fazendo no projeto.

7.1.2.3 A Avaliação

Em geral, quando se avalia algo, procura-se, de uma forma mais consciente ou menos, em primeiro lugar interpretar o que foi dado para depois comparar com alguma escala de valores. Na nossa opinião, na atividade construcionista acontece o mesmo: a Avaliação tem duas partes, uma de Interpretação e outra de Comparação.

A Interpretação. Na Interpretação³⁹, o aprendiz procura interpretar por si só, ou com auxílio de ferramentas ou outras pessoas, o que foi construído até o momento. A meta é levantar pontos que permitam comparar o produto da construção com o que fôra idealizado.

Por exemplo, no caso do Logo Gráfico, o aprendiz constrói um programa reunindo as operações da tartaruga. Este programa pode ser interpretado tanto por ele, imaginando o que cada operação faz, como concretamente pelo computador, gerando uma imagem na real na tela.

No caso da escrita de um texto, um tipo de interpretação seria pedir para alguém ler o texto e dizer o que entendeu. O próprio aprendiz poderia executar ler o texto, mas este seria, possivelmente, um julgamento parcial e tendencioso do que foi feito na Construção.

Normalmente, para a interpretação do produto, o aprendiz conta com alguns métodos e ferramentas que são mais ou menos adequados para cada momento da atividade. No trabalho com o Logo Gráfico, por exemplo, costuma-se fazer dois tipos de Interpretações: o brincar de tartaruga e o via computador.

O brincar de tartaruga é normalmente usado como uma maneira de introduzir os aprendizes aos comandos do Logo Gráfico (ACKERMANN, 1991, p.9) e também de ajudar a depurar o programa sendo desenvolvido. O aprendiz, com ou sem um colega, procura simular com seu corpo a execução dos comandos dados à tartaruga. Por exemplo, se dá ao seu colega o comando 'parafrente 10', o colega caminha 10 passos para a frente. Se o novo comando for 'paradireita 30', o colega irá virar 30 graus para a direita. Assim, ambos vão interpretando os comandos e familiarizando-se com as noções do Logo e da geometria de uma forma bem lúdica e pessoal.

No computador, é a tartaruga quem interpreta os comandos. É ela quem se deslocará na tela sob as ordens do aprendiz.

A Comparação. Na Comparação⁴⁰, o aprendiz compara o resultado da Interpretação com o que havia imaginado na Idealização. Nesta parte ele é forçado

³⁹ Valente (1993) chama de "execução" o que aqui entendemos por "interpretação". Acreditamos que o termo interpretação seja mais genérico – execução parece muito ligado à programação – e adequado para este estudo.

⁴⁰ A "comparação" equivale ao que Valente (1993, p.34) chama de "reflexão". Como o aprendiz reflete tanto durante a comparação quanto em outras ações, achou-se melhor escolher um outro termo.

a refletir sobre o que havia imaginado e sobre o produto que tem em mãos, aprimorando o seu conhecimento do assunto. Se decidir que o produto já se encontra em um estado satisfatório, o objetivo da atividade foi atingido. Se não estiver satisfeito, começará a depurar o produto.

Dependendo da ênfase dada na atividade, pode ser pedido que o aprendiz descreva os resultados obtidos na comparação, identificando os pontos que ele acredita que precisam ser melhorados e aqueles que já estão adequados. É função do designer da atividade propiciar artefatos que possibilitem ao aprendiz efetuar a comparação de modo que os conteúdos esperados do aprendizado sejam ressaltados.

Da mesma forma que a Construção é central para o desenvolvimento do produto, a Comparação é fundamental para o aprendizado dos conceitos relacionados com o projeto. É na Comparação que o aprendiz descobre que havia algo de errado com a abordagem que estava seguindo. As diferenças encontradas geram uma espécie de desequilíbrio cognitivo (ou *breakdown*, segundo a Teoria da Atividade) que, por sua vez, provavelmente irá causar uma reformulação (ou aprendizado) do conceito pelo aprendiz⁴¹.

7.1.2.4 A Depuração.

A meta da Depuração é, a partir de uma análise do que foi feito nas ações anteriores, levantar hipóteses para os problemas detectados na Avaliação. Dependendo da hipótese, o aprendiz poderá reformular sua idéia inicial, reconstruir parte do produto ou refazer a avaliação de outra maneira.

Em termos de descrição, seria interessante que o aprendiz documentasse as hipóteses levantadas na Depuração, as razões de cada hipótese e seus resultados. Nas atividades com o Logo Gráfico isso normalmente é feito no “diário-de-bordo”.

Ciclo fractal. Convém notar que o aprendiz raramente joga fora o que foi feito nas voltas anteriores do ciclo. Ele as reconstrói em cima das novas hipóteses geradas na Depuração. Aprofundando-se nesta idéia, percebe-se que cada nova hipótese gerada na Depuração dispara um pequeno ciclo de aprendizagem dirigido para o problema específico que está sendo trabalhado. É como se a nova atividade tivesse como objetivo não mais a idéia maior buscada inicialmente, mas sim o ponto problemático.

Fatores que facilitam a Depuração. De um modo geral, a Depuração pode ser facilitada quando se possibilita ao aprendiz

⁴¹ Conforme discutido no capítulo 5, no Construcionismo não existe a noção de certo ou errado; pelo menos não com a ênfase moral de bom ou ruim que é normalmente atribuída a estes conceitos na cultura ocidental. O “erro” do aprendiz é entendido apenas como um resultado que ele obteve e não estava de acordo com suas expectativas. E essa diferença entre resultados é benéfica no sentido de atrair sua atenção para algo que talvez ele não tivesse percebido e que, provavelmente, incentivará o aprendizado.

- estabelecer uma relação direta entre os elementos usados na Construção e os resultados da Interpretação;
- dividir o produto em partes significativas que possam ser exploradas independentemente e que possam ser combinadas gradualmente em novas unidades significativas;
- orientar-se na estrutura do produto, localizando os elementos procurados sem muita dificuldade. Essa facilidade depende tanto dos materiais e ferramentas usadas, quanto da qualidade das descrições e meta-descrições desenvolvidas nas demais ações;
- visualizar e gerenciar os caminhos percorridos e as hipóteses já testadas, podendo, inclusive, voltar atrás para o estado anterior a alguma hipótese que não tenha dado certo;
- efetuar modificações na Construção (e nos produtos das demais ações) sem ter que alterar tudo o que já foi feito desde o início;
- trocar idéias com pessoas mais experientes, buscando novos conceitos e estratégias.

Como será visto no próximo capítulo, o computador pode contribuir muito para a implementação destes fatores.

7.1.2.5 A Descrição

Como pode ser visto acima, a Descrição é uma ação efetuada em paralelo ao desenvolvimento do produto do aprendiz. Na nossa opinião, sua meta é, além de descrever o que o aprendiz está fazendo, também descrever os motivos por trás de suas ações.

Sob o ponto de vista da Teoria da Atividade, a descrição no aprendizado construcionista pode ser analisada como um artifício que leva o aprendiz a interromper sua atividade de design para externalizar seus pensamentos. Esta externalização, além de incentivá-lo a tomar consciência do que está fazendo no momento (reflexão na ação), também ajuda nas reflexões feitas à posteriori (reflexão sobre a ação), facilitando a localização e navegação no que foi produzido e a compreensão das decisões tomadas⁴².

O diário-de-bordo. Nos trabalhos com o Logo Gráfico, normalmente pede-se aos alunos, antes, durante e depois das sessões, que façam anotações em um caderninho – o “diário-de-bordo” – descrevendo seus objetivos para o dia, os problemas que estão enfrentando, as mudanças que foram efetuadas e as razões por trás delas. Espera-se que, com isso (e também com o auxílio de outros procedimentos e materiais), os aprendizes se tornem mais conscientes do que estão

⁴² Prado (1996) também discute a questão da reflexão na ação e da reflexão sobre a ação na prática construcionista.

fazendo e que os facilitadores obtenham alguma base mais concreta para orientar o seu trabalho.

Alguns pontos chamam a atenção no uso do diário-de-bordo:

a) Ele enfatiza tanto a descrição quanto a meta-descrição. A descrição aparece em perguntas do tipo “quais os planos para hoje”, “problemas que eu tive hoje” e “planos e idéias para o dia seguinte”. A meta-descrição aparece em tópicos como “quais os motivos das modificações efetuadas hoje” (ver Figura 18).

Projeto “Oficina de Jogos”	
Meu nome: <u>Zedrick R. Mendes</u>	Data de hoje: <u>03/03/97</u>
Problemas que eu tive hoje: <u>Não foi muito fácil fazer algos alguns, quando é</u> <u>a um mesmo, fácil fazer um desenho com um fundo</u> <u>muito bom. O problema é o tempo</u>	
Mudanças que eu fiz hoje e a razão destas mudanças: <u>Tentei mudar a peça “tira-torta” para conto, não</u> <u>o lago, mas não precisei porque o lago precisa pode</u> <u>ficar por cima</u>	
Planos e idéias para amanhã... <u>Continuar o acabamento da apresentação e</u> <u>fazer a lição.</u>	

Figura 18 - Um exemplo de diário-de-bordo usado na Oficina de Jogos (capítulo 9).

b) É uma solução genérica que pode ser usada tanto em trabalhos no computador como fora dele. No entanto, conforme ressaltado por Harel, as crianças costumam ter dificuldades de escrever no diário-de-bordo enquanto estão trabalhando no computador. Os principais problemas são a falta de espaço na mesa (elas têm que apoiar o caderno no colo), a dificuldade de mostrar as relações entre

o que estão descrevendo e o que estão fazendo no computador, a linearidade e gerenciamento das anotações.

O ideal, segundo ela, seria que o diário fosse implementado no computador. Isso resolveria o problema das anotações distanciadas e possibilitaria que as crianças fizessem anotações a todo instante e do jeito que quisessem (HAREL, 1991, p.382). Os aplicativos para atividades construcionistas poderiam oferecer, por exemplo, um mecanismo de anotações que abrisse uma janela para o aprendiz escrever, desenhar ou gravar suas observações quando quisesse. Nesse caso, o aplicativo também deveria oferecer formas de o aprendiz acessar e organizar estas anotações.

Motivação para Descrição. Embora seja um instrumento bastante valioso para o acompanhamento e a compreensão, tanto pelo aprendiz como pelo facilitador e pelo designer da atividade, a Descrição não é uma ação verdadeiramente essencial para o bom desenvolvimento do projeto do aprendiz. Por causa disso, é um esforço normalmente considerado chato e sem sentido por eles (principalmente para os iniciantes), exigindo tempo para provar sua utilidade.

No trabalho desenvolvido por Harel e Kafai isso tornou-se evidente. Ambas trabalharam durante vários meses, diariamente, desenvolvendo projetos em Logo com as crianças. No primeiro caso, as crianças demoraram cerca de 2 semanas para sentirem a utilidade do diário-de-bordo (HAREL, 1991, p.356). No segundo, elas pediram para parar de usar o diário-de-bordo após as férias (KAFAI, 1995, p.40).

Como a descrição pela descrição não é prazerosa, é preciso criar condições em que ela se justifique para o aprendiz. Fazendo-se uma analogia com o que a Teoria da Atividade fala a respeito da externalização (ver capítulo 5), a Descrição se tornaria intrinsecamente motivante para o aprendiz caso

- O projeto fosse longo ou complexo demais e o aprendiz precisasse de suas anotações para se orientar;
- O aprendiz identificasse algum erro que implicasse em alguma revisão passo-a-passo do processo;
- O projeto precisasse ser comunicado a alguém, seja devido a uma apresentação para colegas ou a um desenvolvimento em equipe.

Em outras palavras, dificilmente projetos muito simplificados e individuais oferecerão estímulo para a Descrição. O designer da atividade deve levar em consideração os fatores mencionados acima na criação de situações de aprendizado agradáveis e efetivas.

Porém, não é demais ressaltar que, mesmo nestas condições, o esforço gasto na descrição deve estar bem integrado às demais ações do aprendiz. O ideal seria que ele não tivesse que largar o que estivesse fazendo toda vez que parar para descrever suas ações, fato que ocorre no uso do diário-de-bordo feito em um caderno a parte. A descrição (e a reflexão) deveria ser vista como uma ação natural da atividade do aprendiz.



Outro ponto importante a ser considerado é que, para que a Descrição seja bem feita, é preciso que seja formada toda uma **cultura incentivadora**. É possível (e deve-se) criar mecanismos e ferramentas que facilitem externalização (como o diário-de-bordo), mas o seu bom uso dependerá, exclusivamente, da boa vontade do aprendiz. Nenhum artefato é capaz de garantir a veracidade e a profundidade de expressão por parte do seu usuário – isso ficou aparente no capítulo 9. Se for imposta sem nenhuma razão aparente, a descrição (e, principalmente, a meta-descrição) dificilmente representará o que se passa na mente do aprendiz⁴³.

Por fim, como a descrição do que foi feito serve de base para a atividade do facilitador e do designer da atividade (e talvez até para outros participantes da atividade), deve-se tomar cuidado para não sobrecarregar o aprendiz com a produção de coisas que fujam do escopo do aprendizado. Cada uma destas atividades tem necessidades específicas, como a coleta de dados estatísticos ou amostras de trabalho, que devem ser supridas para o bom andamento da educação, mas que não estão diretamente relacionadas às ações esperadas do aprendiz⁴⁴.

Idealmente, tal como descrito acima, o desenvolvimento da atividade educacional deveria considerar todas estas necessidades e oferecer meios para que elas pudessem ser atendidas sem interferir uma com as outras.

7.1.3 O nível das operações

Tal como descrito na Teoria da Atividade, só é possível refletir sobre as operações uma vez que elas já tenham ocorrido. Isso coloca uma série de dificuldades para que o aprendiz reflita sobre o que fez e para que o facilitador e o designer da atividade possam analisar as operações de cada aprendiz.

Essa problemática pode ser bastante reduzida com o auxílio de pessoas ou ferramentas que funcionem como observadores externos, registrando e depois relatando o que foi feito. Neste caso, apenas as ações e operações externas (não as mentais) poderão ser acessadas e, por mais completa que seja a documentação destas operações, nenhuma conclusão fechada poderá ser tirada.

Em se tratando de atividades em grupo, o aprendiz pode refletir com os companheiros sobre o que foi feito. O problema é que as interpretações podem ser parciais (ver capítulo 8). Na maior parte dos casos, principalmente em situações difíceis de se reproduzir, uma câmera de vídeo pode ajudar bastante. A câmera pode ser deixada em um lado do ambiente ou ser levada para filmar partes

⁴³ Algumas versões do Logo Gráfico permitem que o aprendiz escreva comentários ao lado do programa. Editores de texto como o MS-Word também oferecem facilidades para o que os autores ou revisores insiram anotações a respeito do que estão escrevendo. Estas anotações *in locu* reduzem as limitações dos diários de bordo tradicionais no que toca a referenciar partes específicas de trabalhos feitos no computador. Por isso, na nossa opinião, deveriam ser melhor consideradas no design de novos software para atividades educacionais.

⁴⁴ Este é um exemplo de conflito entre atividades relacionadas.

específicas da atividade. Em seu trabalho, Harel descreve diferentes modalidades de uso da câmera de vídeo e como elas contribuíram para o desenvolvimento de sua pesquisa (HAREL, 1991, p.368).

Na nossa opinião, além de ajudar pesquisadores, o uso de uma câmera também pode ajudar muito aprendizes e facilitadores a reverem a sua prática.

7.1.4 Conceitualização e operacionalização

Idealmente, em se falando de aprendizado construcionista, todas as conceitualizações (tomadas de consciência) deveriam estar relacionadas ao tema que está sendo aprendido e ao próprio processo do aprendizado. As demais conceitualizações, relacionadas aos artefatos em si ou a qualquer outra coisa, deveriam ser minimizadas ou resolvidas o mais rapidamente possível.

Segundo Valente (1993, p.34), os problemas detectados na Depuração estão normalmente relacionados a:

- dificuldades inerentes às ferramentas utilizadas;
- dificuldades inerentes ao domínio conceitual do problema;
- dificuldades inerentes às estratégias utilizadas.

Sob o olhar da Teoria da Atividade, as dificuldades inerentes às ferramentas estão relacionadas com os aspectos físico e de manuseio dos artefatos e, portanto devem ser minimizadas (ver capítulo 4). As demais dificuldades, inerentes ao domínio e às estratégias, estão ligadas aos aspectos dirigidos ao objeto/sujeito e é nelas que a atividade deve se concentrar.

Por exemplo, em uma atividade com o Logo Gráfico, todas as conceitualizações deveriam estar ligadas a conceitos típicos da geometria, como ângulos, distâncias, coordenadas etc.. As conceitualizações relacionadas com os nomes dos comandos, operações de salvamento de arquivos, lentidão de processamento, falta de material disponível, regras incoerentes etc. adicionam maior complexidade à atividade e tendem a desviar a atenção do aprendiz quanto ao objetivo educacional.

É claro que não se pode esperar que, logo na primeira vez em que trabalhar com a ferramenta, o aprendiz não encontre dificuldades com os comandos dados e as características específicas do software. E também não se pode esperar que, pelo simples fato de ele já dominar a funcionalidade básica do software, não encontre mais nenhum problema e acabe aprendendo todos os conceitos desejados. Como será visto na seção 7.4 e no capítulo 9, a aquisição da fluência na ferramenta e no domínio em questão exige tempo e segue todo um processo que deve ser considerado na criação do ambiente educacional.

De qualquer modo, é função do designer da atividade se preocupar em reduzir ao mínimo os problemas relativos aos aspectos físico e de manuseio e isso durante toda a evolução esperada do aprendiz na atividade. Nas situações em que o artefato não der conta dos níveis de competência e problemas esperados, pode ser oferecido treinamento e material de apoio para complementar o suporte aos participantes da atividade.

O mesmo também vale para os facilitadores. Enquanto os designers da atividade trabalham a nível macro com a atividade, são os facilitadores que lidam com os casos específicos que surgem na execução da mesma. Nas situações em que, por exemplo, o aprendiz esteja escrevendo o nome errado de um comando do Logo (como “para_frente” ao invés de “parafrente”), ao invés de ficar esperando que ele descubra a sintaxe correta, algo que foge do domínio da geometria, o facilitador pode intervir e mostrar como se escreve o comando.

Idealmente, não deveria ser dado ao aprendiz a possibilidade do erro de sintaxe (que é uma espécie de conceitualização relacionada ao aspecto de manuseio dos comandos e dos objetos), a não ser que a conceitualização sintática também fosse um dos objetivos da atividade – algo que seria perfeitamente aceitável se o desejado fosse explorar, por exemplo, os problemas sintáticos dos comandos do Logo em português. Isso poderia ser resolvido, no Logo, com a inserção de um menu onde o aprendiz pudesse simplesmente selecionar o comando desejado dentre os disponíveis.

Nos casos em que o erro for inevitável, o software (ou o artefato) deve oferecer recursos como sistemas de ajuda e mensagens de erro claras e específicas para que o aprendiz identifique, facilmente, o que aconteceu e possa voltar à situação normal, sem maiores dificuldades. A maior parte das versões do Logo Gráfico deixa muito a desejar neste aspecto, apresentando mensagens de erro genéricas que, dificilmente, ajudam o aprendiz a identificar o erro cometido. Por exemplo, como pode ser visto na Figura 19, a mensagem de erro “Ainda não aprendi parabaixo” não esclarece se o usuário cometeu um erro de sintaxe ou se o comando “parabaixo” não existe, que é a verdadeira razão do erro.

Ao mesmo tempo em que procura minimizar a conceitualização dos aspectos físico e de manuseio da ferramenta, a atividade com o Logo Gráfico traz uma série de mecanismos que incentivam a conceitualização dos conceitos da geometria e do processo de design.

A primeira se dá pela utilização dos comandos da tartaruga geométrica, que realçam a geometria nos elementos familiares ao aprendiz. Por exemplo, todas as pessoas caminham de um lado para o outro sem se preocupar com ângulos e distâncias. Trabalhando com Logo, o aprendiz se torna consciente destes conceitos quando é obrigado a especificá-los nos comandos que movimentam a tartaruga na tela. A mudança do mundo real para o da tartaruga, com seus comandos e idiossincrasias, é que acaba por incentivar a conceitualização – a reflexão na ação – destes conceitos.

A segunda, conforme discutido anteriormente, é reforçada pelas descrições que o aprendiz tem que fazer, principalmente no diário-de-bordo. De certa forma, o artefato computacional também poderia contribuir para a conceitualização do processo, criando situações que o “forcem” a descrever o que está fazendo. Por exemplo, o computador poderia exigir que ele escrevesse um comentário antes de executar determinadas ações tais como iniciar um projeto, fazer alguma alteração, etc.. Alguns editores de texto abrem uma janela toda vez que o usuário vai gravar

um novo arquivo, pedindo que se faça uma pequena descrição do que foi escrito e quais as palavras-chave.

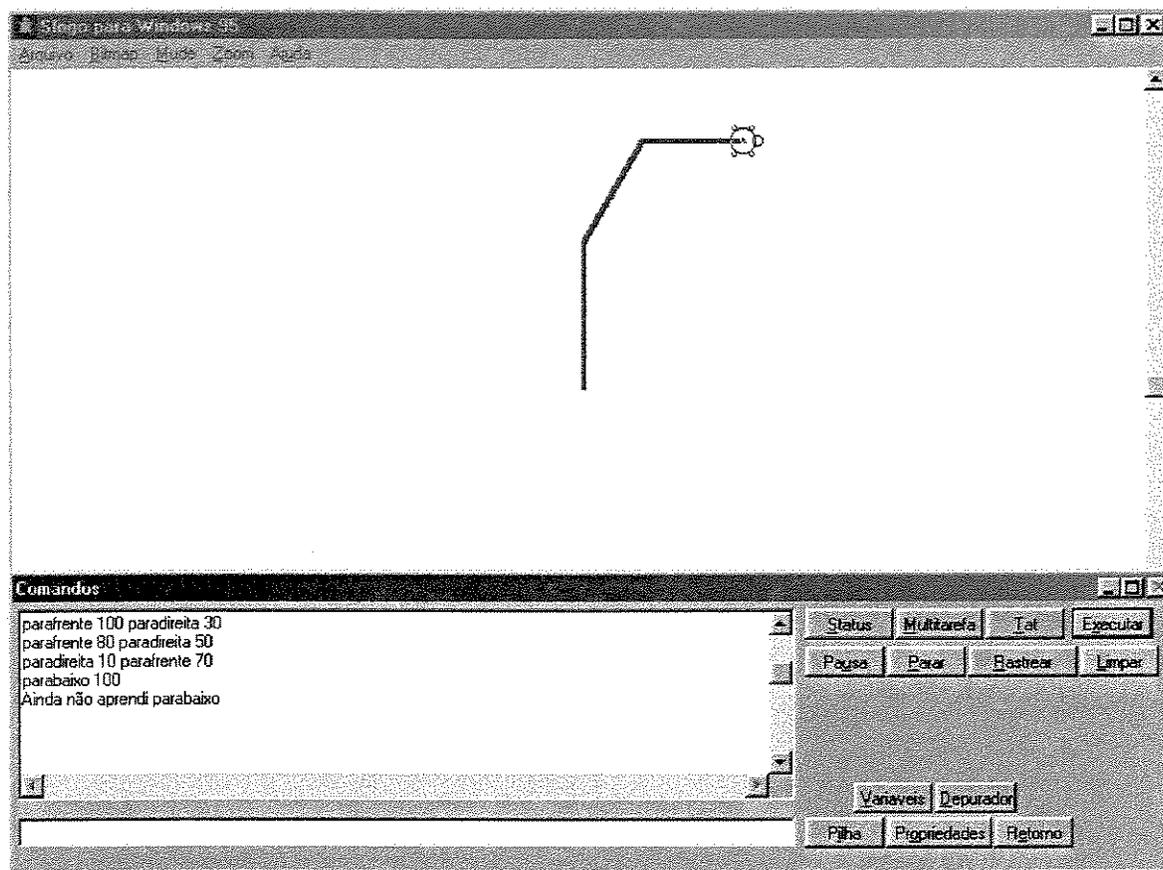


Figura 19 - A mensagem de erro "Ainda não aprendi parabaixo".

De uma forma mais sutil, o computador também poderia emitir mensagens para o usuário, indicando pontos que mereceriam maior atenção. Nenhuma destas alternativas poderia garantir a conceitualização, no entanto. O melhor mesmo seria criar uma cultura e atividades que motivassem os aprendizes a refletir sobre o que estão fazendo por livre e espontânea vontade.

Em linhas gerais, o objetivo das atividades ou micromundos construcionistas é criar situações propícias para a conceitualização dos princípios do domínio que está sendo aprendido. Com o correr da atividade, espera-se que os aprendizes interajam o suficiente com estes conceitos de forma que eles se tornem cada vez mais operacionalizados. Aí sim, pelo menos nas situações abordadas, há maior probabilidade de o aprendizado se concretizar.

7.2 Orientação a objetos

Como apresentado no capítulo 3, a orientação a objetos é uma dimensão da Teoria da Atividade que aborda as relações entre os elementos que compõem o ambiente da atividade. Mais especificamente, se preocupa em determinar que propriedades dos objetos são mais ou menos relevantes e de que forma elas interferem no desenvolvimento da atividade.

A Teoria da Atividade, por estudar atividades intencionais humanas, enfatiza muito o fato de toda atividade ter uma finalidade. É o produto, o objeto produzido pelo sujeito, que a motiva e, sem ele, a atividade perde todo o seu sentido. No Construcionismo, a motivação do aprendiz também está muito relacionada ao produto de seu projeto. Como foi visto no capítulo 5, espera-se que o aprendiz vá explorando o domínio que está sendo estudado à medida em que desenvolve algo significativo, compartilhável e, de preferência, que lhe seja útil.

Por outro lado, o Construcionismo enfatiza que, além da construção de objetos pessoalmente significativos, a motivação do aprendiz também é influenciada pela possibilidade de ele poder atuar no seu próprio estilo e ter seu trabalho reconhecido pelas pessoas à sua volta (dimensão sintônica). Para o Construcionismo, o processo em si também influencia a motivação do aprendiz. Nesse sentido, os artefatos e objetos do meio devem oferecer diferentes possibilidades de combinação e uso para que o aprendiz possa expressar-se da forma que mais lhe convier (PAPERT, 1990) e divulgar o seu trabalho (HAREL, 1991).

Segundo a Teoria da Atividade, além das propriedades físicas, biológicas e químicas, as propriedades culturais dos objetos também influenciam o tipo de interação da atividade. O Construcionismo enfatiza bastante esta idéia, frisando a importância de o aprendiz identificar-se e sentir-se bem com os materiais que está lidando (dimensão sintônica).

A própria noção de objetos transicionais (dimensão semântica) se baseia inclusive na familiaridade cultural do aprendiz com alguns atributos dos objetos da atividade como base para a exploração de novos conceitos. Esse é o caso, por exemplo, das engrenagens que levaram Papert a refletir sobre mecânica na sua infância, ou da própria tartaruga do Logo Gráfico com que os aprendizes se identificam.

A importância do tema. Convém notar que há sempre uma possibilidade de algumas das características motivantes da atividade chamarem por demais a atenção do aprendiz e acabarem por distraí-lo do objetivo educacional para o qual ela fora planejada. Para reduzir a ocorrência destas situações, é muito importante a escolha de temas autênticos, em que a utilização dos novos conceitos já seja por si só intrinsecamente motivante para o aprendiz.

Segundo Kafai (1995, p.296), diferentes temas oferecem diferentes graus de liberdade para os aprendizes, o que pode fazer com que conceitos esperados acabem por não serem explorados. Essa conclusão tornou-se patente nos resultados de um projeto conduzido por ela em que os aprendizes deveriam desenvolver, com o Logo Gráfico, jogos que ensinassem conceitos de frações para

crianças menores. No fim do projeto, foram criados vários jogos mas, a maior parte deles tinha motivação extrínseca, isto é, os conceitos de frações eram apresentados, principalmente, de uma forma descontextualizada e, em muitos casos, usados como obstáculos a serem vencidos para que o jogador ganhasse os prêmios do jogo.

Por outro lado, os temas também não podem ser muito restritos para não limitar a liberdade de exploração dos aprendizes. O designer da atividade deve buscar um equilíbrio que leve o aprendiz a explorar os conceitos esperados dentro de seu interesse e sem se sentir reprimido.

Na nossa opinião, uma boa forma de se obter temas intrinsecamente motivantes para os aprendizes é observar situações do cotidiano deles, onde o domínio estudado seja normalmente aplicado. A atividade deveria incentivar a exploração dessas situações reais e oferecer materiais adequados para isso. Por exemplo, se as frações aparecem na vida dos aprendizes quando eles vão ao mercado, quando têm que preparar uma receita ou alterar a escala de um desenho, a atividade deveria girar em torno deste tipo de temática. Frações não têm nada que as relacione intrinsecamente com a produção de jogos. Talvez seja por isso que elas tenham aparecido tão fora de contexto na pesquisa de Kafai⁴⁵.

Disposição e disponibilidade dos objetos. Além de motivantes e de possuírem características culturais que afetam a interação do sujeito com o mundo, os objetos também constituem a substância com a qual será gerado o produto. A sua maior ou menor disponibilidade também interfere, diretamente, no que poderá ser produzido – sem material suficiente, por mais capaz que seja o sujeito, ninguém consegue construir nada.

A disponibilidade junto aos objetos é afetada por uma série de fatores que vão desde a dificuldade de aquisição (como custo, etc.), manipulação (devido ao tamanho, peso ou fragilidade do objeto), etc. até os horários de funcionamento do local onde será efetuada a atividade. É preciso tempo para que o aprendiz possa explorar domínio, fazer suas experimentações e assim, se desenvolver.

Diversas atividades construcionistas realçam a importância da arquitetura do ambiente nas interações entre seus participantes. Diferente do modelo clássico de aulas em que os alunos ficam sentados lado a lado, quietos, ouvindo um professor falar, o aprendizado construcionista incentiva a criação de ambientes onde haja muita ação e interação entre aprendizes, facilitadores e os objetos do ambiente.

No Projeto *Headlight*, local onde foram desenvolvidas várias pesquisas construcionistas, os computadores foram colocados em uma espécie de pátio fechado entre as salas de aula, dispostos em círculos, com as telas viradas para o

⁴⁵ Outra hipótese para esta questão das frações talvez tenha sido o fato de o conteúdo delas ter sido trabalhado de forma não integrada com o desenvolvimento dos jogos e, provavelmente, distanciada da experiência dos alunos. Caso o aprendizado das frações fosse mais contextualizado, provavelmente elas apareceriam mais intrinsecamente ligadas à temática dos jogos criados.

lado de fora. Segundo Yasmin Kafai, pesquisadora do projeto, esta disposição dos computadores em grupos enfatizou ainda mais os aspectos comunitário e cooperativo do aprendizado, dando maior oportunidade para os alunos verem os projetos dos outros estudantes (KAFAI, 1995, p.34).

Como será discutido no capítulo 9, a preparação da atividade construcionista da Oficina de Jogos também levou em consideração a disposição das mesas e computadores na sala. O reflexo disto pôde inclusive ser notado no trabalho dos alunos.

7.3 Internalização e externalização

No Construcionismo não existe um conceito tão explícito, como o de internalização e externalização da Teoria da Atividade, relacionando as atividades externas com as internas do sujeito. Enquanto a Teoria da Atividade se fundamenta nas idéias de Vygotsky, que se concentram, principalmente, na influência do social (relação do sujeito com os demais sujeitos), o Construcionismo é derivado de Piaget⁴⁶ e a ênfase é dada aos elementos culturais (relação do sujeito com os objetos e ferramentas do ambiente).

Conforme discutido no capítulo 5, isso não significa que o Construcionismo negue a importância do social na atividade de aprendizado. As relações entre o aprendiz e a comunidade são importantes sim, principalmente no que se refere ao intercâmbio de idéias e à motivação oriunda do reconhecimento do trabalho. A diferença é que, ao invés de partir de uma análise social para desenvolver o resto do trabalho, as pesquisas iniciais construcionistas se preocupavam, principalmente, com o desenvolvimento de ferramentas individuais e os aspectos sociais acabavam por aparecer como algo externo e quase que secundário, incentivado pelo facilitador e pelas regras do ambiente. O próprio papel do facilitador só passou a ser valorizado na década de 80, sendo que, inclusive, Papert discute essa “deficiência” no prefácio da segunda edição de seu livro *“Mindstorms: children, computers and powerful ideas”* (PAPERT, 1993a).

7.3.1 Internalização

O equivalente à internalização, no Construcionismo, é o mecanismo de assimilação e acomodação de Piaget, através do qual o sujeito vai desenvolvendo modelos mentais cada vez mais precisos do mundo à medida em que vai interagindo com situações novas e resolvendo os conflitos cognitivos gerados por elas. Enquanto a Teoria da Atividade enfatiza o caráter social, o Construcionismo não parece

⁴⁶ Em sua tese de doutorado, Harel apresenta uma discussão mais detalhada sobre os pensamentos de Vygotsky e Papert (HAREL, 1991).

diferenciar a natureza destas situações: a social é apenas mais um tipo de interação do indivíduo com o mundo.

Assim como Piaget, o Construcionismo também acredita que esta “internalização” depende do estágio de desenvolvimento cognitivo do sujeito. Posto de modo simplificado, isso significa que o nível de compreensão esperado de uma criança na atividade será diferente do nível de compreensão esperado de um adulto. Há uma série de diferenças entre o Construtivismo piagetiano e o Construcionismo de Papert que, no entanto, fogem do escopo deste trabalho e podem ser melhor compreendidas no trabalho de Ackermann (1990).

É interessante notar como, nesse aspecto de internalização, o Construtivismo e a Teoria da Atividade parecem se complementar. O primeiro, focando, principalmente, nos estágios de desenvolvimento cognitivo de um sujeito genérico em relação com percepções genéricas do mundo, independente de situações específicas. O segundo, focando, primordialmente, nas interações sociais e nos efeitos causados pelas diferentes situações culturais.

Ainda que estas teorias apresentem uma série de pontos de discordância (que extrapolam o escopo deste trabalho), o Construcionismo de Papert parece se situar em algum lugar entre elas, combinando o aprendizado piagetiano, onde o aprendiz vai construindo conhecimento à medida em que interage com o mundo e de acordo com seu estágio cognitivo e o aspecto sociocultural da Teoria da Atividade, na criação de micromundos propícios para a construção de determinados tipos de conhecimento.

7.3.2 Externalização

A externalização é provavelmente um dos aspectos mais importantes da atividade construcionista. Partindo do princípio de que o aprendizado se dá pela interação do sujeito com o mundo, os construcionistas acreditam que ele ocorre especialmente bem quando o aprendiz constrói algo concreto e, ainda por cima, reflete sobre o processo. Essa reflexão é facilitada se o aprendiz tiver meios de visualizar a estrutura de seu projeto e as estratégias seguidas ao longo de seu desenvolvimento, como acontece nos projetos que usam o Logo e o diário-de-bordo.

Junto com a descrição textual ou esquemática do processo, como é o caso dos programas Logo, o Construcionismo também valoriza muito a discussão entre os participantes da atividade (PAPERT, 1993b, p.92). Segundo Harel (1991, p.362; HAREL e PAPERT, 1991, p.76), a discussão leva os aprendizes a refletir sobre outras formas de pensar e de expressar seus pensamentos. Isso propicia uma visão mais objetiva e descentralizada da realidade.

Além da correção e do aperfeiçoamento das ações do aprendiz, a externalização no Construcionismo também se enquadra nas demais razões levantadas por Vygotsky para a externalização (ver capítulo 3). As descrições ajudam a orientar o aprendiz ao longo dos projetos desenvolvidos (principalmente no que se refere à visualização passo-a-passo e na execução de cálculos complexos) e ajudam na coordenação das atividades dando, por exemplo, *feedback* ao facilitador e ao designer da atividade com relação ao andamento das coisas.

No entanto, Papert e Harel atentam para o fato de que todas estas interações do sujeito com o meio e as possibilidades de externalização (discussão, etc.) estão sujeitas a **questões políticas**: além de prover material cultural adequado e suficiente para a atividade, é preciso que a organização da atividade dê tempo para as explorações do aprendiz e incentive as discussões e trocas de idéias.

A liberdade de tempo e ritmo são essenciais para que o aprendiz desenvolva seu próprio estilo e possa perceber o que está acontecendo à sua volta (HAREL e PAPERT, 1991, p.68). O **tempo prolongado** também propicia melhores condições para que o facilitador conheça os aprendizes.

7.4 História e desenvolvimento

História. Um dos princípios fundamentais defendido pela Teoria da Atividade é que cada atividade tem um desenvolvimento não linear e específico, resultado de uma série de eventos, planejados ou não, que ocorreram ao longo de sua história. A boa compreensão da essência de uma atividade exige que esta atividade seja analisada frente às diferentes conjecturas que levaram à sua transformação e que diferentes pesos sejam usados em cada situação.

No caso específico do Construcionismo, é interessante notar que a teoria foi nascendo, principalmente, ao redor de estudos efetuados em cima do Logo. Na realidade, ao contrário do que possa parecer, o Logo foi desenvolvido antes do próprio termo Construcionismo ser cunhado. Teoria e software foram desenvolvendo-se lado a lado, com os estudos feitos em cima de um, ajudando e sendo completados pelos conceitos do outro.

É muito importante compreender o contexto histórico em que o Logo foi criado para que se possa perceber quais de suas características são, realmente, essenciais e quais são resultado de limitações específicas de uma época. Do mesmo modo, é importante compreender o contexto atual para aproveitar os recursos e valores correntes no desenvolvimento de atividades construcionistas mais adequadas e efetivas. Também é importante considerar as tendências futuras para incentivar o desenvolvimento do que se achar mais pertinente.

Por exemplo, na década de 70, quando as primeiras versões do Logo Gráfico foram criadas, os recursos computacionais eram extremamente limitados em relação aos atuais. Os computadores não tinham poder suficiente de processamento para trabalhar com recursos gráficos e de áudio tão sofisticados e a comunicação, via computador, não atingia as dimensões e as facilidades de acesso à informações e de troca de idéias possibilitadas pela Internet.

Ainda que a maioria das versões atuais do Logo mantenha boa parte da aparência que tinha no início da década de 80, isso não significa que o Construcionismo seja contra a utilização de multimídia e das redes. Embora estas características não

fossem vislumbradas até alguns anos atrás, aos poucos elas vão amadurecendo e mostrando seu potencial educacional.

Por outro lado, apesar de o Logo ter quebrado uma série de barreiras, trazendo novos conceitos culturais e tecnológicos (como a interface gráfica e o conceito de programação para crianças) para uma época onde o computador era usado, na maioria das vezes, por um pequeno grupo de especialistas para fazer cálculos em algumas indústrias e centros de pesquisa, não significa que as inovações tecnológicas possam ser inseridas sem critérios no ambiente educacional. É preciso, novamente, usar os princípios teóricos como orientação e vesti-los da melhor forma possível nas roupagens dos novos produtos e costumes.

Desenvolvimento. Enquanto a dimensão “história” analisa a evolução da atividade construcionista como um todo, a dimensão “desenvolvimento” permite acompanhar o que acontece com os sujeitos ao longo das atividades, ressaltando suas mudanças de perspectiva, seus problemas e suas necessidades mais comuns.

O aprendizado leva tempo. Em primeiro lugar, deve ser enfatizado que o aprendizado, segundo o Construcionismo, não é algo instantâneo em que basta imergir o aprendiz em um ambiente e em uma dinâmica específicos para que ele aprenda tudo de uma única vez. Para o Construcionismo, o aprendizado também não caminha em linha reta, de um nível de conhecimento para outro. Ele exige tempo para que o aprendiz possa ir explorando, gradualmente, os novos conceitos.

Para que isso dê certo, é preciso não só que a atividade possibilite este tempo de exploração, através de suas regras e da disponibilização de material suficiente, como também que o próprio aprendiz se aceite como um sujeito em formação e tenha a paciência necessária para suportar o processo até o final.

Em certos momentos, é muito difícil para um facilitador perceber quem precisa de mais tempo em um determinado assunto e quem precisa de uma visão mais ampla do que está sendo aprendido. Nestes casos, a duração longa dos projetos ajuda os próprios aprendizes a resolverem este problema e, ao mesmo tempo, cria condições para que facilitadores e aprendizes consigam aproximar-se mais (HAREL, 1991, p.366).

Também é importante que as ferramentas dêem suporte para o desenvolvimento dos aprendizes (e para os demais papéis da atividade) através das múltiplas sessões que, provavelmente, farão parte da atividade educacional. Como será visto no capítulo 8, embora este tipo de funcionalidade não costume estar presente nos aplicativos para a educação, o computador pode contribuir muito para o desenvolvimento de trabalhos longos.

Instrumentalização e fluência. Em segundo lugar, como foi comprovado na Oficina de Jogos (ver capítulo 9) é preciso que sejam dadas condições para que o aprendiz se familiarize com as ferramentas antes de começar a criar. Segundo Papert, é preciso fluência para que o aprendiz consiga expressar-se abertamente. Existe uma diferença muito grande entre saber desenhar as letras do alfabeto e conseguir descrever textualmente uma idéia. A fluência exige toda uma prática e uma experiência desenvolvida em situações diversificadas.

Apesar disso, a literatura construcionista não enfatiza como se dá a aquisição da fluência. Por outro lado, a Teoria da Atividade parece abordar de modo adequado esta questão. Em especial, o trabalho de Bodker (ver capítulo 4) indica, claramente, as etapas que o sujeito passa até operacionalizar os instrumentos e, assim, conseguir se concentrar no domínio em questão. Esta pesquisadora enfatiza que o desenvolvimento dos artefatos deve levar toda esta evolução do sujeito em consideração e suportá-lo em seus diferentes níveis de competência.

Na nossa opinião, o desenvolvimento de artefatos construcionistas também deveria seguir por este caminho e oferecer, principalmente, mais recursos para facilitar os primeiros contatos do aprendiz com as ferramentas. Mesmo os comandos básicos do Logo Gráfico precisam ser aprendidos para que os aprendizes consigam desenvolver projetos que lhes sejam significativos. Não se pode esperar que eles se sentem na frente da máquina e já saiam programando. Como o objetivo do Logo não é fazer com que os aprendizes se distraiam, tentando descobrir quais são os comandos disponíveis, recursos como tutoriais, mecanismos de ajuda e tabelas explicativas dos comandos poderiam facilitar a passagem desta fase instrumental da atividade.

Estilos de aprendizado. Conforme destacado no capítulo 5, segundo o Construcionismo, cada pessoa tem um estilo próprio de aprender, que varia de situação para situação entre algo mais “planejador” e algo mais “escultor” (TURKLE e PAPERT, 1991). É interessante notar que, embora a literatura estudada sobre a Teoria da Atividade fale bastante sobre atividades, ações e operações, nada foi encontrado com relação a diferenças de ênfase ou a possíveis padrões que diferenciariam os sujeitos de uma forma geral, seja por gênero, estágio cognitivo ou qualquer outro fator, na execução de uma mesma atividade. Este é um ponto em que o Construcionismo pode colaborar com a Teoria da Atividade.

Ainda assim, embora o Construcionismo defenda a liberdade de estilos, historicamente sua teoria se prendeu muito à expressão via programação que, embora flexível no nível de construção, é extremamente lógica e abstrata, favorecendo, de certa forma, o pensamento planejador. Os estudos mais recentes, por outro lado, estão demonstrando a importância de que os aprendizes tenham liberdade de se expressar também via narrativa.

Segundo Kafai (1995, p.85), quando o aprendiz situa o seu projeto frente a uma história (textual ou não, como aconteceria na criação de um jogo de computador), ele acaba desenvolvendo aspectos fantásticos e pessoais que complementam a formalidade e a lógica da programação. A narração parece ser uma forma fácil e eficiente de dar um contexto para o projeto e tornar a tarefa toda mais relevante.

Nesse sentido, aplicativos construcionistas como o Logo, que são conhecidos principalmente como ambientes de programação (e não de design), deveriam ser mais abertos e permitir a escrita de textos e outras formas de narrativa que permitissem uma melhor contextualização do programa feito.

7.5 Mediação

A Teoria da Atividade levantou um conceito extremamente importante a ser considerado na criação de atividades educacionais: a **não neutralidade dos artefatos**. Os artefatos podem ser tanto potencializadores quanto limitadores das ações do sujeito e eles trazem dentro de si toda uma cultura que foi amadurecendo ao longo de seu desenvolvimento e que influencia no seu modo de uso.

Ainda assim, não se pode dizer que artefatos sejam educacionalmente “bons” ou “ruins” sem que se leve em consideração os demais elementos da atividade em que eles estiverem sendo usados. O próprio Logo, que é um software tipicamente voltado para atividades construcionistas, também pode ser usado de maneira mais tradicional.

Cabe ao designer da atividade, com o auxílio dos demais participantes, analisar o contexto atual da atividade, avaliar os valores trazidos pelos artefatos e trabalhar com suas potencializações e limitações em proveito da educação do aprendiz.

O Construcionismo se aproveita, por exemplo, deste caráter limitador dos artefatos para restringir as ações do aprendiz às propriedades educacionais desejadas dos objetos que ele manipular durante a atividade – este é o conceito fundamental por trás da idéia dos micromundos.

Conforme explicado anteriormente, os comandos do Logo Gráfico restringem as ações do aprendiz às propriedades geométricas da tartaruga gráfica. Se o objetivo da atividade fosse trabalhar com física dinâmica, provavelmente, a tartaruga teria comandos do tipo “aumente velocidade horizontal”, “aumente velocidade vertical”, etc.. Do mesmo modo, os itens a serem preenchidos no diário-de-bordo procuram levar o aprendiz a refletir sobre o andamento da atividade em si e sobre seus planos futuros.

Por outro lado, o Construcionismo utiliza o papel potencializador das ferramentas para construir situações de aprendizado que, de outra forma, seriam impossíveis ou inviáveis de se desenvolver. São ferramentas como o Logo Gráfico que trazem melhores condições para que os aprendizes possam refletir sobre as estratégias que utilizaram na resolução de seus problemas. São ferramentas como o LEGO-Logo que permitem às crianças terem acesso, de uma maneira prática, aos problemas da mecânica e da robótica. São ferramentas como o MUSIC que ajudam a comunidade a visualizar suas prioridades e discutir o que deve ser feito.

Poder versus facilidade de uso. O Construcionismo também atenta para o compromisso que, normalmente, existe entre o poder propiciado por uma ferramenta e a sua facilidade de uso. Idealmente, as ferramentas deveriam trazer um mínimo de dificuldades e um máximo de potencial de uso (um conceito que é conhecido na literatura construcionista como *low threshold* e o *high ceiling*). De nada adianta fazer com que uma ferramenta tenha mais recursos se ela se tornar mais difícil de ser utilizada (BRUCKMAN, 1997). Segundo Papert, esta foi uma das principais razões para que versões mais modernas da linguagem Logo, como o

ObjectLogo, que traz uma série de conceitos inovadores, não obtivesse o sucesso esperado.

Mediação social. Além de toda esta questão tecnológica, o Construcionismo também destaca a mediação social, principalmente no que se refere aos valores culturais impregnados no ambiente e no papel da comunidade no incentivo e facilitação do aprendizado.

Para Papert, boa parte dos problemas educacionais de hoje são resultantes da falta de uma cultura que valorize o aprendizado. As pessoas falam de suas conquistas e fracassos financeiros e amorosos, de suas férias, etc., mas estratégias e estilos pessoais de aprendizagem não costumam aparecer nas conversas – nem mesmo nos ambientes educacionais.

Nesse sentido, as atividades educacionais deveriam incentivar as discussões e a reflexão sobre aprendizado. As pessoas deveriam conversar mais em casa, na escola e no trabalho – em qualquer lugar – sobre situações que levaram a descobertas interessantes, sobre diferentes abordagens para problemas comuns, etc. (PAPERT, 1996).

A mediação social se torna evidente com a presença do facilitador nas atividades construcionistas. Os estudos têm mostrado que, sem uma pessoa que trabalhe ao lado dos aprendizes, acompanhando e incentivando o seu desenvolvimento e ajudando, principalmente, nas situações em que eles se sentem estagnados, é muito difícil obter sucesso no aprendizado.

Aprender significa lidar com o novo, algo que exige muitas tentativas e erros e que, se não for bem acompanhado e motivado, pode ser extremamente frustrante. Cabe ao facilitador fazer este acompanhamento e orientar o aprendiz no que for preciso para que ele supere os problemas enfrentados no cotidiano da atividade de aprendizado de uma forma positiva.

Mediação nas outras sub-atividades construcionistas. Por fim, embora este trabalho se concentre, principalmente, na atividade de aprendizado construcionista e, por isso, enfatize o desenvolvimento de artefatos dirigidos ao aprendiz, convém lembrar que a atividade educacional também envolve atividades de facilitação, administração, etc. que, idealmente, também deveriam ser consideradas no desenvolvimento dos artefatos.

Por exemplo, como será melhor discutido no capítulo 9, percebe-se hoje em dia uma falta de ferramentas para auxiliar o trabalho dos facilitadores construcionistas. A quantidade de material produzido pelos aprendizes ao longo dos projetos é muito grande e costuma envolver tanto materiais produzidos no computador (como os programas e os desenhos feitos na tela), como fora dele (anotações dos facilitadores, diários de bordo, construções em LEGO, gravações em vídeo, etc.). A análise da evolução dos trabalhos diários e comparação entre o que foi feito por diferentes aprendizes são coisas que poderiam interessar até aos próprios aprendizes e que poderiam exigir muito menos esforço caso pudessem ser efetuadas com o auxílio de ferramentas para este fim.

Só no trabalho de Harel (1991), foram 4 meses coletando informações das sessões diárias com o Logo e, simultaneamente, ajudando os aprendizes no que fosse preciso. Se esta coleta de informações fosse mais automatizada, o facilitador não teria que se esforçar tanto na análise dos trabalhos e poderia se concentrar mais no que os aprendizes estavam fazendo.



8 A análise do artefato computacional na atividade construcionista

No capítulo anterior, foram analisados, à luz da Teoria da Atividade, uma série de aspectos sobre a educação e, mais especificamente, sobre o aprendizado construcionista. No presente capítulo, seguindo a estrutura e as questões propostas na abordagem do capítulo 6, é analisado o papel do artefato computacional frente a cada um dos aspectos da atividade educacional construcionista descrita no capítulo 7.

O objetivo desta discussão é identificar o potencial específico do computador na atividade educacional construcionista, possibilitando condições de comparação com outros artefatos e indicando pontos a serem considerados no desenvolvimento novos artefatos computacionais para este tipo de atividade.

8.1 A estrutura hierárquica da atividade construcionista

Influência do computador no nível da atividade. Realmente, tal como descrito pela Teoria da Atividade, o computador, aqui visto principalmente como um artefato digital, pode influenciar a atividade construcionista em todos os seus níveis. No mais alto, que se refere à atividade em si, ele pode ajudar a viabilizar situações que, dificilmente, poderiam ocorrer nas salas de aula tradicionais, como desenvolver experimentos com materiais perigosos, simular fenômenos que não possam ser visualizados a olho nu, etc.

O computador também possibilita a criação de atividades educacionais que extrapolem as paredes da escola, incentivando um conceito de educação que englobe os lares, as empresas e as instituições formais de ensino, favorecendo a criação de comunidades que compartilhem um mesmo interesse e que não estejam tão limitadas pela distância ou pelas possíveis deficiências físicas ou mentais de seus participantes.

Além de atuar nos diferentes níveis da atividade, a flexibilidade do computador permite ainda que ele ofereça suporte a uma diversidade de papéis da educação construcionista. Dependendo do software, ele pode oferecer funcionalidades específicas para aprendizes, facilitadores, fontes de referência, administradores e outros. Pode ainda ajudar na interligação e na alternância entre os diferentes papéis de um mesmo sujeito.

O computador e as ações construcionistas. No nível das ações do aprendizado construcionista, há diversos pontos a serem considerados. Por exemplo, na **Idealização**, o computador pode ser usado para fazer anotações e representações do que se pretende construir. Estas anotações e representações podem incluir

desenhos, textos, fotos, filmes, gravações sonoras e outras mídias, permitindo uma boa descrição das idéias que forem surgindo, por mais reais ou irreais que pareçam.

Também pode ajudar o aprendiz a divulgar suas idéias e a visualizar o trabalho de outras pessoas. Dependendo do caso, pode inclusive ajudar a desenvolver idéias em conjunto, ou mesmo testar combinações inusitadas que possam ajudar o aprendiz a conceber novas idéias.

Na **Construção**, como será discutido mais adiante, o computador pode ser bastante flexível quanto ao tipo e quantidade do material trabalhado pelo aprendiz. Além disso, ele também pode facilitar o trabalho de descrição da construção realizada ajudando a manter o aprendiz mais focado e motivado.

Para a **Avaliação**, talvez uma das maiores contribuições do computador seja a possibilidade de o aprendiz visualizar e acompanhar o que está acontecendo na **Interpretação** da forma que lhe seja mais representativa e que facilite na **Comparação**.

Por exemplo, quando uma criança executa um programa Logo que desenha uma figura, ela pode visualizar tanto a figura, quanto os movimentos executados pela tartaruga à medida em que os comandos vão sendo interpretados pela máquina. Com isso, ela tem mais recursos para estabelecer relações entre o que desejava, o que fez e o que obteve. Nestes casos, o ideal é o aprendiz poder regular a velocidade da tartaruga e a quantidade de feedback a ser fornecida por cada parte do programa.

Vale notar que, na **Interpretação**, há uma série de aspectos em que as diferenças humano-máquina mereceriam um estudo mais aprofundado. Dentre eles, destacam-se a imparcialidade da máquina e sua incapacidade de analisar significados.

Imparcialidade. Se um programa escrito em Logo fosse executado repetidas vezes no mesmo computador, o resultado obtido na tela seria exatamente o mesmo. Se um aprendiz pedisse para que o seu colega imitasse a tartaruga se movimentando pela sala à medida em que fosse recebendo os comandos, é muito provável que, em cada nova tentativa, obtivesse resultados diferentes dos anteriores. No caso de conjuntos de movimentos simples, isso não seria muito relevante. Porém, se a figura almejada fosse muito grande ou detalhada, a diferença poderia levar a conclusões errôneas.

Enquanto a interpretação efetuada pela máquina é imparcial, a interpretação humana é influenciada pela experiência e estado de espírito do interpretador. No caso da máquina, o aprendiz pode ter certeza de que o produto obtido na tela é resultado único e exclusivo de seu próprio trabalho e de seus conhecimentos.

Análise de intenções e significados. Apesar de sua imparcialidade, em uma situação aberta, a máquina se mostra incapaz de inferir as intenções e os significados subjacentes à construção do aprendiz. Tudo o que ela processa são operações predefinidas. No programa citado, o computador seria incapaz de inferir, a não ser que alguém o tivesse programado para tal, que os comandos executados produziram uma flor ou qualquer outra coisa. Para ele, o desenho seria um mero conjunto de pontos acesos na tela.

Do mesmo modo, se fosse pedido ao computador para ler um texto em voz alta, provavelmente, a única interpretação retornada seria a entonação do texto. Ele não poderia dizer quais as idéias que estavam sendo apresentadas, ou se havia algo sem nexos ou difícil de entender. No máximo, o computador poderia indicar que algumas palavras estão repetindo-se mais do que outras e que algumas regras gramaticais não estão sendo aplicadas – algo que, sem dúvida, apesar da grande valia, não suprime a contribuição humana.

O ideal, portanto, seria que tanto o leitor quanto o computador fossem usados, de acordo com suas limitações, para fazer uma interpretação do construído.

Na **Comparação**, o computador pode ajudar a determinar e a visualizar, de forma precisa e rápida, as diferenças e as similaridades entre os objetos com que o aprendiz estiver lidando e, se necessário, guardar os resultados para futura referência.

A questão seria identificar que tipos de diferenças seriam relevantes para cada papel da atividade. Por exemplo, enquanto os facilitadores poderiam estar interessados nas comparações entre o que foi desenvolvido por um mesmo aprendiz entre uma seção e outra, os administradores poderiam estar interessados na diferença de uso da máquina entre turmas inteiras de aprendizes.

Na **Depuração**, a flexibilidade do computador pode contribuir muito para a implantação dos elementos facilitadores de depuração identificados no capítulo 7. Em especial, aplicativos de suporte ao design reflexivo como o Logo Gráfico trazem diversos atributos que ajudam o aprendiz a depurar o seu programa. Destacam-se:

- O formalismo dos seus comandos (cada um tem um resultado específico), a imparcialidade da máquina e a possibilidade de visualizar graficamente o que está ocorrendo na interpretação, que facilitam no estabelecimento das relações Construção-Interpretação.
- A organização dos comandos em grupos (os procedimentos) que podem ser testados isoladamente e depois recombinações, ajuda tanto na compreensão do que foi feito quanto na identificação de possíveis regiões problemáticas.
- A possibilidade de alterar o que foi feito sem ter que refazer tudo de novo.
- A disponibilidade de comandos e ferramentas específicos para depuração. Diversas versões do Logo trazem, além dos comandos básicos da tartaruga, comandos especiais que permitem visualizar o valor de variáveis internas (como as coordenadas da tartaruga na tela) e controlar o modo de execução do programa. O Slogo, do NIED (ROCHA, 1993), traz todo um conjunto de ferramentas especiais para depuração que permite observar diferentes detalhes da execução do programa (como a seqüência de chamada de procedimentos) sem ter que alterá-lo.

Ainda assim, o trabalho de gerenciamento das versões e tudo o que se refere à comunicação entre os participantes da atividade são exemplos de atributos que poderiam ser melhor desenvolvidos neste software.

Na **Descrição**, conforme discutido no capítulo 7, o computador pode complementar o diário-de-bordo feito em papel, permitindo a criação e o gerenciamento de anotações não-lineares, multimeios e diretamente integradas ao produto das demais ações feitas no computador. Ele também pode automatizar parte do processo descritivo, principalmente no que se refere à anotação dos comandos efetuados e à geração de estatísticas, ajudando a não desviar o aprendiz do objeto de seu interesse ao mesmo tempo em que fornece dados específicos para os demais papéis da atividade.

O programa como descrição. No entanto, além destas características levantadas acima que podem ser implementadas nos mais variados artefatos digitais, o Construcionismo dá uma atenção especial ao potencial de programação dos artefatos computacionais. Isso porque, para o Construcionismo, os programas desenvolvidos pelo aprendiz expressam os conceitos, estratégias e estilos usados por ele durante a atividade. Esta descrição simples (no caso do Logo), precisa e formal, serve como que um mapa que ajuda o aprendiz (e os demais participantes da atividade) a visualizar e refletir sobre o que foi feito.

Além disso, conforme descrito anteriormente, embora o formalismo e a precisão sejam atributos de qualquer linguagem de programação, o Logo Gráfico se destaca pela familiaridade do nome dos comandos (que se assemelham a termos usados no dia-a-dia do próprio usuário) e pelo tipo de feedback retornado aos aprendizes. O Logo Gráfico foi sendo aprimorado com os estudos construcionistas, ajudando a valorizar também o processo além do produto e é essa sua principal diferença com relação às demais ferramentas de programação encontradas no mercado (VALENTE, 1995).

Convém notar que, embora seja uma descrição completa e precisa da estrutura do produto construído pelo aprendiz, o programa por si só não permite identificar os diferentes caminhos percorridos na construção. Isso só seria possível através da comparação dos diversos programas intermediários desenvolvidos por ele ao longo da atividade inteira.

Neste caso, o software poderia salvar, periodicamente, o trabalho do aprendiz e oferecer ferramentas que mostrassem as diferenças entre as versões e também estatísticas que fossem de interesse do aprendiz e/ou do facilitador. Algumas versões do Logo salvam todos os comandos dados pelo aprendiz em um arquivo especial que pode ser analisado à posteriori. Recursos de análise mais avançados ainda estão por ser implementados.

Ainda assim, por mais que o software permitisse armazenar e rever os caminhos percorridos pelo aprendiz, a análise do programa e destes dados não seria suficiente para determinar os motivos que levaram-no a seguir determinadas estratégias, ou a desenvolver determinados produtos. Conforme discutido anteriormente, isso só seria possível através do desenvolvimento de uma cultura que incentivasse a meta-descrição e de ferramentas que reduzissem o esforço de descrição/meta-descrição ao mínimo.

Além da qualidade e precisão da descrição propiciados pelo Logo, o Construcionismo também enfatiza o fato de, na programação, a descrição ser algo praticamente indissociável da construção (PAPERT, 1993a; VALENTE, 1993). Ainda que incompleta, a descrição do programa faz parte do próprio programa e o aprendiz não tem que interromper nada (ou desviar sua atenção) para fazê-la.

Descrição para os outros papéis da atividade. Por fim, ainda com o intuito de evitar que o aprendiz distraia-se de sua atividade, na nossa opinião, o desenvolvimento do artefato computacional deveria considerar as necessidades de cada um dos papéis da atividade e oferecer meios para que elas pudessem ser atendidas sem interferir com o aprendiz. Neste caso, parte das informações específicas poderia ser gerada, automaticamente, sem que ele tomasse consciência, para que os interessados pudessem acessá-la via mecanismos especiais.

Por exemplo, embora os programas do Logo Gráfico sejam, relativamente, de fácil compreensão e representem uma boa descrição da Construção, é muito difícil para um facilitador acompanhar o trabalho de vários aprendizes ao longo de uma sessão. Possivelmente, uma extensão para o Logo Gráfico seria incorporar facilidades que reportassem ao facilitador (ou mesmo para pesquisadores) pontos específicos nos quais ele estaria interessado.

No **nível das operações**, além de, como mencionado acima, automatizar as operações que não interessarem pedagogicamente para a atividade, o computador pode ser usado para registrar o que foi feito pelo aprendiz (ou outro participante da atividade) e depois repassar o que foi armazenado para o facilitador, ou mesmo para o aprendiz, a fim de que ele reflita sobre o que foi feito.

No Logo, por exemplo, costuma haver diferentes maneiras de se enviar comandos para a tartaruga. A indireta (ou assíncrona), que é feita pela escrita de procedimentos que depois serão executados e a direta (ou síncrona), por meio da digitação de comandos que serão executados, imediatamente, pela tartaruga.

Algumas versões do Logo, por exemplo, possuem uma área onde são armazenados os últimos comandos. A qualquer momento o aprendiz pode entrar nessa área e rever (e reaproveitar) os últimos comandos dados em modo direto.

Há também versões que permitem movimentar a tartaruga na tela por meio do mouse. O problema levantado nestas versões é como registrar os movimentos efetuados para que eles possam ser analisados depois.

Convém notar que, mesmo com o auxílio de recursos do computador, é importante que se tenha um **observador externo**. As facilidades do computador normalmente só registram os comandos que o aprendiz efetuou no software que estiver sendo utilizado. Todas as operações que ele pratica fora do computador, ou mesmo fora de um software específico – como anotações em papel no diário-de-bordo, discussões com os amigos e outras coisas – embora também façam parte da atividade, não são registradas.

8.1.1 Conceitualização e operacionalização

No capítulo 7, foram levantados aspectos do computador que interferem no processo de conceitualização e operacionalização. Também foi discutido como estes aspectos podem ser utilizados para focar o aprendiz na parte pedagógica da atividade educacional, diminuindo os problemas de sintaxe, melhorando os mecanismos de suporte, explicitando os pontos desejados, etc.

De fato, estes aspectos podem ajudar muito no desenvolvimento de atividades educacionais como um todo, não apenas construcionistas. Também é interessante notar que os aspectos levantados valem, em sua maioria, para artefatos em geral, sejam computacionais ou não.

Além destes pontos gerais, no entanto, o Construcionismo chama a atenção para alguns atributos específicos do artefato computacional que também parecem estar diretamente relacionados ao mecanismo de operacionalização e conceitualização.

Conforme discutido anteriormente, os artefatos computacionais, pelo seu potencial de programação, podem trazer grande flexibilidade e alcance às atividades educacionais, chegando até a propiciar uma forma mais concreta de se visualizar os processos mentais do indivíduo.

Por exemplo, a programação com o Logo Gráfico parece ser um ótimo meio para representar as operacionalizações e conceitualizações do aprendiz. À medida em que vão sendo operacionalizados, os comandos básicos do Logo Gráfico vão sendo combinados em procedimentos que efetuam ações mais complexas. Depois, estes procedimentos podem ser divididos ou reagrupados para serem utilizados em situações que não tenham sido anteriormente previstas.

De fato, para o Construcionismo, a possibilidade de se combinar e descombinar operações – isto é, esta facilidade de poder lidar com operações como se fossem objetos – ajuda a tornar mais concretas a estrutura e a dinâmica dos processos mentais, podendo, com isso, ajudar na sua análise e possível aprimoramento. Segundo Ackermann (1998), os artefatos computacionais permitem com que os sujeitos possam, além de delegar suas ações mentais para a máquina, determinar o quanto de trabalho mental deverá ser automatizado a cada momento.

Esta possibilidade de projetar nossas mentes para a máquina – que Ackermann chamou de *mental teleportation* – é, na nossa opinião, conforme já discutido no capítulo 6, uma característica única do artefato computacional e é isso que torna este artefato **uma extensão “artificial” direta do mecanismo de conceitualização e operacionalização humano**. Outros artefatos automatizam operações, mas as operações automatizadas perdem sua capacidade de conceitualização. Elas ficam fechadas na estrutura do artefato. O artefato computacional permite combiná-las e descombiná-las como a própria mente, mas de uma forma externalizada e mais objetiva.

8.2 Orientação a objetos

Os artefatos digitais (incluindo os artefatos computacionais digitais) têm uma série de atributos a serem considerados na dimensão “Orientação a Objetos” da atividade construcionista.

Em primeiro lugar, para o Construcionismo, o computador pode até ser usado, em casos muito especiais, como um elemento motivador da atividade educacional. Do mesmo modo que ele já vem sendo utilizado, devido ao seu status cultural, como um chamariz mercadológico para a atração de novos alunos para as escolas e como um elemento motivador para práticas educacionais ultrapassadas, ele também pode ser utilizado pelos educadores construcionistas como um pretexto para a mudança educacional. Neste sentido, os educadores poderiam usá-lo como uma forma de atrair a atenção dos alunos e, com isto, abrir espaço para uma exploração construcionista de temas que estes alunos, por uma razão ou outra, já tenham desistido de aprender.

Os educadores devem aproveitar a chance e, junto com a entrada dos computadores, divulgar as idéias construcionistas e propor atividades em que o computador fosse utilizado de modo mais adequado (PAPERT, 1993a, p.222).

Idealmente, no entanto, espera-se que o tema abordado e a dinâmica criada em torno dos objetos transicionais disponíveis na atividade sejam interessantes por si sós. O computador deve ser visto como um elemento motivador, no sentido em que ajuda a viabilizar uma gama de situações educacionais mais atraentes.

Em segundo lugar, entrando mais a fundo na questão dos materiais disponibilizados, a possibilidade de se trabalhar com objetos digitais torna o computador (o artefato digital) um elemento extremamente poderoso para a criação de micromundos construcionistas. No Logo Gráfico, como já discutido, a tartaruga digital ajuda o aprendiz a perceber relações espaciais.

Em outra ferramenta, o LEGO-Logo (RESNICK, 1993), os aprendizes exploram conceitos de robótica manipulando, por meio de uma versão especial do Logo, objetos digitais diretamente associados a motores e sensores acoplados à suas construções feitas em LEGO. Construindo casinhas, carrinhos e guindastes que funcionam, na atividade com o LEGO/Logo o aprendiz integra o caráter essencialmente simbólico do computador com o físico das pecinhas do LEGO e explora diversas propriedades mecânicas e sensoriais que, dificilmente, poderiam ser experimentadas nas atividades em que só o meio digital fosse utilizado.

Já no software StarLogo, (RESNICK, 1991; RESNICK, 1994), que é uma variação da linguagem Logo para o estudo de sistemas descentralizados, aproveita-se o fato de a quantidade de objetos digitais manipulados na atividade poder ser virtualmente infinita – estando apenas limitada à quantidade de memória e à capacidade de processamento do computador – para abrir novos horizontes.

Com o StarLogo, o aprendiz pode criar milhares de tartarugas gráficas que funcionam em paralelo e interagem entre si. Estes recursos tornam viáveis o estudo e o teste de hipóteses em cima de formações como engarrafamentos de trânsito,

colônias de formigas, bandos de pássaros e outros que, com muita dificuldade, poderiam ser abordados de forma prática e pessoal pelos alunos de uma sala de aula.

De qualquer modo, apesar do enorme potencial dos objetos digitais, conforme discutido no capítulo 7, o Construcionismo enfatiza que o tema da atividade também tem um papel muito importante no incentivo dado aos aprendizes para explorarem os atributos desejados dos objetos disponibilizados.

Faltam pesquisas construcionistas comparando objetos digitais e não digitais. E no que se refere a uma comparação entre objetos materiais e digitais, falta, em termos de pesquisas construcionistas, subsídios que ajudem a orientar, por exemplo, em que situações seria suficiente usar as representações digitais, em que situações elas não seriam recomendadas e em que situações o melhor seria uma combinação de representações e realidade. Mesmo questões relacionadas com ambientes distribuídos de aprendizagem também mereceriam um maior aprofundamento.

Em terceiro lugar, segundo Turkle, em se falando de estilos de aprendizagem (ver capítulo 5), os objetos digitais têm atrativos que os tornam interessantes tanto para escultores como para planejadores. Em especial, objetos como a tartaruga gráfica combinam uma representação familiar mais concreta com uma definição lógico-matemática mais abstrata e formal que faz com que eles se situem entre os dois estilos. Esta ambivalência, além de fazer com que sujeitos de ambos os estilos se sintam à vontade, também serve de ponte, facilitando a circulação de um estilo para outro (TURKLE e PAPERT, 1991).

Por fim, conforme discutido na seção 8.1.1, o Construcionismo enfatiza muito esta capacidade do artefato computacional “tornar mais concretos” os conceitos que antes dele eram muito abstratos e difíceis de serem aprendidos. Em especial, a possibilidade de visualizar as estratégias adotadas na solução de problemas e, como ocorre na programação, lidar com as operações usadas como se fossem objetos, é vista como algo que pode contribuir bastante para o desenvolvimento cognitivo.

8.3 Internalização e externalização

Internalização. Como foi discutido na Análise da Atividade Construcionista, para o Construcionismo, o desenvolvimento das atividades mentais do sujeito é resultado, principalmente, de suas interações com o mundo e de seu estágio cognitivo. Os micromundos são criados em torno desta filosofia, representando situações socioculturais propícias para que o aprendiz se desenvolva em determinadas direções.

O computador, neste contexto, é visto como um material extremamente interessante para a construção destes micromundos. No que se refere aos aspectos culturais, tal

como discutido na seção acima, o computador possibilita a criação e o acesso a diferentes objetos e representações que, dificilmente, poderiam ser conseguidos com outras ferramentas.

No que se refere ao aspecto social, que é o ponto mais enfatizado pelo conceito de internalização da Teoria da Atividade, as facilidades de comunicação do computador ajudam a ampliar os limites alcançados pelos micromundos, colocando os participantes da atividade em contato com outras práticas sociais e viabilizando a formação de novas comunidades.

Com a Internet, por exemplo, começou a tornar-se mais popular o conceito de comunidades unidas não pela sua localização geográfica, mas por temas de interesse comum. Existem milhares de grupos de discussão voltados para temas que incluem desde religião e desenvolvimento de software, até formação de empresas e criação de animais.

Dependendo da forma com que são organizadas e do tipo de atividade sendo desenvolvida, estas comunidades podem contribuir bastante para que os aprendizes possam investigar suas áreas de interesse e também já possam ir se integrando às rodas sociais onde estas áreas se desenvolvem.

E o que serve para os aprendizes também é válido para os facilitadores, administradores e demais papéis encontrados na educação construcionista. As facilidades de correio eletrônico (e-mail), discussões online (chats) e videoconferência são exemplos de formas de comunicação via computador que podem ser usadas para a troca de experiências, resolução de questões e até mesmo para conversas informais entre todos – algo que pode ajudar a manter a motivação e dar subsídios para o desenvolvimento das práticas construcionistas nas instituições educacionais.

Em especial, o Construcionismo Social de Alan Shaw (ver capítulo 7) usa o computador tanto como um pretexto, quanto como um catalisador para o estreitamento das relações sociais de comunidades locais urbanas. Sob o ponto de vista da Teoria da Atividade, o software MUSIC criado por ele evidencia algumas práticas sociais específicas, como a troca de idéias para a resolução de problemas comuns e dá condições para que os membros destas comunidades participem ativamente destas práticas (via troca de mensagens, publicação de observações em painéis de avisos, etc.).

Aos poucos, segundo a Teoria da Atividade, estas interações sociais vão sendo internalizadas e os membros da comunidade começam a extrapolar o uso inicial do computador, identificando novos pontos de interesse comunitário e criando reuniões e eventos presenciais como gincanas e debates visando o desenvolvimento da comunidade.

Externalização. Conforme discutido anteriormente, a externalização das atividades mentais (dos conhecimentos e das estratégias de resolução de problemas) é um dos elementos chave para o bom desenvolvimento da educação construcionista. Dentre outras coisas, a externalização do pensamento ajuda o aprendiz a ordenar suas idéias e a refletir sobre as estratégias adotadas. Ela também serve de base

para que o facilitador acompanhe o desenvolvimento do aprendiz e possa intervir quando necessário.

Também foi visto que, dependendo de como a atividade for planejada, o esforço efetuado pelo aprendiz para descrever tudo o que foi feito pode afetar sua motivação. Segundo a Teoria da Atividade, a externalização só se mostra intrinsecamente motivante em determinadas situações que, na nossa opinião, deveriam ser consideradas no design das atividades construcionistas:

- Quando o aprendiz tem necessidade de visualizar passo-a-passo o que foi feito, como ocorre quando ele detecta um problema e tem que depurar o que foi feito.
- Quando ele tem que efetuar cálculos grandes ou complexos demais, como no caso de projetos longos.
- Quando ele tem que necessidade de comunicar as idéias, como se dá em um projeto em equipe ou quando ele tem que apresentar o que fez para outras pessoas.

Na nossa opinião, o computador é, provavelmente, um dos poucos artefatos que consegue, de forma integrada, dar suporte a estas 3 necessidades de externalização e, ao mesmo tempo, ainda oferecer espaço para que o sujeito construa seus projetos e teste suas idéias. Software como o Logo Gráfico teoricamente poderiam oferecer, por exemplo, facilidades de programação, depuração, anotações e de intercâmbio de mensagens e pedaços de código totalmente integradas que, dificilmente, poderiam ser obtidas com outras ferramentas.

Ainda que o computador não seja o artefato mais indicado para todas as situações de externalização, essa sua característica polivalente o torna uma ferramenta bem atraente para o desenvolvimento de atividades construcionistas.

No que se refere ao aspecto político da externalização, o artefato computacional digital parece ser flexível o bastante para permitir a expressão de uma infinidade de conceitos e estilos mas, dependendo de como for implementado, também pode ser bem restritivo. Por exemplo, aplicativos que ofereçam poucos recursos para guardar versões intermediárias de projetos, ou que ofereçam poucas facilidades para que o sujeito possa alterar o que fez, restringem bastante o trabalho de sujeitos que tenham estilo de aprendizado escultor.

Esta é uma questão que depende, principalmente, das idéias consideradas pelos designers durante a concepção do software – algo que acontece principalmente na fase de Análise – e é, segundo acreditamos, uma das maiores contribuições que o presente trabalho tem para oferecer quando propõe uma metodologia que abranja estes pontos.

8.4 História e desenvolvimento

História. A questão central por trás da análise histórica de uma atividade é poder identificar as características essenciais dessa atividade e determinar regras gerais que auxiliem na incorporação de novos elementos a essa atividade.

No caso, por exemplo, deve-se compreender que o Construcionismo não se limita nem ao computador e nem à linguagem Logo. O objetivo do Construcionismo gira em torno da criação de ambientes educacionais mais efetivos e adequados para as necessidades de hoje e, para isso, estuda o papel da tecnologia na criação destes ambientes.

O computador, assim como o vídeo e o LEGO, é tido como um ótimo material para a criação de micromundos construcionistas e o Logo representa uma de suas aplicações mais famosas neste sentido.

Prosseguindo nesta visão histórica, mesmo o Logo não precisa se limitar aos recursos da época em que foi criado. A questão – e este é um dos pilares deste trabalho – determinar quais de suas características são as mais relevantes e trabalhar as novidades tecnológicas de acordo com os critérios norteadores da filosofia construcionista, tendo sempre como perspectiva a atividade educacional como um todo.

Desenvolvimento. A análise do artefato computacional frente o desenrolar de uma atividade exige compreender como o artefato pode suportar o aprendizado do sujeito e suas passagens de um nível de competência para outro no correr desta atividade. No caso da atividade educacional construcionista isto implica, por exemplo, em considerações relacionadas à duração da atividade e às variações do nível de competência e das necessidades do sujeito ao longo dela.

Suporte para atividades longas. Como o aprendizado, segundo o Construcionismo, é algo que leva tempo (ver capítulo 7), o artefato computacional deve oferecer suporte para que a atividade possa ser efetuada em sessões múltiplas, ao longo de vários dias. Exemplos de suporte seriam: facilidades para armazenamento e cópias de segurança dos trabalhos de cada sessão; facilidades de gerenciamento e comparação de diferentes versões dos trabalhos, etc.

É interessante notar que, por exemplo, embora os projetos desenvolvidos com o Logo Gráfico costumem durar vários dias, as versões deste software oferecem pouca ou nenhuma facilidade para que o aprendiz, ou outro interessado, possa fazer um acompanhamento do que foi feito a cada dia. Teoricamente, o software poderia salvar, automaticamente, os trabalhos e organizá-lo de modo a facilitar futuras análises sem que o aprendiz precisasse se preocupar com isso.

Suporte para manutenção e desenvolvimento do nível de competência. Tal como discutido no capítulo 4, a Teoria da Atividade apresenta uma análise detalhada das mudanças qualitativas que ocorrem na interação do sujeito com o artefato à medida em a atividade vai se desenrolando. Segundo ela, a transparência do artefato é dinâmica (mesmo um especialista pode voltar a tomar consciência do artefato) e depende, principalmente, de seu uso e de um bom treinamento.

O dinamismo de consciência implica que o artefato computacional deve prover (ou vir acompanhado de) material de suporte que indique ao sujeito como proceder, no caso de um desequilíbrio cognitivo, para retornar ao seu nível original de competência. Este tipo de suporte, assim como o planejamento de uma fase inicial da atividade dedicada ao aprendizado do artefato são pontos que, apesar de serem extremamente importantes para o bom andamento de atividades educacionais que envolvam o computador, raramente são abordados na literatura construcionista;

Flexibilidade do artefato. À medida em que o sujeito se desenvolve, suas ações vão se transformando e, com elas, as formas de uso e a necessidade de novas ferramentas. Para que possam se manter úteis o máximo possível ao longo da atividade, Harel sugere que os artefatos construcionistas deveriam permitir com que o próprio aprendiz pudesse ir adicionando funcionalidade neles à medida em que fosse sentindo necessidade.

Segundo ela, os artefatos computacionais deveriam oferecer uma linguagem de programação com características semelhantes ao Logo (em termos de facilidade de uso) que permitissem ao aprendiz criar novos comandos e incorporá-los aos já existentes no artefato (HAREL, 1991, p.384). É interessante notar como são raros os aplicativos que oferecem esta funcionalidade.

Suporte aos diferentes estilos de aprendizagem. Para concluir, dado a percepção de que as pessoas possuem estilos diversos de aprendizado, tais como o planejador e o escultor (ver capítulo 5), o desenvolvimento de artefatos computacionais construcionistas deveria levar ambos os estilos em consideração. Isso implicaria, por exemplo, na inclusão de facilidades para planejamento e abstração, possibilidade de expressão via narrativa (textual, gráfica, etc.) e programação, possibilidade de desfazer e refazer o que foi feito, etc.

8.5 Mediação

Conforme discutido anteriormente, o computador tem, de fato, características limitadoras e potencializadoras que podem ser bem utilizadas na atividade educacional construcionista. Como limitador, o artefato computacional permite restringir o que pode ser acessado e manipulado pelo aprendiz às propriedades e operações dos objetos que sejam de interesse da atividade educacional, ajudando-o a perceber o ponto em questão e evitando que ele se disperse. No Logo Gráfico, por exemplo, foram preservados somente os atributos da tartaruga que eram relevantes para a exploração dos conceitos matemáticos.

Como potencializador, o computador pode ajudar a criar condições para que o aprendiz construa produtos que, dificilmente, conseguiria com outros materiais, pode facilitar o intercâmbio e exposição de idéias (inclusive a nível mundial), etc.

Neste sentido, é interessante notar que, dado o encapsulamento natural dos objetos digitais por trás da interface, basta variar os aspectos físico e de manuseio dos

artefatos, para tornar possível que diferentes portadores de deficiências físicas e mentais consigam interagir com os mesmos objetos que as pessoas não portadoras destas deficiências. Esta característica dos artefatos digitais diminui as restrições impostas para participação das atividades educacionais e incentiva as políticas integradoras na educação.

Em especial, pela sua capacidade de ajudar na visualização das estratégias adotadas na solução de problemas, o artefato computacional em si é visto como um grande catalisador do raciocínio.

Por isso e pelas características acima descritas, o aprendizado da computação e das ciências da informação é visto pelo Construcionismo como “uma atividade sinergeticamente reflexiva do aprendizado”, ou seja, é um tipo de aprendizado que facilita outros tipos de aprendizado. Quando feito de modo integrado, aprendizado da informática não compete em tempo com as outras matérias do currículo, representando assim uma possível solução para o aparentemente lotado dia escolar. *“Se algum conhecimento facilita outro conhecimento, então, de uma forma maravilhosamente paradoxal, mais pode significar menos!”* (HAREL e PAPERT, 1991, p.75)

Mesmo assim, tal como discutido no capítulo 7, o computador sozinho não é um elemento mediador suficiente que ofereça todas as condições para que o aprendiz consiga desenvolver seu projeto. É preciso um mediador humano – o facilitador – que ajude o aprendiz a resolver problemas específicos, principalmente no que se refere aos aspectos psicológicos, à integração com o resto da turma, ao estabelecimento de relações com o domínio e, também, ao próprio uso do artefato computacional.

Ainda assim, o computador também pode servir como mediador da atividade do facilitador, ajudando-o a acompanhar a atividade do aprendiz, acessar referências, produzir relatórios e até se comunicar.

Além disso, conforme discutido no capítulo 6, vale notar que todo o potencial e as limitações geradas pelo artefato computacionais são resultado direto do tipo de operações que o artefato automatizar e das facilidades que disponibilizar para o usuário acesse e recombinar estas operações. A seleção de um conjunto adequado de operações no desenvolvimento do artefato é fundamental para a sua boa utilização na atividade educacional. Seria muito interessante levantar que operações seriam as mais indicadas para atividades construcionistas em geral (como o levantamento de estatísticas para os facilitadores, suporte às anotações dos aprendizes, operações que salvem periodicamente o material desenvolvido, etc.) e para o domínio específico a ser trabalhado.



9 Um caso prático: a Oficina de Jogos

Este capítulo retrata a experiência prática da Oficina de Jogos, uma atividade educacional construcionista centrada no Cocoa, um software feito para a criação de jogos e simulações que é tido por muitos como um sucessor moderno do Logo Gráfico.

Inicialmente, a dinâmica e os elementos da Oficina são discutidos de uma forma geral, segundo os critérios levantados no capítulo 7. Depois, é feita uma análise detalhada do papel do artefato computacional e, mais especificamente, do software Cocoa nessa atividade.

No final, são reunidas as principais conclusões levantadas ao longo da Oficina.

9.1 Introdução

A Oficina de Jogos foi uma atividade educacional efetuada entre fevereiro e março de 1997, no Parque do Ibirapuera em São Paulo. A idéia era criar um ambiente construcionista onde os participantes pudessem, utilizando o software Cocoa, planejar e desenvolver seus próprios projetos, no caso, um jogo de computador sobre o parque.

Além disso, a Oficina de Jogos também buscou servir, seguindo na linha construcionista, como um objeto que ajudasse a pensar os estudos do presente trabalho. Na época de sua realização, ainda não tínhamos uma visão mais estruturada nem do Construcionismo e nem da Teoria da Atividade. Nossa abordagem era, essencialmente, técnica e teórica. Faltava uma sensibilização prática para se perceber quais aspectos realmente interessavam e até que ponto a Teoria da Atividade poderia ser utilizada para descrevê-los.

Mais especificamente, a Oficina de Jogos foi um experimento aberto que teve como objetivos:

- Compreender as dificuldades inerentes à criação e ao desenvolvimento de uma atividade educacional construcionista.
- Entender quais são e como se relacionam os principais elementos-chave desta atividade.
- Perceber o papel da tecnologia.
- Levantar pontos para a avaliação e desenvolvimento de software para a educação.
- Analisar a versão corrente do software *Cocoa*, um moderno ambiente de programação dirigido às crianças, frente a uma atividade educacional real.

A organização e as idéias por trás da Oficina de Jogos foram baseadas, principalmente, no trabalho de Idit Harel e Yasmin Kafai, do *MIT Media Lab* (HAREL, 1991; HAREL e PAPERT, 1991; KAFAI, 1995). Estas pesquisadoras criaram atividades construcionistas onde crianças de uma escola americana foram colocadas no papel de designers de software para a educação que ensinassem frações para crianças menores.

O trabalho de Kafai seguiu, basicamente, o mesmo esquema de Harel, inclusive no que se refere à duração, escola e faixa etária. Ambas trabalharam durante 4 meses em sessões diárias de 1 hora com alunos do *4th grade* (faixa etária dos 10 anos) do *Project Headlight*, uma espécie de escola modelo, onde os alunos tinham uma grande quantidade de computadores à sua disposição.

A principal diferença é que, na pesquisa de Kafai, as crianças, ao invés de desenvolverem um software qualquer para a educação, deveriam criar um que fosse um jogo. Ambas as pesquisas obtiveram resultados interessantes, principalmente no que se refere à motivação, dinâmica social e importância da tecnologia nessas atividades.

Ainda que a Oficina de Jogos tivesse uma curta duração (apenas 10 sessões) e não se preocupasse com o aprendizado de algum tema curricular tão específico quanto frações, procurou-se implementar uma estrutura de trabalho semelhante à americana, propiciando a disponibilidade de um computador para cada aluno e incentivando a iniciativa, a criatividade, a reflexão e a troca de idéias entre todos.

Para que isto fosse possível, a Oficina de Jogos se estendeu ao longo de 3 semanas, entre os meses de fevereiro e março de 1997, no final das férias escolares de verão – uma época que não entraria em conflito com as demais atividades escolares dos participantes.

Foi realizada nas dependências do Instituto do III Millennium, na Marquise do Parque do Ibirapuera, um dos centros de lazer mais importantes da cidade de São Paulo. O Instituto do III Millennium é uma entidade não governamental e sem fins lucrativos que atende a milhares de crianças e adolescentes, em sua maioria carentes, com cursos de formação educacional e profissional usando software como a Linguagem Logo e aplicativos abertos (processador de texto e planilha). Seu trabalho é tido como referência na área de tecnologia para a educação fora do contexto da escola, servindo de base para uma grande quantidade de pesquisas nacionais e internacionais.

De acordo com a metáfora de Papert (1977, 1993a), o Instituto do III Millennium pode ser considerado como uma "escola de samba de informática". O objetivo comum de todos é desvendar os mistérios da informática e isto é conseguido através do desenvolvimento de projetos pessoais que têm o acompanhamento de profissionais qualificados que se colocam no papel de facilitadores da aprendizagem.

Os participantes vão ao III Millennium quando podem e por livre iniciativa. O custo das atividades é nulo ou simbólico. Os requisitos mais fortes para admissão são a faixa etária e a disponibilidade de computadores. No caso do trabalho com o Logo

Gráfico, a participação é gratuita, o indivíduo tem que ter entre 10 e 18 anos e só pode usar o computador durante 1 hora por dia.

Devido às suas características únicas e à abertura demonstrada com relação a novas idéias, o Instituto do III Millennium se mostrou um local bastante adequado para uma iniciativa como a Oficina de Jogos.

9.2 A atividade educacional da Oficina de Jogos

9.2.1 A estrutura hierárquica da Oficina de Jogos

9.2.1.1 A atividade

Sob o ponto de vista pedagógico, a atividade educacional da Oficina de Jogos tinha como principais motivos incentivar os aprendizes a refletirem sobre o Parque do Ibirapuera, conhecerem uma tecnologia diferente da que estavam habituados a mexer e se tornarem ainda mais motivados para continuar freqüentando o Instituto do III Millennium. Sob o ponto de vista técnico, conforme descrito acima, a idéia era que a Oficina de Jogos trouxesse subsídios para o desenvolvimento do presente trabalho.

De fato, a própria percepção da necessidade de um motivo educacional mais explícito e a diferenciação entre os motivos pedagógicos da atividade (conhecimentos, procedimentos e atitudes desenvolvidos) e seus produtos (elementos concretos resultantes da atividade) só vieram à tona a partir desta experiência.

De uma forma geral, na concepção da Oficina, houve uma preocupação muito grande em torná-la uma verdadeira atividade construcionista, isto é, uma atividade de design reflexivo, onde os participantes não só construíssem projetos, como também refletissem sobre o que estavam fazendo e como.

Com isto em mente, no desenvolvimento da atividade educacional da Oficina de Jogos, foram enfatizados os seguintes atributos construcionistas:

- A preocupação em se ter os aprendizes na posição de construtores de seus próprios projetos, sem que lhes fosse impostas muitas regras de conduta e procedimentos predeterminados.
- Um tema interessante e aberto – desenvolvimento de jogos e o Parque do Ibirapuera – que incentivasse o trabalho e a criatividade dos aprendizes dentro da área que se buscava aprender.
- A utilização de materiais atraentes e ao alcance físico e cognitivo de todos os participantes (facilitadores ou aprendizes).
- O incentivo à exploração, criatividade, troca de idéias, ao repensar e ao desenvolvimento de estilos pessoais.

- A não preocupação moral com as soluções "certas" ou "erradas". Cada aprendiz tinha seu estilo e ritmo de aprendizado respeitado e apenas as críticas construtivas eram incentivadas.
- A aproximação entre facilitadores e aprendizes.

No planejamento da Oficina, a maior parte do esforço foi dirigida à criação de uma estrutura de suporte ao aprendizado, mais do que à facilitação ou outras sub-atividades da educação construcionista. Também houve uma preocupação em se documentar o que foi feito e, com isso, esperava-se dar condições para que o facilitador exercesse sua prática e o designer da atividade conseguisse analisar o trabalho como um todo.

Como será percebido ao longo das próximas seções, a atividade educacional da Oficina de Jogos se desenrolou quase que totalmente ao redor do Cocoa. De uma forma simplificada, o Cocoa pode ser entendido como um conjunto de ferramentas utilizado inclusive por crianças para a construção de jogos, animações e simulações envolvendo figuras, sons e personagens que funcionam em paralelo. Tanto o Cocoa, quanto o hardware e demais aplicativos utilizados na Oficina serão discutidos em maior profundidade na seção sobre a análise do artefato computacional.

9.2.1.2 Os componentes da Oficina de Jogos

Seguindo a linha proposta no capítulo 7, a atividade educacional da Oficina de Jogos pode ser representada através de um diagrama de Engeström como o da Figura 20.

Nela, o aprendiz, no caso, um freqüentador do III Millennium, utiliza uma série de artefatos, dentre eles o software Cocoa e um diário-de-bordo, para desenvolver um jogo sobre o Parque do Ibirapuera. Os objetos manipulados pelos aprendizes consistiam, basicamente, nos disponíveis no Cocoa, que eram suficientes para a construção de um jogo de computador e nas folhas de planejamento.

As sessões tinham uma organização predefinida que procurava incentivar a criatividade e a reflexão. A divisão do trabalho refletia diretamente a discutida na análise da atividade construcionista (ver capítulo 7), com as figuras de aprendizes, facilitadores, designers da atividade e fontes de referência. A comunidade consistia essencialmente dos participantes de cada sessão, isto é, cerca de 4 freqüentadores do III Millennium, que eram os aprendizes e mais 1 "professor", que assumia os papéis de facilitador e designer da atividade. Todos também assumiam o papel de fontes de referência em maior ou menor escala. Por vezes, surgia um pesquisador visitante que, sem compromisso, atuava como ator e designer da atividade ajudando na análise das sessões.

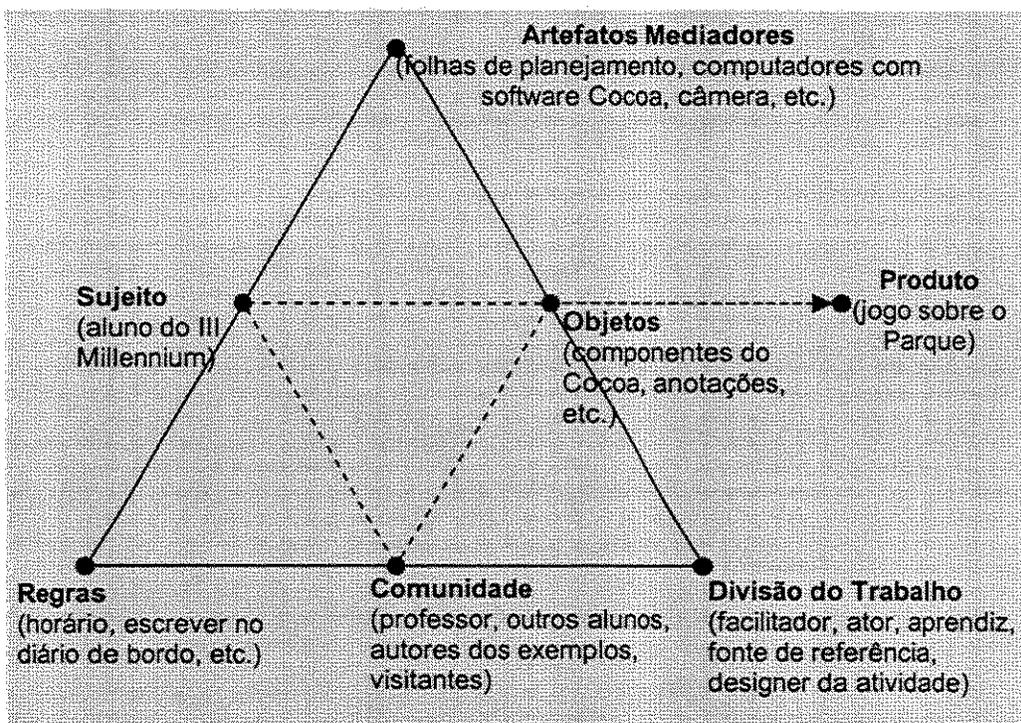


Figura 20 - O diagrama da atividade educacional construcionista da Oficina de Jogos

Cada um destes componentes será melhor detalhado a seguir.

Os sujeitos. A Oficina de Jogos trabalhou com 2 turmas de 4 aprendizes, cada uma sob a orientação de um professor. As duas turmas trabalhavam da mesma forma, porém em horários diferentes.

A Turma 1, que trabalhava das 14:00 às 15:20, contou com os seguintes participantes:

- J.S.S., de 12 anos, estudante da 6ª série. Ele já tinha 51 sessões de Logo e o III Millennium era o único lugar em que tinha acesso ao computador;
- M.Y.M.O., de 19 anos, aluno da 8ª série. Ele já tinha feito 74 lições de Logo e também participado de um curso de MS-Works dado no III Millennium. O Instituto era o único lugar onde acessava o computador;
- C.M.V., de 20 anos, aluna do 1º colegial. Ela já havia feito 60 sessões com o Logo e não tinha acesso ao computador em nenhum outro lugar. C.M.V. assistiu as 5 primeiras sessões da Oficina e depois faltou as seguintes sem deixar nenhum aviso;
- R.F., de 18 anos, aluno da 8ª série. Ele já tinha feito 42 sessões com o Logo e também os cursos de MS-Windows, MS-Works, MS-Word e robótica no III Millennium. O R.F. participou de 2 sessões, faltou 1, voltou a comparecer na Oficina e, depois, faltou nas 6 sessões seguintes sem dar notícias.

A Turma 2 trabalhava das 15:40 às 17:00. Ela foi composta por:

- R.N.M., 13 anos, aluno da 8ª série. Ele tinha 120 sessões de Logo e já havia feito os cursos de Introdução ao Windows e Robótica (LEGO/Logo). Tinha oportunidade de trabalhar com o computador em sua escola toda semana e freqüentava o III Millennium, principalmente, nas férias;
- R.H.M., 11 anos, cursando a 6ª série. Irmão de R.N.M.. Também já havia feito o curso de Robótica e tinha 115 sessões de Logo. Na escola tinha contato, principalmente, com o Windows e com jogos de computador;
- M.S.S.P., 16 anos, aluna da 8ª série. Foi a única participante mulher a completar a Oficina. Embora houvesse começado a freqüentar o III Millennium há pouco mais de 2 meses, já havia completado 75 sessões de Logo e também o curso de Robótica. Só acessava o computador no III Millennium;
- Z.R.M., 18 anos, 2º grau completo. Já havia feito 60 sessões de Logo, cursos profissionalizantes de Windows e Word e também o curso de Robótica. Tinha acesso a computadores no III Millennium, onde ia sempre que possível e na casa de uma amiga, freqüentemente.

Com exceção de já possuírem alguma familiaridade com o computador – especialmente com o Windows e o Logo – e serem freqüentadores assíduos do Instituto do III Millennium, os aprendizes que participaram da Oficina tinham uma grande diferença de faixa etária (dos 12 aos 20 anos), de defasagem escolar (vários estavam atrasados de 4 a 5 anos com relação à sua série, outros estavam de acordo e um estava adiantado uma série) e de possibilidade de uso do computador fora do Instituto.

Embora estas características estejam aqui descritas para indicar um perfil dos estudantes, uma análise mais detalhada das relações entre elas e o desempenho que cada um obteve ao longo da Oficina foge do escopo deste trabalho.

O professor da Oficina de Jogos foi o próprio autor do presente trabalho. Com experiência maior em desenvolvimento de software, era a primeira vez que criava uma atividade educacional construcionista. Na Oficina, atuou como facilitador, designer da atividade, fonte de referência e, em muitas ocasiões, como aprendiz.

O produto. Como já foi dito, o objetivo da atividade dos aprendizes era desenvolver um jogo de computador sobre o Parque do Ibirapuera. Como será aprofundado mais adiante, esperava-se que a criação do jogo propiciasse condições para que os motivos pedagógicos e técnicos por trás da criação da Oficina fossem atingidos, isto é, o incentivo à reflexão sobre o Parque e o subsídio a uma análise do Cocoa frente a uma atividade construcionista.

Os artefatos. Os principais artefatos usados pelo aprendiz na atividade eram, do lado tecnológico, um microcomputador Macintosh munido do software Cocoa, exemplos de uso do Cocoa, um software que permitia compartilhar dados com outros participantes e um software que possibilitava a visualização online de um guia de referência sobre o Cocoa. Do lado não-digital, o aprendiz contava, principalmente, com o diário-de-bordo para fazer suas anotações e uma versão impressa do guia de referência.

Já o facilitador pôde contar, no lado digital, além do computador do aprendiz, com um software gerenciador de redes que facilitava as cópias do trabalho diário. No nível não-digital, utilizava do material escrito pelos aprendizes no diário-de-bordo. Diariamente, o facilitador também escrevia seu próprio diário-de-bordo com reflexões sobre cada aprendiz e sobre o andamento da Oficina. No último dia, pôde contar com uma câmara de vídeo para registrar o depoimento de cada um dos aprendizes.

Os objetos. Basicamente, os aprendizes só manipulavam os objetos do Cocoa, isto é, os tabuleiros e os personagens, com suas regras, aparências, sons e propriedades. De vez em quando, utilizavam as folhas de planejamento do diário-de-bordo ou consultavam o guia de utilização do Cocoa. Fora isso, nenhum outro objeto material específico foi utilizado durante a Oficina de Jogos.

O professor, atuando como facilitador, fonte de referência e designer da atividade, usou as próprias observações anotadas antes, durante e após as sessões. Também contou com os programas desenvolvidos, as folhas de planejamento e as explicações dadas por cada aluno.

As regras. A Oficina de Jogos procurou, dentro do possível, trabalhar de forma bem aberta e informal. Dentre as regras que foram implantadas, destacaram-se:

a) A quantidade, o horário e a duração das sessões. A Oficina de Jogos foi distribuída em 10 sessões ao longo de 10 dias. Cada sessão durava 80 minutos, tempo estimado para não ser nem muito curto para os objetivos do dia e nem cansativo demais. Os alunos foram divididos em duas turmas que, apesar de trabalharem separadas, tinham os mesmos objetivos e recebiam o mesmo tratamento;

b) A organização das sessões. Como será melhor discutido na seção sobre o desenvolvimento da atividade, a Oficina de Jogos foi planejada para ter 2 momentos: um de introdução ao computador e ao Cocoa e outro para o desenvolvimento dos projetos. As sessões de cada um destes momentos tinham sua estrutura própria, definida de antemão pelo designer da atividade;

Apesar desta aparente rigidez, no entanto, havia uma certa flexibilidade para acomodar o que deveria ser feito conforme as necessidades e interesses da turma no momento.

c) O incentivo à discussão e à troca de idéias. Na Oficina de Jogos, também diferente do que costuma ocorrer nas escolas tradicionais, os alunos podiam e eram incentivados a trocar idéias com os colegas sobre o que estavam fazendo. A única restrição é que cada um deveria desenvolver seu próprio projeto.

A divisão do trabalho. Conforme mencionado anteriormente, a divisão do trabalho na Oficina de Jogos seguiu a divisão proposta para a atividade construcionista (ver capítulo 7), com os papéis de aprendiz, facilitador, designer da atividade, ator e fonte de referência. Outros papéis, como o de patrocinador, administrador etc., que participaram da organização da Oficina, mas não de sua execução, não foram considerados.

Basicamente, conforme mencionado anteriormente, os alunos assumiram os papéis de aprendizes e fontes de referência, enquanto o professor assumiu os papéis de facilitador, designer da atividade e fonte de referência. Todos, na medida em que davam opiniões sobre a atividade, também cumpriram o papel de ator.

A comunidade. Como a Oficina aconteceu em uma das salas fechadas do Instituto do III Millennium, a comunidade ficou restrita, principalmente, ao professor e aos 4 alunos de cada turma.

No caso de uma análise mais detalhada, também deveriam ser considerados os autores dos exemplos do Cocoa que foram usados como referência, as demais pessoas que visitavam o III Millennium, os familiares e os colegas dos participantes. No entanto, a influência destes sujeitos não apareceu explicitamente na Oficina e julgou-se desnecessário incluí-los no trabalho.

9.2.1.3 As ações

Neste trabalho, procuramos nos concentrar nas ações relacionadas ao aprendizado construcionista, sub-atividade principal da educação construcionista. Trabalhos futuros poderão analisar outras sub-atividades importantes, como a do facilitador ou a da fonte de referência.

Idealização. Antes de se pedir que os aprendizes idealizassem o tipo de jogo que gostariam de desenvolver, julgou-se necessário fazer uma introdução às potencialidades do software Cocoa. Na primeira sessão da Oficina foi feita uma introdução geral ao tipo de trabalho que seria feito e às ferramentas que seriam utilizadas. Depois, foram necessárias mais 5 sessões discutindo características básicas do software até sentir que os aprendizes já tinham condições de propor seus próprios projetos⁴⁷.

As propostas iniciais foram apresentadas pelos aprendizes em uma folha de planejamento que perguntava “Que jogo eu pretendo construir?” (ver Figura 21). Foi enfatizado que esta proposta poderia ser alterada com o correr da Oficina, mas que as mudanças teriam que ser justificadas no diário-de-bordo.

Construção. A construção dos jogos foi feita quase que totalmente no computador. Os aprendizes não tinham outros materiais à disposição com que pudessem brincar ou construir protótipos. Na maior parte dos casos, a atividade da Oficina se resumia a escrever um pouco no diário-de-bordo e ficar a maior parte do tempo programando no Cocoa.

Avaliação. A parte de interpretação do que estava sendo feito também ficou bastante centrada no computador. O aprendiz programava um pouco e logo em seguida testava o que havia feito. Em raros momentos pedia opiniões aos demais participantes. Com isso, a interpretação mais relacionada com conteúdos

⁴⁷ Esta divisão da Oficina em duas etapas, uma de introdução ao Cocoa e outra mais focada na construção dos jogos, será melhor descrita mais adiante, na seção sobre história e desenvolvimento da atividade.

específicos (como ecologia, reciclagem de lixo, saúde e outros que poderiam estar relacionados ao Parque) e atitudes foi relegada a um segundo plano. A falta de incentivo à discussão entre os aprendizes foi, na nossa opinião, um dos aspectos da Oficina que poderiam ser aprimorados em uma próxima vez em que ela fosse desenvolvida.

E no que se refere à comparação, como a idealização estava no papel e a construção era feita no computador, não havia nenhum artefato externalizado específico que permitisse comparar o que havia sido idealizado com o que foi feito. Cada aprendiz fazia sua própria análise subjetiva da situação.

Depuração. Na depuração, raramente os aprendizes se voltavam para o diário-de-bordo. Eles programavam um pouco do jogo, executavam, viam o resultado e voltavam a programar até ficarem satisfeitos. A maior parte das questões era respondida com auxílio das próprias facilidades de depuração presentes no Cocoa. Quando ocorria um problema mais complexo, o facilitador era chamado. Não foram feitas discussões específicas sobre os temas escolhidos, nem sobre as diferentes maneiras de se testar um software .

Descrição. Além da descrição efetuada no próprio ato de programar, os aprendizes eram incentivados a escrever no diário-de-bordo antes, durante e depois das sessões.

Na Oficina de Jogos foram usados dois tipos de diários-de-bordo: o dos aprendizes e o do professor. O dos aprendizes foi baseado no *Designer's Notebook* usado por Yasmin Kafai e Idit Harel em suas pesquisas com as crianças do Projeto *Headlight* (HAREL, 1991; KAFAI, 1995). Tanto na Oficina quanto nos EUA, ele cumpria dois objetivos: o primeiro, criar um espaço para os aprendizes documentarem as idéias e mudanças que fossem surgindo em seus projetos e, com isso, incentivar a reflexão sobre o estava sendo feito; o segundo, auxiliar o facilitador na avaliação do andamento da atividade e na determinação de quando e como intervir.

O diário-de-bordo dos aprendizes só foi utilizado nas sessões da Oficina onde eles estavam desenvolvendo seus jogos. Ele era composto por 4 tipos de folhas que ficavam à disposição dos participantes. O primeiro, descrito acima, na idealização, incentivava a externalização das idéias que o aprendiz gostaria de desenvolver como objetivo da Oficina (ver Figura 21).

O segundo, que era entregue no início de cada sessão, pedia que o aprendiz descrevesse os seus planos para aquele dia e deixava espaço para ele escrever e desenhar o que desejasse (ver Figura 22).

O terceiro, que era entregue ao final de cada sessão, pedia para o aprendiz descrever os problemas que teve durante a sessão, as mudanças efetuadas e suas razões e os planos e idéias que ele gostaria de desenvolver na sessão seguinte (ver Figura 18).

O quarto, era uma folha padrão do Instituto do III Millennium que pedia um depoimento dos participantes sobre o que eles haviam achado da Oficina de Jogos (ver Figura 23).

Na realidade, segundo Holland e Reeves (1996), esta não utilização do diário-de-bordo do modo esperado pode ser explicada pela diferença de perspectiva entre os participantes da atividade. Para o facilitador e para o designer da atividade, o diário-de-bordo era uma necessidade clara. Para os aprendizes, era mais uma tarefa a ser cumprida. A solução, tal como colocada por estes autores, seria promover, durante a própria atividade, o desenvolvimento de uma perspectiva comum a todos e que fosse própria do grupo (pg. 275).

Projeto
"Oficina de Jogos"

Meu nome: Zedivil R Mendes **Data de hoje:** 03/03/97

Meus Planos para hoje:
Nossa primeira etapa vai fazer a entrada
do jogo (apresentação) e a primeira fase.
Na 1ª fase o personagem principal deve ser
liberado de uma caixa apenas se todos os links
for ligado para os lados.

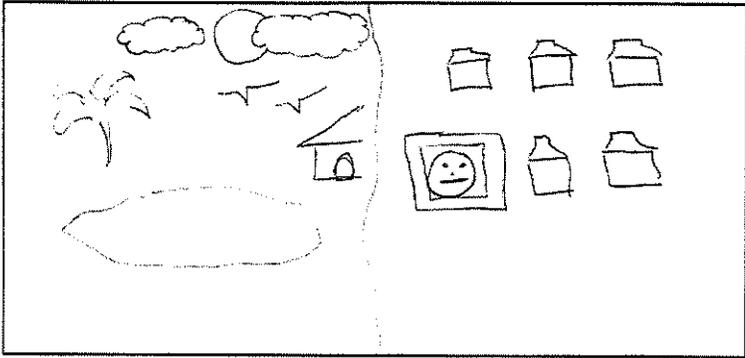


Figura 22 – Folha do diário-de-bordo com as idéias do dia

O diário-de-bordo do professor não foi planejado de antemão. Ele surgiu durante a própria Oficina como resultado das anotações que o professor fazia, em um caderno comum, antes, durante e depois das sessões.

Refletindo bem os papéis de facilitador e designer da atividade exercidos pelo professor, estas anotações acabaram por registrar tanto observações pedagógicas quanto técnicas. Nas pedagógicas eram descritos os objetivos esperados para cada sessão, observações sobre a evolução de cada aluno e sobre a turma em si e

também sobre algumas atitudes que o professor havia tomado durante a aula. Nas técnicas, eram descritos alguns problemas detectados na organização das sessões, na utilização do software e também como determinados acontecimentos repercutiam na dinâmica da sala.



CURSO - OFICINA DE JOGOS
PROFESSOR - LEO BURD
PERÍODO - 12/02 - 6/03 (10 sessões)
DATA - 06/03/97
HORÁRIO - 15:00
NOME - Zedson de Paiva Mendes

DEPOIMENTO

É muito bom eu poder fazer meu próprio jogo, aquele que eu vivo procurando de nos jogos e não ensentis.

O jogo é muito fácil de fazer mas para ter muito ~~de~~ adrenalina é preciso muito tempo.

O curso foi muito bom, e se fosse possível gostaria de fazê-lo mais vezes para poder me "furar".

[Assinatura]

Figura 23 – A folha para depoimento dos participantes da Oficina de Jogos

Em várias situações, para não esquecer o que havia acontecido, o professor tinha que parar de observar os alunos para anotar algo que lhe tivesse chamado a atenção e que poderia ser melhorado na atividade. Nestes casos, ele tinha que gerenciar a sua dupla função de facilitador e designer.

Após o término da atividade, sentiu-se falta de relatórios impressos sobre a utilização do software que pudessem ser referenciados na análise. Qualquer observação que iria ser feita dependia de se ter um computador com o Cocoa e os trabalhos dos alunos instalados. Ajudaria muito se fosse possível imprimir o tabuleiro e as descrições dos componentes que faziam parte dos jogos.

9.2.1.4 As operações

Como explicado pela Teoria da Atividade, não é possível identificar os conjuntos de operações que compõem uma determinada ação antes de ela ser executada. O máximo que se consegue é, à posteriori, a partir de descrições e da própria visualização da ação, levantar um repertório das operações mais utilizadas.

Infelizmente, na Oficina, não havia nenhum mecanismo específico para o registro das operações. A câmera de vídeo, que poderia servir para isso, só foi utilizada no final para registrar a entrevista com os participantes. O próprio computador também poderia registrar os principais comandos dados e gerar um relatório. Percebe-se, no entanto, que a quantidade de operações a ser analisada seria gigantesca e que, antes de mais nada, seria necessário estipular critérios que orientassem a sua observação. Idealmente, o facilitador deveria ter mecanismos para analisar os dados coletados conforme sua necessidade.

Operacionalização e conceitualização. Conforme discutido anteriormente, o principal objetivo pedagógico da Oficina de Jogos era incentivar reflexões sobre o Parque do Ibirapuera. Para que isso desse certo, era importante que os aspectos relacionados a procedimentos da Oficina e uso dos artefatos fossem operacionalizados o mais facilmente possível. Por outro lado, como boa parte das ferramentas e conceitos eram novos, esta operacionalização acabou tomando um tempo considerável da atividade. Só nas últimas sessões os aprendizes se sentiram mais fluentes para tratar diretamente de seus projetos.

Também se esperava que na Oficina, as principais conceitualizações fossem mais dirigidas às questões relacionadas ao Parque do Ibirapuera. Na nossa opinião, isso não se deu da forma esperada e o maior foco de atenção dos aprendizes acabou recaindo sobre o Cocoa em si e as características específicas de sua programação.

Na realidade, talvez devido à curta duração da Oficina e à falta de incentivo à discussões sobre o Parque, os aprendizes acabaram trabalhando questões que já lhes eram conhecidas, sem, necessariamente, pararem para refletir sobre isso. Maiores detalhes sobre estas conceitualizações serão apresentados na seção sobre a análise do artefato computacional.

9.2.2 Orientação a objetos

Sobre o produto e a motivação da atividade. Sob o ponto de vista mais concreto, o objetivo da Oficina de Jogos era possibilitar a criação de jogos de computador pelos alunos. Porém, atendo-se à idéia de que o produto, para o Construcionismo, pode ser entendido como que um pretexto para que o aprendiz descubra determinados conceitos e reflita sobre o seu desenvolvimento na atividade, não se esperava que os aprendizes concluíssem a Oficina tendo construído um jogo completo ou sofisticado. Isso, provavelmente, seria bom para atender à expectativa que eles mesmos haviam se colocado no início do trabalho – o que deveria ser muito respeitado – mas não era algo considerado essencial para a atividade.

Na realidade, ao término da Oficina, praticamente nenhum aprendiz havia conseguido finalizar totalmente o que havia proposto. Isso fez com que, na maioria

das vezes, ao perceberem que não haveria tempo suficiente, os próprios aprendizes optaram por simplificar seus projetos.

Houve um caso, no entanto, em que mesmo simplificando ao máximo o que desejava construir, as dificuldades encontradas por um aprendiz foram tão grandes que o facilitador teve que intervir e ajudá-lo a fazer algo que fosse minimamente satisfatório e que o mantivesse motivado a continuar freqüentando o Instituto do III Millennium⁴⁸. Nestes casos, na nossa opinião, intervenções como esta por parte do facilitador são de fundamental importância.

Na Oficina de Jogos, tal como defendido pelo Construcionismo, ficou evidente que a motivação da atividade não estava unicamente relacionada ao produto que os aprendizes estavam construindo, mas também ao processo e à forma com que as coisas eram encaminhadas. Para o pessoal da Oficina, o fato de terem sido selecionados por mérito, a liberdade de opção na criação dos jogos, a possibilidade de lidar com ferramentas avançadas, ser bem tratado e não ser forçado a andar num ritmo que não fosse o seu contribuíram muito para a continuidade do trabalho.

Embora dois dos alunos tenham abandonado a Oficina no meio, em várias ocasiões ficou patente que os participantes realmente estavam motivados com a atividade: presença maciça nos dias de chuva, a vontade de levar material para estudar em casa e a espontaneidade de chegar mais cedo e começar a trabalhar antes da aula e outras. Situações como esta raramente se vêem em atividades educacionais tradicionais.

Reforçando esta idéia da motivação pelo processo, vale notar que mesmo o produto da atividade era virtual: um software. O único elemento concreto que os aprendizes levaram da Oficina de Jogos foi um certificado de conclusão.

Ainda assim, concordamos que seu grau de satisfação com a atividade teria sido bem maior se tivessem conseguido finalizar os seus jogos. Se o jogo produzido funcionasse nas máquinas que eles acessam normalmente (máquinas com MS-Windows), certamente uma boa recordação da Oficina seria um diskette com o jogo desenvolvido pelo aprendiz.

Também é interessante notar que, mesmo com a motivação originando-se de fatores tecnológicos e pessoais e mesmo não havendo discussões mais explícitas

⁴⁸ Assim foi com M.Y.M.O., que apresentou muitas dificuldades de raciocinar com o Cocoa. Inicialmente, ele queria fazer um jogo de basquete com torcida, jogadores, etc.. Depois de um tempo achou melhor mudar para um jogo de futebol e acabou contentando-se em fazer só um jogador, a bola e metade do campo. Ainda assim, não conseguia fazer as coisas funcionarem e ficava brincando com o editor de figuras do Cocoa. Com ajuda, ele conseguiu finalizar sua última proposta e saiu extremamente contente com o trabalho desenvolvido na Oficina.

Não acreditamos, no entanto, que as dificuldades encontradas por este aprendiz sejam representativas da escala do que normalmente se encontrará em atividades como a Oficina de Jogos. Pelo seu próprio perfil e atitudes na Oficina, percebe-se que ele deveria possuir algum distúrbio de aprendizagem que exigiria uma atenção mais especializada.

sobre o que estava sendo feito, uma breve análise dos projetos desenvolvidos demonstra que os alunos chegaram a refletir sobre características do Parque que lhes chamavam mais a atenção.

A Figura 24 mostra alguns exemplos de jogos que foram desenvolvidos durante a Oficina: (a) um jogo de futebol; (b) um jogo ecológico geral; (c) um jogo falando sobre ladrões; (d) um jogo sobre o Instituto do III Millennium; (e) um jogo sobre a pista de cooper; (f) um jogo sobre lixo reciclável.

Neste aspecto, a escolha do tema foi de fundamental importância. O conceito de se “criar um jogo sobre o Parque” apareceu como algo significativo para todos e o fato de este tema ter sido deixado meio aberto permitiu que os aprendizes explorassem esta idéia da forma que lhes fosse mais conveniente. O software também contribuiu para isso, mostrando-se flexível o suficiente para atender a multiplicidade de idéias que apareceram.

Por outro lado, por mais que tenha trazido bons resultados, questionamos um pouco esta idéia de criar jogos como uma forma de levar os aprendizes a perceberem determinados conteúdos. É como se o conteúdo em si não oferecesse nenhum mérito por si só e que, por isso, precisasse ser enfeitado com algo para que lhe fosse dado valor.

Segundo o Construcionismo, o ideal seria que os aprendizes experimentassem uma situação onde o conhecimento-tema da atividade fosse intrinsecamente motivador, isto é, uma situação em que, no caso, os problemas do parque se mostrassem realmente importantes para eles e não fossem utilizados como simples complementos ou obstáculos para o jogo. Talvez, partindo-se de um histórico sobre os parques urbanos, sua importância e formas de engajamento, fosse possível se obter um tema mais intrinsecamente motivante para a atividade.

Sobre os materiais utilizados. Basicamente, os participantes da atividade só tinham contato com o computador e com o diário-de-bordo, os quais estavam à sua disposição no horário das sessões. Não era utilizado nenhum material que não estivesse presente na sala da atividade.

E, mesmo dentro da sala, raramente os aprendizes saíam do computador para fazer alguma coisa. De um modo geral, todo o material usado na construção dos jogos era digital e era constituído, principalmente, pelos objetos e artefatos disponibilizados pelo Cocoa, isto é, tabuleiros, personagens, etc.

Uma discussão mais aprofundada sobre esta questão pode ser encontrada neste mesmo capítulo, na seção sobre a análise do artefato computacional.

Sobre as relações entre os sujeitos. No caso da relação entre os sujeitos, na Oficina de Jogos o que mais se encontrava eram conversas pontuais e espontâneas entre os próprios aprendizes ou entre os aprendizes e o facilitador. Não houve nenhum contato planejado com pessoas de fora ou mesmo com os participantes de outras turmas. O contato era pessoal e direto.

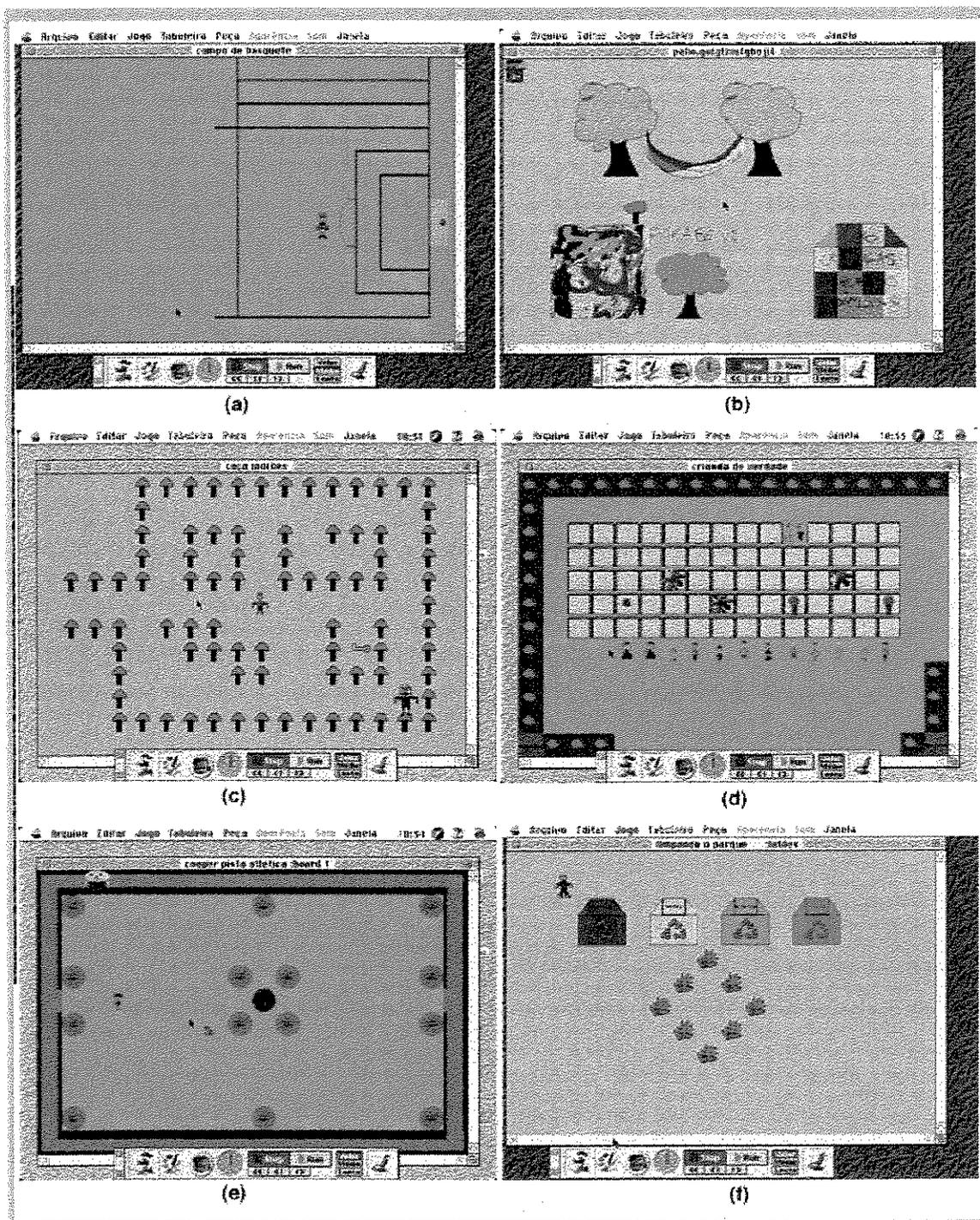


Figura 24 – Telas dos jogos criados durante a Oficina

A partir de uma análise à posteriori, percebe-se que houve uma falta de incentivo às discussões sobre questões relacionadas com o Parque ou com os métodos utilizados na programação dos jogos. De uma forma geral, o tempo curto, a atração das novas ferramentas e o desafio de fazer seu próprio jogo pareceram motivos suficientes para manter os aprendizes centrados mais no computador do que na opinião de seus colegas.

Arquitetura da sala. Vale notar que, para facilitar o fluxo de idéias, os computadores dos alunos foram dispostos, propositadamente, próximos uns dos outros e de forma que os alunos pudessem ver facilmente o que os demais estavam fazendo (ver Figura 25). Para aumentar a diversidade das trocas, pediu-se aos alunos que não sentassem ao lado do mesmo colega dois dias seguidos. Com o auxílio da rede de computadores, eles podiam continuar seu trabalho de qualquer máquina que desejassem e ainda tinham de, se quisessem, trocar arquivos com os colegas.



Figura 25 – Disposição dos participantes na sala

Este incentivo realmente repercutiu nos trabalhos desenvolvidos ao longo da Oficina. Era fácil perceber, por exemplo, que quando um aluno aprendia uma nova funcionalidade do software (como trabalhar com sons, ou controlar o movimento dos personagens com o teclado) ou explorava um novo conceito interessante sobre o parque, rapidamente os alunos que sentavam ao seu lado já se motivavam e acabavam incorporando as novas idéias em seus próprios projetos.

Percebeu-se que, por outro lado, se os aprendizes fossem mantidos mais afastados e isolados, este intercâmbio e cooperativismo teria sido muito diferente ou mesmo não ter ocorrido. Assim sendo, dependendo do tipo de interação desejada, a forma com que a sala for disposta pode ter muito a contribuir.

Outras questões relacionadas à orientação a objetos serão abordadas na seção que trata da análise do artefato computacional.

9.2.3 Internalização e externalização

Internalização. A estrutura da Oficina de Jogos incentivou uma série de práticas sociais com o intuito de que elas fossem internalizadas pelos participantes. A principal delas foi o incentivo ao planejamento e à reflexão sobre os projetos que estavam sendo criados. Com este fim, especialmente nas sessões onde já estavam desenvolvendo os jogos, os aprendizes eram convidados a fazer uma descrição, no diário-de-bordo, do que pretendiam fazer, dos problemas encontrados durante o dia e das estratégias pretendiam adotar.

A reflexão sobre a atividade também foi motivada pelas perguntas efetuadas durante o depoimento em vídeo, que foram:

- Gostou da atividade? O que foi bom/ruim?
- Agora que você já está mais acostumado com o Cocoa, se fosse começar o jogo novamente, o que você faria de diferente?
- Mostre o que você fez: no geral, tipos, aparências, ...
- Como você compara o trabalho no Cocoa com aquele que vocês estavam acostumados no Logo?

Como será melhor aprofundado na seção 9.2.4, com o intuito de facilitar a internalização das operações do Cocoa, as sessões iniciais da Oficina foram organizadas de modo especial. Nelas, primeiramente os aprendizes acompanhavam uma exposição do professor mostrando quais era e como funcionavam as operações básicas do software. A seguir, os aprendizes repetiam o que havia sido feito e tentavam criar algo que utilizasse o conceito passado.

Já com o intuito de incentivar o intercâmbio de idéias, como discutido na seção anterior, os computadores dos alunos foram colocados bem próximos uns dos outros. Isso contribuiu para que alguns conceitos realmente circulassem pela turma e passassem de um projeto para outro.

Refletindo-se sobre estes procedimentos e sobre os objetivos originais da Oficina de Jogos, percebe-se que não houve muitos incentivos para que os alunos discutissem ou ponderassem sobre as características do Parque, ou mesmo da utilização do Cocoa, que eles considerassem mais relevantes. Na nossa opinião, a Oficina de Jogos poderia ter incentivado mais a discussão sobre estes temas. Talvez, se os aprendizes fossem motivados a trabalhar em grupo, os debates aparecessem mais naturalmente.

Externalização. No que se refere à externalização, algo semelhante também aconteceu. Houve uma grande preocupação com a descrição das estratégias sendo adotadas, ou dos objetivos propostos por cada um (isso tudo no programa sendo feito e no diário-de-bordo), mas não uma ênfase nos temas escolhidos, na sua importância para os aprendizes e nos sentimentos que eles estavam tendo com relação à atividade.

De um modo geral, saiu-se da Oficina com a impressão de que a qualidade da descrição poderia ser aprimorada. Talvez, se os estudantes tivessem uma maior

quantidade de dias para desenvolverem propostas mais ousadas, se o espaçamento entre as sessões fosse maior, ou se eles tivessem que passar diretrizes para um parceiro sobre o que deveria ser feito, as descrições ganhariam importância e seriam feitas com mais afinco.

Mesmo o incentivo às discussões poderia ter contribuído mais para a externalização do pensamento dos aprendizes.

Apesar disso, por mais precários que tenham sido, os diários-de-bordo dos aprendizes e do professor, junto com a coletânea dos programas desenvolvidos e a gravação em vídeo dos depoimentos, se mostraram bastante importantes para o facilitador e o designer da atividade e contribuíram muito para o desenvolvimento do presente trabalho.

9.2.4 História e desenvolvimento

9.2.4.1 História

Historicamente, tal como discutido no início deste capítulo, a Oficina foi baseada nas pesquisas de Harel e Kafai e procurou trabalhar dentro do esquema de funcionamento do Instituto do III Millennium. Do trabalho de Harel e Kafai, a Oficina aproveitou a idéia de se ter alunos como construtores de jogos, de possibilitar 1 computador por aluno, de se usar o diário-de-bordo (inclusive a estrutura do diário-de-bordo foi aproveitada) e de incentivar a livre circulação de idéias.

Do trabalho do III Millennium, procurou-se respeitar o acesso gratuito aos computadores, o incentivo aos alunos que mais freqüentavam o Instituto e a inserção do Instituto no Parque do Ibirapuera. Provavelmente, se a Oficina fosse efetuada em outro lugar, estas características teriam que ser repensadas.

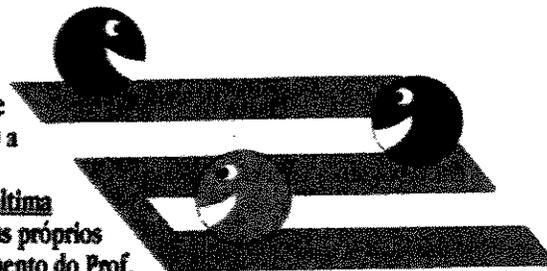
De qualquer jeito, esta foi a primeira vez em que a Oficina foi executada e, dada às características experimentais do presente trabalho e das necessidades específicas da versão utilizada do Cocoa (só funcionava em computadores Macintosh), não se tinha nenhuma expectativa de que, tão cedo, ela viesse a ocorrer novamente.

Sobre a estruturação da Oficina. Devido às particularidades do projeto, foi muito difícil conseguir os computadores utilizados na Oficina. Um fator importante foi a falta de popularidade da Apple, o que diminuiu muito a quantidade de potenciais patrocinadores que trabalhassem com esta marca.

Outro fator, mais complexo, foi o fato de ainda não haver, no país, uma cultura de integração universidade-indústria que incentivasse projetos deste tipo, em que ambos os parceiros muito têm a ganhar. De um lado, o meio acadêmico lucraria em poder aplicar conhecimento em atividades práticas e em, de certa forma, embasar os rumos que serão tomados no desenvolvimento de produtos como o Cocoa. De outro, a indústria consegue uma avaliação de alto nível e uma fonte de referência neutra e idônea para seus produtos.

OFICINA DE JOGOS

Durante o mês de fevereiro o Instituto do III Millennium está apoiando uma nova atividade dirigida às crianças. Trata-se do projeto "Oficina de Jogos" onde as crianças, de 10 a 14 anos, terão contato com computadores Macintosh de última geração e poderão montar seus próprios jogos. Tudo com acompanhamento do Prof. Leo Burd, pesquisador da UNICAMP. Participe deste novo projeto procurando o Instituto do III Millennium que fica no Show Room da Marquise do Ibirapuera. Não perca tempo! As vagas são limitadas.



Apoio Cultural:

**INSTITUTO DO III MILLENNIUM
SHOW ROOM DA MARQUISE
PARQUE IBIRAPUERA**

MacWorld
(011) 531-9644



E durante o mês de fevereiro a MacWorld estará com uma oferta imperdível:

Você compra o Performa 5215CD com 12 MB de memória RAM por apenas R\$ 1.999! E mais: Acrescente R\$ 300,00 ao seu pedido e leve uma impressora colorida Apple Color StyleWriter 2500 mais R\$ 100,00, leve uma câmera digital QuickCam mais R\$ 100,00, leve uma placa MPEG para captura de vídeo

GRÁTIS! Capa, mouse pad e curso de 3 horas do Sistema Operacional.

O Performa 5215CD já vem com Fax Modem e carregado com diversos softwares para trabalho, educação e diversão!

Apenas
R\$ 1.999



Figura 26 – Cartaz anunciando a Oficina de Jogos

Felizmente, conseguiu-se manter contato com uma das maiores revendas Apple no Brasil, a MacWorld, que se ofereceu para emprestar os computadores e para cuidar de sua manutenção pelo tempo da Oficina. Em troca, esta empresa entrou como "Apoio Cultural" à Oficina e, mediante consulta caso a caso, poderia utilizar o evento em seu plano de marketing (ver Figura 26).

Além de trabalhar com computadores emprestados por um curto período de tempo, a Oficina teria que se encerrar antes do término das férias escolares de seus alunos. Estas restrições fizeram com que o seu processo de preparação fosse acelerado e, por isso, apenas parte do material de apoio (manuais e tutoriais) foi traduzida. Com base na pequena experiência prévia que já havíamos tido com o

software, estimou-se que o material disponível e as 10 sessões seriam suficientes para um bom desenvolvimento do trabalho.

Já a seleção dos aprendizes ficou a cargo do Instituto do III Millennium. Desejava-se trabalhar com freqüentadores típicos do Instituto, tanto no que se referia ao gênero, quanto à diversidade sócio-econômica. Conforme já foi dito, a equipe do III Millennium optou por convocar, em primeiro lugar, aqueles membros que tinham o maior número de sessões com o Logo. Isso, segundo eles, serviria de estímulo para que as demais crianças viessem mais vezes participar das atividades do local.

Porém, devido ao curto espaço de tempo deixado para a convocação, não foi possível entrar em contato com os participantes que não tinham telefone, dentre eles, os meninos de rua.

Felizmente, a Oficina acabou concretizando-se e, de uma forma ou de outra, terminou atingindo seus objetivos e fornecendo subsídios para o presente trabalho.

9.2.4.2 Desenvolvimento

Como pré-requisitos para a Oficina de Jogos, esperava-se que os participantes fossem freqüentadores do III Millennium, tivessem familiaridade com algum sistema operacional como o MS-Windows e se mostrassem comprometidos com a atividade que seria desenvolvida. Ao término da Oficina, esperava-se que eles tivessem produzido minimamente, com o Cocoa, um jogo sobre o Parque e tivessem, com isso, adquirido mais motivação para continuar indo ao Instituto, aprendido a lidar com o software e estivessem mais conscientes do Parque do Ibirapuera.

Além do acompanhamento do professor e dos depoimentos no diário-de-bordo e em vídeo, não foi feita nenhuma avaliação mais formal da Oficina.

Organização das sessões. Em termos de organização, a Oficina de Jogos foi dividida em 2 fases principais: a fase de instrumentalização e a fase de fluência. A fase de instrumentalização tinha por objetivo apresentar a atividade e familiarizar os alunos com a funcionalidade e o potencial das ferramentas que seriam utilizadas, principalmente do Cocoa. A fase de fluência contava com esta experiência dos alunos e a transparência do Cocoa para que eles se concentrassem no desenvolvimento de seus projetos.

Cada uma destas fases tinha uma estrutura e um encaminhamento próprios que serão discutidos a seguir (ver Tabela 2).

A **fase de Instrumentalização**, inicialmente prevista para ocupar de 3 a 4 sessões, devido à quantidade de conceitos e ao ritmo da turma, acabou por usar 6 das 10 sessões da Oficina. Esta fase, sob o ponto de vista educacional, poderia ser considerada como um "degrau necessário" uma vez que, nela, os participantes deveriam estar, principalmente, focados no aprendizado dos procedimentos da atividade e no uso do ferramental, não no conteúdo a ser explorado.

Nos termos da Teoria da Atividade, o objetivo desta fase era operacionalizar as ações dirigidas ao computador e internalizar as práticas da Oficina. Idealmente, tanto a estrutura da atividade quanto as ferramentas a serem usadas (ou o software)

deveriam ser o mais intuitivos possível, o que reduziria em muito o esforço gasto nesta fase.

Tabela 2 - O enfoque da Oficina de Jogos ao longo do tempo.

Enfoque	Dia	Conteúdo
Instrumentalização	17/02	<ul style="list-style-type: none"> • Introdução à atividade • Criação e movimentação de personagens • Entrada e saída do sistema
	18/02	<ul style="list-style-type: none"> • Combinação de regras e mudança de aparência dos personagens
	25/02	<ul style="list-style-type: none"> • Seqüência de ativação dos personagens • Seqüência da ativação das regras • Conjuntos de regras
	26/02	<ul style="list-style-type: none"> • Variáveis
	27/02	<ul style="list-style-type: none"> • Turma 1: variáveis e gravação de sons • Turma 2: som e manipulação dos personagens via mouse e teclado
	28/02	<ul style="list-style-type: none"> • Turma 1: manipulação dos personagens via mouse e teclado; Início do projeto do jogo. • Turma 2: Som e projeto do jogo
	Fluência	03/03
04/03		<ul style="list-style-type: none"> • Desenvolvimento dos projetos pessoais
05/03		<ul style="list-style-type: none"> • Desenvolvimento dos projetos
06/03		<ul style="list-style-type: none"> • Desenvolvimento dos projetos • Encerramento da Oficina com depoimentos dos participantes

Por outro lado, é bom notar que uma mesma fase de instrumentalização pode ser utilizada para todas as atividades que envolvam as mesmas ferramentas. Como, no exemplo da Oficina de Jogos, ela serviu como uma introdução ao software Cocoa, provavelmente outras atividades em que os participantes da Oficina fossem utilizar este software teriam uma fase de instrumentalização muito reduzida.

Além disso, na Oficina de Jogos, a fase de instrumentalização se mostrou extremamente importante não apenas por abordar os problemas mais técnicos, mas também por ser um momento onde os participantes estavam se familiarizando com os colegas e com o professor. Sentindo-se entre amigos, eles ficavam mais à vontade para fazer perguntas e participar mais abertamente da atividade.

Sob o ponto de vista pedagógico, as sessões da fase de instrumentalização foram essencialmente instrucionistas. Na primeira parte da sessão, o professor apresentava um exemplo prático de algum conceito do Cocoa que os participantes ainda não conhecessem. Na segunda parte, os participantes deveriam reproduzir o exemplo e depois criar uma situação diferente em que este conceito pudesse ser usado.

Devido a este caráter instrucionista, esta fase da atividade exigiu um bom esforço por parte do professor para apresentar as características do software e, mesmo com a pouca quantidade de sessões, diversas dificuldades se mostraram aparentes.

No início, os participantes estavam, praticamente, no mesmo nível. Com o passar dos dias, no entanto, as diferenças entre e intra-turmas já começavam a aparecer – na 5ª sessão já houve necessidade de se trabalhar conceitos diferentes entre as turmas – e uma aula única para todos não parecia mais uma solução adequada. Este problema acabou agravando-se ainda mais com as faltas de alguns alunos e ajudou a levantar o questionamento sobre como oferecer orientação e suporte para aprendizes com problemas de horários e, mesmo, com velocidades ou estilos de aprendizado diferentes.

Provavelmente, a utilização de um material didático (algo como tutoriais específicos e materiais de referência) que apresentasse o software de modo adequado permitiria que cada um fosse caminhando segundo seu próprio ritmo e, além disso, deixaria o professor mais livre para atender aqueles que precisassem de uma ajuda mais específica.

Na época da Oficina, no site do Cocoa havia um conjunto de pequenos filmes e tutoriais explicando a utilização básica do Cocoa. Infelizmente, este material estava em inglês e, devido a restrições momentâneas, não foi possível traduzi-lo a tempo para a Oficina de Jogos.

É interessante notar que, embora a fase de instrumentalização seja algo necessário em toda nova atividade educacional, não encontramos, ao longo do desenvolvimento do presente trabalho, nenhuma referência construcionista indicando como ela deveria ocorrer. As referências consultadas, como as de Harel e Kafai, costumavam descrever o andamento dos alunos já na fase de fluência.

Por outro lado, a Teoria da Atividade enfatiza muito que, na criação de novas atividades, ou mesmo na inclusão de novos elementos em atividades já existentes, os participantes precisam de um tempo para operacionalizarem as novas ações e internalizarem a nova dinâmica. Como no caso da Oficina, boa parte dos elementos eram novos (computadores, software, tipo de proposta, etc.), era de se esperar que a instrumentalização demandasse grande parte da atividade.

Também convém realçar que, na fase de instrumentalização, não era esperado que os participantes se apropriassem de todos os conceitos passados a eles. A idéia era de que eles aprendessem a funcionalidade básica do software, ganhassem uma noção geral do que poderia ser feito com ele e onde conseguir maiores informações. Quando isso acontecia, os participantes passavam para a segunda fase da Oficina, a fase de fluência.

A **fase de fluência**, por ter uma ênfase mais centrada no planejamento e no desenvolvimento dos jogos dos participantes, teve uma organização diferente da fase de instrumentalização. Seguindo o modelo proposto por Kafai e Harel, em toda sessão os participantes se reuniam na "mesa de planejamento" para anotar suas idéias do dia. Depois, eles iam trabalhar no computador e, no fim da sessão, voltavam para a mesa de planejamento a fim de escrever o que haviam feito, suas dificuldades e novas idéias. Dos 80 minutos de cada sessão, praticamente 10 eram gastos no planejamento inicial, 60 no computador e mais 10 nas anotações finais⁴⁹. As anotações foram feitas em folhas especialmente criadas para este fim que depois eram reunidas formando os diários de bordo.

Apesar desta aparente rigidez, os alunos tinham flexibilidade para se estenderem por mais ou menos tempo com o diário-de-bordo e, inclusive, eram incentivados a fazer suas anotações no meio da sessão. Infelizmente, não havia espaço na mesa do computador para eles escreverem e isso provavelmente atrapalhou um pouco a dinâmica das descrições.

A relação entre as fases. Convém notar que a separação entre as fases de instrumentalização e de fluência não é estanque. Na fase de instrumentalização, os participantes eram incentivados a ir pensando nos jogos que iriam criar e em como aquela funcionalidade que estava sendo apresentada poderia ser utilizada. Na fase de fluência, toda vez que aparecia uma dúvida com relação à organização da atividade ou ao uso do software, o mediador dava explicações individuais ou em grupo.

Enquanto na fase de instrumentalização o professor atuava como fonte de referência e era muito exigido dos aprendizes, na fase de fluência as dúvidas apresentadas não eram tão freqüentes e o professor tinha mais tempo para dedicar-se a questões ou alunos específicos. Esta diferença de carga de trabalho do professor é importante e pode ser considerada no planejamento de aulas mais eficientes, onde ele trabalha com grupos de alunos em níveis diferentes da mesma atividade.

⁴⁹ Refletindo à posteriori, esta idéia de se separar o planejamento da execução do software é questionável. Por um lado, é bom gastar um tempo pensando antes de se ir para o computador, principalmente se o tempo e a quantidade de recursos é pouca. Por outro lado, esta prática parece não favorecer muito os aprendizes de estilo mais "escultor" e seria bom se os alunos pudessem fazer suas anotações também na hora em que estivessem programando. Às vezes, uma anotação indicando, no próprio programa, os pontos que deveriam ser alterados seria melhor tanto para o aprendiz quanto para o facilitador.

Na última sessão da Oficina, além de darem os últimos retoques em seus projetos, os participantes também refletiram e deram depoimentos sobre o trabalho desenvolvido. Parte destes depoimentos foi gravado em vídeo e parte escrita em folhas de papel que o Instituto do III Millennium utiliza exclusivamente para isso. Nos depoimentos em vídeo, os participantes demonstraram no computador o projeto que estavam desenvolvendo e respondiam algumas perguntas sobre o que tinham achado da atividade e quais eram suas sugestões.

Outras considerações relacionadas com o tempo da Oficina. Com o decorrer da Oficina, a duração das sessões se mostrou compatível com a sua periodicidade. Provavelmente, se fossem mais espaçadas, a sua duração deveria ser maior para permitir uma melhor recapitulação dos alunos.

O intervalo entre as turmas do mesmo dia, que era de 20 minutos, também se mostrou adequado e extremamente necessário para que o facilitador pudesse fazer anotações e refletir sobre sua prática. Por várias vezes, esta reflexão ajudou a melhorar a qualidade da sessão seguinte. O período no final do dia também era importante. Nele, além de fazer novas anotações, o professor também fazia cópias dos trabalhos dos alunos por segurança e para futura análise.

É interessante notar que, nas escolas tradicionais, normalmente o professor vai diretamente de uma aula para outra, sem tempo para pensar no que fez e melhorar o seu trabalho.

Mesmo durante as sessões, era importante que os aprendizes não fossem cobrados o tempo todo e se sentissem mais à vontade para trocar idéias, testar hipóteses, errar e fazer de novo. A duração e dinâmica das sessões da Oficina até que permitiam isso. Infelizmente, a quantidade de sessões é que se mostrou muito reduzida para um trabalho de melhor qualidade.

Sobre os estilos de aprendizagem. Complementando o que foi dito acima acerca dos diferentes ritmos e necessidades individuais percebidos, no decorrer da Oficina também ficou aparente a diferença entre estilos de aprendizado dos alunos. Por exemplo, uns gastavam mais tempo com a aparência de seus personagens, outros se preocupavam mais com os movimentos, outros com os sons e outros com a quantidade de fases e as teclas de controle. Alguns tinham mais facilidade e seguiam sozinhos com o auxílio de materiais de apoio. Outros precisavam de alguém ao seu lado, orientando cada passo.

Ao invés de uma diferenciação mais exata, cada aprendiz aparentava uma variedade de combinações entre um estilo mais “escultor” ou mais “planejador”. Neste sentido, a diferença entre o Z.R.M. e R.N.M. foi a mais evidente.

Z.R.M. era mais estético, enquanto que R.N.M. era mais funcional. Z.R.M. criava aparências fantásticas e não abria mão das idéias que estava seguindo no projeto. R.N.M. não parecia preocupar-se com os desenhos e tão pouco prendia-se às idéias iniciais que havia proposto.

Enquanto o trabalho de Z.R.M. parecia ter começo, meio e fim, o de R.N.M. era uma coletânea de idéias. Ele fazia uns rabiscos e já prosseguia trabalhando. Seu objetivo parecia mais ser explorar as possibilidades do software do que fazer algo mais

coerente. Seu jogo tinha 7 níveis, dúzias de tipos de personagens e fazia um monte de coisas diferentes. Sua curiosidade ultrapassava o computador e chegava ao monitor e à câmera de vídeo. Não havia sessão em que ele não perguntasse para que serviam os demais botões e conexões do computador, algo que, constantemente, o distraía da atividade.

Felizmente, a infra-estrutura preparada e o software tinham amplitude suficiente para suportar o trabalho de ambos os alunos. Ainda assim, com esta diversidade de encaminhamentos, restava a questão sobre como avaliar os projetos desenvolvidos. Eles deveriam ser analisados frente aos conhecimentos técnicos aplicados? Pela criatividade na exploração do tema? Pelos conhecimentos do parque? Pela interação com os colegas?

Hoje, de acordo até com as novas diretrizes educacionais vigentes (MINISTÉRIO DA EDUCAÇÃO, 1997), as práticas educacionais devem ser analisadas quanto aos conteúdos explorados, habilidades aplicadas e atitudes dos participantes. Todos estes aspectos deveriam ter sido melhor explicitados no planejamento da Oficina de Jogos.

Refúgios cognitivos. Por fim, a Oficina também mostrou que, independente de estilos, todos aprendizes tendiam a acomodar-se, fazendo coisas que já estavam habituados antes de avançar para algo que não tivessem compreendido bem. Por exemplo, uns ficavam mais no editor de figuras, outros brincando com regras simples. Parecia que só com o incentivo do facilitador é que eles se arriscavam a explorar novas funcionalidades do software. Nestas situações, era muito importante que o facilitador reconhecesse estes “refúgios cognitivos” e não forçasse os aprendizes a avançarem demais, fora do ritmo deles, o que poderia frustrá-los ou inibi-los.

9.2.5 Mediação

Conforme já pode ser depreendido do texto acima, o trabalho dos aprendizes na Oficina de Jogos sofreu influência das regras e divisão do trabalho da atividade e, mais evidentemente, pela mediação dos artefatos digitais e não-digitais (como o Cocoa e o diário-de-bordo, respectivamente) e pela do facilitador, o qual fazia uma “mediação humana” mais direta e ajudava na adequação da Oficina às restrições que apareciam a cada momento.

Na nossa opinião, cada tipo de mediador tem uma contribuição para dar e, idealmente, todos deveriam complementar os trabalhos uns dos outros. O tipo de contribuição que artefatos digitais e computacionais podem oferecer já foi discutido nos capítulos 6 e 8. A das regras da divisão do trabalho como um todo precisariam ser melhor exploradas.

De qualquer jeito, percebeu-se que o facilitador teve uma série de dificuldades para acompanhar o que os aprendizes estavam fazendo, orientá-los de acordo com suas necessidades e ainda registrar o seu desenvolvimento. Por outro lado, o simples acesso ao computador não era suficiente para que os aprendizes tocassem seus projetos. Dificuldades como esta poderiam ser minimizadas se houvesse uma maior integração entre os materiais didáticos desenvolvidos, o software disponível e o

próprio facilitador. Uma integração como esta, ou uma melhor divisão de funções entre os diferentes componentes da Oficina, no entanto, exigiria uma análise mais ampla da atividade educacional como um todo, uma análise como esta a que se propõe o presente trabalho.

9.3 O artefato computacional frente à Oficina de Jogos

Esta seção analisa o papel do artefato computacional e, mais especificamente, o do software Cocoa em relação a cada um dos aspectos da Oficina de Jogos discutidos na seção 9.2.

9.3.1 A infra-estrutura computacional básica

Cada aprendiz tinha à sua disposição, durante o horário da Oficina, um computador Macintosh Performa 5215CD ou similar com todos os aplicativos instalados (ver Figura 27). Esta família de computadores tem sido muito utilizada em escolas americanas e tem como principais atributos os recursos multimídia (leitor de CDROM, microfone e caixas acústicas) e a integração física dos componentes: monitor, CPU, microfone, caixas acústicas e modem compõem uma única peça, sem fios de conexão ou componentes móveis que poderiam se quebrar mais facilmente.



Figura 27 - O computador usado na Oficina de Jogos

Na Oficina, estes computadores estavam com uma versão em português do sistema operacional (no caso, o System 7.5.3 da Apple) e do teclado (com teclas para todos os caracteres e acentos da língua portuguesa). Estes dois recursos foram considerados essenciais para reduzir a dificuldade dos participantes com os novos computadores e também para facilitar a escrita com os acentos brasileiros.

Os 4 computadores dos participantes estavam ligados em rede a um outro computador que armazenava todo o trabalho desenvolvido. Isso permitia que os participantes não precisassem trabalhar todos os dias no mesmo computador e dava maior flexibilidade ao facilitador para alterar a disposição dos participantes na sala. A centralização dos dados também facilitou o trabalho diário de fazer cópias de segurança do material produzido pelos alunos.

O software escolhido para gerenciar a rede foi o *At Ease For Workgroups*, versão 3.0, desenvolvido pela *Claris Corporation*. Este software, além de tornar o acesso à rede quase que totalmente transparente para os participantes da Oficina, também permitia controlar a interação deles com o computador, restringindo seu acesso a apenas os itens escolhidos pelo designer da atividade. Sob o ponto de vista pedagógico, ambas as facilidades ajudavam a focar a atenção dos participantes nos elementos pertinentes à atividade. Sob o ponto de vista de manutenção, essa facilidade diminuía os riscos de problemas com partes do sistema não diretamente ligadas à atividade.

Além de contarem com essa infra-estrutura computacional básica descrita acima, que é genérica e básica o suficiente para servir um sem número de atividades computacionais diferentes, os computadores dos participantes tinham instalado uma versão parcialmente traduzida do Cocoa e mais material de suporte a este software. Este material era composto por um guia de referência rápida online explicando os principais comandos e uma série de exemplos de programas desenvolvidos com o Cocoa que ilustravam algumas de suas possíveis aplicações.

9.3.1.1 O software Cocoa

Originariamente conhecido como KidSim (CYPHER e SMITH, 1995; SMITH et al., 1994), o Cocoa é um software que permite às crianças criarem suas próprias simulações e jogos computacionais. Com o Cocoa, elas constroem mundos onde os personagens se movem e interagem entre si em cima de uma espécie de tabuleiros ou cenários bidimensionais.

A Figura 28 mostra a tela do software Cocoa com um dos jogos criados durante a Oficina. Nele o herói precisa fugir dos bandidos do parque para recuperar uma chave e, com ela, salvar sua amada.

Os personagens são compostos por regras, aparências, sons e propriedades (ver Figura 29). As regras definem como o personagem deve se comportar em cada situação específica. As aparências permitem que ele mude a forma com que é visto na tela, os sons definem os ruídos que pode emitir e as propriedades são usadas para guardar dados a respeito do personagem. Algumas propriedades já vêm predefinidas, como é o caso do nome, do tipo e da aparência atual do personagem. Outras podem ser criadas pelo usuário para armazenar informações específicas, como número de vidas que ainda restam (para um personagem de joguinho), quantidade de energia, direção do movimento, etc.

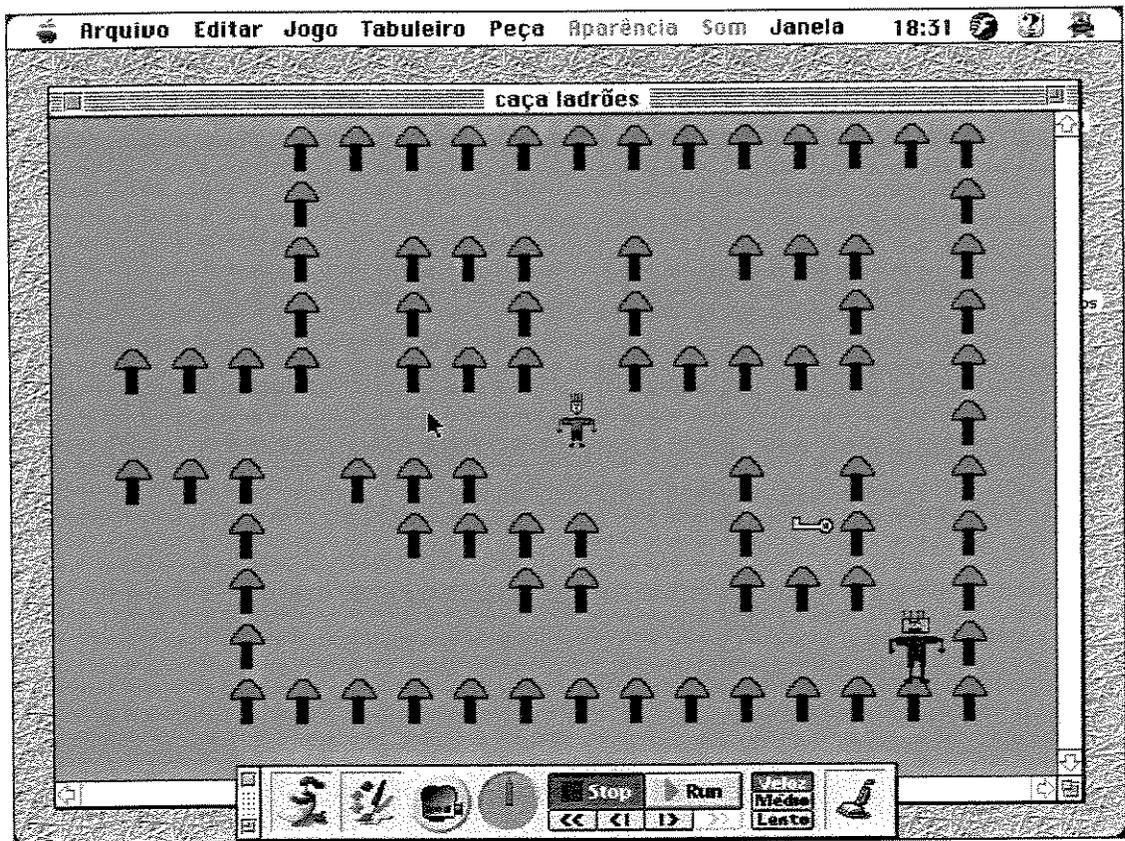


Figura 28 – A tela do software Cocoa.

Por exemplo, a criança poderia criar uma regra que fizesse com que, toda vez que se encontrasse ao lado de uma pedra, o personagem mudasse de cor, emitisse um grunhido e saltasse a pedra. Este personagem também poderia ter uma propriedade associada com a sua quantidade de energia que fosse decrementada toda vez que ele saltasse a pedra.

Os mundos criados no Cocoa têm um relógio cuja velocidade pode ser controlada pelo usuário. A cada tique do relógio é dada uma chance para os personagens avaliarem sua situação e executarem a primeira de suas regras que for aplicável. Quando todos os personagens tiverem executado suas regras, o relógio avança e um novo ciclo de interações é processado.

O Cocoa é um software essencialmente gráfico e de manipulação direta. A criança, praticamente, só precisa usar o teclado para dar nomes aos elementos do jogo e para fazer algumas operações específicas com as propriedades.

As regras são criadas através de um mecanismo chamado Programação por Demonstração (*Programming by Demonstration*), onde a criança coloca o sistema em modo de gravação, seleciona com o mouse uma região do tabuleiro que demarque a condição na qual a regra deverá ser disparada (como o personagem ao lado da pedra) e executa, manualmente, todas as ações que a regra deverá efetuar (como alterar o valor de alguma propriedade ou arrastar, com o mouse, o personagem por cima da pedra). O Cocoa então generaliza estas ações para que

elas sejam executadas toda vez que qualquer personagem do tipo em questão se encontrar em uma situação similar àquela para qual a regra foi definida.



Figura 29 – As propriedades e as regras de um personagem criado no Cocoa

A Figura 30 mostra a criação de uma regra que ensina o personagem a se mover para a direita toda vez que o espaço ao seu lado estiver vazio. No passo (a), o personagem já está selecionado. No passo (b), a área a ser considerada na ativação da regra é demarcada. No passo (c), o usuário demonstra a ação que deverá ser efetuada pela regra, isto é, mover o personagem. A Figura 31 mostra como ficou a regra depois de pronta e já com um comentário inserido pelo aprendiz.

As aparências são criadas através de um editor gráfico repleto de ferramentas de desenho que permite, inclusive, que o usuário importe figuras externas, como fotos digitalizadas ou desenhos produzidos em outros aplicativos. Os sons são gravados através de um mecanismo que simula um gravador digital com o microfone do computador. Com o editor gráfico e o gravador, a criança consegue trazer, para os mundos criados no Cocoa, sons e imagens que façam parte de seu cotidiano.

Uma vez completos, os mundos Cocoa podem ser salvos e distribuídos para os colegas. O Cocoa também traz facilidades para que os mesmos possam ser incorporados em páginas da WWW para serem acessados via Internet.

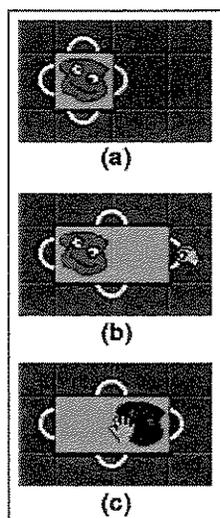


Figura 30 – Passos da criação de uma regra no Cocoa.

Convém notar que, na época em que este trabalho foi escrito, o Cocoa ainda estava em desenvolvimento e não havia nenhuma versão comercial disponível. Logo que as primeiras versões minimamente funcionais do software foram lançadas, diversos estudos informais foram realizados. Em especial, merecem destaque as pesquisas desenvolvidas na Universidade de Nottingham no Reino Unido (GILMORE et al., 1995) e na Universidade do Colorado em Boulder, nos EUA (RADER et al., 1997; BRAND et al., 1997; LEWIS et al., 1997; BRAND e RADER, 1996), que buscavam analisar o potencial educacional do Cocoa e fazer sugestões para o seu design.



Figura 31 – A nova regra depois de pronta.

Boa parte das recomendações propostas já foi incorporada ao software em suas versões mais recentes. Outras ainda estão por ser estudadas e, à medida em que novas pesquisas e alterações forem surgindo, o Cocoa vai sendo aprimorado.

Na Oficina de Jogos foi utilizada uma cópia da versão DR1, parcialmente traduzida para o português especialmente para esta atividade. Junto com o software, também foi feita uma tradução parcial de seu guia de referência rápida, que traz uma breve descrição de cada um dos comandos disponíveis.

Autonomia com relação ao professor. Devido ao esforço que seria gasto na tradução e a pouca quantidade de alunos por turma, optou-se pela não utilização do tutorial e nem do material de apoio que estava disponível junto com o Cocoa.

Acreditava-se que o professor, como fonte de referência, poderia suprir esta deficiência. No entanto, como será discutido mais adiante, mesmo com a turma pouco numerosa, houve momentos, principalmente quando os aprendizes que haviam faltado retornavam, em que este material de apoio poderia ser de grande utilidade. Por essa razão, na nossa opinião, a avaliação do uso educacional de um artefato computacional deve levar em consideração o material de suporte que acompanha este artefato.

Atualmente, devido a questões internas, o Cocoa saiu da Apple e está sendo desenvolvido sob o nome de StageCast por uma empresa de mesmo nome fundada pela equipe original de desenvolvedores da Apple. Sua primeira versão oficial foi lançada recentemente e funciona tanto em PCs como em computadores Macintosh. Por meio do site <http://www.stagecast.com/> está sendo fomentada a discussão sobre a utilização educacional deste software, dúvidas estão sendo respondidas e exemplos de utilização estão sendo disponibilizados.

Nas seções que se seguem, é feita uma análise do papel do computador e, mais especificamente, do Cocoa, na Oficina de Jogos.

9.3.2 A estrutura hierárquica da atividade

9.3.2.1 A atividade em si

De uma forma geral, pode-se dizer que é impossível pensar a Oficina de Jogos sem o computador e sem o Cocoa. Na realidade, ela foi criada com este software em vista e o resultado de sua utilização se mostrou muito positivo. O Cocoa possibilitou a criação de jogos pelos aprendizes e ainda se mostrou extremamente atraente tanto pelo fato de ser algo novo, quanto pelo estilo e abertura de opções que deixava à disposição.

Analisando a contribuição do Cocoa frente à atividade como um todo, pode-se dizer que ele é um software desenvolvido para suportar o trabalho individualizado de seus usuários na construção de jogos. O usuário do Cocoa é visto como um "construtor de mundos", não especificamente como um aprendiz ou facilitador (pelo menos não nos moldes construcionistas), daí a necessidade de complementar este software com outros artefatos na execução destes papéis.

9.3.2.2 As ações

Conforme discutido anteriormente, neste trabalho estamos nos concentrando principalmente na atividade dos aprendizes e em escala menor, na atividade dos designers, dos facilitadores, fontes de referência e demais papéis que compõem a atividade educacional construcionista. O tipo de suporte computacional oferecido na Oficina de Jogos também seguiu esta ênfase .

Idealização. Conforme descrito anteriormente, a idealização do jogo a ser desenvolvido só foi pedida depois de os alunos terem passado um tempo ganhando familiaridade com o Cocoa e descobrindo que tipo de coisas poderiam ser feitas com este software. Neste sentido, exemplos disponíveis e os exercícios desenvolvidos nas primeiras sessões da Oficina contribuíram bastante.

Ainda assim, a externalização das idéias que o aprendiz gostaria de desenvolver foram feitas no papel, em uma ficha especialmente criada com este fim. Embora houvesse a possibilidade de, no Cocoa, os usuários fazerem um pequeno sumário, descrevendo o mundo que estavam construindo, por falta de conhecimento ela acabou não sendo utilizada na Oficina.

Construção. Praticamente todo o material utilizado na construção dos jogos era composto por objetos e artefatos digitais. Os aprendizes não criavam protótipos de nenhum tipo antes de partirem para a implementação no computador. No máximo, descreviam parte de suas idéias no diário-de-bordo e seguiam trabalhando.

Conforme demonstrado na Oficina, o Cocoa aparentemente dispunha de recursos suficientes para a implementação de projetos interessantes. No entanto, durante o processo de construção, foram detectadas algumas dificuldades. As que mais chamaram a atenção estão descritas a seguir:

- Na criação de novos personagens, as aparências geradas, automaticamente, pelo Cocoa eram muito semelhantes. Houve vários casos em que os aprendizes confundiram o comando Criar, que cria um novo personagem de um novo tipo, com o comando Duplicar, que cria um novo personagem do mesmo tipo do personagem selecionado.
- A versão testada do software não tinha o comando "desfaz" (*undo*). Se o aprendiz cometesse um engano como, por exemplo, ter trocado o nome de uma variável em parte do programa, teria que se lembrar exatamente o que havia feito para poder consertar o que fez.

Avaliação (Interpretação e comparação). Como se espera de uma linguagem de programação, todo comando dado no Cocoa tinha uma resposta única que era executada. No entanto, devido ao processamento paralelo dos personagens, quando havia muitos comandos sendo **interpretados** ao mesmo tempo no tabuleiro, às vezes não era tão fácil perceber o que de fato estava acontecendo e o aprendiz ficava com a impressão de que alguma coisa não estava funcionando bem no sistema.

Para que o aprendiz possa questionar suas próprias estratégias, é muito importante que ele tenha confiança total no software que estiver usando e esteja familiarizado com o seu funcionamento, algo que só é conseguido com a prática e a experimentação.

Por exemplo, no Cocoa, a cada tique do relógio, o sistema tentava executar, em cada um dos personagens, a primeira regra que possuísse uma condição válida. As regras eram testadas de cima para baixo, de acordo com sua posição na "caixa de regras" e, uma vez concluídas, o sistema passava para o personagem seguinte.

A ordem de ativação dos personagens, por sua vez, estava relacionada com a seqüência em que os personagens foram criados. Assim, se em um jogo como o Pac Man fossem inseridos vários monstrinhos e, só depois, o bichinho come-come, todos os monstrinhos se movimentariam antes do come-come.

No projeto desenvolvido por M.S.S.P., a idéia era que cada personagem saísse de um desenho representando o III Millennium e falasse uma frase educacional. A autora queria que, em primeiro lugar, o personagem da esquerda falasse e que, depois, viessem aqueles que estavam mais à sua direita. Como os personagens não haviam sido criados na ordem em que iriam falar, o resultado não estava saindo conforme o esperado.

Para resolver problemas como este, seria interessante se os usuários pudessem visualizar, ou mesmo alterar, a ordem em que os personagens seriam ativados. Felizmente, o Cocoa oferecia uma série de possibilidades para a visualização do que estava acontecendo e para o acompanhamento detalhado de cada personagem. Ainda assim, este problema demorou um bom tempo até ser detectado.

Como discutido no capítulo 8, o computador tem uma série de limites quanto a interpretação dos significados do que foi feito pelo seu usuário. No Cocoa isto também era verdade. O software era capaz de interpretar as regras uma a uma, mas não poderia inferir se o conjunto fazia algum sentido. Por exemplo, se fosse criada uma regra chamada "leve o macaquinho para cima" que, na realidade, levasse o macaquinho para baixo, nada seria indicado pelo sistema. Esta inconsistência teria que ser detectada pelo facilitador ou outro colega do aprendiz.

Uma solução seria permitir a criação, pelo designer da atividade, de objetos mais específicos que já viessem com um pouco mais de significado embutido e verificassem, por exemplo, determinadas características materiais dos objetos (como peso, volume, resistência, etc.). Com isso, o software poderia detectar a ocorrência de situações insólitas e tomar alguma providência, como notificar o usuário. No caso da Oficina de Jogos, se possuísse personagens como árvores, pessoas, animais, etc. talvez o Cocoa pudesse fazer alguns tipos de inferências mais dirigidos às questões do Parque do Ibirapuera.

Em relação à ação da **comparação**, conforme descrito anteriormente, não havia nenhum mecanismo automatizado que facilitasse a identificação das diferenças ou similaridades entre o que havia sido idealizado e o que havia sido feito, entre as diferentes versões do trabalho de um mesmo aluno e entre trabalhos de diferentes alunos.

Também às vezes era muito complicado motivar os aprendizes a atualizarem a idealização que haviam feito no papel ao mesmo tempo em que estavam empolgados no computador. Uma possibilidade talvez fosse incrementar o Cocoa com alguma funcionalidade que incentivasse os aprendizes a descreverem melhor suas idéias antes de implementá-las e que produzisse relatórios, discriminando o nome do aluno, a data, as idéias desenvolvidas e uma descrição esquemática do estado corrente dos tabuleiros e dos personagens com suas respectivas propriedades. Este tipo de funcionalidade poderia ajudar tanto o aprendiz quanto o facilitador a compreender melhor o desenvolvimento dos projetos.

Depuração. Além de toda a parte gráfica e dos mecanismos de interação centrados no mouse, uma característica que se sobressai no Cocoa são as facilidades que

este software oferece para a depuração. Há botões que permitem controlar se a execução do programa deve ser interrompida, se deve prosseguir mais rápida ou mais lentamente ou, inclusive, se deve prosseguir ao contrário, retrocedendo no tempo todas as ações que já tenham sido executadas (também com as mesmas possibilidades de controle de velocidade) para que o aprendiz possa ver novamente o que já foi feito (ver Figura 32).

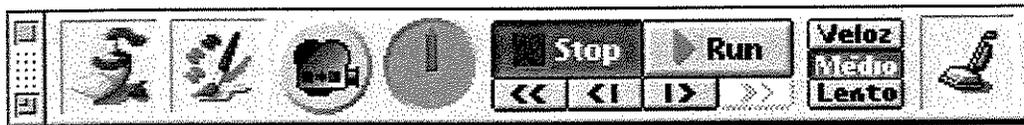


Figura 32 – Os botões que controlam a execução dos programas

Com 2 cliques do mouse, também é possível acompanhar o funcionamento das regras de qualquer personagem que se deseje e, com isso, analisar que regras foram ativadas em cada novo avanço do relógio. O usuário pode ainda determinar que certas regras não sejam executadas e, com isso, forçar situações específicas de teste.

O único porém de se acompanhar o funcionamento detalhado de vários personagens ao mesmo tempo era o de que o número de janelas abertas na tela podia atrapalhar a visão do que estava acontecendo no tabuleiro e ainda tornar o processamento do jogo mais lento. Para minimizar este problema, uma alternativa seria o Cocoa criar uma espécie de relatório histórico, descrevendo textualmente ou esquematicamente a seqüência das operações executadas pelo programa do aprendiz.

A relativa independência dos personagens dos mundos Cocoa e a facilidade de se poder dispô-los no tabuleiro com o mouse ajuda muito na criação de situações de teste. Por exemplo, o aprendiz poderia criar um cenário de teste colocando dois personagens um do lado do outro e, ativando o relógio, verificar se eles agiram conforme o esperado. Mais adiante, ele poderia ir, gradativamente, criando situações mais complexas, acrescentando novos personagens ou mudando a disposição no tabuleiro.

Nos casos específicos, no entanto, seria bom que o Cocoa permitisse gravar o estado inicial dos cenários para que o usuário pudesse voltar a eles quando sentisse necessidade.

Ainda que uma comparação direta talvez não seja pertinente, vale lembrar que, na maioria das variações do Logo, os programas criados pelos aprendizes apareciam, seqüencialmente, com formato texto, o que não tornava nada fácil isolar os componentes para gerar casos de teste.

Esta idéia de cenários também poderia ser utilizada por professores para incentivar os alunos a trabalhar com hipóteses. Aliás, o Cocoa em si poderia ser usado por alunos para construir e demonstrar suas próprias teorias.

No entanto, vale notar que, para que o tipo de análise isolada dos cenários possa funcionar, é preciso que todas as regras testadas estejam contidas nos

personagens que estiverem presentes no tabuleiro. Apenas eles é que serão executados quando o relógio começar a funcionar.

Outra característica do Cocoa é que as regras que afetam um personagem não precisam, necessariamente, estar contidas dentro do personagem, o que poderia gerar algumas confusões. Por exemplo, na Oficina, um dos jogos mostrava um personagem correndo em uma pista. As regras de movimentação do corredor podiam estar presentes tanto na pista, como no corredor. No caso, elas estavam uma parte em cada lugar e, quando o aprendiz foi alterar a forma da corrida, já não se lembrava onde procurar.

Para minimizar este problema, seria interessante contar com ferramentas que permitissem ao usuário fazer buscas por tipo de personagem (ou mesmo por aparências, nomes de variáveis, nomes de regras, comentários, etc.) que retornassem uma lista indicando todos os lugares aonde este personagem era referenciado. Uma ferramenta como esta não só facilitaria na compreensão do programa, como também ajudaria na hora em que se desejasse aproveitar um personagem em novos mundos do Cocoa.

Descrição. A distinção entre construção e descrição proposta no capítulo 7 se fez muito importante na análise do Cocoa. Enquanto neste software se mostrava muito fácil criar novas regras, a sua compreensão, ou a compreensão do conjunto como um todo, por alguém externo, ou mesmo pelo próprio criador depois de um tempo, tendia a ser complicada e exigia uma série de cuidados.

Os programas desenvolvidos foram, na nossa opinião, a principal fonte de descrição do trabalho dos alunos na Oficina de Jogos. Ainda que eles também escrevessem no diário-de-bordo, estes comentários não pareciam ter sido muito valorizados e nem sempre pareciam refletir o que acontecia ao longo da sessão com o software.

Embora o Cocoa também fosse uma linguagem de programação, o tipo de descrição resultante era muito diferente do Logo e chamou a atenção por diversos aspectos, dentre eles:

a) Processamento paralelo e programa distribuído entre os personagens.

Enquanto na maioria das versões do Logo os programas têm uma estrutura seqüencial única, organizada hierarquicamente em procedimentos com ordem preestabelecida de execução, os programas do Cocoa são organizados em regras. As regras se encontram distribuídas pelos diferentes tipos criados e são executadas uma de cada personagem de cada vez, passando a impressão de que os personagens são processados paralelamente.

Este tipo de organização do programa, ainda que muito diferenciado do Logo tradicional, não é exclusivo do Cocoa. Várias linguagens de programação dirigidas à educação, como o HyperStudio e o MicroMundos, utilizam estrutura similar e têm sofrido tanto críticas quanto elogios. As críticas se referem, em maioria, à dificuldade de se orientar no programa e compreender o seu funcionamento. Os elogios retratam a possibilidade de se criar, com maior facilidade, representações mais próximas da realidade, onde os acontecimentos não acontecem seqüencialmente.

Na nossa opinião, ambas as colocações também valem para o Cocoa. De qualquer forma, na Oficina de Jogos, o paralelismo do Cocoa possibilitou aos aprendizes trabalharem, em seus programas, com movimentos e combinações que, dificilmente, conseguiriam de outro jeito. Já a distribuição das regras foi um problema, principalmente nos programas maiores, onde se gastava muito tempo procurando-se uma determinada regra. Nestes casos, a inclusão de novas ferramentas de busca e de geração de relatórios, como as descritas acima, poderia ser de grande utilidade.

Conforme discutido no capítulo 7, o tipo de descrição construcionista deve variar conforme aquele que a for utilizar. A descrição para um aprendiz é diferente da descrição procurada por um facilitador. O tipo de descrição propiciada por um programa feito no Cocoa é mais dirigido para aprendizes. Já os facilitadores têm dificuldade de compreender o que foi feito e, assim, avaliar ou mesmo contribuir com o aprendizado.

b) A descrição das regras era essencialmente gráfica. Em alguns casos, também incluía comparações numéricas ou textuais. Opcionalmente, podiam receber um nome e comentários que facilitassem o seu entendimento.

No caso das regras simples, que só dependiam de uma condição espacial para serem ativadas, a descrição gráfica bastava e era logo compreendida. Já para as regras que precisavam verificar o estado de variáveis, onde as comparações matemáticas eram necessárias, a compreensão era um pouco mais complicada.

Por exemplo, a Figura 33 ilustra uma regra onde um personagem salta um obstáculo. Olhando-se só para a parte (a) da figura, não é possível saber se a regra verifica alguma variável antes de ser executada. No caso, como fica aparente na parte (b), é preciso clicar no botão "And check" para perceber que a regra só funciona se o valor da variável "Energia" for diferente de zero.

Na nossa opinião, o Cocoa deveria deixar mais explícitas as condições de disparo das regras. Da forma em que está, o usuário é obrigado a clicar toda vez no botão "And check" de cada uma das regras para compreender o funcionamento do programa.

Nestes casos, uma boa **descrição textual** no nome ou no comentário também ajudaria. No entanto, na Oficina de Jogos os aprendizes tinham uma tendência de não descrever o que suas regras faziam. Em alguns casos, quando escreviam, usavam nomes que não estavam nenhum pouco relacionados ao que a regra fazia. Um aluno, por exemplo, dava nomes de carros às regras que criava.

Para minimizar estes problemas, o facilitador optou por pedir para que eles dessem nomes a todas as regras e que explicassem, em voz alta, as operações que elas executavam. Na nossa opinião, este exercício de verbalização oral e escrita ajudou-os a entender melhor seus próprios programas e a detectar inconsistências em seu próprio raciocínio. No entanto, como discutido no capítulo 7, pareceu faltar uma cultura que incentivasse, de fato, os aprendizes a descreverem melhor seus programas. As facilidades de descrição textual do Cocoa não eram suficientes para assegurar isso.

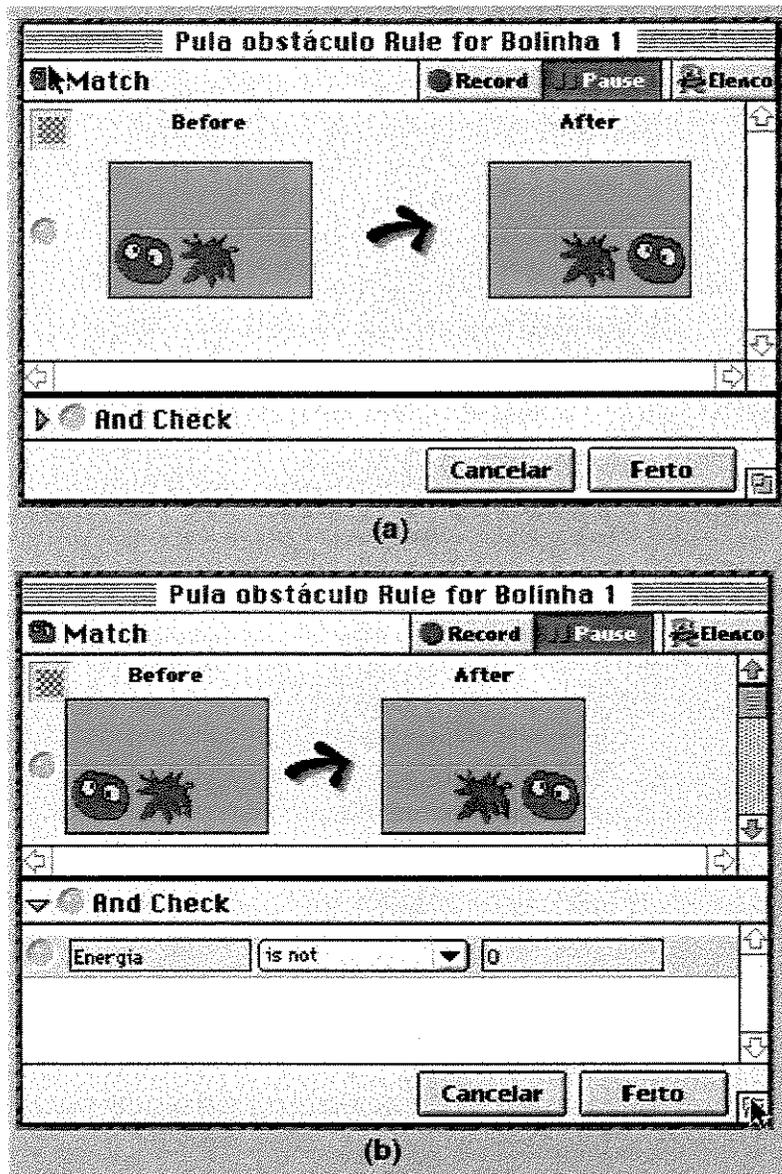


Figura 33 – Visualizando as condições de disparo de uma regra

É interessante notar que, apesar das regras aparentarem ser essencialmente gráficas, sob os aspectos mais avançados da programação do Cocoa, os nomes das variáveis, dos sons, das aparências e dos tipos dos personagens são mais importantes do que o conteúdo associado a eles. Quando, por exemplo, o Cocoa compara aparências, não são os pontinhos coloridos da imagem que estão sendo analisados, mas o nome das aparências e os tipos aos quais elas pertencem. Duas aparências que tenham a mesma figura (por exemplo, uma bolinha), se tiverem nomes ou tipos diferentes serão consideradas diferentes.

Na nossa opinião, dada sua importância para o processamento do programa, acreditamos que os nomes dos objetos do Cocoa deveriam ser mais evidenciados e esclarecidos. Por exemplo, a janela para criação de aparências já poderia ter um campo para o nome da aparência e indicar o tipo ao qual ela está associada. O

mesmo poderia aplicar-se para a criação de tipos e de sons. Outra solução seria eliminar esta relação de dependência entre aparências (e sons) e tipos e deixá-los em uma galeria à parte que fosse comum a qualquer tipo.

c) A descrição de um programa vai além de seus componentes básicos. Além das regras e dos personagens criados, os mundos do Cocoa são compostos por tabuleiros e pela distribuição dos personagens sobre eles. Não basta olhar as regras ou os personagens do Cocoa para se entender o que o jogo criado faz. É preciso ver quantos personagens de cada tipo existem, que tipo de situações o aprendiz criou, como ele organizou suas regras, etc.

Por exemplo, na nossa opinião, o modo com que os personagens foram distribuídos nos tabuleiros cria cenários que ajudam a compreender o contexto no qual se insere o programa e ajudam muito o facilitador a obter uma visão geral das coisas. De certa forma, eles complementam a descrição lógica e permitem uma abertura para fantasia e a exploração de outros estilos de aprendizado, tal como as descrições em formato de narrativa escrita mencionadas no trabalho de Kafai (ver capítulo 7).

Do mesmo modo, regras mais ou menos genéricas, conjuntos de regras, personagens com aparências específicas, comentários escritos, etc. também ajudam a perceber o grau de abstração do pensamento do participante. O uso de uma maior variedade de ferramentas também pode indicar o grau de domínio do software.

Ainda assim, é muito difícil compreender o que foi feito só pelo programa que os alunos escreveram no computador. Neste ponto, a descrição por escrito no planejamento é fundamental. Sem ela e sem conversar pessoalmente com os alunos, o facilitador não teria condições de entender o que foi feito, o que era desejado e o que de fato aconteceu durante a Oficina. Apenas parte das ações era dirigida para o Cocoa. Os alunos conversavam, pensavam, iam de um lado para outro, faziam várias tentativas, etc.

Além disso, é importante frisar que o Cocoa é um aplicativo muito novo e ainda não existem muitos estudos indicando que pontos são mais ou menos relevantes para se fazer uma análise dos trabalhos desenvolvidos pelos alunos. Na nossa opinião, estudos como estes seriam muito importantes para uma melhor adequação do software para atividades educacionais.

Por fim, conforme discutido no capítulo 7, o programa final, por si só, não traz as marcas das tentativas que foram arriscadas e depois abandonadas pelos aprendizes. Isso exigiria quase que uma filmagem de todos os passos seguidos por cada um, ou pelo menos um retrato sucinto e periódico do que estava acontecendo. Às vezes, para descobrir se um aprendiz estava tendo muitas dificuldades, bastava olhar a latinha de lixo do Cocoa e ver o que o aprendiz já havia jogado fora.

9.3.2.3 As operações

O tipo de interface do Cocoa, bem como a facilidade de se criar regras por meio de movimentos do mouse e demonstrações do que deveria ser feito, se mostrou bem amigável para os aprendizes e permitiu com que eles fizessem coisas bem

sofisticadas. Como os elementos que seriam utilizados nas regras podiam ser visualizados no tabuleiro e não havia muitas possibilidades de erro de sintaxe, rapidamente os aprendizes operacionalizavam as operações básicas do software. As maiores dificuldades surgiam na hora de combinar intencionalmente estas operações e utilizá-las para fazer algo que se desejasse.

Tal como discutido acima, o único registro que o Cocoa guardava do jogo desenvolvido era o programa em si. Não havia um histórico das operações efetuadas pelo usuário até que ele chegasse ao término do seu jogo.

Percebeu-se, de qualquer modo, por meio da observação, que os aprendizes passavam a maior parte do tempo criando ou modificando regras e aparências dos personagens. Recursos mais sofisticados, como grupos de regras, regras que fizessem comparação de variáveis específicas, comentários, etc. foram pouco utilizados.

9.3.2.4 Conceitualização e operacionalização

A seguir são discutidas as principais conceitualizações indesejadas, isto é, não relacionadas com “a criação de um jogo sobre o parque” percebidas no uso do computador na Oficina de Jogos.

No que se refere aos **aspectos físicos**, o que mais chamou a atenção dos aprendizes foi o mouse dos computadores usados. Diferentes dos mouses utilizados em computadores do tipo PC, que têm 2 ou 3 botões, os mouses dos computadores Apple têm apenas 1 botão. Fora isso, o teclado e a forma do computador em si não pareceram trazer nenhuma distração para a atividade.

Já nos **aspectos de manuseio**, foram detectados os seguintes pontos:

a) Diferenças entre sistemas operacionais. O sistema operacional do Macintosh, embora fosse gráfico e trabalhasse com janelas, não era igual ao sistema com que os aprendizes estavam familiarizados, o MS-Windows. A maior parte dos aprendizes teve que aprender comandos básicos (ou operacionalizar as ações dirigidas ao artefato) como movimentar janelas, passar de um aplicativo para outro, alterar o tamanho das janelas, etc.. Idealmente, sob o ponto de vista da atividade desejada, o usuário não deveria preocupar-se com questões deste tipo.

b) Operações com arquivos. Como ocorre com a maior parte dos aplicativos, no Cocoa os aprendizes também tinham que se preocupar em salvar seus mundos em arquivos para não perderem o que haviam feito e poderem continuar seu trabalho na sessão seguinte. Embora este tipo de procedimento não fosse anormal para eles, sob o ponto de vista do desenvolvimento de jogos, operações com arquivos não precisariam existir e poderiam ser automatizadas pelo próprio Cocoa.

c) Tempo de salvamento de arquivos. Além de terem que salvar os arquivos, o tempo de salvamento na rede às vezes era muito lento e chegava a incomodar. Este ponto ajuda a reforçar a idéia de que, no desenvolvimento de um software, é muito importante considerar o artefato computacional como um todo no qual ele se insere. Talvez o Cocoa funcionasse muito bem se estivesse em um computador isolado. No

entanto, como estava em um computador de rede, o seu funcionamento acabou sendo parcialmente comprometido.

d) Inserção de novas regras em um conjuntos de regras. No Cocoa era possível criar conjuntos que agrupassem regras e, assim, conseguir executar mais do que uma única regra em um mesmo tique do relógio. Para isso, o aprendiz criava o conjunto e ia arrastando com o mouse as regras que lhe interessassem para dentro do conjunto, na posição de execução que fosse mais conveniente. Infelizmente, os aprendizes tinham muita dificuldade de efetuar este procedimento. Eles tinham que fazer várias tentativas até conseguir inserir a nova regra na sua posição correta.

e) O lixo e a velocidade de processamento. Para apagar uma regra, uma propriedade ou um personagem, bastava o usuário selecionar o objeto com o mouse e arrastá-lo para dentro do aspirador que aparecia desenhado na tela. Com dois cliques sobre o aspirador era possível visualizar o que havia sido jogado fora e, se necessário, trazer de volta para o tabuleiro. No entanto, à medida em que o aspirador ia ficando cheia, o jogo ia ficando mais lento e passava a incomodar os alunos. Para completar a situação, na versão testada, não parecia haver nenhuma forma de se esvaziar o aspirador para acelerar o processamento do jogo.

Por outro lado, havia no Cocoa uma série de elementos que contribuíam para minimizar os aspectos de manuseio. Destaca-se, por exemplo, o fato de que o software praticamente não deixava com que o usuário cometesse erros de sintaxe na escrita dos programas. Todos os comandos eram criados a partir da combinação de operações preexistentes selecionadas com o mouse. Os poucos erros que poderiam acontecer estariam relacionados à escrita errada de nomes de variáveis, tipos e aparências. Mesmo estes poderiam ser minimizados caso estes nomes também pudessem ser selecionados dentre os já criados pelo usuário.

f) Suporte aos aprendizes. Para os casos imprevistos, em que o usuário tivesse cometido um engano ou precisasse obter alguma informação específica sobre os elementos do Cocoa, não havia muitos recursos de suporte, ainda mais em português. Nestes casos, o maior auxílio vinha diretamente do professor que, atuando como fonte de referência, por vezes viu-se sobrecarregado e não conseguia atender a todos os participantes de modo adequado.

g) Problemas de programação X exploração do domínio. No desenvolvimento de seus projetos, os aprendizes esbarravam em uma série de problemas lógicos, inerentes às facilidades de programação disponíveis, que prendiam bastante a sua atenção e acabavam concorrendo com o principal objetivo pedagógico da atividade, que era discutir o Parque do Ibirapuera.

Um destes problemas aparecia quando se desejava fazer com que um personagem andasse até um obstáculo e voltasse andando no sentido contrário. A solução mais direta era fazer uma regra bem específica que executasse todas estas operações de uma só vez. Porém, ela apresentava algumas deficiências: além de só funcionar nesta situação, todas as operações acabavam sendo executadas em um único tique do relógio enquanto todos os demais personagens do mundo ficavam esperando.

Outra solução, mais genérica, seria fazer uma regra para cada tipo de movimento: uma que fizesse o personagem andar no sentido do obstáculo, outra que percebesse o obstáculo e virasse o personagem e outra que o fizesse andar no sentido contrário. Diferente da anterior, esta poderia ser aplicada para qualquer distância em que se encontrasse o personagem do obstáculo. O ponto é que, com o personagem distante do obstáculo, como é que o computador saberia se o personagem estava indo para a direita ou para a esquerda? Se as regras são testadas seqüencialmente, sempre seria executada aquela que viesse antes, independente do movimento.

Para resolver esta questão, o aprendiz deveria criar algum mecanismo que indicasse o sentido do movimento. Isso poderia ser feito por meio de 2 aparências distintas para o personagem – uma mostrando ele indo para um lado, outra indicando que ele estava seguindo para o outro – ou por meio da criação de uma variável específica para este fim.

Qualquer que fosse a alternativa seguida, toda a concentração do aprendiz estaria dirigida a estratégias de programação em si, o que seria perfeito se o objetivo principal da atividade fosse aprender a programar no Cocoa. No entanto, como a Oficina de Jogos tinha um interesse maior em usar o computador como um pretexto para uma discussão sobre pontos relacionados ao Ibirapuera, o esforço gasto em problemas como este deveria ser minimizado.

O ideal seria operacionalizar todas estas questões estratégicas que não estivessem relacionadas com a intenção da atividade. Isso poderia ser feito pela implementação de novos comandos no Cocoa ou através de um treinamento mais intenso com os participantes.

Vale lembrar que, trabalhando com o Logo Gráfico, volta e meia o aprendiz também se depara com questões que, aparentemente, não sabe resolver utilizando apenas as operações básicas da linguagem. O exemplo mais comum é fazer a tartaruga desenhar um círculo na tela, usando apenas os comandos que movimentam a tartaruga para frente e a rotacionam para o lado. A descoberta da solução final, de que um círculo pode ser entendido como um polígono regular de muitos lados bem pequeninos, exige um grande esforço de programação, mas tudo dentro do domínio da geometria. Talvez, se as operações do Cocoa fossem mais específicas para temas relacionados ao Parque, o esforço de programação estaria mais próximo de questões relacionadas ao tema da Oficina.

Felizmente, na documentação do Cocoa em inglês havia uma série de exemplos de desafios de programação tais como os descritos acima. Com base neles, a abordagem adotada na Oficina foi, logo de início, mostrar aos aprendizes como resolver os problemas mais comuns. A filosofia adotada era a de que os aprendizes deveriam focar o máximo possível no domínio educacional da atividade. Quando a ferramenta não permitisse isso de modo adequado, o professor, atuando como fonte de referência, poderia ajudar a minimizar a dificuldade. Neste caso, sua interação complementar a interface do software, reduzindo suas conceitualizações de programação e tornando-o mais dirigido ao tema desejado.

9.3.3 Orientação a objetos

Motivação. Além de um tema que estivesse relacionado com o dia-a-dia dos alunos e a liberdade dada para o desenvolvimento dos projetos, na nossa opinião, o artefato computacional em si também foi um elemento que contribuiu para a motivação dos participantes da Oficina de Jogos. Sem dúvida, o simples fato de se trabalhar com computadores Apple (os alunos nunca haviam lidado com computadores que não fossem PC) e ainda ter a oportunidade de mexer com um software inovador e atraente como o Cocoa já garantiriam a participação em atividades como a Oficina por um bom tempo.

Além do fato de possibilitar a criação de jogos – algo que interessa aos jovens – um aspecto do Cocoa que contribuiu para a motivação foi a possibilidade de se trabalhar, nos personagens criados, com sons e imagens produzidos pelos próprios aprendizes. Isso parecia mexer muito com a auto-estima deles e fazia o clima da atividade ficar mais descontraído.

Em especial, a possibilidade de trabalhar com som pareceu ótima para integrar a turma e diminuir a inibição. Provavelmente, a maioria dos aprendizes pouco teve oportunidade de ouvir sua própria voz. Por exemplo, quando M.S.S.P. foi produzir as mensagens educacionais que os personagens de sua animação iriam falar, a maior dificuldade foi ela conseguir parar de rir e segurar a agitação do pessoal no momento da gravação. No final da atividade, a turma já estava bem entrosada e um pedia para os demais ficarem calados enquanto o outro fazia gravações.

Na nossa opinião, o trabalho com imagens poderia produzir um efeito parecido. Para isso, a disponibilidade de uma câmara fotográfica digital seria imprescindível.

Outro aspecto interessante do Cocoa era a possibilidade de controlar um dos personagens do tabuleiro com o auxílio do teclado. Dependendo de como programasse, o aprendiz podia, por exemplo, dirigir a movimentação do personagem usando as setas, mudar sua aparência usando a tecla espaço, etc.

Com isso, o aprendiz tinha a oportunidade de participar em primeira pessoa do jogo e interagir mais diretamente com os personagens e situações que ele mesmo havia criado. No jargão do Construcionismo, esta funcionalidade contribuiu para ampliar a “sintonicidade com o ego” na atividade.

No que se refere aos **objetos manipulados**, tal como descrito anteriormente, na Oficina de Jogos os participantes estavam quase que totalmente centrados no Cocoa. Com este fim, ele disponibilizava um arsenal de artefatos que permitiam criar uma infinidade de personagens e cenários, determinar suas aparências, definir como eles deveriam funcionar, etc.

Uma das características principais do Cocoa é que tanto estes artefatos quanto os objetos digitais por eles criados tinham uma representação gráfica e formas de manipulação, via mouse, que os tornavam, praticamente, concretos para o sujeito. Depois de certo tempo, o sujeito conseguia abstrair do computador e se concentrar mais nos elementos que fariam parte de seu novo mundo.

Falta de letras e números. No que se refere ao manuseio dos objetos digitais do Cocoa, os participantes sentiram falta de objetos como letras e números que pudessem ser utilizados em seus jogos para criar placares, títulos, etc.. R.N.M. até que conseguiu uma solução bem interessante, desenhando as letras cursivamente com o mouse (ver Figura 34). Para resolver este problema, seria muito interessante que o Cocoa já trouxesse objetos prontos com esta finalidade.

Falta de objetos transicionais específicos. Na nossa opinião, uma das maiores discrepâncias do Cocoa com relação ao Logo, foi o fato de ele não trabalhar com objetos transicionais dirigidos a algum domínio específico. Os objetos do Cocoa, com sua aparência de “massa de modelar digital”, são realmente familiares para o sujeito mas, conforme discutido na seção anterior, não parecem ser representativos do domínio explorado na Oficina de Jogos. Na realidade, o Cocoa não foi feito especialmente para a Oficina de Jogos, nem para atividades construcionistas em geral, mas sim como algo mais genérico.

Tal como percebido após a Oficina, se a intenção de uma atividade educacional fosse trabalhar com alguma temática específica e também reduzir a carga técnica imposta aos alunos, o professor deveria propor algo mais fechado e disponibilizar alguns personagens e cenários mais relacionados com este tema para os aprendizes trabalharem. Esta idéia está de acordo com o conceito de micromundos descrito no capítulo 5.

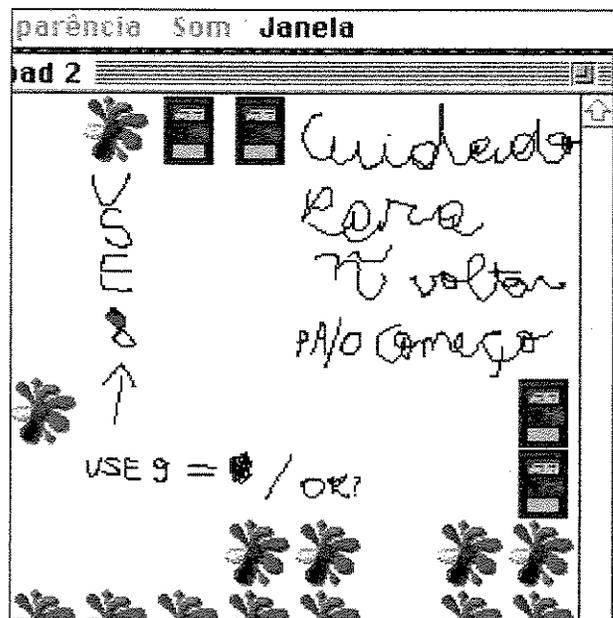


Figura 34 – Detalhe de um jogo onde o aprendiz inseriu um texto escrito à mão no tabuleiro do Cocoa

No caso, estes objetos mais específicos poderiam ser pré-construídos usando o próprio Cocoa. Para isso, no entanto, este software deveria oferecer alguns recursos específicos para designers de atividades educacionais como, por exemplo, mecanismos que restringissem as modificações, pelos aprendizes, nos objetos criados pelo professor, formas de se catalogar, guardar e intercambiar objetos criados, etc.

Vale notar que, ainda que aparentemente contradizendo o que foi dito acima, apesar da generalidade dos objetos do Cocoa e da temática relativamente aberta da Oficina, os jogos e animações criados foram, em sua maioria, intrinsecamente motivantes. Como foi discutido na seção 9.2.2, questões relacionadas com reciclagem de lixo, segurança, esporte e o Instituto do III Millennium eram centrais e apareciam de modo indissociável nas produções dos aprendizes.

Em contrapartida, na pesquisa de Kafai (1995), onde os aprendizes trabalhavam com o Logo, os jogos criados foram, em sua maioria, extrinsecamente motivantes. Na nossa opinião, além da influência do tema (ver discussão no capítulo 7), o fato de o Cocoa praticamente não trabalhar com textos, dificultou a criação de jogos do tipo pergunta e resposta, que acabavam motivando, de certa forma, estruturas mais descontextualizadas.

De qualquer modo, acreditamos que, se os objetos do Cocoa fossem mais dirigidos ao Parque, a Oficina poderia ter transcorrido de uma forma completamente diferente.

No que se refere ao **contato de um participante com os demais**, a infra-estrutura tecnológica da Oficina pouco tinha a oferecer para a troca de idéias ou trabalhos entre os participantes da mesma turma ou de turmas diferentes. Embora fosse permitido e possível, através da rede, copiar mundos de um participante da Oficina para outro, o Cocoa não permitia ao aprendiz ter 2 mundos abertos ao mesmo tempo – nem que um deles fosse só para leitura – e nem tão pouco copiar objetos de um mundo para outro. A única maneira de um aprendiz reaproveitar um personagem criado por outro aprendiz era refazer este personagem com todos os detalhes em seu próprio mundo.

Como um artefato computacional, o Cocoa também trazia uma série de **facilidades para que o usuário criasse novas operações no software** a partir das já implementadas. Estas facilidades serão discutidas na seção sobre mediação.

9.3.4 Internalização e externalização

9.3.4.1 Internalização

Conforme discutido na seção anterior, o computador não teve uma participação ativa na comunicação entre os participantes da Oficina de Jogos, isto é, a interação entre os participantes era feita diretamente entre eles. Eles não compartilharam informações através da rede ou de algum mecanismo que facilitasse a troca de mensagens. Aparentemente, as práticas mais internalizadas foram as incentivadas pelo professor, as descobertas observando os trabalhos dos colegas e as encontradas nos poucos exemplos que já estavam disponíveis.

No entanto, ainda que o software não facilitasse muito a interação social, de uma forma mais sutil, as próprias facilidades de programação propiciadas pelo Cocoa incentivavam determinadas formas de ação e, conseqüentemente, de raciocínio por parte dos aprendizes. Embora já existam alguns estudos analisando as formas de raciocínio associadas a diferentes paradigmas de programação (BARANAUSKAS,

1993), no caso do Cocoa, pesquisas com este objetivo ainda têm que ser desenvolvidas.

Tal como percebido na Oficina, o Cocoa parecia incentivar um tipo de raciocínio dirigido ao processamento paralelo de operações. Um raciocínio onde cada objeto tem vida própria e interage com outros em situações nem sempre determinísticas.

Além disso, dadas as facilidades para criação, alteração e visualização da execução dos programas, percebeu-se também que os aprendizes seguiram um estilo de programação com pouco planejamento antecipado e muita construção por tentativa e erro. Neste sentido, ainda que fosse possível escrever todo o programa antes de processá-lo, como fariam, provavelmente, os aprendizes mais “planejadores”, o software pareceu incentivar mais aqueles de estilo mais “escultor”, que iam decidindo o que fazer à medida em que iam criando seus mundos. Talvez, se a Oficina fosse mais longa, os aprendizes sentissem necessidade de planejar um pouco melhor o que gostariam de implementar.

9.3.4.2 Externalização

Tal como discutido anteriormente, os personagens, as regras e os demais objetos criados com auxílio do Cocoa representaram uma das fontes de externalização mais importantes da atividade. Com base nas regras que construíam, os aprendizes podiam visualizar o que haviam feito, refletir sobre o que poderia ser aprimorado e implementar uma nova versão de seus programas.

Para os facilitadores, no entanto, a análise do programa construído dava subsídios, mas não era suficiente à compreensão total do pensamento dos aprendizes. Era necessário ter a companhia do diário-de-bordo e, ainda assim, estar presente no dia-a-dia da atividade. Talvez, no futuro, com a produção de uma maior quantidade de estudos baseados no Cocoa, seja possível fundamentar conclusões mais completas sobre o aprendizado a partir da análise dos programas desenvolvidos.

Uma funcionalidade do Cocoa que não foi aproveitada na Oficina de Jogos por falta de tempo e disponibilidade de linhas telefônicas era a publicação, na Internet, dos mundos criados pelos aprendizes. Na nossa opinião, esta divulgação dos trabalhos desenvolvidos, aliada a uma catalogação do que já estivesse exposto, incentivaria um intercâmbio maior entre os usuários e, conseqüentemente, o aprendizado de novas técnicas e usos do software.

9.3.5 História e desenvolvimento

9.3.5.1 História

Por que computadores Apple? Conforme já foi discutido acima, um dos principais complicadores para o desenvolvimento da Oficina de Jogos foi obter computadores Apple. Na realidade, esta escolha só ocorreu porque, na época, o Cocoa só funcionava em computadores deste tipo. Não é de nosso interesse, neste trabalho, discutir as vantagens e desvantagens de cada computador.

Originalmente, o Cocoa nasceu de um software chamado KidSim, desenvolvido no centro de pesquisas da Apple. Nos Estados Unidos, Japão e Europa os computadores Apple são bastante utilizados com finalidade educacional. No Brasil, por outro lado, esta plataforma ainda é pouco abrangente. Daí a dificuldade em se conseguir as máquinas. De qualquer modo, como descrito acima, a versão atual do Cocoa, lançada sob o nome de StageCast Creator, já está sendo comercializada tanto para computadores Apple como para PCs.

Por que o Cocoa? A própria seleção do Cocoa também sofreu por razões históricas. Dentre os vários aplicativos de programação dirigidos para o público leigo (usários não-profissionais de programação) sendo analisados, ele se destacou pela aparente facilidade de uso e também porque não exigiria muito esforço de tradução. Outros aplicativos candidatos foram o AgentSheets (<http://www.agentsheets.com/>) e o ToonTalk (<http://www.toontalk.com/>), mas não seria possível deixá-los operacionais em tempo para a Oficina.

9.3.5.2 Desenvolvimento

Instrumentalização e fluência. Acima, na análise da atividade educacional, discutiu-se que a Oficina havia sido dividida em 2 fases. A primeira, de instrumentalização, focada, principalmente, na operacionalização das ações dirigidas ao Cocoa. A segunda, de fluência, centrada no desenvolvimento dos jogos.

Para facilitar o processo de instrumentalização, foi necessário a disponibilização de uma série de exemplos, orientações e suporte técnico contínuo aos alunos. O Cocoa em si, não oferecia nenhuma funcionalidade específica para facilitar este trabalho. O seu site, no entanto, trazia tutoriais e exemplos já prontos que poderiam ser utilizados por quem desejasse.

Infelizmente, devido a restrições de tempo, apenas parte deste material pôde ser traduzido para a Oficina. Esta falta acabou sobrecarregando o professor e forçando uma dinâmica mais centralizadora da turma. Acreditamos que a existência deste material de suporte traria maior flexibilidade para a Oficina e, provavelmente, possibilitaria que cada aluno tivesse uma fase de instrumentalização mais de acordo com o seu ritmo próprio, não algo amarrado aos demais alunos de sua turma. Com isso, as fases de instrumentalização e fluência teriam um caráter mais pessoal do que grupal.

Por isso, na nossa opinião, orientações sobre o uso do aplicativo, descrevendo como dar os primeiros passos e mostrando os principais tipos de atividades para as quais ele foi criado, deveriam fazer parte obrigatória do desenvolvimento de software para a educação.

Na fase de fluência, a demanda dos alunos mudou. Eles ainda poderiam levantar questões esporádicas sobre o software, mas a maior parte das dúvidas era voltada para estratégias de implementação mais avançadas. Nesta fase, exemplos mais avançados e trocas com os colegas de turma eram muito importantes.

O professor passou a ter um papel maior como facilitador e o tipo de demanda que exigia do computador também se alterou. Na fluência, ele passou a acompanhar

mais o desenvolvimento dos projetos dos alunos, dando dicas e incentivando a continuidade do trabalho. Para isso, conforme já discutido anteriormente, seriam necessárias ferramentas que ajudassem a levantar os pontos relevantes de cada trabalho, documentar a sua evolução e facilitassem anotações e registros diversos.

Na Oficina, boa parte deste trabalho do facilitador teve que ser feito a mão, no papel. Talvez esta fosse uma boa funcionalidade a ser incorporada em versões futuras do Cocoa.

Estilos. Conforme discutido na seção sobre internalização, no nosso entender, o Cocoa pareceu ser uma ferramenta na qual os aprendizes de estilo “escultor” se sentiam mais a vontade do que no Logo. No Cocoa, estes aprendizes podiam, facilmente, a trabalhar a estética do seu mundo, criar novos personagens, testar uma nova combinação de operações, voltar a criar, testar novamente, etc.

Pelo menos durante o curto período da Oficina, o Cocoa não pareceu ser uma ferramenta onde os elementos usados na construção fossem mais difíceis de manusear e, portanto, exigisse um maior planejamento à priori.

Por outro lado, o Cocoa também não oferecia restrições para os aprendizes mais planejadores. Tão pouco ele oferecia facilidades específicas para este público. Tal como na Oficina, o projeto teria que ser feito no papel mesmo.

Backup. Por questões de segurança e para possibilitar uma futura análise do desenvolvimento da Oficina, todos os dias eram feitas cópias do material desenvolvido no computador pelos alunos. Estas cópias eram feitas através da rede pelo próprio professor e não havia mecanismos que automatizassem o processo ou mesmo que ajudassem a analisar tudo o que foi coletado.

Faltam ferramentas para análise pedagógica e para segurança do material produzido. É interessante notar que tanto a análise como a proteção do material dos alunos são ações comuns que se aplicam a qualquer atividade educacional. Ainda assim, há uma aparente escassez de mecanismos que facilitem a automação deste processo. Na nossa opinião, este é um tipo de observação que só pôde ser levantada a partir da análise da atividade educacional e do artefato computacional como um todo. Outras abordagens provavelmente teriam uma visão mais isolada de cada componente e não ressaltariam a integração entre eles.

9.3.6 Mediação

Conforme discutido ao longo de todo este capítulo, realmente o artefato computacional não foi uma ferramenta que passou incólume pela Oficina de Jogos e nem era esperado que isto acontecesse. O Cocoa e a demais ferramentas utilizadas realmente ajudaram os aprendizes a fazerem programas interessantes. Idealmente, porém, havia uma expectativa de que as conceitualizações geradas pelo artefato atraíssem a atenção do aprendiz, principalmente para alguma característica do domínio sendo discutido, não tanto para o próprio artefato. Infelizmente, conforme discutido anteriormente, a fase de instrumentalização acabou por tomar um tempo da atividade muito maior do que o esperado.

De uma forma geral, criar novos personagens, modificar aparências e produzir movimentos simples pode ser facilmente aprendido no Cocoa. Com poucos minutos de prática, os alunos já estavam aptos a criar personagens que pulavam obstáculos ou emitiam sons em determinadas situações. Nos termos da Teoria da Atividade, estas ações eram rapidamente operacionalizadas pelo aprendiz e facilmente se constituíam em órgãos funcionais. Fazer combinações mais avançadas de regras e forçar seqüências de processamento exigia o conceito de variáveis, que era um pouco mais complexo.

Já projetos grandes, com muitos personagens e regras, eram bem mais difíceis de serem implementados. O aplicativo oferecia poucos recursos para navegação e localização de partes específicas do programa. Além disso, ele também não oferecia nenhuma funcionalidade especial para a combinação e o reaproveitamento das novas operações criadas pelo usuário.

De fato, no correr da Oficina, sentiu-se falta de alguma facilidade do software que ajudasse no reaproveitamento das regras construídas. Se, por exemplo, em determinadas situações (como a morte de um monstinho no jogo do Pac Man), o personagem precisasse executar uma seqüência fechada de operações (como, no caso, mudar de aparência, emitir um som diferente, ganhar pontos e mudar de nível), todas as regras onde isso pudesse acontecer deveriam ter uma cópia da mesma seqüência. Esta seqüência, por si só, a não ser por meio de mecanismos mais complexos de programação, não poderia ser tratada como uma nova operação pelo sujeito.

Sob estes aspectos, o Cocoa pareceu especialmente indicado para projetos pequenos e médios.

Além disso, assim como acontece com a maioria dos aplicativos de programação para usuários "leigos", percebeu-se que as possibilidades de programação do Cocoa se restringiam única e exclusivamente às operações dirigidas aos personagens criados. Não era permitido ao usuário criar novas operações a partir de operações de manuseio como as de salvar ou fechar arquivos, fazer cópias e alterações de regras, etc.

Por um lado, percebe-se que o verdadeiro foco do software era justamente a programação dos personagens. Por outro, entendendo-se a atividade do Cocoa como uma atividade de projeto que deverá se repetir várias vezes, talvez fosse interessante, para os usuários mais avançados, que o próprio Cocoa pudesse ser programado e alterado conforme a necessidade.

9.4 Reflexões gerais sobre a Oficina

A seguir, são apresentadas as principais conclusões tiradas da execução da Oficina de Jogos como um todo. Ao final, são destacadas as principais conclusões relacionadas ao uso do computador nesta atividade.

9.4.1 Sobre o planejamento e a execução da Oficina

Em primeiro lugar, deve ser notado que o tipo de proposta da Oficina de Jogos, como uma atividade construcionista, incentivava um relacionamento mais pessoal entre os participantes, não algo em massa como o que se presencia nas escolas tradicionais. Pôde-se perceber que cada aluno trazia uma experiência própria de vida, um estilo e um ritmo particular de aprendizado, um interesse e um grau de motivação específicos.

Estas características transpareciam o tempo todo e o desafio de lidar com elas esteve presente em todos os aspectos da atividade. Por exemplo, de início, nenhum dos alunos estava familiarizado nem com a proposta da atividade, nem com as ferramentas que seriam utilizadas. A Oficina de Jogos ressaltou a importância de se ter uma “fase de instrumentalização”, focada na própria atividade, antes que os alunos possam sair produzindo de fato, que é o que se faz na “fase de fluência”.

Cada uma destas fases impunha uma dinâmica e uma demanda específica que tinham que ser cuidadosamente planejadas. Na instrumentalização, os alunos eram levados a conhecer a atividade e o software a partir de exemplos já prontos e explicações do professor. Na fluência, eles já caminhavam mais por conta própria e o foco eram os seus projetos pessoais. As dúvidas eram mais específicas e o professor atuava mais como um orientador dos trabalhos.

Sob o ponto de vista construcionista, a fase de instrumentalização pode ser considerada como um patamar necessário a ser escalado para que a atividade se concentre em questões mais educacionalmente interessantes. Felizmente, à medida em que as ferramentas e a proposta da atividade se tornem mais familiares, a fase de instrumentalização tende a diminuir cada vez mais.

É interessante notar que, apesar de ser algo comum a todas as atividades, a maior parte da literatura construcionista deixa a instrumentalização praticamente de lado e descreve as atividades já na fase de fluência.

Em segundo lugar, apesar de a Oficina de Jogos ter sido uma atividade essencialmente centrada no computador, o papel do facilitador humano foi algo que acabou se sobressaindo. Conforme discutido no capítulo 9, apesar de toda a estrutura armada, era ele quem cuidava do encaminhamento diário da atividade, resolvendo todos os imprevistos e deficiências que surgissem. Além disso, ele também demonstrou um papel fundamental de orientação e incentivo aos alunos na busca por novas soluções e na superação de seus limites.

Mesmo com toda esta sua importância para o desenvolvimento dos alunos, na Oficina de Jogos percebeu-se como praticamente não existem ferramentas especialmente feitas para ajudar o facilitador a executar o seu trabalho. À medida em que a atividade ia se desenrolando, os alunos iam desenvolvendo projetos cada vez mais complexos e diversificados. Depois de um tempo, mesmo com apenas 5 alunos por sessão, tornava-se difícil acompanhar o que cada um estava fazendo e ainda oferecer alguma contribuição. Como será discutido mais adiante, a incorporação de ferramentas específicas para o facilitador talvez ajudassem muito para a resolução deste problema.

Em terceiro lugar, confirmando a idéia do ciclo de vida apresentada no capítulo 6, a Oficina de Jogos ajudou a perceber o quanto a atividade educacional é algo dinâmico que exige um olhar crítico e um remodelamento constantes. Por mais que se planeje, sempre haverá pontos que não foram previstos, ou que precisariam ser melhorados. Por exemplo, não se imaginava, na Oficina, que a fase de instrumentalização fosse exigir tanto tempo, que os participantes fossem se desenvolver de forma tão diversificada, ou que mais discussões entre os alunos deveriam ser incentivadas. Estes problemas só foram notados durante a própria Oficina e, na medida do possível, deveriam ser resolvidos em uma próxima vez que ela fosse efetuada.

Para facilitar esta reflexão e aprimoramento sobre a própria atividade, as anotações efetuadas pelo professor antes, durante e após as sessões foram fundamentais. Em especial, o tempo de 20' deixado entre as turmas se mostrou bastante adequado para o replanejamento constante que era exigido. Durante as sessões, pelo menos na fase de instrumentalização, era mais difícil escrever e prestar atenção no que estava sendo feito. Aqui também ferramentas específicas seriam muito bem-vindas.

Outro elemento que também contribuiria para a reflexão seria a possibilidade de o professor trocar idéias com outros professores a respeito do que foi feito. Isso ocorreu de forma esporádica e não planejada na Oficina e, ainda assim, foi muito recompensador. Vale notar que, nas escolas tradicionais, normalmente o professor também acaba como responsável pelos mesmo papéis e trabalha a maior parte do tempo sozinho. Valeria a pena estudar como seria um tipo de escola onde os professores trabalhassem mais próximos uns dos outros⁵⁰.

9.4.2 Sobre o uso do computador na Oficina de Jogos

Embora o computador seja um elemento dentre os vários que compõem a atividade, é muito importante que ele esteja bem integrado ao resto do contexto para não se transformar em um obstáculo. Na Oficina de Jogos, o artefato computacional composto pelo software Cocoa, os computadores Macintosh e os demais componentes da rede se mostraram adequados o suficiente para que o trabalho fosse desenvolvido com êxito. Os aspectos que chamaram mais a atenção serão discutidos a seguir.

Vale frisar que o tipo de análise proposto pela Teoria da Atividade, de se considerar o artefato computacional como um todo, sem separar hardware de software, ajudou a perceber diversos pontos levantados na Oficina de Jogos. Dentre estes pontos, destacam-se as dificuldades que os alunos tiveram para lidar com o mouse e o sistema operacional da Apple (eles estavam familiarizados com o de computadores da linha PC) e as reclamações que vieram da demora em se conseguir salvar um

⁵⁰ Na Open Charter School, escola pública de Los Angeles, as salas de aula são agrupadas duas a duas, sem divisão entre elas. Os professores das salas agrupadas trabalham com crianças de um mesmo ciclo (2 anos) e têm a liberdade de, quando julgarem necessário, combinarem as turmas e desenvolverem trabalhos em conjunto.

trabalho (devido a rede). Nenhum destes pontos teria sido levantado na análise isolada de qualquer software utilizado.

De qualquer modo, o principal software utilizado na Oficina foi o Cocoa, que trazia uma série de funcionalidades para a criação de jogos e simulações. Os demais aplicativos da atividade só estavam presentes para complementá-lo, daí não terem sido tão enfocados na análise efetuada.

Na realidade, o Cocoa não foi criado com a atividade educacional construcionista em mente. Seu usuário é visto como um “construtor de mundos”, não como um aprendiz ou facilitador. Ainda assim, sua análise ajudou a perceber muitos aspectos interessantes para o desenvolvimento de software para atividades construcionistas. Por exemplo, no Cocoa, era muito fácil construir personagens com aparências gráficas e sons, dar-lhes movimento e testá-los em situações específicas.

Neste processo, os alunos praticamente não tinham que escrever nada, bastava arrastar e apertar o mouse. No entanto, à medida em que os jogos fossem ficando um pouco mais complexos, não era fácil entender o que cada personagem fazia e tanto aprendizes quanto facilitadores tinham dificuldade para identificar os problemas que apareciam.

Esta dificuldade ajudou a reforçar a idéia proposta no ciclo das ações construcionistas (ver capítulo 7) de que, mesmo na programação, a ação da construção deve ser tratada de forma independente da descrição. No caso do Cocoa, seria interessante que o aplicativo incorporasse ferramentas que ajudassem o aprendiz a localizar-se e o incentivassem a fazer descrições textuais sobre o seu projeto.

Outra característica que chamava a atenção no Cocoa era o fato de, por ser um software de caráter genérico, seus objetos (os tabuleiros e personagens) não estarem associados a nenhum domínio específico. Nem tão pouco os seus comandos foram criados de forma a enfatizar determinados tipos de conhecimento, como acontece no Logo.

Isso, por um lado, trazia uma grande flexibilidade de utilização ao software. Por outro, exigia um esforço maior por parte do facilitador e do designer da atividade para que o tema desejado fosse explorado pelos aprendizes. No caso da Oficina de Jogos, por exemplo, cujo objetivo era trabalhar como tema o Parque do Ibirapuera, sentiu-se falta de objetos que estivessem mais relacionados ao próprio Parque, como árvores, animais, etc.

O manuseio de objetos como estes poderia diminuir um pouco da complexidade lógica específica da programação e incentivaria o aprendiz a refletir sobre os aspectos planejados para a atividade. Por isso, na nossa opinião, a possibilidade de criar, incluir ou mesmo alterar objetos de domínios específicos talvez fosse uma característica que pudesse ser melhor trabalhada em versões futuras do Cocoa.

Suporte aos facilitadores. Por fim, saindo do caso específico do Cocoa e retomando à questão apresentada na seção anterior, a Oficina também deixou evidente a falta de ferramentas para o suporte dos facilitadores da atividade.

Todo o trabalho de backup do que havia sido feito pelos alunos tinha que ser feito manualmente, todos os dias, pelo professor. Também não havia ferramentas que o ajudassem a fazer anotações e nem a analisar os aspectos mais importantes do trabalho que seus alunos faziam.

Na realidade, no planejamento da Oficina de Jogos não foram definidos quais seriam os critérios de avaliação dos trabalhos dos alunos. E mesmo no que se refere ao Cocoa, o software era tão novo que ainda não havia estudos ressaltando seu uso educacional e os pontos que deveriam ser considerados.

De qualquer modo, na nossa opinião, é muito importante que estes critérios sejam definidos e que ferramentas sejam desenvolvidas para ajudar a identificá-los nos projetos dos alunos. Conforme discutido no capítulo 8, cada papel da atividade educacional exige um tipo de descrição (ou externalização) para que possa trabalhar. O tipo de informação requerida pelo aprendiz para refletir sobre seu projeto é diferente do tipo de informação requerida pelo facilitador para avaliar e acompanhar o que está acontecendo. No caso do facilitador na Oficina de Jogos, seria muito interessante que fossem gerados relatórios com o nome do aprendiz, a quantidade de sessões e a relação dos personagens e tabuleiros que foram por ele criados.

Além destas ferramentas que seriam de grande valia para o acompanhamento e avaliação da atividade, outra necessidade grande levantada pela Oficina de Jogos foi a de material que ajudasse na introdução das ferramentas que foram utilizadas.

Boa parte das dúvidas que surgiam na fase de instrumentalização eram comuns à maioria dos aprendizes e poderiam ser facilmente sanadas com a presença de tutoriais automatizados, tabelas com os procedimentos mais usados, manuais de referência etc.. Embora existisse este tipo de material para o software Cocoa, ele não chegou a ser traduzido por falta de tempo.

Caso ele estivesse disponível, provavelmente a demanda de trabalho exigida do facilitador nesta fase fosse completamente diferente e os alunos teriam evoluído cada um no seu próprio ritmo, ao invés de terem que esperar a aula que era dada para todos ao mesmo tempo.

Com base nesta experiência, concordamos com a abordagem da Teoria da Atividade cuja proposta é a de que tudo o que for de treinamento e suporte relacionado com o software em si (ou do artefato computacional) e sua utilização na atividade, deva ser considerado no design deste software e da própria atividade.

Além disso, acreditamos que, de uma forma geral, para efeitos da atividade educacional, a tecnologia deve ser vista como uma extensão que potencializa e foca o trabalho de todos os envolvidos, ajudando a minimizar as sobrecargas que as atividades de uns colocam sobre as dos outros.

O próprio pensamento de que a combinação entre facilitador, computador e demais elementos do contexto deveria constituir um órgão funcional cujo objetivo fosse gerar condições propícias para o aprendizado, abre algumas possibilidades bem interessantes. Uma delas seria, por exemplo, reduzir a complexidade dos sistemas de suporte de alguns aplicativos, passando uma maior formação aos professores

que o utilizarão. Outra seria criar aplicativos que funcionassem como componentes que pudessem ser combinados pelo professor, conforme a necessidade.

10 Conclusões

Conforme colocado no capítulo 2, o objetivo principal deste trabalho era propor uma abordagem para o desenvolvimento de software para a educação que considerasse o contexto sociocultural onde este software será utilizado. A idéia era basear a nova abordagem nos conceitos da Teoria da Atividade e verificar como esta abordagem se sairia junto à educação Construcionista.

A partir desta estratégia, buscava-se também descobrir como seria trabalhar com a Teoria da Atividade e verificar até que ponto a abordagem proposta poderia contribuir para uma visão mais integrada do Construcionismo que orientasse o desenvolvimento e a análise de software para atividades construcionistas.

Este capítulo apresenta as conclusões dessa empreitada. Ele está organizado da seguinte forma:

- A seção 10.1 faz considerações gerais a respeito da interdisciplinariedade envolvida na abordagem proposta de desenvolvimento de software para a educação e identifica as principais inter-relações entre e sinergias percebidas entre a Engenharia de Software, a Teoria da Atividade e o Construcionismo.
- A seção 10.2 discute as principais dificuldades inerentes à Teoria da Atividade. Nela também são apresentadas as principais contribuições que a Teoria da Atividade trouxe, no âmbito do trabalho, para a compreensão do que é o computador e para o desenvolvimento de software para a educação.
- A seção 10.3 discorre sobre como foi lidar com o Construcionismo e o tipo de contribuições que o presente trabalho trouxe para este campo de pesquisa.
- A seção 10.4, faz uma análise da abordagem proposta com relação aos critérios norteadores do desenvolvimento de software para a educação apresentados no capítulo 6 e levanta uma discussão sobre a aplicabilidade da abordagem em outras teorias educacionais que não o Construcionismo e mesmo em outros domínios que não sejam o da educação.
- A seção 10.5 descreve como as questões levantadas pela abordagem ajudaram a estruturar os conceitos construcionistas em torno da definição de uma atividade educacional genérica e o como esta organização pode contribuir para o desenvolvimento de software.
- A seção 10.6 reflete sobre a análise feita no capítulo 8 a respeito do papel do artefato computacional na atividade construcionista e realça como os pontos da abordagem proposta no capítulo 6 ajudaram na formação de uma visão mais objetiva da importância do computador para o Construcionismo.
- A seção 10.7 apresenta os principais pontos percebidos com a Oficina de Jogos.
- A seção 10.8 discorre sobre quais seriam os próximos passos para a continuidade do trabalho.

10.1 Sobre o trabalho em si

Na nossa opinião, a maior contribuição deste trabalho foi trazer à tona a questão do software para a educação e, junto com isto, ressaltar a necessidade de se incluir, em sua análise e desenvolvimento, um conhecimento amplo das condições sócio-histórico-culturais do contexto em que ele será utilizado.

O problema, tal como colocado no capítulo 2, era identificar estas condições e descrevê-las de modo que os especialistas das áreas envolvidas conseguissem reconhecer os seus pontos de vista, perceber as interdependências com as áreas correlatas e buscar a solução que fosse mais eficaz, eficiente e viável.

Como resposta a esta questão, foi proposta uma abordagem de desenvolvimento de software fundamentada em conceitos da Teoria da Atividade e que integrasse conceitos de Engenharia de Software e da educação (ver capítulo 6).

Desafios inerentes à interdisciplinariedade do trabalho. De fato, como pôde ser percebido ao longo deste documento, o desenvolvimento de software para a atividades educacionais é uma atividade que, além de recente, é interdisciplinar por natureza. Estas suas peculiaridades apresentam uma série de desafios, dentre os quais se destacam os de gerenciar a riqueza e a diversidade dos campos relacionados e, ao mesmo tempo, combinar conceitos, esclarecer conflitos e identificar os espaços vazios que às vezes não são tratados por nenhum destes campos.

Em especial, foi muito difícil encontrar pessoas com quem discutir o trabalho. A Engenharia de Software é, de uma forma geral, uma área relativamente recente, principalmente na parte que trata dos aspectos humanos relacionados ao uso do computador. Muitas idéias novas surgem constantemente, procurando acompanhar e ditar o ritmo da tecnologia, mas principalmente no que se refere à tecnologia para a educação, ainda está dando seus primeiros passos.

A Teoria da Atividade, embora já tenha uma ampla quantidade acumulada de estudos, só agora está sendo mais valorizada no ocidente. Sua comunidade e trabalhos estão dispersos em várias localidades e, às vezes, com traduções diferentes para os mesmos termos. Encontrar artigos específicos não se mostrou uma tarefa nem um pouco trivial, principalmente sobre o desenvolvimento de software e sobre o uso do computador na educação.

O Construcionismo também é um campo de pesquisa bastante novo, concentrado em uns poucos centros, com trabalhos específicos e em diferentes línguas. Raras eram as referências que discutiam a teoria em si ao invés de uma de suas aplicações ou extensões. Parecia não haver uma comunidade de pesquisa que reunisse os conceitos fundamentais do Construcionismo e incentivasse a sua discussão e disseminação.

Assim, não parecia haver um único especialista com quem se pudesse discutir o trabalho como um todo. Conversava-se com alguém sobre Teoria da Atividade, Construcionismo ou Engenharia de Software. No máximo, com algumas pessoas

que entendiam de Engenharia de Software e Teoria da Atividade, ou que já haviam desenvolvido software com características construcionistas.

Felizmente, a maior parte das pessoas mostrou-se aberta para discussões. O próprio tempo de escrita do trabalho deu oportunidade para muitos *insights* e também favoreceu o amadurecimento e a reflexão sobre as idéias. Depois de um tempo, o próprio texto inacabado servia de base para as conversas.

Sinergia entre as áreas. Apesar da dificuldade inicial de integração, sentiu-se um enorme potencial de sinergia entre as áreas abordadas (ver Figura 35). A combinação Engenharia de Software, Teoria da Atividade e Construcionismo mostrou-se muito positiva. Cada área complementou a outra em uma variedade de aspectos. Por exemplo, na nossa opinião,

- A Engenharia de Software contribuiu com a Teoria da Atividade, passando sua experiência acumulada (técnicas, metodologias) de desenvolvimento de software, complementando, principalmente, pontos onde os aspectos técnicos eram mais relevantes.

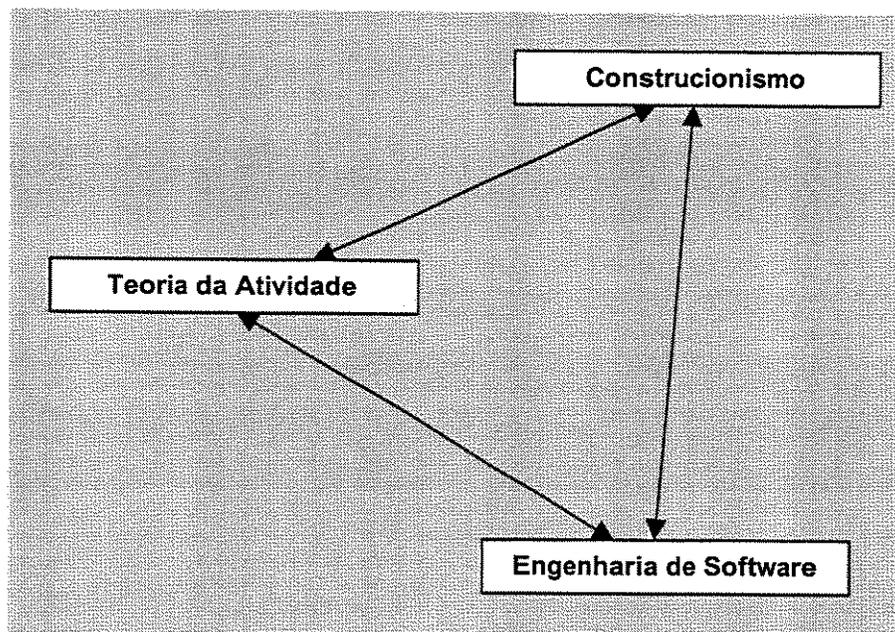


Figura 35 – Construcionismo, Engenharia de Software e Teoria da Atividade contribuíram mutuamente no desenvolvimento do trabalho.

- A Teoria da Atividade contribuiu com a Engenharia de Software, abrindo sua experiência integradora de diversas áreas do saber, ajudando na definição de uma terminologia comum (artefatos computacionais, objetos, etc.) e trazendo maiores subsídios, principalmente para a incorporação de fatores humanos como o aprendizado, a dinâmica social, os elementos culturais e históricos do meio etc. que, ultimamente, estão sendo cada vez mais valorizados no desenvolvimento de novas tecnologias.

Muitos pontos poderiam ainda ser explorados na relação entre essas duas áreas. Em especial, o conceito de orientação a objetos da Teoria da Atividade parece ter vários pontos similares e complementares ao de orientação a objetos da

Engenharia de Software. Neste sentido, um estudo mais específico, que ultrapassaria o escopo do presente trabalho, mereceria ser efetuado.

- A Teoria da Atividade contribuiu com o Construcionismo, ajudando na estruturação e organização das idéias, no levantamento de questões teóricas (como a importância dos aspectos sociais e históricos e do motivo da atividade), na integração com outras áreas do conhecimento e, principalmente, no desenvolvimento de uma visão de Construcionismo que fosse além da relação aprendiz-computador.
- O Construcionismo contribuiu com a Teoria da Atividade, disponibilizando toda sua bagagem acumulada da aplicação do computador na educação e também levantando discussões teóricas sobre a programação como uma representação dos processos mentais do aprendiz, sobre a motivação do aprendiz pelo processo além de pelo produto de sua construção e sobre a questão dos estilos de aprendizagem encontrados nas atividades construcionistas;
- O Construcionismo contribuiu com a Engenharia de Software, mostrando exemplos de software e explicitando os pontos mais importantes a serem considerados no desenvolvimento para a educação (como as propriedades e significados dos objetos manipulados, as facilidades de visualização e combinação das operações da máquina, etc.).
- A Engenharia de Software contribuiu com o Construcionismo, trazendo uma forma organizada de desenvolver e garantir a qualidade das ferramentas computacionais para as atividades construcionistas.

10.2 Sobre a Teoria da Atividade

Dificuldades no trabalho com a Teoria da Atividade. Retomando o que foi dito acima, não foi fácil encontrar referências sobre a Teoria da Atividade, ainda mais em português, sobre sua aplicação na educação ou no desenvolvimento de tecnologia. Não há muita bibliografia e a comunidade parece ser bem pequena. Felizmente, a boa receptividade desta comunidade e a crescente demanda, tanto na engenharia quanto na educação, por abordagens que incluam fatores socioculturais está incentivando um maior número de publicações e discussões sobre a área.

Mesmo conseguindo as referências, outra grande dificuldade encontrada foi a falta de uma padronização nos termos usados. Ao que parece, já existe uma terminologia aceita em russo e talvez em outras línguas. No caso do inglês, pelo menos nos textos consultados, tal como discutido no capítulo 3, ainda permanece uma certa confusão.

De qualquer modo, ainda que restritos a estas referências, ao nosso ver, os capítulos 3 e 4 parecem reunir os conceitos mais importantes da Teoria da Atividade

relacionados ao desenvolvimento deste trabalho e podem servir como orientação para os pesquisadores que estiverem se adentrando agora na área.

Contribuições da Teoria da Atividade para a compreensão do computador. Na nossa opinião, as definições de computador e os trabalhos estudados sobre a sua aplicação pela Teoria da Atividade foram muito interessantes e extremamente úteis, mas não pareciam se encaixar de modo completo. Enquanto algumas referências enfatizavam a flexibilidade e o potencial desta máquina, outras enalteciam a sua capacidade de trabalhar como extensão da mente humana. Não se achava, no entanto, referências que explicassem que atributos do computador o tornavam capaz destas coisas.

Felizmente, a leitura conjunta dos textos sobre a Teoria da Atividade e sobre o Construcionismo criou um solo fértil para as definições de “artefato computacional”, “artefato digital”, “objeto digital” e “aplicativo” introduzidas no capítulo 6. Estas definições, na nossa opinião, além de permitirem a integração das características acima mencionadas, também parecem constituir um elo comum entre ambas as teorias.

Por exemplo, o entendimento de artefatos computacionais como “máquinas que permitem combinar operações previamente definidas” (ver o capítulo 6) garante a questão da flexibilidade e da multiplicidade de usos do computador e ainda o apresenta como uma máquina que, similar ao que acontece na mente, porém de forma externalizada, permite dosar o grau de automação e consciência com que o usuário deseja trabalhar em cada momento.

Para o Construcionismo, a externalização desta combinação de operações, obtida por meio do “programa”, funciona como um retrato do caminho mental seguido pelo indivíduo na resolução de um problema. Quanto mais concreto e significativo ele for, mais subsídios dará para que este indivíduo reflita sobre as estratégias adotadas e aprimore o seu modo de pensar. Da mesma forma, por ser um “retrato externalizado da mente”, o programa também possibilita a análise do que foi feito por outras pessoas, oferecendo recursos para a discussão e subsidiando meios para intervenções cognitivas.

Também a idéia de que o computador permite manipular objetos que podem existir ou não na realidade fora da máquina contribuiu diretamente para um conceito de “objetos digitais” muito similar ao da discussão construcionista sobre “objetos transicionais” encontrada no capítulo 8.

Fora esta problemática conceitual, que tratava, principalmente, dos conceitos de “artefato computacional” e “artefato digital”, as demais idéias apresentadas sobre a atividade de uso e a atividade de desenvolvimento do artefato computacional (ver capítulo 4) foram aproveitadas quase que diretamente pela abordagem proposta no capítulo 6.

Contribuições da Teoria da Atividade para o desenvolvimento de software para a educação. Na nossa opinião, a Teoria da Atividade ajudou a mapear o contexto educacional de uma forma bastante completa, permitindo equacionar boa parte das questões inerentes ao desenvolvimento e uso da tecnologia na educação

levantadas no capítulo 2. Por exemplo, elementos como motivação e relacionamento social são intrínsecos à própria definição de atividade, recebendo destaque especial nas dimensões que analisam a “estrutura hierárquica da atividade” e a “orientação a objetos” (ver capítulo 3). Da mesma forma, independente do tipo de atividade, o aprendizado humano e as restrições específicas do ambiente também são considerados na dimensão que trata da “história e desenvolvimento” ou mesmo na “externalização e internalização”. O uso de artefatos e suas implicações aparece mais destacado na dimensão “mediação”.

Além disso, como mencionado anteriormente, já existe uma série de artigos da Teoria da Atividade discutindo atividades educacionais e mesmo o uso do computador. No entanto, havia pouco material descrevendo o desenvolvimento de software para a educação, muito menos uma proposta de abordagem mais completa como a aqui apresentada.

Por fim, é importante frisar que, apesar das contribuições inúmeras acima, a utilização da Teoria da Atividade como um elemento estruturador e organizador de outras teorias educacionais tem que ser feito com muito cuidado. A Teoria da Atividade parte de pressupostos epistemológicos muito próprios que podem não ser compatíveis com os da teoria educacional sendo analisada. Nestes casos, as questões levantadas pela abordagem aqui proposta devem ser vistas como mera referência, não como um fator para se determinar que uma teoria educacional é mais “adequada” do que outra.

Pode ocorrer, e é até provável, que a teoria analisada enfatize conceitos que nem sequer sejam mencionados pela Teoria da Atividade. No caso do Construcionismo, isso aconteceu, por exemplo, em toda a parte relacionada aos estilos de aprendizado e à motivação que os aprendizes obtêm no próprio processo de construção de seus projetos.

10.3 Sobre o Construcionismo

Conforme discutido nos capítulos 1 e 5, a pesquisa construcionista está restrita a uns poucos centros de pesquisa e a maior parte dos artigos se concentra em um aspecto diferente da teoria. Isso de fato dificulta bastante a formação de uma visão geral e de uma comunidade que incentive o seu desenvolvimento.

O capítulo 5 deste trabalho procurou minimizar este problema. Ele constitui uma revisão das principais referências teóricas e apresenta o Construcionismo tanto como uma teoria de aprendizado (baseada nas idéias do construtivismo piagetiano), quanto como uma teoria educacional que aplica estes princípios de aprendizado na criação de micromundos educacionais.

Mais adiante, nos capítulos 7 e 8, estes conceitos foram discutidos em maior profundidade, agora sob o olhar da Teoria da Atividade, procurando-se determinar até que ponto eles seriam, realmente, dependentes do Logo, ou inclusive do

computador. Na nossa opinião, esta intenção foi atingida com êxito, conseguindo-se separar um pouco “Construcionismo” e “computador” e apresentando-se observações quanto ao desenvolvimento de artefatos para atividades construcionistas.

Dadas as dificuldades relativas à dispersão da literatura construcionista, o capítulo 5 foi um dos mais difíceis e demorados de ser escrito. Ainda assim, apesar do esforço, acreditamos que ainda possa haver uma série de conceitos importantes a serem incorporados. Procuramos nos concentrar nas referências básicas, principalmente nas escritas pelo próprio Papert, mas sabemos que muito material já foi publicado em países como a Argentina, República Tcheca e outros. Espera-se que, com o tempo, este material possa ser acessado e incorporado ao corpo deste trabalho.

10.4 Sobre a abordagem proposta

A abordagem proposta no capítulo 6 parece englobar a maioria dos atributos identificados como importantes para o desenvolvimento de software para a educação (também descritos no capítulo 6). Ela parte de princípios teóricos (descritos na “análise da atividade genérica”), analisa os prós e contras de se usar o computador (descritos na “análise do artefato computacional”) e descreve as características específicas do ambiente onde ele será utilizado (na “análise da atividade específica”). Tudo isso considerando os aspectos sócio-histórico-culturais do ambiente educacional e utilizando da terminologia definida pela Teoria da Atividade para facilitar o intercâmbio de idéias.

Já o lado mais técnico do desenvolvimento, que inclui a análise financeira, o projeto estrutural do software em si e tudo o mais relacionado à sua codificação, teste, otimização, manutenção, distribuição, etc. não foi enfatizado. No máximo, defendeu-se a idéia de que o artefato computacional, ao menos inicialmente, deveria ser tratado como um todo, sem separar o que é hardware do que é software e que o desenvolvimento da tecnologia deveria se basear em protótipos e estudos o mais realistas possível.

De fato, a idéia básica por trás da abordagem proposta era identificar, com auxílio dos conceitos da Teoria da Atividade, os fatores humanos que mais poderiam influenciar a utilização do computador na educação e apresentá-los de modo que as outras metodologias da Engenharia de Software, que abordam os aspectos essencialmente técnicos, pudessem prosseguir seu trabalho. Assim sendo, as principais contribuições da abordagem proposta se concentram nas fases de “análise da atividade educacional” e “análise do artefato computacional”.

Vale notar que a idéia de orientar o desenvolvimento do software ao redor de protótipos não é originária da Teoria da Atividade e já vem sendo estimulada por várias abordagens da Engenharia de Software. Por outro lado, o conceito de

artefato computacional é próprio da Teoria e parece ser bastante interessante para as novas modalidades de computadores e artefatos digitais (como agendas e brinquedos eletrônicos) que, atualmente, estão sendo viabilizados.

Como aplicação prática da abordagem proposta, foi feito um estudo mais detalhado em cima do que seria uma atividade educacional construcionista (ver capítulos 7, 8 e 9). Na nossa opinião, neste caso, como já foi levantado acima e como será discutido nas seções 10.5 e 10.6 deste mesmo capítulo, a utilização da abordagem se mostrou bastante positiva ajudando a levantar e a esclarecer uma série de aspectos importantes.

Ainda assim, conforme discutido na seção 10.2, é preciso se tomar uma série de precauções para se evitar que a aplicação da abordagem acabe desvirtuando a teoria educacional que estiver sendo trabalhada, principalmente se ela for baseada em pressupostos epistemológicos diferentes dos da Teoria da Atividade.

De fato, ao nosso ver, o Construcionismo, ainda que uma teoria de origem construtivista, já trazia em si uma preocupação sociocultural (como os conceitos de micromundos e Construcionismo Social discutidos no capítulo 5) que facilitou a sua análise pela Teoria da Atividade. Resta a dúvida de se a abordagem proposta poderia ser utilizada com outras teorias educacionais que não o Construcionismo.

De qualquer forma, os princípios que orientaram a criação da abordagem (discutidos nos capítulos 2 e 6) são de ordem genérica, independentes de teoria educacional. Apesar disso, por mais que se apresente conjecturas, só a tentativa real de aplicar esta abordagem com outras teorias educacionais (como o instrucionismo) é que poderá trazer dados mais concretos a esta pergunta.

Também poderia ser colocada como questão a possível aplicabilidade da proposta em domínios que não fossem o da educação. Na nossa opinião, a resposta seria sim e não. Sim, porque seus princípios fundamentais de eficácia, eficiência e viabilidade, tal como apresentados no capítulo 2, podem ser aplicados ao desenvolvimento de qualquer tecnologia (mesmo não computacional, desde que adaptada) e para qualquer área, lembrando que a educação, ou pelo menos o aprendizado, conforme defende a Teoria da Atividade, é uma característica presente em todas as atividades humanas.

E não, porque ela parece ser especialmente indicada para atividades onde o fator humano seja central. Para o desenvolvimento de aplicativos de controle de produção, processamento mecânico e etc. há metodologias de desenvolvimento de software mais indicadas a serem utilizadas.

Por fim, vale notar que, como são muitos os pontos a serem considerados (ver capítulo 6), facilmente a descrição da atividade poderá atingir uma complexidade ou um tamanho inadmissíveis para um determinado projeto que se tenha em mente. Para que este problema seja tratável, é imprescindível uma boa dosagem das características a serem analisadas. Talvez, restringir as características mais importantes de cada componente da atividade, ou delimitar as sub-atividades a serem consideradas, sejam boas saídas. No caso da análise da atividade

construcionista, por exemplo, tivemos que nos concentrar na atividade do aprendiz e deixar a do facilitador e dos demais papéis em segundo plano (ver capítulo 7).

Uma boa notação e uma boa ferramenta que facilite as verificações e o tratamento das informações também ajudariam bastante a lidar com esta dificuldade.

10.5 Sobre a análise da atividade construcionista

Enquanto no capítulo 5 foram apresentados os princípios fundamentais do aprendizado e da educação construcionista, o capítulo 7 procurou combinar estes princípios na definição do que seria uma atividade educacional construcionista genérica. Na nossa opinião, este processo ajudou a perceber uma série de elementos interessantes do Construcionismo e deu base para uma análise mais detalhada do papel do Logo e do computador (análise esta complementada no capítulo 8) frente a esta atividade.

De fato, embora o Construcionismo atualmente esteja muito centrado no Logo e suas variações, tentar compreender a teoria pela simples análise desta ferramenta, desvinculada de seu contexto de uso, é uma tarefa que tende a resultar em conclusões errôneas. Mesmo o Logo, um artefato que se presta muito bem a atividades construcionistas, também pode ser usado de modo instrucionista. Basta um professor pedir para que seus alunos copiem programas já prontos para dentro do computador.

No capítulo 7, feitas as ressalvas discutidas nas seções acima, a Teoria da Atividade deu subsídios para que o Logo pudesse ser analisado frente a um contexto construcionista e, com isso, permitiu entender que aspectos da educação construcionista realmente estavam amarradas a ele e que aspectos eram inerentes ao contexto criado fora da máquina.

De certa forma, cada aspecto analisado da atividade contribuiu para a formação de uma imagem mais objetiva e estruturada do Construcionismo. A **análise da estrutura hierárquica** da atividade detalhou quais eram os principais componentes de uma atividade construcionista, como se relacionam, que tipo de papéis cada participante poderia assumir e que tipo de conceitos deveriam ser enfatizados.

Além disso, esta dimensão da análise descreveu em bastante profundidade as principais ações efetuadas pelos aprendizes, propondo uma nova versão do tradicional ciclo construcionista da descrição-execução-reflexão-depuração-descrição explicado no capítulo 5. Nesta nova versão, conhecida pelo ciclo da idealização-construção-avaliação-depuração-descrição, uma ênfase maior foi dada à descrição das idéias originais do aprendiz, à descrição dos motivos que o levaram a seguir determinados caminhos, a uma independência maior do computador e a um maior espaço para o papel da comunidade.

A análise da dimensão “**orientação a objetos**” ajudou a identificar que características dos objetos deveriam ser consideradas na construção dos micromundos construcionistas e a importância do tema da atividade. Neste ponto também foi ressaltado que, diferente do que é enfatizado pela Teoria da Atividade, para o Construcionismo não só o produto da atividade, mas o processo de sua construção (a liberdade e o tipo de incentivo dado) também é fundamental para a motivação dos participantes.

A dimensão “**internalização e externalização**” ressaltou como o Construcionismo vê a influência das relações sociais na formação do indivíduo, explicitou a importância da descrição das ações dos aprendizes e descreveu o elemento político que existe por trás de tudo isso.

A dimensão **história e desenvolvimento**, por um lado, facilitou a análise do Construcionismo e do Logo frente a época em que surgiram levantando os pontos que deveriam ser mantidos ou substituídos com a evolução teoria e da tecnologia. Ressaltou-se aí, por exemplo, a questão de que, quando o Logo foi criado, os computadores não tinham multimídia, interfaces gráficas sofisticadas e nem tão pouco a Internet.

Por outro lado, em uma escala temporal mais microscópica, esta parte do trabalho discutiu os diferentes estilos de aprendizado – aspecto este que não parece ser discutido pela Teoria da Atividade – e também ajudou a classificar os diferentes momentos que compõem uma atividade educacional construcionista: a fase de instrumentalização, onde os participantes estão se conhecendo e aprendendo a lidar com os artefatos e objetos do ambiente e a fase de fluência, onde já estão mais concentrados no desenvolvimento de seus projetos.

Por fim, a dimensão **mediação** ajudou perceber os diferentes tipos de mediadores (computador, facilitador e outros) encontrados na relação dos participantes da atividade com o mundo e como cada um pode ser utilizado para explicitar ou mascarar os pontos mais ou menos relevantes para a educação. Foi mostrado, por exemplo, como os comandos do Logo Gráfico ajudam a realçar as características geométricas da tartaruga e de como existem poucas ferramentas para auxiliar no trabalho dos facilitadores na atividade construcionista.

Devido a restrições de tempo, o presente trabalho acabou focando-se, principalmente, na análise do aprendizado construcionista. Na medida do possível, comentários sobre as atividades do facilitador e do designer da atividade também foram inseridos. Ainda assim, para que a análise da atividade educacional ficasse completa, seria necessário descrever estas outras atividades com maior grau de detalhe, algo que só será possível de se fazer no futuro.

De qualquer modo, como discutido no capítulo 7, o aprendizado é a sub-atividade mais importante da educação construcionista e, com base na descrição apresentada, já se torna possível tirar uma série de diretrizes para o desenvolvimento de novas atividades construcionistas que sejam, inclusive, independentes do computador ou da matemática (que estão muito relacionados ao Logo).

Por exemplo, na nossa opinião, antes de mais nada, o designer da atividade deveria identificar o domínio educacional que pretende abordar (que será o objetivo pedagógico da atividade) e os aprendizes (qual a sua idade e que tipo de conhecimento já possuem?) que participarão da atividade. A seguir, ele deveria encontrar um exemplo real que fosse bem significativo da aplicação deste domínio para os aprendizes. A partir daí, o tema da atividade já poderia ser derivado e o designer poderia ir definindo cada aspecto da atividade com a ajuda do questionário apresentado no capítulo 6 e das considerações dos capítulos 7 e 8.

Idealmente, no entanto, estas considerações talvez pudessem ser reunidas em um novo questionário de referência. Faltaria, no entanto, uma maior quantidade de experimentações práticas para refiná-las. A definição deste conjunto de diretrizes para a criação de atividades e artefatos construcionistas também poderiam ser exploradas em uma possível continuidade do trabalho.

10.6 Sobre o papel do computador no Construcionismo

Conforme discutido ao longo do trabalho, o computador executa um papel tão central dentro do Construcionismo que às vezes fica difícil se imaginar o que seria do Construcionismo sem esta máquina. Realmente, como foi visto, de uma forma geral, o computador é um instrumento viabilizador e facilitador de uma série de ações da atividade educacional construcionista, ajudando na expressão, visualização e execução de processos os mais variados.

Além disso, para o Construcionismo, a possibilidade de combinar as operações predefinidas da máquina em novas operações mais específicas, de forma quase tátil e transparente, torna o computador uma extensão da mente que possibilita a visualização do que foi feito e como, abrindo espaço para a discussão e reflexão da solução e, provavelmente, do aprimoramento das estratégias adotadas.

Por outro lado, com o subsídio da Teoria da Atividade, este trabalho também ajudou a desmistificar e esclarecer um pouco mais esta relação do computador com a educação construcionista. Ainda que ele seja, por definição, o artefato que permite esta combinação de operações (ver discussão no capítulo 6), não é qualquer computador que faz isto da mesma forma e também há uma série de fatores externos que influenciam este processo. O aprendiz precisa estar motivado a entrar na empreitada, os elementos manipulados tem que ser significativos e importantes para ele, tem que haver incentivo para a discussão, etc.

Em especial, no capítulo 8 foi feita uma análise detalhada do artefato computacional frente a cada uma das dimensões da atividade educacional construcionista. Percebeu-se, por exemplo, que:

a) Na **estrutura hierárquica da atividade**, o computador pode oferecer suporte a cada um dos papéis identificados na divisão do trabalho do aprendizado

construcionista. Além disso, ele pode ainda facilitar a transição entre os papéis executados por um mesmo sujeito.

No caso do aprendiz, o computador pode ainda contribuir de modos diferentes para cada ação. Por exemplo: na idealização, pode ajudar nas anotações, representações e discussões; na construção, trazer toda uma flexibilidade aos materiais utilizados; na avaliação, ajudar a visualizar que estiver sendo interpretado e comparar os resultados; na depuração, ajudar a localizar e resolver os pontos mais problemáticos e, na descrição, gerar materiais específicos conforme o tipo de necessidade de cada momento e de quem for usá-las.

Destacou-se também que, na “avaliação”, apesar de o computador sempre fazer interpretações imparciais do que foi programado, o que facilita a compreensão do que foi feito, ele tem uma série de limitações quanto à compreensão do significado geral do resultado do programa e às intenções originais do aprendiz. Apesar de cada operação computacional já estar amarrada a um ou mais significados, é muito difícil que o computador consiga emitir opiniões sobre as inúmeras combinações possíveis com estas operações. Nestes casos, concluiu-se que o melhor talvez fosse uma combinação de interpretações feitas pelo computador com as feitas pelos demais participantes da atividade.

Na ação da “descrição”, também foram levantados os principais pontos positivos e negativos de se ter o programa em si como representação do que foi feito pelo aprendiz. Embora o programa possa dizer muito sobre as operações que foram utilizadas pelo aprendiz, ele nada informa sobre as ações efetuadas fora do computador, nem tão pouco sobre os caminhos percorridos ou os motivos que levaram o aprendiz a seguir uma determinada linha de raciocínio.

Por fim, também foi enfatizado que o artefato computacional pode funcionar como uma extensão do mecanismo de internalização e externalização da Teoria da Atividade, porém com a particularidade de oferecer algo externalizado e concreto que, segundo o Construcionismo, pode incentivar a tomada de consciência dos processos mentais.

b) Pelo viés da **orientação a objetos**, o computador em si, por ser um elemento novo e atraente, pode servir como motivador da atividade educacional, ajudando a disseminar inclusive práticas construcionistas em ambientes mais tradicionais. No entanto, para o Construcionismo, o ideal mesmo seria que o computador não fosse o objetivo da atividade, mas um meio que ajudasse na criação de situações educacionais em que o próprio tema e os conceitos utilizados fossem interessantes para os participantes.

Também foi levantado que a possibilidade de se trabalhar com objetos digitais por si só já traz toda uma abertura para o desenvolvimento de novas atividades como, por exemplo, aquelas que lidam com objetos que, dificilmente, poderiam estar presentes ou trabalhados de outro modo (devido ao seu tamanho, complexidade, custo, localização ou perigo de manipulação). Em especial, a possibilidade de, na programação, manipular-se as próprias operações da máquina como se fossem objetos traz ainda mais perspectivas para o desenvolvimento e análise da cognição.

c) Na dimensão **internalização e externalização**, o computador, principalmente com o auxílio das redes, pode ajudar a viabilizar novas práticas sociais e facilitar a construção de micromundos bastante diversos e não tão restritos aos limites geográficos e temporais. Também foi levantado que, dependendo do que for considerado no design da atividade e do artefato, determinados tipos de externalização poderiam ser mais incentivados do que outros.

d) Na dimensão **história e desenvolvimento**, o Construcionismo não necessariamente precisa depender do computador. Nem os aplicativos para atividades construcionistas precisam se limitar aos recursos tecnológicos da época em que o Logo foi criado.

Também percebeu-se que, embora as atividades construcionistas costumem demorar vários dias, normalmente não é oferecido muito suporte tecnológico para facilitar a continuidade e o acompanhamento do trabalho ao longo das múltiplas sessões.

e) Na dimensão **mediação**, o computador pode ampliar as oportunidades para que portadores de deficiências ou necessidades especiais também participem da atividade educacional.

Ficou patente ainda que o computador por si só não parece ser um mediador suficiente para a atividade educacional. É necessário um facilitador humano, mais pessoal e afetivo, que incentive os aprendizes e os oriente nas questões mais complexas encontradas no dia-a-dia do contexto educacional.

Por fim, na nossa opinião, esta análise do artefato computacional frente à atividade educacional construcionista ajudou a explicitar o fato de que o computador, por mais importante que seja, é mais um componente dentre vários da atividade educacional. Ajudou ainda a identificar alguns de seus principais potenciais e limitações, fortalecendo a conclusão de que, em alguns casos, a melhor aplicação da tecnologia se dá quando ela é combinada com os demais elementos humanos do contexto. Cada parte contribuindo com o que tem de melhor.

10.7 Sobre a Oficina de Jogos

Segundo o jargão construcionista, a Oficina de Jogos foi um “objeto que ajudou a pensar” o trabalho como um todo. Ela ajudou a perceber os diferentes aspectos práticos das atividades educacionais construcionistas – principalmente os extra-tecnológicos, como a importância de um suporte aos alunos, o tipo de carga exigido do professor, a importância das discussões, etc. – e motivou a busca pela abordagem aqui proposta.

Realmente, a Oficina de Jogos ajudou a tornar muito claro que, apesar da importância do computador para o Construcionismo, o conjunto hardware mais software representava apenas um pequeno componente de um processo muito

maior. O grau de liberdade dado aos participantes, a postura do professor, o tema desenvolvido, etc. também influenciavam muito o andamento das coisas.

Esta inter-relação entre os elementos foi tão aparente que, por mais que a análise tenha focado, principalmente, na atividade dos aprendizes, muitos aspectos das atividades do “facilitador” e do “designer da atividade” também foram levantados. Felizmente, análise a partir da Teoria da Atividade deu abertura para descrever todos estes aspectos de uma forma integrada. De fato, sentiu-se que procurar isolar a análise do aprendiz da dos elementos com quem ele interage era um procedimento forçado que pouco retrataria o que acontece na prática.

Outro aspecto complementar ressaltado pela Oficina de Jogos foi o de que, ainda que o computador seja apenas um de seus componentes, ele é um elemento que tem muito a contribuir para a atividade educacional – tanto que a Oficina de Jogos se desenvolveu ao redor dele.

Mesmo assim, apesar dos benefícios que o computador trouxe para a Oficina, a análise criteriosa desta atividade ajudou a perceber diversos pontos que, embora a tecnologia pudesse ajudar, ainda não estavam sendo cobertos. Exemplos disso são a falta de ferramentas para ajudar o facilitador, mecanismos de segurança, etc.

10.8 Sobre a continuidade do trabalho

Este trabalho constituiu uma primeira tentativa de se criar uma abordagem de desenvolvimento de software para a educação baseada na Teoria da Atividade. Para que as idéias aqui apresentadas possam se consolidar em uma metodologia de fato, muito resta a ser feito. Será necessário fazer experimentações com mais casos práticos, promover discussões, refinar os conceitos, formalizar notações, etc.

Como estas idéias basearam-se em casos de software já existentes, seria interessante verificar como a abordagem se sairia na criação de um software (e de uma atividade) a partir do zero e como ela se enquadraria no novo perfil de desenvolvimento de software a partir de componentes (ver capítulo 2). Mesmo para o caso estudado, do Construcionismo, falta ainda explorar com mais detalhes a atividade do facilitador e suas principais relações com a atividade do aprendiz.

A curto prazo, no entanto, talvez o mais indicado seja divulgar ao máximo as idéias aqui propostas para interessados e especialistas das diversas áreas abrangidas e promover uma discussão contínua, visando o refinamento do trabalho. Neste caso, para facilitar a discussão, provavelmente o melhor seria escrever um ou dois artigos focando os principais conceitos apresentados.

Espera-se, por exemplo, que especialistas em Engenharia de Software contribuam com propostas sobre como aperfeiçoar a abordagem apresentada com notações e detalhes para que ela venha a se constituir uma verdadeira metodologia de desenvolvimento de software para a educação. Outros pontos interessantes a

serem explorados talvez sejam o detalhamento das relações entre a abordagem aqui proposta e outras metodologias de desenvolvimento de software e, em especial, o levantamento dos pontos comuns, complementares e divergentes entre a visão de orientação a objetos proposta pela Teoria da Atividade e a da Engenharia de Software.

Especialistas da Teoria da Atividade e do Construcionismo também poderiam ajudar trazendo novos conceitos ou esclarecendo ainda mais os que já foram apresentados. Em especial, ao longo deste trabalho, foram levantadas várias questões que mereceriam um maior aprofundamento. “Até que ponto o manuseio de objetos digitais, representações, objetos materiais ou a combinação entre eles influencia na eficácia das atividades educacionais?”, “em que situações as comunidades distribuídas ou não-presenciais contribuem para o aprendizado?”, “em que situações a presença física de professores e colegas é insubstituível?” são apenas alguns dos exemplos que poderiam ser melhor explorados.



Referências bibliográficas

- ACKERMANN, Edith. Aprendizagem colaborativa. In: *Workshop Internacional sobre Formação de Professores via Telemática*, 2, 1998. Campinas, NIED/UNICAMP, novembro, 1998. 1 videocassete: VHS, son., color.
- _____. From decontextualized to situated knowledge: revisiting Piaget's water-level experiment. In: HAREL, Idit, Ed. *Constructionist learning*. Cambridge: The Media Laboratory, Massachusetts Institute of Technology, 1990.
- _____. Who is designing what for whom? Who is learning what from whom? or The clinical method as a tool for rethinking learning and teaching. In: Annual Meeting, American Research Association, 43.01. *Kids as Designers: Exploring New Frameworks for Learning*. Chicago, 1991.
- BANNON, Liam J. *Computer-mediated communication*. In: NORMAN, Donald A.; DRAPER, Stephen W., Eds. *User centered system design*. Hillsdale, Lawrence Erlbaum Associates, 1986a. p. 433-452
- _____. Issues in design: some notes. In: NORMAN, Donald A.; DRAPER, Stephen W., Eds. *User centered system design*. Hillsdale, Lawrence Erlbaum Associates, 1986b. p. 25-30
- BARANAUSKAS, Maria Cecília Calani. Procedimento, função, objeto ou lógica: linguagens de programação vistas pelos seus paradigmas. In: VALENTE, José Armando, Org. *Computadores e conhecimento: repensando a educação*. Campinas, NIED/UNICAMP, 1993. p.45-63
- BELLAMY, Rachel K. E. Designing Educational Technology: Computer-Mediated Change. In: NARDI, Bonnie A., Ed. *Context and consciousness: activity theory and human-computer interaction*. Cambridge, MIT Press, 1996. p. 123-146
- BODKER, Susanne. A Human Activity Approach to User Interfaces. *Human-Computer Interaction*, v. 4, p. 171-195, 1989.
- _____. *Through the interface: a human activity approach to user interface design*. Hillsdale, Lawrence Erlbaum Associates. 1991. 186 p.
- BORGES, Marcos Augusto Francisco. *O Design Centrado no Aprendiz no Sistema Jonas: uma experiência de desenvolvimento de um sistema para formação na empresa*. Campinas, 1997. (Dissertação mestrado) - IMECC - UNICAMP.
- BRAND, C.; PULVER, P.; RADER, C. Using visual models in elementary classrooms. In: *NARST 1997 Annual Meeting*. Chicago, National Association for Research in Science Teaching, 1997.
- BRAND, C.; RADER, C. How does a simulation program support students creating science models? In: IEEE, *Symposium on Visual Languages*. Boulder, IEEE Computer Society, 1996. p.102-109.
- BRASIL EM EXAME 97. São Paulo, Editora Abril, 1997. (Parte integrante do número 645 da revista Exame)

- BROWN, John Seely. From cognitive to social ergonomics and beyond. In: NORMAN, Donald A.; DRAPER, Stephen W., Eds. *User centered system design*. Hillsdale, Lawrence Erlbaum Associates, 1986. p. 457-486
- BRUCKMAN, Amy Susan. *MOOSE Crossing: construction, community, and learning in a networked virtual world for kids*. Cambridge, 1997. (Tese doutorado) - Media Laboratory - Massachusetts Institute of Technology
- CARD, Stuart K.; MORAN, Thomas P.; NEWELL, Allen. *The psychology of human-computer interaction*. Hillsdale, Lawrence Erlbaum Associates, 1983.
- COLE, Michael; ENGESTRÖM, Yrjö. A cultural-historical approach to distributed cognition. In: SOLOMON, Gavriel, Ed. *Distributed cognitions: psychological and educational considerations*. Cambridge University Press, 1993.
- CUBAN, Larry. *Teachers and machines*. New York, Teachers College Press, 1986. 134 p.
- CYPHER, Allen; SMITH, David Canfield. KidSim: end user programming of simulations. In: *Proceedings of CHI, 1995*. New York, ACM, 1995. p. 27-34
- DEWEY, John. *Experience and education*. New York, Macmillan, 1938.
- ENGESTRÖM, Yrjö. *Learning by expanding: an activity-theoretical approach to developmental research*. Helsinki, 1987. 372 p. (Tese doutorado) - University of Helsinki
- FALBEL, Aaron. *Constructionism: tools to build (and think) with*. Klejs, Lego Group, 1993.
- FREIRE, F. M. P.; PRADO, M. E. B. B. Professores construcionistas: a formação em serviço. In: *Congresso Internacional de Logos, 6 e Congresso de Informática Educativa do Mercosul, 1*, Porto Alegre, 1995. Anais. Porto Alegre, 1995. p. 229-236
- GILMORE, David J. et al. Learning graphical programming: an evaluation of KidSim. In: NORDBY, K. et al, Eds. *Human-computer interaction: Interact'95*. London, Chapman & Hall, 1995. p. 145-150
- GUOHUA. Bai. Contradition approach to IS design and use activities. In: BJERKNES, Gro; BRATTETEIG, Tone; KAUTZ, Karlheinz, Eds. *Proceedings of IRIS, 15*, 1992. Oslo, University of Oslo, 1992. p.171-188
- HAREL, Idit. *Children designers: Interdisciplinary constructions for learning and knowing mathematics in a computer-rich school*. Norwood, Ablex, 1991.
- HAREL, Idit; PAPERT, Seymour. Software design as a learning environment. In: HAREL, Idit; PAPERT, Seymour, Eds. *Constructionism*. Norwood, Ablex, 1991.
- HOLLAND, Dorothy; REEVES, James R. Activity theory and the view from somewhere: team perspectives on the intellectual work of programming. In: NARDI, Bonnie A., Ed. *Context and consciousness: activity theory and human-computer interaction*. Cambridge, MIT Press, 1996. p. 257-282

- JOURNAL OF INTERACTIVE MEDIA IN EDUCATION. Special Issue on authoring tools and the Educational Object Economy, maio, 1998. (Disponível em <http://www-jime.open.ac.uk/98/ew-editorial/ew-editorial.html>)
- KAFAI, Yasmin B. *Minds in play: computer game design as a context for children's learning*. Hillsdale, Lawrence Erlbaum Associates, 1995. 339 p.
- KAPTELININ, Victor. Activity Theory: implications for Human-Computer Interaction. In: NARDI, Bonnie A., Ed. *Context and consciousness: activity theory and human-computer interaction*. Cambridge, MIT Press, 1996a. p. 103-116
- _____. Computer-mediated activity: functional organs in social and developmental contexts. In: NARDI, Bonnie A., Ed. *Context and consciousness: activity theory and human-computer interaction*. Cambridge, MIT Press, 1996b. p. 45-68
- KAPTELININ, Victor; COLE, Michael. Individual and collective activities in educational computer game playing. In: *Computer Support For Collaborative Learning 97*. Ontario, Ontario Institute for Studies in Education, University of Toronto, 1997. (Disponível em <http://www.oise.utoronto.ca.csci/papers/kaptelinin.pdf>)
- KAPTELININ, Victor; NARDI, Bonnie A. Activity Theory: Basic Concepts and Applications. In: CHI97 Conference on Human Factors in Computing Systems. Atlanta, ACM/SIGCHI, 1997. (Tutorial)
- KAPTELININ, V.; NARDI, B.; MACAULAY, C. The Activity Checklist: A Tool for Representing the "Space" of Context. *interactions*, v.6, n.4, p.27-39, Jul./Aug. 1999.
- KAY, Alan C. Computers, networks and education. *Scientific American*, v.265, n.3, p.138-148, 1991.
- KUUTI, Kari. Activity Theory as a Potential Framework for Human-Computer Interaction Research. In: NARDI, Bonnie A., Ed. *Context and consciousness: activity theory and human-computer interaction*. Cambridge, MIT Press, 1996. p.17-44
- LEWIS, C. et al. Models children build: content, logic and educational impact. In: NARST. *1997 Annual Meeting*. Chicago, National Association for Research in Science Teaching, 1997.
- LAUREL, Brenda, Ed. *The art of human-computer interface design*. Menlo Park, Addison-Wesley, 1990. 523 p.
- LEONTIEV, A. N. The problem of activity in psychology. *Soviet Psychology*, v.13, n.2, p.4-33, 1974.
- LEONTYEV, A. N. Problems of the development of the mind. Moscow, Progress, 1981. Apud ENGSTRÖM, Yrjö, *Learning by expanding: an activity-theoretical approach to developmental research*. Helsinki, University of Helsinki, 1987. p.69

- MAZZONE, Jaures Salvatore. 2012: Educação na sociedade enxuta. Campinas, NIED/UNICAMP, 1995.
- MCARTHUR, David; LEWIS, Matthew; BISHAY Miriam. *The Roles of Artificial Intelligence in Education: Current Progress and Future Prospects*. Santa Monica, RAND, 1993. (Disponível em <http://www.rand.org/hot/mcarthur/Papers/role.html>)
- _____. *ESSCOTS for Learning: Transforming Commercial Software into Powerful Educational Tools*. Santa Monica, RAND, 1994. (Disponível em <http://www.rand.org/hot/mcarthur/Papers/esscots.html>)
- MINISTÉRIO DA EDUCAÇÃO. Secretaria de Educação Fundamental. *Parâmetros curriculares nacionais: introdução aos parâmetros curriculares nacionais*. Brasília, 1997.
- NARDI, Bonnie A. *A small matter of programming: perspectives on end user computing*. Cambridge, MIT Press, 1993. 162 p.
- _____. Studying context: A comparison of activity theory, situated action models, and distributed cognition. In: NARDI, Bonnie A., Ed. *Context and consciousness: activity theory and human-computer interaction*. Cambridge, MIT Press, 1996a. p.69-102.
- _____, Ed. *Context and consciousness: activity theory and human-computer interaction*. Cambridge, MIT Press, 1996b. 400 p.
- NEGROPONTE, Nicholas. *Being digital*. New York, Vintage Books, 1995. 255p.
- NEWMAN, Dennis. *Functional environments for microcomputers in education*. In: COLE, Michael; ENGSTRÖM, Yrjö; VASQUEZ, Olga, Eds. *Mind, culture and activity: seminal papers from the Laboratory of Comparative Human Cognition*. Cambridge, Cambridge University Press, 1997. p.279-291
- NIELSEN, Jakob. *Usability Engineering*. Orlando, AP Professional, 1993. 362 p.
- NORMAN, Donald A. *The invisible computer: why good products can fail, the personal computer is so complex, and information appliances are the solution*. Cambridge, MIT Press, 1998. 302 p.
- NORMAN, Donald A.; DRAPER, Stephen W., Eds. *User centered system design*. Hillsdale, Lawrence Erlbaum Associates, 1986. 526 p.
- NORMAN, Donald A.; SPOHRER, James C. Learner-Centered Education. *Communications of the ACM*, v.39, n.4, p.24-27, 1996.
- PAPERT, Seymour. A learning environment for children. In: SEIDEL, Robert J., RUBIN, Martin L., Eds. *Computers and communications: implications for education*. New York, Academic Press, 1977.
- _____. *Constructionism: a new opportunity for elementary science education*. Cambridge, Epistemology and Learning Group, Massachusetts Institute of Technology, Nov. 1986. (Proposta para The National Science Foundation)

- _____. *Computer criticism vs. technocentric thinking*. Cambridge, Epistemology and Learning Group, Massachusetts Institute of Technology, Nov. 1990. (E&L Memo número 1)
- _____. *Mindstorms: children, computers and powerful ideas*. 2. ed. New York, BasicBooks, 1993a. 230 p.
- _____. *The children's machine: rethinking school in the age of the computer*. New York, BasicBooks, 1993b. 241p.
- _____. *The connected family: bridging the digital generation gap*. Atlanta, Longstreet Press, 1996. 211 p.
- PERKINS, Michael; NÚÑEZ, Celia. Is there a market for learning software? *The Red Herring*, n. 98, Jan. 1994. (Disponível em <http://www.herring.com/mag/issue98/market.html>)
- PRADO, Maria Elisabette B. B. *O uso do computador na formação do professor: um enfoque reflexivo da prática pedagógica*. Campinas, 1996. (Dissertação mestrado) - Faculdade de Educação - UNICAMP
- PRADO, Maria Elisabette B. B.; BARELLA, F. M. F. Da repetição à recriação: uma análise da formação do professor para uma informática na educação. In: Congresso Ibero-Americano de Informática na Educação, 2, Lisboa, 1994. *Atas*. Lisboa, 1994. p.138-152. v.2
- RADER, C.; BRAND, C.; LEWIS, C. Degrees of comprehension: children's mental models of a visual programming environment. In: CHI97 Conference on Human Factors in Computing Systems. *Proceedings*. Atlanta, ACM/SIGCHI, 1997. p.351-358.
- REBER, Arthur S. *The Penguin Dictionary of Psychology*. London, Penguin Books, 1985. 848p.
- RESNICK, Mitchel. Beyond the centralized mindset. In: *Proceedings of the International Conference on the Learning Sciences*, 1991.
- _____. Behavior construction kits. *Communications of the ACM*, v.36, n.7, p.64-71, July, 1993.
- _____. Changing the centralized mind. *Technology Review*, p.32-40, July 1994.
- RESNICK, Mitchel; BRUCKMAN, Amy; MARTIN, Fredy. Pianos not stereos: creating computational construction kits. *interactions*, v.3, n.5, p.40-50, Sept./Oct. 1996.
- ROCHA, Heloisa Vieira. Representações computacionais auxiliares ao entendimento de conceitos de programação. In: VALENTE, José Armando, Org. *Computadores e conhecimento: repensando a educação*. Campinas, NIED/UNICAMP, 1993. p.24-44
- ROSHELLE, Jeremy et al. Scalable integration of educational software: exploring the promise of component architectures. *Journal of Interactive Media in Education*, May, 1998. (Disponível em <http://www-jime.open.ac.uk/>)

- SCHNASE, John L.; CUNNIUS, Edward L. *CSCL '95 The First International Conference on Computer Support for Collaborative Learning*. Bloomington, Indiana University, Oct. 1995. (Disponível em <http://www-cscl95.indiana.edu/cscl95>)
- SHAW, Alan Clinton. *Social constructionism and the inner city: designing environments for social development and urban renewal*. Cambridge, 1995. (Tese doutorado) - Epistemology and Learning Group - Massachusetts Institute of Technology
- SHUKLA, Shilpa. Context and consciousness: activity theory and human-computer interaction; book review. *SIGCHI Bulletin*, v.29, n.2, April 1997.
- SMITH, David Canfield; CYPHER, Allen; SPOHRER, Jim. KidSim: programming agents without a programming language. *Communications of the ACM*, v.37, n.7, p. 54-67, July 1994.
- SMITH, David Canfield et al. Designing the Star user interface. *Byte Magazine*, v.7, n.2, p.242-283, April 1982.
- SOLOWAY, Elliot; GUZDIAL, Mark; HAY, Kenneth E. Learner-Centered Design: The Challenge for HCI In The 21st Century. *interactions*, v.1, n.2, p.36-48, April 1994.
- SOLOWAY, Elliot; PRYOR, Amanda. The next generation in Human-Computer Interaction. *Communications of the ACM*, v.39, n.4, p.16-18, 1996.
- SOLOWAY, Elliot et al. Science in the palms of their hands. *Communications of the ACM*, v.42, n.8, p.21-27, 1999.
- SPITULNIK, Jeff et al. *The RiverMUD design rationale: Scaffolding for scientific inquiry through modeling, discourse, and decision making in community-based issues*. Atlanta, Highly Interactive Computing Research Group, Michigan University, 1995. (Relatório técnico)
- TIKHOMIROV, O. K. The psychological consequences of computerization. In: Wertsch, James V, Ed. *The concept of activity in Soviet psychology*. Armonk, M.E. Sharpe, 1981. p.256-278
- TURKLE, Sherry; PAPERT, Seymour. Epistemological pluralism and the revaluation of the concrete. In: HAREL, Idit; PAPERT, Seymour, Eds. *Constructionism*. Norwood, Ablex, 1991. p.161-191
- UMASCHI, Marina. Soft toys with computer hearts: building personal storytelling environments. In: CHI97 Conference on Human Factors in Computing Systems. *Proceedings*. Atlanta, ACM/SIGCHI, 1997. p.20-21 (Disponível em <http://www.acm.org/sigchi/chi97/proceedings/demo/mu.htm>)
- VALENTE, José Armando. Logo and Freire's Educational Paradigm. *Logo Exchange*, v.11, n.1, 1992.
- _____. Logo as a window into the mind. *Logo Update*, v.4, n.1, 1995.

- _____. Por quê o computador na educação? In: VALENTE, José Armando ,
Org. *Computadores e conhecimento: repensando a educação*. Campinas,
NIED/UNICAMP, 1993. p.24-44
- _____. O papel do professor no ambiente Logo. In: VALENTE, José Armando ,
Org. *O professor no ambiente Logo: formação e atuação*. Campinas,
NIED/UNICAMP, 1996a. p.1-34
- _____, Org. *O professor no ambiente Logo: formação e atuação*. Campinas,
NIED/UNICAMP, 1996b. 435p.
- VIGOTSKI, Liev Semionovitch; [COLE, M. et al., Orgs.]. *A formação social da mente: o desenvolvimento dos processos psicológicos superiores*. 5.ed.
Tradução de José Cipolla Neto, Luis Silveira Menna Barreto, Solange Castro Afeche. São Paulo, Martins Fontes, 1994.
- VYGOTSKY, Liev Semionovitch. The instrumental method in psychology. In:
WERTSCH, James V., Ed. *The concept of activity in Soviet psychology*. Armonk,
M.E. Sharpe, 1981. p.134-146
- WERTSCH, James V. The concept of activity in Soviet psychology: An introduction.
In: WERTSCH, James V., Ed. *The concept of activity in Soviet psychology*.
Armonk, M.E. Sharpe, 1981. p.3-36

