

Análise Lógico-Operatória do Ambiente de Desenvolvimento Cooperativo de Programação ENVY/400

Dra. Patricia Alejandra Behar

Professora do Departamento de Estudos Especializados
Faculdade de Educação
Universidade Federal do Rio Grande do Sul
behar@{inf, edu}.ufrgs.br

Dr. Antônio Carlos da Rocha Costa

Escola de Informática - ESIN/UCPEL
Universidade Católica de Pelotas
rocha@atlas.ucpel.tche.br

Abstract

In the present work we describe the Cooperative Programming Development Environment ENVY/400 and its piagetian logical-operatory analysis. Thus, it's presented a model that represent the interactive process between the collective subject and the related tool. Based on it, we describe some of the logic and/or infralogic operations of the environment.

Keywords: collective subject, logical-operatory modeling, cooperative computational tools.

1 Introdução

Este estudo é um dos resultados do trabalho de investigação desenvolvido em Behar (1998), que utiliza a teoria piagetiana (Piaget, 1972, 1973), (Piaget & Inhelder, 1976), (Castorina & Palau, 1982) como uma das formas de realizar uma análise operatória de ferramentas computacionais de uso individual e coletivo.

Para entender o papel do sujeito em relação ao computador ou, mais especificamente, a uma determinada ferramenta, os psicólogos piagetianos acompanham e estudam a evolução cognitiva do sujeito, coletando os dados necessários até chegar no seu desenvolvimento máximo, com o objetivo de analisá-lo. Neste caso, ele é interpretado em termos da lógica-operatória. Mas, para compreender de forma completa essa interação, também é preciso entender a ferramenta que está envolvida na mesma. Quem faz esse estudo é a Ciência da Computação e, para tornar essa interação compatível, isto é, descrita nos mesmos termos, o computador deve ser reinterpretado na lógica-operatória. Ou seja, para avaliar o sujeito em relação a este objeto peculiar, é

muito conveniente caracterizar esse objeto no que se refere ao estado atual dos conhecimentos existentes a respeito dele, utilizando uma linguagem semelhante à que é usada para a análise do sujeito.

Como este estudo trata, especificamente, de uma ferramenta computacional cooperativa, o sujeito que se encontra envolvido nesta interação é um sujeito coletivo, como definido em (Behar & Costa, 1997b). Portanto, foi preciso investigar as características deste tipo de sujeito, os elementos e as funções que devem ser levados em conta para depois construir os modelos de interação, do ponto de vista da lógica operatória (Behar & Costa, 1997a). A partir destes, foi aplicado um dos modelos descritos no Ambiente de Desenvolvimento Cooperativo de Programação ENVY/400 (IBM, 1993), para poder analisá-lo.

Como se trata de dar continuidade ao trabalho apresentado em (Behar & Costa, 1997b), não serão descritos aqui os conceitos necessários para compreender a análise operatória do ambiente cooperativo em questão, pois já foram abordados anteriormente. Partimos, portanto, para uma breve descrição do Ambiente de Desenvolvimento Cooperativo de Programação ENVY/400, apresentamos seu modelo de interação e algumas operações lógicas e/ou infralógicas identificadas na ferramenta. Cabe enfatizar que a descrição completa desta análise encontra-se em (Behar, 1998).

2 Descrição do Ambiente de Desenvolvimento Cooperativo de Programação ENVY/400

Este ambiente pode ser utilizado por equipes de desenvolvimento de programas compostas por um número variado de sujeitos, que são responsáveis pela realização de programas em conjunto (IBM, 1993). O ambiente proporciona um extenso conjunto de ferramentas de programação em equipe, serviços de acesso à objetos AS/400 e, ainda, ferramentas de programação visual para aplicações de interfaces do tipo CUA 91. O código das aplicações é escrito em Smalltalk, uma linguagem orientada a objetos, que inclui uma coleção ampla de códigos fonte. Esta linguagem suporta classes definidas pelo próprio usuário, que permitem a criação de objetos de trabalho e gerenciamento automático de memória. A biblioteca compartilhada baseada em AS/400, que é um arquivo armazenado em *folders* compartilhados do tipo AS/400, mantém uma história completa de todas as mudanças da aplicação que acontecem durante o seu ciclo

de vida. As facilidades de exportação/importação seletivas suportam a transferência de versões de aplicações e componentes de aplicações entre organizações de desenvolvimento dispersas geograficamente. Utilizando este ambiente, as equipes de desenvolvimento de programas podem criar cooperativamente aplicações combinando interfaces orientadas a objetos do tipo CUA 91 com objetos AS/400 - todos vindos de um ambiente de desenvolvimento individual. O ambiente de desenvolvimento ENVY/400 gerencia as configurações das aplicações para aliviar o desenvolvimento multiplataforma e o empacotamento dos sistemas de produção final, isto é, o conjunto de aplicações. Existem três tipos de configuração da aplicação: a) uma configuração sem nenhum tipo de conexão com AS/400, ou seja, de uso individual, onde a estação de trabalho realiza todas as funções, independentemente do hospedeiro (AS/400); b) uma aplicação rodando em uma estação privada de trabalho e acessando os dados do hospedeiro através dos serviços disponíveis (compartilhados); e c) uma aplicação cooperativa completa, com estações de trabalho (PWS) e componentes do hospedeiro, isto é, o AS/400 funciona como controlador. Como pode ser observado, o ambiente pode ser utilizado de diversas formas, de acordo com a sua configuração. Cada estação de trabalho tem a sua própria imagem em relação ao repositório, ou seja, à biblioteca compartilhada, e é a partir desta que são criados os componentes da aplicação. A aplicação ENVY/400 provê uma interface consistente para disparar recursos distribuídos para o desenvolvimento de aplicações distribuídas através de estações de trabalho rodando em Windows ou OS/2 e AS/400 sobre uma conexão de rede. Os objetos são denominados de agentes e estes utilizam os despachantes para realizar o trabalho. A tríade agente-despachante-serviço caracteriza a aplicação. Quem faz o processo de empacotamento da aplicação para a liberação final é o Empacotador/ENVY, isto é, este torna a aplicação pública. Várias aplicações individuais são empacotadas em uma aplicação cooperativa. O ENVY/400 é integrado com um Gerenciador ENVY de programação cooperativa e facilidades de controle de versões. Como pode ser visto, o sistema cooperativo ENVY/400, além de ser um ambiente cooperativo de desenvolvimento de programas, é também um ambiente de aplicação.

Foram utilizados como base os modelos de interação descritos em (Behar & Costa, 1997a). Somente um destes foi aplicado ao ambiente em questão, resultando, em termos gerais, no modelo de interação dos sujeitos usuários/programadores (sujeito coletivo) com o ENVY/400, mostrado na figura 1. Cabe enfatizar que a explicação deste

modelo, assim como o processo de construção do mesmo, encontra-se detalhado em (Behar & Costa, 1997b). Este pode ser lido da seguinte forma:

- O sujeito (S) é composto por estruturas afetivas (EA), cognitivas (EC) e simbólicas (ES). Levamos em conta o fator mental de um sujeito, que é formado, segundo Piaget (1973), por três aspectos indissociáveis: o estrutural (cognitivo-O), o energético (afetivo-V) e os sistemas de símbolos (simbólico-L), servindo de significantes a estas estruturas operatórias ou a estes valores individuais. Cada sujeito tem a sua estação de trabalho Windows 3.1 ou OS/2 v2.0 e um suporte de PC (SPC). Através desta ferramenta, eles interagem com os serviços AS/400, que também têm um suporte de PC. Cada sujeito cria os seus componentes, que são unidades de programas, desenvolvendo a sua própria representação individual (RI). Tanto esta quanto os componentes entram no módulo Gerenciador/Empacotador ENVY (G-ENVY, E-ENVY), empacotando todas as aplicações geradas nas estações de trabalho e liberando a representação cooperativa (RCoop), que é o produto final, desenvolvido pelo sujeito coletivo (SC). Todos os sujeitos compartilham a biblioteca do ambiente ENVY/400 (BC-ENVY). Os componentes e as aplicações podem ser manipuladas pelos programadores de forma direta ou indireta, como definido em (Behar, 1998).

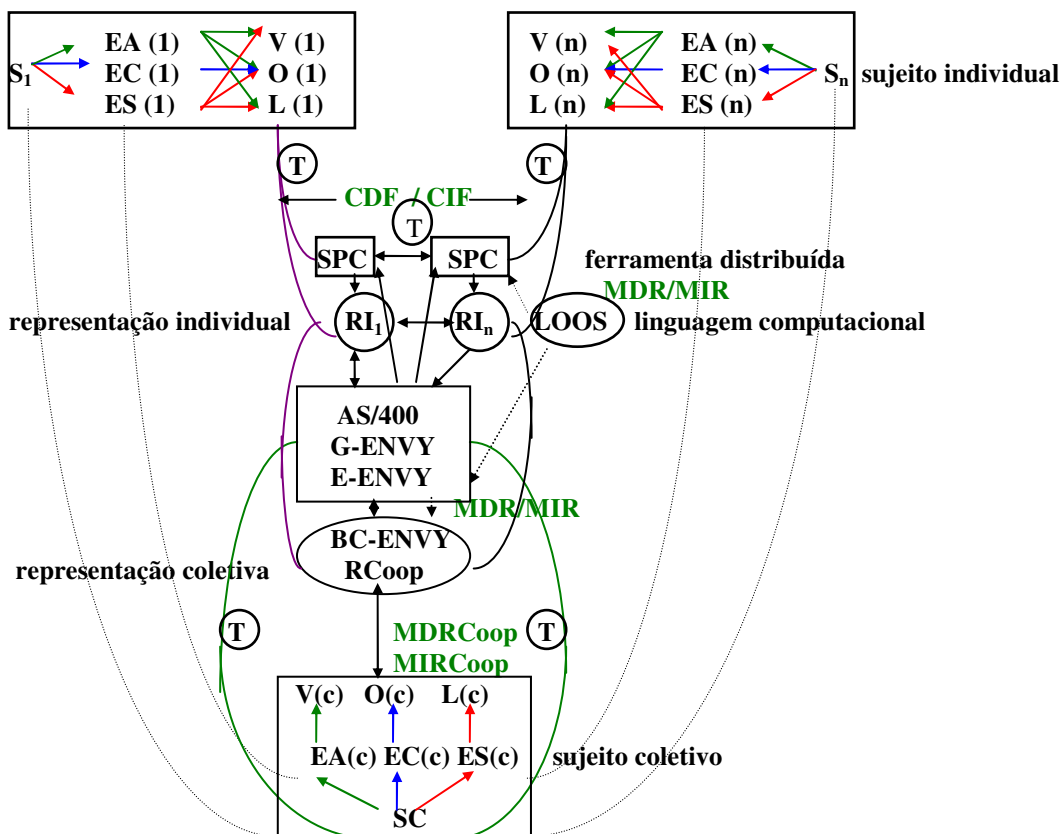


Figura 1- Modelo de interação do sujeito coletivo com o ambiente ENVY/400

onde: **S₁**: sujeito₁; **S_n**: sujeito_n; **SC**: sujeito coletivo; **EA**: estrutura afetiva; **EC**: estrutura cognitiva; **ES**: estrutura simbólica; **V**: valores; **O**: objetos; **L**: linguagem; **AS/400**: serviços do hospedeiro AS/400; **SPC**: suporte PC; **T**: tela, teclado, etc...(canal de comunicação sujeito-ferramenta); **:** elemento ativo (agente, programa, sujeito); **○**: elemento passivo (memória, banco de dados, texto); **RI₁**: representação do sujeito₁ (unidades de programas); **RI_n**: representação do sujeito_n; **BC-ENVY**: biblioteca compartilhada baseada em AS/400 do ambiente ENVY; **G-ENVY**: Gerenciador ENVY **E-ENVY**: Empacotador ENVY; **LOOS**: linguagem orientada a objetos Smalltalk; **RCoop**: representação cooperativa; **CDF/CIF**: controle direto/indireto sobre a ferramenta; **MDR/MIR**: manipulação direta/indireta sobre a representação; **MDRCoop/MIRCoop**: manipulação direta/indireta sobre a representação cooperativa.

Uma biblioteca desenvolvida coletivamente é uma base de dados do gerenciador ENVY em formato de arquivo PC e é armazenada em um *folder* compartilhado AS/400. Ela armazena versões e edições de todos os objetos ordenados, desde aplicações, sub-aplicações, classes, até métodos individuais. Somente o dono de uma aplicação pode liberar uma nova versão para o resto da equipe de desenvolvedores. Dessa forma, são substituídos, em alta velocidade, os estilos antigos registrados/enviados de versões gerenciadas. A figura 2 mostra como uma aplicação é aberta para modificações (novas edições), como as classes e os métodos são alterados e como a aplicação é versionada e liberada para o resto da equipe de desenvolvedores.

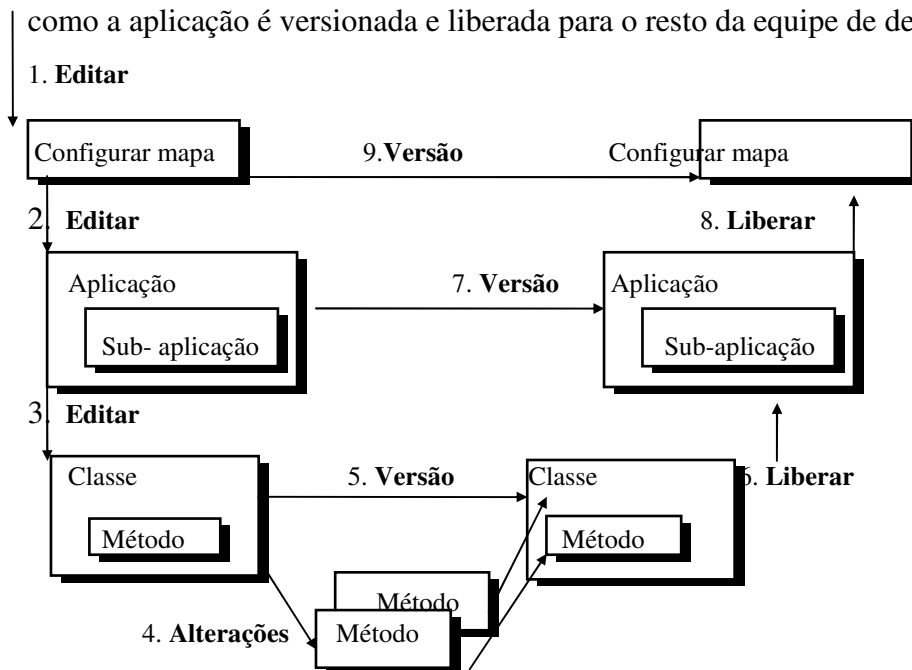


Figura 2 - Edições e versões

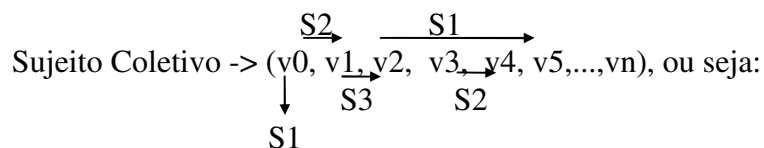
3 Análise lógico-operatória do Ambiente de Desenvolvimento Cooperativo de Programação ENVY/400

Em primeiro lugar, é preciso caracterizar a operação individual e/ou coletiva e o sujeito coletivo nesta ferramenta em particular, para depois realizar a análise lógico-operatória da mesma. Quando se fala em programação em equipe, estamos nos referindo

às *co-operações* (Piaget, 1973), isto é, operações realizadas em conjunto, visando um objetivo comum que, neste caso, é o desenvolvimento de programas. Neste sistema, a operação coletiva nada mais é do que a combinação de operações individuais. Assim, pode-se dizer que todas as operações realizadas pela equipe de desenvolvedores são do tipo individual, mas todas as ações refletem sobre o coletivo no momento em que os desenvolvedores têm que compartilhar e liberar (valor) tudo o que eles estão fazendo de forma imediata. O produto final é cooperativo porque todos irão utilizar nos seus componentes de software elementos desenvolvidos pela equipe. A operação coletiva diz respeito à criação de versões de uma representação coletiva. A criação de versões é o versionamento, e este pode ser feito tanto pelo sujeito individual quanto pelo sujeito coletivo. O sujeito individual pode criar diferentes versões, por exemplo: $S_1 (v_0, v_1, v_2, v_3, \dots, v_n)$; $S_2 (v_0, v_1, v_2, v_3, \dots, v_n)$; $S_3 (v_0, v_1, v_2, v_3, \dots, v_n)$, onde: S_1 : sujeito₁; S_2 : sujeito₂; S_3 : sujeito₃; v_0 : versão 0; v_1 : versão 1; v_2 : versão 2; v_3 : versão 3; v_n : versão n. Por outro lado, o grupo de desenvolvedores (sujeito coletivo) estará criando diferentes versões, portanto:

$$SC: \left(\begin{array}{l} S_1 (v_0, v_1, v_2, v_3, \dots, v_n) \\ S_2 (v_0, v_1, v_2, v_3, \dots, v_n) \\ S_3 (v_0, v_1, v_2, v_3, \dots, v_n), \dots \end{array} \right) \text{ então:}$$

Sujeito Coletivo: $\{[v_0, S_1], [v_0, S_2], [v_0, S_3], [v_1, S_1], [v_1, S_2], [v_1, S_3], [v_2, S_1], \dots\}$, ou seja, o sujeito coletivo é a combinação dos diferentes sujeitos individuais e das versões criadas por eles. Este nada mais é do que a biblioteca compartilhada. Sendo assim, as versões individuais são criadas a partir da biblioteca compartilhada e, portanto, chamam o sujeito coletivo. Ou seja, este é criado a partir do individual e vice-versa. O sujeito coletivo também pode ser visto como a combinação das ações individuais da seguinte forma:



O S_1 desenvolve a versão 0 (v_0), o S_2 carrega a versão 0 e cria a partir dela a versão 1 (v_1), o S_3 pega a versão 1 criada pelo S_2 e desenvolve a versão 2 (v_2), o S_1 pega a versão 2 e cria a versão 5 (v_5) e o S_2 cria, a partir da versão 3 (v_3), a versão 4 (v_4). Com isso, pretende-se mostrar que as operações individuais nada mais são do que operações realizadas em cima de operações, ou seja, *co-operações* (Piaget, 1973). A

ação cooperativa está presente continuamente no momento em que um sujeito individual utiliza uma versão que já se encontra na biblioteca compartilhada e, portanto, no sujeito coletivo, para criar uma nova.

O próximo passo é identificar na ferramenta ENVY/400 algumas operações lógicas e/ou infralógicas, do ponto de vista da lógica operatória piagetiana. Estas operações foram retiradas de (Piaget, 1973) e (Piaget & Inhelder, 1976, 1993) e se encontram explicitadas em (Behar, 1998).

1) *Operação infralógica de adição e partição primitiva*: Falamos de uma operação infralógica porque um componente de software (todo) é formado de partes (mapas, aplicações, sub-aplicações, classes, métodos,...) que dependem umas das outras. Esta operação pode ser identificada no ambiente ENVY/400 na criação de novos componentes. Ela descreve as operações requeridas para a criação de novas instâncias de componentes de software armazenadas no gerenciador/ENVY. Por exemplo, na *configuração de mapas*, é preciso ativar o menu *Create Configuration Map*, digitar o nome do novo mapa e uma nova configuração de mapa será criada. A nova edição é gerenciada pela pessoa que a criou. Quando o mapa for criado, a edição inicial contém todas as aplicações que foram carregadas na imagem. Como pode ser observado, este caso de operação trata de um nome a ser dado à representação de um objeto que, neste caso, é o mapa a ser configurado. Portanto, é uma *operação individual de manipulação direta da representação* (MDR-mapa), como definido em (Behar & Costa, 1997b). O próximo passo é ativar os itens do menu *Add New Application..., Release Other...* e *Delete Application*, para modificar o mapa editado a fim de que contenha as aplicações desejadas. É possível adicionar aplicações no mapa que não são carregadas na imagem. Logo, é uma *operação elementar infralógica de adição primitiva*, pois será adicionada ao mapa uma aplicação. Portanto, o mapa (todo) já existe com suas aplicações (partes), e será colocada mais uma, para depois liberá-la ou até deletar uma aplicação. Portanto: $C: \text{todo} = \text{mapa}; B: \text{partes} = \text{aplicações}; B': \text{aplicação adicionada}$, logo: $B + B' = C$. Se for utilizada a operação *Delete Application* para subtrair uma aplicação da lista: $C - B' = B$ (novo mapa, somente com o conjunto de aplicações contido em B) \Rightarrow *operação infralógica de partição primitiva*.

2) *Operação infralógica de ordem de colocação*: Para criar uma aplicação, ativar o menu *Applications* \Rightarrow *Create* e digitar o nome da nova aplicação. Será aberta a opção de *PrerequisitesDialog*, que permite que o usuário defina os pré-requisitos para uma

nova aplicação. Assim, uma nova aplicação será criada. Esta nova edição é gerenciada pela pessoa que a criou; portanto, percebe-se uma relação de *poder* sobre o objeto que é o valor sobre a aplicação. O valor poder é uma *operação coletiva* porque diz respeito a um grupo. Os pré-requisitos descrevem a situação na qual um componente deve existir em uma definição do sistema antes de vir um outro. Por exemplo: aplicações definem pré-requisitos de aplicações, indicando que esses pré-requisitos são requeridos pela aplicação antes desta funcionar. Portanto, está se referindo à criação de uma aplicação que deverá seguir uma ordem para ser criada e poder funcionar. Neste caso, é uma *operação de manipulação indireta da representação* (aplicação), onde esta é criada através do nome que se dá a ela. Estamos frente a uma *operação infralógica de ordem de colocação* porque os pré-requisitos se referem a uma ordem que tem que ser seguida pelos elementos do sistema para que a aplicação possa funcionar.

3) *Operação lógica de correspondência*: Para criar um método da *ApplicationsBrowser*, é preciso selecionar a aplicação à qual se deseja adicionar o método, selecionar a classe a que se deseja adicionar o método, selecionar a opção *New Method Template* e, na janela de texto, editar o texto que corresponde ao ambiente do método desejado. Depois salvar (*Save*) o texto do menu. Selecionar uma classe (todo) onde será adicionado o método (parte) significa que, além de ser uma *operação infralógica de adição primitiva*, deverá ser escrito um texto seguindo certas regras sintáticas da linguagem computacional. Portanto, deverá existir uma *correspondência* e uma ordem de colocação dos elementos da linguagem.

4) *Operação com valores*: Para criar uma nova edição de uma sub-aplicação, deve ser selecionada a versão da sub-aplicação de onde será copiada a nova edição. Novas edições somente podem ser criadas de versões, devendo ser selecionado do menu a opção *Create New Edition*. A nova edição conterà a mesma versão de classe que a versão de onde foi criada. É recomendado que quando for criada uma nova edição de sub-aplicação, que esta seja liberada imediatamente para a edição de aplicação que fará parte dela. Esta nova edição de sub-aplicação tem que seguir uma *regra coletiva* (regra no sentido descrito em (Behar & Costa, 1997b)) do ambiente, que é a *liberação imediata* para ser utilizada também por outros programadores.

5) *Operação lógica inversa: Recuperando imagens “crashes”*: Quando as alterações são perdidas, por não terem sido salvas ou porque ocorreu algum erro inesperado no sistema, todas as modificações podem ser recuperadas. Em todos os casos de

recuperação de aplicações, classes e métodos, trata-se da *operação lógica inversa*, pois estará havendo compensação de algo que foi perdido, revertendo a ação anterior.

4 Conclusões

O objetivo deste trabalho foi dar ênfase à possibilidade de análise, a nível lógico-operatório, de um ambiente computacional cooperativo. Para isso, foi necessário fazer a integração dos conhecimentos da lógica operatória piagetiana (Piaget, 1972), (Piaget & Inhelder, 1976) e (Castorina & Palau, 1982) com a Computação Cooperativa ou Computer Supported Cooperative Work (CSCW) (Ellis et al., 1993), (Greenberg, 1991) e (Kirsche et al., 1993). Também deve ser destacado o processo de aplicação do modelo de análise lógico-operatória do sujeito coletivo em relação ao ENVY/400. A partir disso, foi possível caracterizar o “objeto” operatoriamente, da mesma forma que pôde ser realizado com o sujeito (Behar et al., 1995). Cabe destacar que a leitura do ambiente cooperativo em questão, em termos operatórios, não esgota todas as possibilidades de análise; tampouco foram abordadas exhaustivamente todas as operações lógicas e/ou infralógicas do mesmo. A aplicação e análise da ferramenta estão baseadas nos modelos descritos em (Behar & Costa, 1997a).

Foi desenvolvido, portanto, um instrumento que poderá servir, sob a ótica do sujeito, como uma contribuição para os educadores e pesquisadores em geral, já que estes poderão utilizá-lo como guia para auxiliar no desenvolvimento lógico-operatório de sujeitos envolvidos em experiências através de ferramentas computacionais de uso cooperativo.

Por outro lado, do ponto de vista do objeto, este instrumento poderá ser utilizado para avaliar softwares e groupwares, de acordo com as operações lógicas e infralógicas. Além disso, o projeto e desenvolvimento de ferramentas computacionais poderá ser baseado na lógica operatória piagetiana.

O resultado deste trabalho (Behar, 1998) deu origem a um novo projeto de pesquisa que está sendo desenvolvido a partir deste ano (Behar & Costa, 1998b).

5 Referências Bibliográficas

- Behar, Patricia Alejandra (1998). *Análise operatória de ferramentas computacionais de uso individual e cooperativo. Tese de doutorado*. Porto Alegre: Curso de Pós-Graduação em Ciência da Computação - CPGCC - da Universidade Federal do Rio Grande do Sul - UFRGS, 1998.
- Behar, Patricia A; Costa, Antônio C.R. (1997a). *Base models for organizing the logical-operatory analysis of cooperative computational environments*. In: Ed-Media & Ed-Telecom 97- World Conference on Educational Multimedia and Hypermedia and World Conferences on Educational Telecommunications, 1997, Calgary, Canadá. *Proceedings...*Calgary: University of Calgary.
- Behar, Patricia A; Costa, Antônio C.R. (1997b). *Caracterização operatória do processo interativo de um sujeito coletivo com ferramentas computacionais cooperativas*. In: Simpósio Brasileiro de Informática na Educação, 8., 1997, São José dos Campos, São Paulo. *Anais...* São José dos Campos: ITA, 1997.
- Behar, Patricia A; Costa, Antônio C.R. (1998a). *Análise lógico-operatória do Microsoft Netmeeting 2.0 Beta 4*. In: IV Congresso Ibero-Americano de Informática na Educação, 1998, Brasília, Brasil. *Anais...* Brasília: RIBIE, 1998.
- Behar, Patricia A; Costa, Antônio C.R. (1998b). *Método de análise operatória de ferramentas computacionais de uso individual e cooperativo*. Projeto de pesquisa UFRGS/CNPq. Porto Alegre: Faculdade de Educação/UFRGS; Pelotas: Escola de Informática/UCPel, 1998.
- Behar, Patricia A.; Ceron, Maria T.; Costa, Antonio C. R. (1995). *Uma Reconstrução do Método Piagetiano de Análise Lógico-Operatória de Experiências*. In: Congresso Internacional Logo, 7., e Congresso de Informática Educativa do Mercosul, 1., 1995, Porto Alegre, RS. *Anais...* Porto Alegre: UFRGS.
- Castorina, José Antonio; Palau, Gladys Dora. (1982). *Introducción a la lógica operatória de Piaget-Alcances y significado para la psicología genética*. Buenos Aires: Ediciones Paidós.
- Ellis, C.A. et al. (1993). *Groupware: some issues and experiences*. In: *Groupware and Computer-Supported Cooperative Work*, Baecker, Ronald. USA: Morgan Kaufmann Publishers.
- Greenberg, Saul. (1991). *Personalizable groupware: Acomodating individual roles and group differences*. In: European Conference of Computer Supported Cooperative Work - ECSCW'91-, 1991, Amsterdam, Holanda. *Proceedings...* Amsterdam: Kluwer Press.
- IBM, Envy/400. (1993). *Developer Guide & ENVY/400 User's Guide*, v. 2, release 2, modification 0. USA: IBM.
- Kirsche, T. et al. (1993). *Communication Support for Cooperative Work*. *Computer Communications*, v. 16, n. 9.
- Piaget, J. (1958). *Psicologia da Inteligência*. Trad: Egléa de Alencar. Rio de Janeiro: Fundo de Cultura.
- Piaget, J. (1971). *A formação do símbolo na criança*. Rio de Janeiro: Zahar.
- Piaget, J. (1972). *Ensaio da lógica operatória*. São Paulo: Ed. da USP.

Piaget, J. (1973). *Estudos Sociológicos*. Rio de Janeiro: Forense.

Piaget, J. (1983). *Psicologia da Inteligência*. Rio de Janeiro: Zahar.

Piaget, J; Inhelder, B. (1976). *Da lógica da criança à lógica do adolescente*. São Paulo: Pioneira.

Piaget, J & Inhelder, B. (1993). *A representação do espaço na criança*. Porto Alegre: Artes Médicas.