

Modelagem Adaptativa de Aplicações Complexas

Almir Rogério Camolesi^{1,2}, João José Neto¹

⁽¹⁾ Departamento de Engenharia de Computação e Sistemas Digitais (PCS),
Escola Politécnica (POLI), Universidade de São Paulo (USP),
Av. Luciano Gualberto, Trav. 3, nº 380 - 05.508-900 - São Paulo - SP – Brasil

⁽²⁾ Coordenadoria de Informática, Instituto Municipal de Ensino Superior de Assis (IMESA),
Fundação Educacional do Município de Assis (FEMA)
Av. Getúlio Vargas, 1200 – 19.807-634 – Assis – SP - Brasil

almir.camolesi, joao.jose@poli.usp.br

Abstract

This paper presents the ISDL-Adp model and its adaptive actions. This model is the result of extending the concepts of non-adaptive ISDL devices by using the concepts of adaptive devices. This paper shows the basic structure of adaptive actions and its use in modeling of behavior of distributed adaptive system.

Key words: non adaptive devices, adaptive devices, ISDL, ISDL-Adp, system modeling.

Resumo

Este artigo apresenta a estrutura do modelo ISDL-Adp e de suas ações adaptativas. Tal modelo é fruto da extensão do modelo não-adaptativo ISDL aos conceitos de mecanismos adaptativos. Neste trabalho serão apresentadas a estrutura básica das funções adaptativas e a sua utilização na modelagem de sistemas distribuídos adaptativos.

Palavras-chave: dispositivos não-adaptativos, dispositivos adaptativos, ISDL, ISDL-Adp, modelagem de sistemas.

1. Introdução

Ao desenvolver uma aplicação os projetistas não desejam construir sistemas complexos, mas devido ao grande número de funcionalidades que um sistema deve desempenhar e as necessidades requeridas pelos usuários a maioria das aplicações iniciam com um ciclo de vida contendo um mínimo de funcionalidades requeridas para execução. Entretanto muitos softwares estão acoplados a ambientes complexos e, mais importante ainda, à ambientes que incluem seres humanos. A introdução de novos sistemas de software e as mudanças no ambiente de operação criando mudanças recíprocas nas interações do sistema tornam o desenvolvimento das aplicações cada dia mais complexo. Todos estes fatores motivam os projetistas a buscarem técnicas que facilitem o projeto de tais aplicações. Muito trabalho vem sendo realizado nos últimos anos em buscas de técnicas e ferramentas para o auxílio aos projetistas no desenvolvimento de suas aplicações. Dentre os modelos empregados pode-se citar metodologias e técnicas como *UML* [1], *Statecharts* [2] e Técnicas de Descrição Formal (*TDFs*) como *LOTOS* [3], *Estelle* [4], *SDL* [5], etc..

Na década de 90 um grupo de pesquisadores iniciou um estudo buscando algumas deficiências em relação a aplicações de TDF no projeto de sistemas, gerando ao mesmo tempo um conhecimento mais profundo das necessidades de modelagem de sistemas distribuídos. Fundamentado nos estudos realizados desenvolveram um conjunto de conceitos arquitetônicos [6] que formaram a base conceitual do modelo *Interaction System Design Language (ISDL)* [7] para suportar o projeto de sistemas telemáticos, como, por exemplo, serviços e protocolos do modelo *OSI*. Atualmente este modelo é utilizado em diversas áreas de conhecimento, como sistemas distribuídos, telemáticos e reengenharia de processos de negócios.

Outros trabalhos focados na especificação de aplicações que tem o seu comportamento modificado durante o tempo de execução foram desenvolvidos. Em [8] é apresentado um mecanismo fundamentado nos conceitos básicos de Autômatos Finitos. Em [9] foi desenvolvida uma ferramenta para especificação de aplicações dinâmicas utilizando a técnica de *Statechart* como base. Atualmente pode-se citar trabalhos como [10] e [11], que buscam o desenvolvimento de técnicas e metodologias para o desenvolvimento de aplicações dinâmicas e reconfiguráveis.

Neste contexto é apresentado o modelo *Adaptive Interaction System Design Language (ISDL-Adp)* que tem por objetivo a extensão do Modelo ISDL aos conceitos de mecanismos adaptativos visando a especificação do comportamento de sistemas distribuídos complexos que possuem características reconfiguráveis presentes em seu comportamento. Tal extensão fundamenta-se na estrutura básica do Modelo ISDL [7] estendida aos conceitos de Mecanismos Adaptativos proposto em [12]. Este artigo encontra-se estruturado da seguinte forma: a seção 2 ilustra os conceitos básicos do Modelo ISDL; na seção 3 é descrita a estrutura do modelo ISDL-Adp e de suas

funções adaptativas, na seção 4 é realizado um exemplo ilustrativo e, por fim, na seção 5, são tecidas algumas conclusões e trabalhos futuros.

2. Modelo não adaptativo ISDL

O principal aspecto a ser considerado no modelo formal desenvolvido é a possibilidade de representação das possíveis mudanças que ocorrem em um determinado comportamento durante o tempo de execução. Tais mudanças podem ocorrer no conjunto de atividades desempenhado por uma aplicação, no conjunto de dados manipulados, no tempo em que uma determinada atividade é executada ou na localização de ocorrência da mesma. Como modelo subjacente ao mecanismo proposto foi utilizado o *Interaction System Design Language* (ISDL).

ISDL é um modelo genérico que foi desenvolvido para modelagem de diversos tipos de sistemas distribuídos, como processos de negócios, aplicações telemáticas e redes de comunicação. Tal modelo fundamenta-se na busca de minimização de limitações existentes em métodos formais [13]. A modelagem de um sistema em ISDL consiste basicamente de dois modelos: um modelo de entidades e um modelo de comportamentos.

O modelo de entidade representa quais partes do sistema que são consideradas, e como elas são interconectadas. Dois conceitos são usados em um modelo de entidade: entidade e pontos de interação. Uma entidade representa um sistema (parte) que desempenha alguma função ou comportamento, como, por exemplo, um componente de software ou departamento de uma empresa. Um ponto de interação representa um mecanismo que possibilita que uma entidade possa interagir com outras entidades, como, por exemplo, um cliente de correio eletrônico.

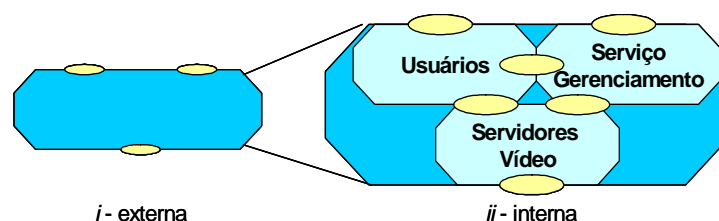


Figura 1 - Exemplo de Modelo de Entidade.

De uma perspectiva externa, um sistema é modelado por uma simples entidade, tendo um ou mais pontos de interação. A Figura 1*i*, por exemplo, representa um modelo de entidade de sistema de vídeo sob demanda (VoD) [17] onde as interações ocorrem com o sistema por meio de três pontos de interação. Estes pontos de interação poderiam, por exemplo, representar, as requisições de serviços realizadas por um usuário, as interações realizadas pelo administrador do sistema e os serviços de manutenção de mídias. Uma entidade é graficamente representada por um retângulo com os cantos entrecortados. Um ponto de interação é graficamente representado por uma elipse sobreposta às entidades que compartilham (eg., via de comunicação) os pontos de interação.

De uma perspectiva interna, um sistema é modelado com uma composição de partes funcionais. Por exemplo, estas partes podem representar sub componentes ou departamentos de uma empresa. A perspectiva interna do VoD é apresentada na Figura 1*ii*. Esta figura ilustra o VoD dividido em três partes: um serviço genérico do usuário, um provedor que notifica os usuários sobre os serviços que estão disponíveis: autenticação, mídias disponíveis, etc..., e um serviço provedor de mídias que são requisitadas pelos usuários.

O modelo de comportamento representa o comportamento, ou funcionalidade, de cada entidade em um correspondente modelo de entidade. Três conceitos são usados: ação, interação e relações de causalidade. Uma ação representa alguma atividade executada por uma entidade simples. Considere por exemplo, a ação do serviço provedor de mídias que é responsável pela entrega das mídias aos usuários do serviço. Uma interação representa uma atividade executada por duas (ou mais) entidades. Uma contribuição de interação representa a participação de uma entidade individual em uma atividade comum. Considere, por exemplo, a interação entre o serviço de requisição de usuários e a apresentação de uma lista de filmes (mídias) disponíveis para apresentação.

Fundamentado nos conceitos apresentados em [12] o conjunto de conceitos arquitetônicos do modelo ISDL constituem um mecanismo não-adaptativo dirigido por regras cujo comportamento depende exclusivamente de um conjunto finito de regras que determinam, para cada possível configuração corrente do mecanismo, a sua próxima configuração. Desta forma, o mecanismo não-adaptativo ISDL é descrito como sendo uma sêxtupla, e é representado formalmente por $ISDL = (Ac, RC, \Sigma, c_0, A, NA)$, onde:

- ISDL é um mecanismo não-adaptativo dirigido por regras cuja operação define um comportamento constituído de um conjunto de Relações de Causalidades RC;
- Ac é o conjunto de todas as possíveis ocorrências de ações, e $c_0 \in Ac$ determina o conjunto de ações do comportamento inicial. Os possíveis comportamentos alcançados são representados por intermédio da conjunção cruzada de suas execuções parciais. A conjunção cruzada e as execuções parciais são denotadas formalmente por \otimes e $e\chi$. Vamos supor a conjunção cruzada de dois comportamentos parciais $EE1$ e $EE2$ que constituem o comportamento EE . EE consiste de todas as possíveis execuções $e\chi$, sendo que $e\chi$ é a conjunção de duas execuções compatíveis $e\chi1$ e $e\chi2$, com $e\chi1 \in EE1$ e $e\chi2 \in EE2$. Conseqüentemente, o comportamento EE consiste de todas as possíveis construções alternativas que satisfaçam ao mesmo tempo

uma (sub-) construção de $EE1$ e uma (sub-) construção de $EE2$.

- ε denota a cadeia vazia, que representa o elemento neutro do conjunto a que pertence, em relação à operação de concatenação;
- Σ é o conjunto finito de todos os possíveis eventos que formam as cadeias de entrada válidas para ISDL, com $\varepsilon \in \Sigma$;
- $A \subseteq Ac$ é o conjunto de ocorrências de ações;
- $F = Ac - A$ é o subconjunto das não ocorrências de ações;
- $w = w_1 \dots w_n$ é uma cadeia de entrada, onde $w_k \in \Sigma - \{\varepsilon\}$, $k = 1, \dots, n$ com $n \geq 0$;
- NA é um conjunto finito (com $\varepsilon \in NA$) de todos os possíveis símbolos gerados como saída do mecanismo ISDL. Na prática, os símbolos de saída contidos em NA podem ser obtidos por meio de chamadas de procedimento, assim uma saída gerada por uma aplicação de qualquer relação de causalidade básica pode ser interpretada como chamada do procedimento correspondente;
- RC é um comportamento definido por um conjunto de ações e interações de comportamentos e suas relações de causalidades. Um comportamento representa um conjunto de atividades que a entidade (conceito abstrato que modela um sistema, ou parte de um sistema) pode executar. O conceito de ação foi introduzido para representar uma atividade executada por um único sistema em um dado nível de abstração.

Atributos de informação, tempo e localização podem ser adicionados a uma (inter)ação para modelar, respectivamente, os resultados estabelecidos por alguma atividade, num determinado instante de tempo, em uma determinada localização. A ocorrência de uma (inter)ação representa o término com sucesso de uma atividade. Uma inter(ação) é atômica desde que a mesma ocorra, algum resultado é estabelecido e torna-se disponível em um determinado tempo numa determinada localidade para todas as entidades envolvidas na atividade. Caso contrário, nenhum resultado é estabelecido e nenhuma atividade pode se referir a qualquer resultado intermediário de uma atividade.

Uma ação é graficamente ilustrada por um círculo (ou elipse). Uma contribuição de interação é graficamente representada por um segmento de círculo (ou elipse), que reflete que múltiplas entidades contribuem para a interação. Os atributos de informação (ι), de tempo (τ) e de localização (λ) são representados dentro de caixas-texto anexadas às (inter)ações. Construções podem ser definidas nas possíveis saídas dos valores de ι , τ e λ . No caso de uma interação, cada contribuição de interação define a construção da correspondente entidade, cada uma com os respectivos valores de ι , τ e λ devem satisfazer todas as construções das entidades envolvidas, caso contrário a interação não pode ocorrer. Também são permitidos múltiplos valores para algum atributo, e uma escolha não determinística entre estes valores é assumida.

Desta forma, o conjunto de ações e interações que definem um comportamento ISDL é dado por uma relação de causalidade:

$$\langle\langle a, I, T, \Lambda, \zeta \rangle, \Gamma, \upsilon, \langle I\text{-Refs}, T\text{-Refs}, L\text{-Refs}, ITL\text{Refs} \rangle, \langle I\text{-Caus}, T\text{-Caus}, L\text{-Caus}, ITL\text{-Caus} \rangle\rangle$$

onde:

- $a \in A$ é o nome da ação, que a identifica unicamente no sistema, e faz parte de A (conjunto de ocorrências de ações);
- I, T, Λ são, respectivamente, os valores dos domínios de informação, tempo e localização da ação a ;
- $\zeta \subseteq I \times T \times \Lambda$ é a combinação dos valores dos domínios da ação a ;
- $\Gamma \in CC$ é a condição de causalidade de a e faz parte de CC (conjunto de condições de causalidades disjuntivas);
- $I\text{-Refs}, T\text{-Refs}, L\text{-Refs}, ITL\text{Refs}$ são, respectivamente, os conjuntos dos atributos de informação, tempo, localização e de combinação destes atributos;
- $I\text{-Caus}, T\text{-Caus}, L\text{-Caus}, ITL\text{-Caus}$ são, respectivamente, os conjuntos de relações de causalidade de informação, de tempo, de localização e de combinações destes atributos;

Uma relação de causalidade é associada com cada (inter)ação, modelando a condição para esta inter(ação) ocorrer. Três condições básicas para a ocorrência de uma ação a são identificadas: $b \rightarrow a$; a ação b deve ocorrer antes da ação a ; $\neg b \rightarrow a$; a ação b não deve ocorrer antes, nem simultaneamente com a ação a ; e $\sqrt{\quad} \rightarrow a$; a ação a é habilitada a ocorrer.

O operador e (\wedge) e o operador ou (\vee) podem ser usados para modelar condições de causalidades complexas. Por exemplo, $b \vee \neg c \rightarrow a$ representa que a ação a ocorre depois da ação b ter ocorrido e a ação c ainda não ter ocorrido. Além disso, um atributo de probabilidade para cada condição pode ser usado para representar a probabilidade da inter(ação) acontecer quando a condição é satisfeita.

O conceito de relação de causalidade permite a modelagem de muitas diferentes relações entre ações. Isto é conveniente para representar estes relacionamentos diretamente, ao invés da composição de relações de causalidade de ações individuais. A Figura 2 ilustra graficamente algumas relações comuns entre duas ou três ações. Os

operadores \wedge e \vee são graficamente representados, respectivamente, pelos símbolos \blacksquare e \square . A condição inicial $\sqrt{}$ é representada por uma seta com nenhuma ação anterior ligada a mesma.

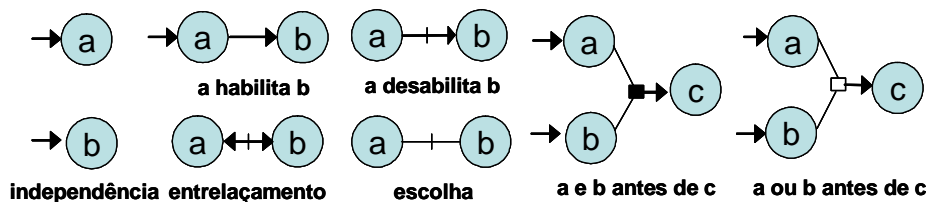


Figura 2 - Algumas ações e relações de causalidade.

O modelo ISDL suporta duas técnicas ortogonais para estruturação de comportamentos por meio da composição de subcomportamentos menores e mais simples: estrutura orientada a causalidade e estrutura orientada a restrição. A estrutura de comportamentos orientada a causalidade é fundamentada na decomposição de uma relação de causalidade em uma construção sintática, que permite definir uma ação e suas relações de causalidades em diferentes subcomportamentos.

A estrutura de comportamentos orientada a restrição é fundamentada na decomposição de uma ação em uma ou mais interações, que permite que uma ação seja definida para um comportamento como uma composição de interações de sub-comportamentos. Esta técnica pode ser usada para decompor condições e construções complexas na execução de uma ação em simples sub-condições e sub-construções que são assumidas para contribuições de interações definidas em sub-comportamentos distintos. Além disso, a estrutura orientada a restrição é necessária para estruturar um comportamento em sub-comportamentos de forma que o mesmo possa ser atribuído a diferentes entidades, desde que as entidades possam se comunicar por meio de interações.

A Figura 3 mostra a modelagem do acesso, da requisição de serviços e do gerenciamento de requisições em que ambas às técnicas de estruturação de comportamento são usadas. Por meio da estrutura orientada à restrição é modelado o acesso do usuário ao sistema, o envio e recebimento de requisições e a execução dos serviços solicitados. Na referida modelagem o usuário realiza a solicitação de acesso ao sistema por meio da interação *Login*. O sistema após receber e validar a requisição de acesso habilita o usuário a fazer requisições ao subcomportamento *Gerenciamento Serviços* que os disponibiliza aos usuários (ex: Fim Serviço). Utilizando-se da estruturação de comportamento orientada a causalidade foi modelada as interações existentes entre os *Usuários* e os subcomportamentos *Serviço Autenticação* e *Serviço Gerenciamento*. Enquanto que a estruturação orientada a causalidade foi utilizada para representar a habilitação da interação *Requisição* do subcomportamento *Serviço Gerenciamento* por meio de pontos de entrada e saída.

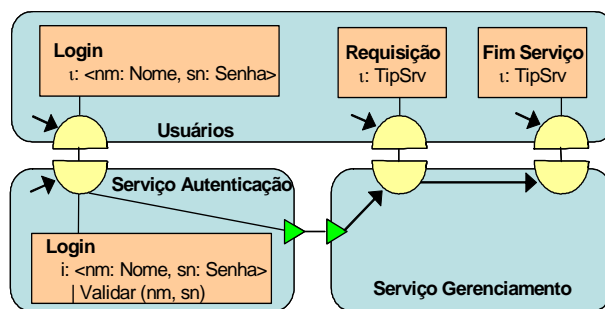


Figura 3 - Exemplo de Estruturação de Comportamento.

Em [14] uma ferramenta gráfica para AMBER, um dialeto de ISDL constituído para reengenharia de processos, foi desenvolvido e o Projeto Friends [15] tem adaptado e estendido esta ferramenta para suportar a modelagem de componentes de software e suas composições para sistemas telemáticos.

3. Modelo Adaptativo ISDL-Adp

A arquitetura básica do modelo *Adaptive Interaction System Design Language* (ISDL-Adp) [16] representada na Figura 4 constitui-se de um núcleo não-adaptativo ISDL [7], envolvido por uma camada de ações adaptativas que tem por finalidade a realização de mudanças das características do sistema representado. A camada adaptativa é formada por um conjunto de ações adaptativas anteriores e posteriores semelhantes as ações definidas para os Autômatos Adaptativos [8]. As ações adaptativas têm por objetivo realizar mudanças no comportamento da especificação ISDL.

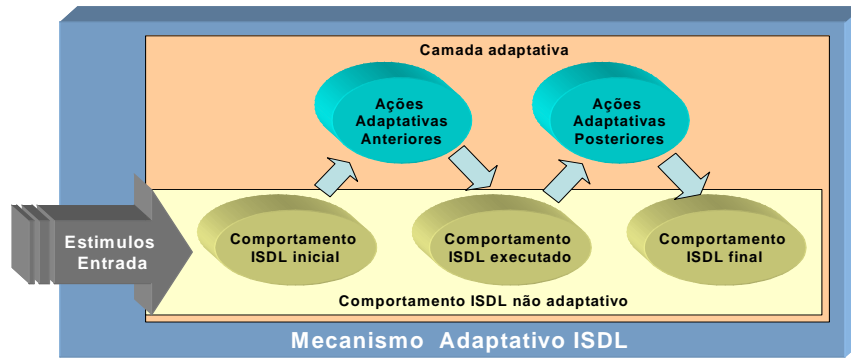


Figura 4 - Mecanismo Adaptativo ISDL.

Tomando-se como base o mecanismo adaptativo geral apresentado em [12], pode-se definir o modelo ISDL-Adp iniciando sua operação na configuração inicial c_0 na forma $ISDL-Adp_0 = (C_0, AR_0, \Sigma, c_0, A, NA, BA, AA)$. No passo $k \geq 0$, um estímulo de entrada movimentada ISDL-Adp para uma próxima configuração e então prossegue sua operação no passo $k+1$ se e somente se uma ação adaptativa não vazia for executada. Assim, estando ISDL-Adp em seu passo k , na forma $ISDL-Adp_k = (C_k, ARC_k, \Sigma, c_k, A, NA, BA, AA)$, a execução de uma ação adaptativa não-vazia o leva à forma $ISDL-Adp_{k+1} = (C_{k+1}, AR_{k+1}, \Sigma, c_{k+1}, A, NA, BA, AA)$. Nesta formulação:

- $ISDL-Adp = (ISDL_0, AM)$ é um mecanismo adaptativo formado por um mecanismo inicial subjacente $ISDL_0$ e um mecanismo adaptativo AM ;
- $ISDL$ é o modelo subjacente não-adaptativo, cujo funcionamento foi descrito na seção 2, no passo k . $ISDL_0$ é o mecanismo subjacente, definido no conjunto inicial RC_0 (conjunto de ações ou relações de causalidades representando um comportamento não-adaptativo). Por definição, qualquer ação ou relação de causalidade não adaptativa em RC_k tem sua regra correspondente adaptativa em ARC_k ;
- C_k é o conjunto de todos os possíveis comportamentos de ISDL no passo k , e $c_k \in C_k$ é o seu comportamento inicial no passo k . Para $k=0$, tem-se, respectivamente, C_0 , o conjunto inicial de ações e relações de causalidade válidas e $c_0 \in C_0$, a configuração inicial de $ISDL_0$ e de $ISDL$.
- ε (“cadeia vazia”) denota a ausência de qualquer outro elemento válido do conjunto correspondente;
- Σ é o conjunto (finito, fixo) de todos os possíveis eventos (inclusive o evento vazio ε) de que se compõe a cadeia de entrada AD ;
- $A \subseteq C$ é o subconjunto de ações e relações de causalidade de aceitação do comportamento ISDL;
- $F = C - A$ é o conjunto das configurações de rejeição, ou seja, é o conjunto resultante da subtração das configurações de aceitação do conjunto de todas as possíveis configurações de comportamento;
- BA e AA são conjuntos de ações adaptativas. Ambas incluem a ação vazia ($\varepsilon \in BA \cap AA$)
- $w = w_1 w_2 \dots w_n$, é a cadeia de entrada, onde $w_k \in \Sigma - \{\varepsilon\}$, $k = 1, \dots, n$ com $n \geq 0$;
- NA , com $\varepsilon \in NA$, é um conjunto (finito, fixo) de todos os símbolos que podem ser gerados como saídas por ISDL-Adp, em resposta à aplicação de regras (ações e relações) adaptativas. Analogamente ao que ocorre com os mecanismos não adaptativos, a cadeia de saída assim obtida pode (se for conveniente) ser interpretada como uma sucessão de chamadas dos procedimentos correspondentes;
- ARC_k é o conjunto das regras adaptativas que definem o comportamento adaptativo de ISDL-Adp no passo K e é dado por uma relação $ARC_k \subseteq BA \times C \times \Sigma \times C \times ISDL \times AA$.

Em particular, ARC_0 define o comportamento inicial de ISDL-Adp. Ações adaptativas modificam o comportamento adaptativo corrente ARC_k de ISDL-Adp para um novo comportamento ARC_{k+1} adicionando e/ou eliminando inter(ações) e relações de causalidade em ARC_k . Regras $arc \in ARC_k$ são da forma:

$\langle\langle ba \rangle, \langle\langle a, I, T, \Lambda, \zeta \rangle, \Gamma, \upsilon, \langle I-Refs, T-Refs, L-Refs, ITLRefs \rangle, \langle I-Caus, T-Caus, L-Caus, ITL-Caus \rangle\rangle, \langle aa \rangle\rangle$

significando que, em resposta a algum símbolo da cadeia de entrada $\sigma \in \Sigma$, arc inicialmente executa a ação adaptativa $ba \in BA$. Se a execução de ba eliminar ar de ARC_k , a execução de arc é abortada; caso contrário, aplica-se a regra subjacente não-adaptativa:

- $arc = \langle\langle a, I, T, \Lambda, \zeta \rangle, \Gamma, \upsilon, \langle I-Refs, T-Refs, L-Refs, ITLRefs \rangle, \langle I-Caus, T-Caus, L-Caus, ITL-Caus \rangle\rangle \in RC_k$, conforme descrito anteriormente; e finalmente, executa-se a ação adaptativa $aa \in AA$.

Define-se ARC como o conjunto de todas as possíveis regras adaptativas para ISDL-Adp;

Define-se RC como o conjunto de todas as possíveis ações e relações de causalidade não-adaptativas para ISDL-Adp;

- $AM \subseteq BA \times RC \times AA$, definido para um particular mecanismo adaptativo ISDL-Adp, é um mecanismo adaptativo aplicado a todas as regras no passo k em $RC_k \subseteq RC$. AM deve ser interpretado da mesma forma como se fosse aplicada a qualquer sub domínio $RC_k \subseteq RC$. Isso determinará um único par de ações adaptativas associadas a cada regra não adaptativa.

O conjunto $ARC_k \subseteq ARC$ pode ser obtido colecionando-se todas as regras adaptativas construídas pela associação

de cada par de ações adaptativas às correspondentes regras não-adaptativas em RC_k .

O comportamento não-adaptativo $ISDL_i$, em cada passo de execução do mecanismo adaptativo, constitui o modelo ISDL-Adp por meio de suas ações, suas condições de causalidade básica, das relações entre ações. Uma ação ISDL-Adp pode ser representada por:

$\langle\langle a, I, T, \Lambda, \zeta \rangle : \mathcal{A} \rangle \rightarrow \langle\langle a', \Gamma, T', \Lambda', \zeta' \rangle, \Gamma', v', \langle I\text{-Refs}', T\text{-Refs}', L\text{-Refs}', ITLRefs' \rangle, \langle I\text{-Caus}', T\text{-Caus}', L\text{-Caus}', ITL\text{-Caus}' \rangle\rangle$

Onde: $\langle\langle a', \Gamma, T', \Lambda', \zeta' \rangle, \Gamma', v', \langle I\text{-Refs}', T\text{-Refs}', L\text{-Refs}', ITLRefs' \rangle, \langle I\text{-Caus}', T\text{-Caus}', L\text{-Caus}', ITL\text{-Caus}' \rangle\rangle$ representa a situação do comportamento C_i antes da execução de uma ação adaptativa e é determinado de acordo com a definição anterior para comportamentos ISDL.

A Figura 5 ilustra a representação de uma ação ISDL-Adp genérica e suas funções adaptativas anteriores (\mathcal{A}) e posteriores (\mathcal{B}). Graficamente estas funções são representadas em caixas textos acopladas a parte superior e inferior das ações ISDL tradicionais.

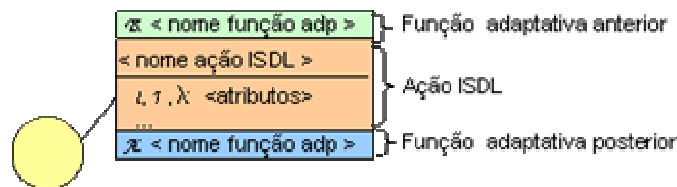


Figura 5 - Representação de ações adaptativas.

Visando diminuir a complexidade e facilitar a leitura das especificações ISDL-Adp geradas, as funções adaptativas são especificadas separadamente das ações ISDL-Adp. Textualmente, foram adicionadas as especificações ISDL tradicionais duas seções: *Before* e *After* que servem, respectivamente, para modelar as funções adaptativas anteriores e posteriores. No interior de cada seção são declaradas as funções adaptativas associadas a cada inter(ação). A Figura 6 ilustra a estrutura básica definida para as seções adaptativas anteriores e posteriores e suas respectivas funções e ações adaptativas.

.... *outras seções ISDL-Adp*

Before

[<nome da função adaptativa anterior>] :
(ação adaptativa elementar)

....

... (declaração de outras funções adaptativas)

After

[<nome da função adaptativa posterior>] :
(ação adaptativa elementar)

....

... (declaração de outras funções adaptativas)

.... *demais seções ISDL-Adp*

Figura 6 – Representação de funções e ações elementares adaptativas ISDL.-Adp.

As ações adaptativas elementares ISDL-Adp são definidas de forma semelhante as ações adaptativas propostas para Autômatos Adaptativos [8]. As ações adaptativas ISDL-Adp elementares assumem o seguinte formato: *prefixo* [*padrão da produção*], onde *prefixo* representa um dos três tipos de ações adaptativas elementares a serem executadas: "?" (ação de inspeção), "-" (ação de eliminação) e "+" (ação de inserção), enquanto que o *padrão da produção* corresponde um nome da função que representa a ação ISDL tradicional. Desta forma temos uma ação adaptativa elementar representada da seguinte forma:

[prefixo] $\langle\langle a, I, T, \Lambda, \zeta \rangle, \Gamma, v, \langle I\text{-Refs}', T\text{-Refs}', L\text{-Refs}', ITLRefs' \rangle, \langle I\text{-Caus}', T\text{-Caus}', L\text{-Caus}', ITL\text{-Caus}' \rangle\rangle$

onde: $\langle\langle a, I, T, \Lambda, \zeta \rangle, \Gamma, v, \langle I\text{-Refs}', T\text{-Refs}', L\text{-Refs}', ITLRefs' \rangle, \langle I\text{-Caus}', T\text{-Caus}', L\text{-Caus}', ITL\text{-Caus}' \rangle\rangle$ faz parte do comportamento ISDL definido na seção 2 e o prefixo da ação adaptativa é definido conforme o tipo de uma ação adaptativa (prefixo), esta desempenha uma função que pode ter as seguintes características:

- ação adaptativa de inspeção: tem função de inspecionar/verificar se determinadas estruturas ISDL fazem parte da especificação do comportamento modelado.
- ação adaptativa de eliminação: tal estrutura é utilizada para eliminar estruturas ISDL necessárias as modificações que devem ocorrer no sistema modelado. Ao executar uma ação adaptativa de eliminação estruturas são eliminadas do comportamento ISDL
- ação adaptativa de inserção: serve para adição de novas estruturas ao comportamento ISDL. É por meio da execução de ações adaptativas de adição que novas estruturas são adicionadas ao comportamento ISDL

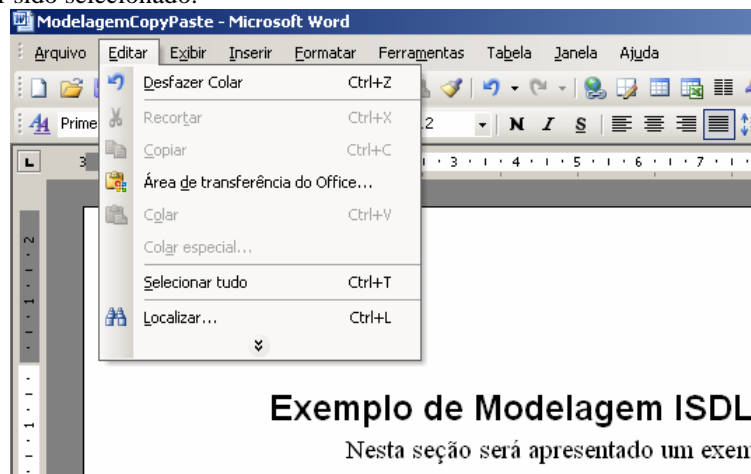
A operação do modelo ISDL-Adp ocorre por meio da modificação das ações pertencentes a um comportamento pela

evolução gradual do seu conjunto de regras, com a adição ou remoção de ações efetuadas por meio da execução das ações adaptativas.

Estando o ISDL-Adp em sua situação inicial, recebe uma seqüência de eventos externos. As transições são executadas conforme essa seqüência, consumindo os eventos externos, repetindo-se o processo até o final da seqüência. Para cada evento recebido verifica-se qual ou quais são as produções a serem executadas. No que se refere à execução do ISDL-Adp, se a ação adaptativa A estiver presente, será executada em primeiro lugar, enquanto que se B estiver declarada, será executada por último.

4. Estudo de Caso

Nesta seção será apresentado um exemplo que ilustrará a utilização do modelo e linguagem ISDL-Adp na modelagem de aplicações complexas. O exemplo escolhido para ser modelado é comum entre os usuários de computadores pessoais e de tecnologia Microsoft Windows [18] e tem por objetivo ilustrar o funcionamento das opções “Copiar”, “Colar” e “Colar Especial” oferecidas na maioria dos produtos disponíveis para esta tecnologia. Na Figura 7 é ilustrado parte da interface da ferramenta Microsoft Office Word 2003 [19] onde aparecem desabilitadas as opções “Copiar”, “Colar” e “Colar Especial” devido o conteúdo da área de transferência estar vazio e nenhum conteúdo ter sido selecionado.

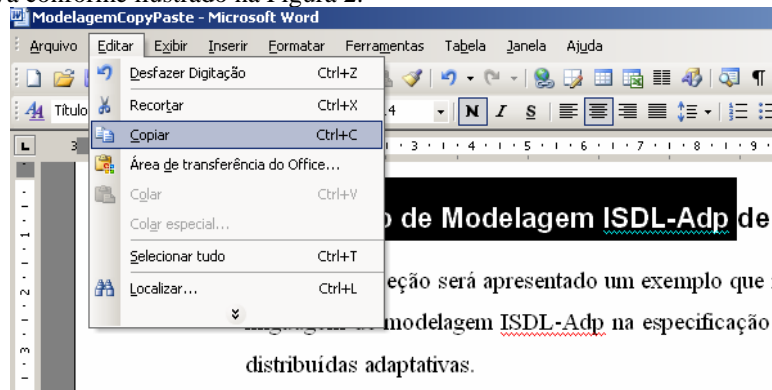


Exemplo de Modelagem ISDL

Nesta seção será apresentado um exemu

Figura 7- Situação inicial MS-Word Office XP 2003.

Para a opção “Colar” ser ativada faz-se necessário que o usuário selecione um texto, imagem ou algum outro objeto que torne a opção ativa conforme ilustrado na Figura 2.



modelagem ISDL-Adp na especificação distribuídas adaptativas.

Figura 8 - Habilitação da opção “Copiar”.

Ao ser selecionada a opção “Copiar” ou ativada a mesma por teclas de atalho é copiado o conteúdo selecionado para a Área de Transferência e habilitada a ocorrência das opções “Colar” e “Colar Especial” do menu de opções. Pode-se observar que ao ser selecionado um texto e copiado o seu conteúdo para a área de transferência são atribuídas a operação de colar “Texto Formatado” para a opção “Colar” e as operações de colar “Texto unicode sem formatação”, “Formato Html”, “Texto Formatado”, “Texto sem formatação”, “Texto Formatado RTF”, etc. para a opção “Colar Especial”. Tais opções são habilitadas conforme o conjunto de programas instalados anteriormente que permitiram que determinadas operações fossem disponibilizadas. A Figura 3 ilustra a interface que é apresentada após a escolha da opção “Colar Especial”. A interface habilitada apresenta ao usuário uma lista de possíveis operações para serem realizadas conforme sua necessidade e escolha.

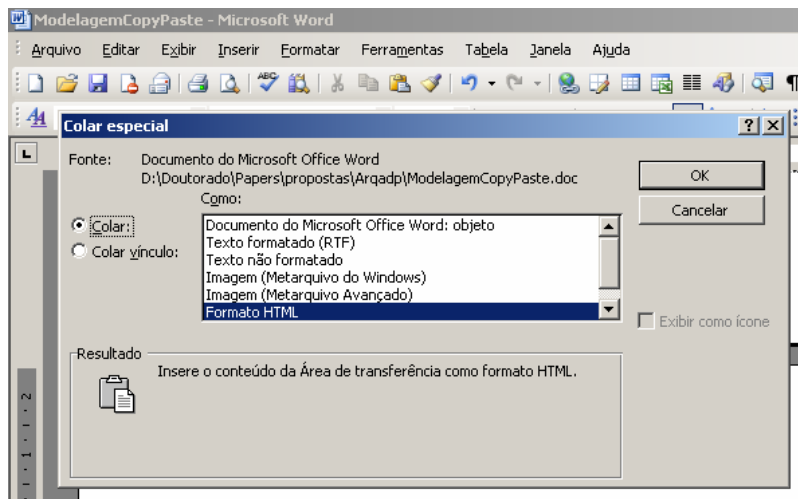


Figura 9 - Operações disponíveis ao escolher a opção "Colar Especial" para objetos do tipo Texto.

Em relação a cópia do conteúdo de uma imagem para a área de transferência ocorre situação semelhante a que ocorre ao ser copiado um texto. Ou seja, ao ser selecionada uma imagem é executada a ação de cópia de conteúdo para a área de transferência disponível na opção “Copiar” e atribuídas as operações “Colar Bitmap” para a opção “Colar” e as operações: “Bitmap”, “Imagem (Meta Arquivo Avançado)”, “Imagem GIF”, etc. para a opção “Colar Especial”. Da mesma forma que para a opção de cópia de um texto as operações são disponibilizadas conforme o conjunto de programas instalados anteriormente no equipamento, o tipo de imagem selecionado e a origem do conteúdo a ser copiado para a área de transferência (programa em que foi gerada a cópia). Desta forma pode-se observar que o conjunto de operações disponibilizadas pelo comando “Colar Especial” pode ser alterado em tempo de execução e disponibilizar diferentes operações conforme características da origem e do conteúdo copiado. Tal situação ilustra a aplicação de tecnologia adaptativa e demonstra o seu uso na prática. Vale ressaltar que este trabalho não tem foco principal na modelagem de características das opções “Copiar”, “Colar” e “Colar Especial” e sim na aplicação de tecnologia adaptativa na modelagem de aplicações complexas. Outras opções de cópias de conteúdos como imagem tipo vetorial, áudio, vídeo, etc.. são sugeridas para efeitos de estudos futuros.

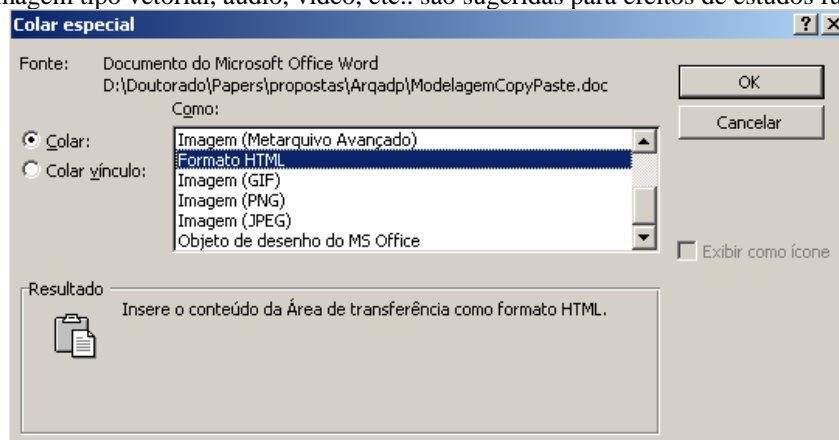


Figura 10 - Operações disponíveis ao escolher a opção "Colar Especial" para objetos tipo Imagem.

4.1 Modelagem ISDL-Adp de Copiar/Colar conteúdo Área de Transferência.

Fundamentado no problema descrito anteriormente é apresentada a modelagem ISDL-Adp de Copiar/Colar conteúdo Área de Transferência. A Figura 11 representa a modelagem de entidades do *Copiar/Colar conteúdo Área de Transferência* que é representada pela na visão externa por um *Usuário* que interage com um *Sistema de Computação*. Na visão interna do sistema de computação foram abstraídos alguns detalhes visando simplificar a sua apresentação. Desta forma temos o *Sistema de Computação* composto por três subentidades: *Interface da Aplicação*, *Sistema Operacional* e *Área de Transferência*. A *Interface da Aplicação* tem por finalidade receber interações vindas do usuário e enviar ao sistema operacional objetivando a sua execução. Esta entidade também recebe mensagens vindas do *Sistema Operacional* e da *Área de Transferência* e as apresenta aos usuários. A subentidade *Sistema Operacional* tem por objetivo realizar o gerenciamento do *Sistema de Computação* e a subentidade *Área de Transferência* é um espaço de memória que permite ao usuário realizar cópias de dados ao utilizar suas aplicações ou intercambiar dados entre as aplicações utilizadas.

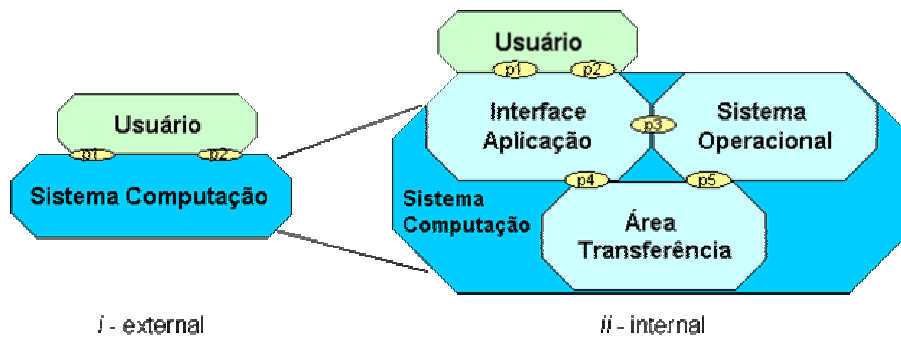


Figura 11 - Modelo Entidades Exemplo Copiar-Colar Área Transferência.

A subentidade *Interface da Aplicação* possui diversas funcionalidades conforme a aplicação que é executada. Na Figura 12 é ilustrada parte do comportamento *C_Copiar_Colar*. Tal comportamento representa as funcionalidades de Copiar e Colar conteúdo para Área de Transferência e é comum para a maioria das aplicações existentes. O comportamento ilustrado representa a configuração inicial da aplicação e constitui-se da interação *i_Cop* que tem por objetivo modelar o recebimento das interações vindas dos usuários referentes a requisição de cópia de um determinado conteúdo selecionado para a área de transferência. Ao ser ativada a interação *i_Cop* ocorre a habilitação das ações *aCopTxt* e *aCopImg* que possuem a função adaptativa anterior *FReset(str)* responsável por restaurar o comportamento em execução para a situação inicial, caso este tenha sido adaptado anteriormente, e a função adaptativa posterior *FCop(str,str)* responsável por realizar mudanças no comportamento atual conforme o conteúdo selecionado pelo usuário.

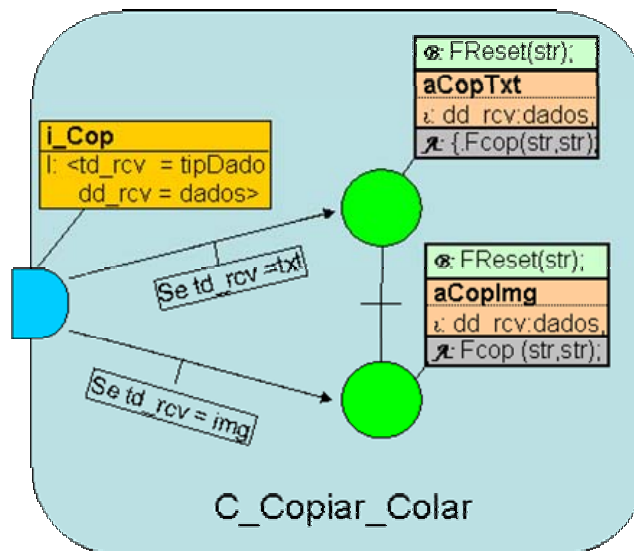


Figura 12 - Comportamento inicial funcionalidade Copiar/Colar conteúdo para Área de Transferência.

Na Figura 13 são apresentados os subcomportamentos *C_Cop_Txt* e *C_Cop_Img*. O subcomportamento *C_Cop_Txt* é responsável por adicionar ou remover do comportamento inicial funcionalidades relacionadas a cópia do conteúdo do tipo texto. Este subcomportamento ao receber uma mensagem oriunda da interação *icl* no ponto de entrada *e1*, referente a colar o conteúdo da área de transferência, habilita a ocorrência da ação *aCl_TxtFmt* responsável por colar o conteúdo da área de transferência como sendo um texto formatado. Ao receber um mensagem no ponto de entrada *e2*, oriunda da interação *icle* (colar especial), ocorre a habilitação da escolha de ocorrência de uma das ações *aCl_TxtFmt*, *aCl_TxtSemFmt* e *aCl_TxtHtml* responsáveis respectivamente por colar textos formatados, sem formatação ou no formato HTML. Após execução de uma das possíveis ações é enviada mensagem por meio do ponto de saída *s1* contendo informações para serem apresentadas ao usuário por meio da interação *i_apu*. O subcomportamento *C_Cop_Img* é constituído de forma semelhante ao comportamento *C_Cop_Txt* diferenciado-se no conjunto de ações a serem desempenhadas que estão relacionadas a operação de colar imagens. A ação *aCl_ImgBmp* é responsável por colar imagens do tipo *Bitmap* e é padrão para a interação *icl*. As ações *aClImgBmp*, *aCl_ImgMtArAvc* e *aCl_ImgGif* são responsáveis respectivamente por colar imagem do tipo *Bitmap*, *Meta Arquivo Avançado* e *Gif* estão relacionadas a interação *icle* e ocorrem em escolha.

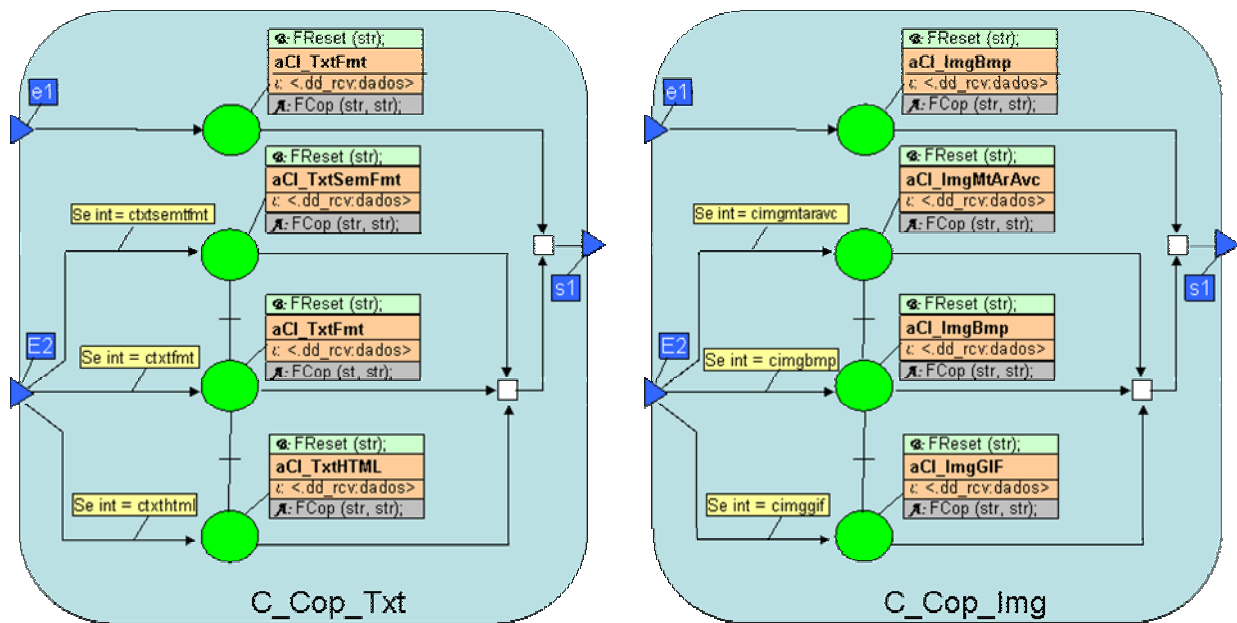


Figura 13 – Subcomportamentos *C_Cop_Img* e *C_Cop_Txt*.

Na Figura 14 são apresentadas as funções adaptativa *FCop(str, str)* e *FReset(str)*. Tais funções são responsáveis por realizar mudanças no comportamento corrente em um determinado instante de execução. A função *Fcop(str, str)* recebe dois parâmetros referente ao nome do comportamento inicial e comportamento que será inserido (adaptado). A ser chamada esta função ocorre a execução de um conjunto de ações adaptativas elementares que tem por objetivo inserir um novo subcomportamento que deverá desempenhar a função de colar conteúdo da área de transferência. Após a inserção do novo subcomportamento ocorre a ligação de seus pontos de entrada e de saída com as respectivas interações do comportamento inicial de forma que a aplicação possa disponibilizar as novas funcionalidades inseridas em seu conjunto de regras. A função *FReset(str)* executa inicialmente uma ação elementar de consulta e verifica no conjunto de regras a existência de uma regra que atenda as condições exigidas. Na existência da regra, a variável *adaptador* recebe o nome do respectivo subcomportamento que atende a referida consulta. Após ser definido qual subcomportamento pertence a configuração atual este e suas ligações de entrada e de saída devem ser removidas de forma a aplicação retornar a sua situação de configuração inicial.

<pre> FCop (inicial: char, adaptador: char); { + adaptador; + icl.inicial ← e1.adaptador; + icle.inicial ← e2.adaptador; + s1.inicial ← i_apu.adaptador; + aCopTxt.inicial ∨ aCopImg.inicial ← icl.inicial; + aCopTxt.inicial ∨ aCopImg.inicial ← icle.inicial; } </pre>	<pre> FReset (inicial: char) var adaptador:char; { ? icl.inicial ← e1.adaptador; - icl.inicial ← e1.adaptador; - icle.inicial ← e2.adaptador; - s1.adaptador ← i_apu.adaptador; - aCopTxt.inicial ∨ aCopImg.inicial ← icl.inicial; - aCopTxt.inicial ∨ aCopImg.inicial ← icle.inicial; - adaptador; } </pre>
---	---

Figura 14 - Funções Adaptativas *FCop(...)* e *FReset(...)*.

Na Figura 15 é apresentado o comportamento *C_Copiar_Colar* que foi modificado ao ser habilitada a execução da ação *aCopImg* e suas respectivas ações adaptativas anterior e posterior. Anteriormente a execução de *aCopImg* foi executada a função adaptativa *FReset(str)* que restaurou o comportamento para a configuração inicial, na seqüência ocorreu a execução da ação *aCopImg* que copiou o conteúdo selecionado para a área de transferência e após a execução de *aCopImg* foi executada a ação adaptativa *FCop(str, str)* que realizou mudanças no comportamento inicial adicionando novas funcionalidades e permitindo a realização de colar conteúdo da área de transferência do tipo imagem.

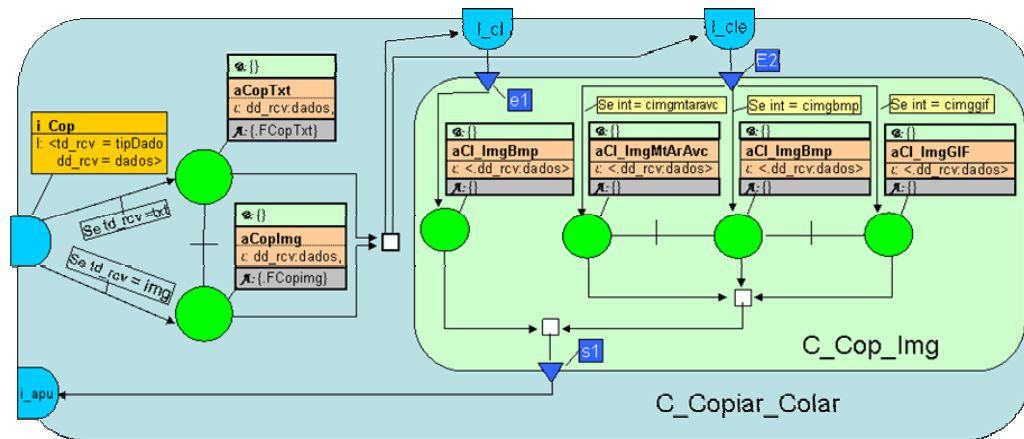


Figura 15 - Comportamento Comp_Copiar_Colar após execução ação Adaptativa.

5. Conclusão

Neste trabalho foi apresentada a estrutura básica do Modelo ISDL-Adp que foi projetado para a especificação de sistemas com características distribuídas e reconfiguráveis. Tal modelo é fruto da união dos conceitos do modelo não-adaptativo ISDL aos conceitos de mecanismos adaptativos.

No modelo ISDL-Adp, de forma análoga ao que ocorre com o modelo ISDL, uma entidade modela um objeto e é caracterizada por seu comportamento (métodos) e estado (atributos). Isso permite, por exemplo, que se estruture um sistema como uma hierarquia de abstrações (entidades) e que se encapsule o comportamento de entidades, pois estes somente são acessíveis por meio dos pontos de interação definidos em sua interface.

Por sua vez, a definição de ações adaptativas na estruturação de comportamentos permite o uso da adição e da otimização de ações e relações de causalidade, de forma dinâmica, o que torna o modelo ISDL mais flexível e com maior poder de expressão para problemas que apresentam características reconfiguráveis. Utilizando-se do modelo ISDL-Adp o projetista pode modelar comportamentos que tenham sua estrutura modificada durante a execução do sistema. Os recursos adaptativos permitem que sejam inseridas ou eliminadas ações e relações de causalidade a cada execução do comportamento, o que permite que aplicações com grande complexidade de representação sejam expressas com o auxílio deste modelo.

O exemplo ilustrativo fundamentado na modelagem de funcionalidade de copia/colar conteúdo para a área de transferência serviu para ilustrar a utilização do modelo no projeto de tais aplicações. Tal exemplo permitiu verificar que o modelo permite representar aplicações que necessitam realizar mudanças em seu comportamento durante o tempo de execução. A modelagem também permitiu que o projeto fosse realizado de forma mais organizada e que somente os recursos necessários sejam utilizados a cada execução. Dessa forma ocorreu uma otimização dos recursos e um melhor desempenho e utilização dos dispositivos móveis.

Como continuação deste trabalho deverá ser investigadas a possibilidade de utilização de conceitos de tecnologia adaptativa na modelagem de interações entre comportamento e na estrutura do modelo de entidades. Também faz-se necessário a utilização de uma metodologia para dar suporte no projeto de aplicações que possuem características adaptativas. Tal metodologia deverá ser fundamentada no modelo ISDL-Adp e ferramentas deverão ser construídas para dar suporte a esta metodologia. Também serão realizados estudos referentes a utilização do modelo no projeto de outros tipos de aplicação.

Referências Bibliográficas

- [1] OMG. UML Profile for Enterprise Distributed Object Computing Specification, ptc/02-02-05, (2002).
- [2] Harel, D., Pnueli, A., Schimidit, J.P. and Sherman, R.,. On the formal semantics of statecharts, In: Symposium on logic in Computer Science, 2º, Ithaca, *Proceedings, IEEE Press, New York*, p.p.54-64, (1984).
- [3] ISO IS 8807, Information Processing Systems - Open Systems Interconnection - LOTOS - A Formal Description Technique Based on the Temporal Ordering of Observational Behaviour, *ISO*, 1989.
- [4] ISO/IEC 9074 (1989) Information Processing Systems - Open System Interconnection - Estelle - A Formal Description Technique Based on an Extended State Transition Model, *ISO*, (1989).
- [5] ITU-T "Recommendation Z.100", White Book and Annexes, 1994
- [6] Pires L.F. Archictetural Notes: A Framework for Distributed Systems Development, *Ph.D Thesis Twente University, Netherlands*, (1994).

- [7] Quartel, D. Actions Relations – Basic design concepts for behaviour modelling and refinement. *Ph.D Thesis Twente University, Netherlands*, (1997).
- [8] Neto, J.J. Contribuição à metodologia de construção de compiladores. *Tese de Livre Docência, Escola Politécnica, Universidade de São Paulo*, São Paulo, (1993).
- [9] Rady, J. STAD-Uma ferramenta para representação e simulação de sistemas através de statecharts adaptativos, *Tese de Doutorado, Escola Politécnica, Universidade de São Paulo*, São Paulo, (1995).
- [10] Allen, R.J., Douence, R. and Garlan, D. Specifying and Analyzing Dynamic Software Architectures. *Proceedings of the 1998 Conference on Fundamental Approaches to Software Engineering* Lisboa, Portugal, (1998).
- [11] Revault N. and Yoder, J. Adaptive Object-Models and Metamodeling Techniques *ECOOP '2001 Workshop Reader; Lecture Notes in Computer Science; Springer Verlag*, (June 2001).
- [12] Neto, J.J. Adaptive rule-driven devices - general formulation and case study, *CIAA 2001 - Sixth International Conference on Implementation and Application of Automata, Pretoria University, Pretoria-África do Sul*, (2001).
- [13] Sinderen M.van, Pires, L.F., Vissers, C.A.and Katoen, C.A A design model for open distributed processing systems. *Computer Networks and ISDN Systems*, (1995), pp. 1263-1285.
- [14] Testbed Studio Tool, <http://www.bizzdesign.com>. (2004).
- [15] Friends Project.<http://www.telin.nl/Middleware/FRIENDS/ENindex.htm>. (Maio 2004)
- [16] Camolesi, A.R. and Neto, J.J. An adaptive model for modelling of distributed system, *Conference Argentina in la Ciência da Computacion (CACIC)*, La Plata, Argentina, (Outubro 2003).
- [17] Camolesi, A.R. Uma metodologia para o Design de Serviços de TV-Interativa, *Dissertação de Mestrado, PPG-CC, UFSCar*, (Fevereiro 2000).
- [18] Microsoft Windows XP disponível em: <http://www.microsoft.com/brasil/windows/default.asp>
- [19] Microsoft Office Word 2003 disponível em <http://www.microsoft.com/brasil/office/editions/standard.asp>