Um Ambiente Para Desenvolvimento De Interfaces Homem-Computador Distribuídas em Rede de Longa Distância

Carlos Neves Júnior[+]
Hans Kurt Edmund Liesenberg[*]

[+]931669@dcc.unicamp.br
[*]hans@dcc.unicamp.br
Instituto de Computação
Universidade Estadual de Campinas
Caixa Postal 6176, 13083-970 Campinas, SP - Brasil

Resumo

O objetivo deste artigo é apresentar um ambiente para desenvolvimento de interfaces homem-computador distribuídas, o Ambiente Xchart. O ambiente Xchart contém um conjunto de ferramentas que facilitam o processo de construção e execução de interfaces homem-computador concorrentes (possivelmente distribuídas). Algumas aplicações distribuídas não exigem desempenho em tempo real e poderiam perfeitamente fazer uso dos recursos de distribuição deste ambiente para executarem de forma distribuída em rede de longa distância.

Palavras Chave: IHC, Sistemas Distribuídos, Sistemas Reativos, Rede de Longa Distância

1. INTRODUÇÃO

O crescente aumento dos recursos computacionais tem permitido a construção de sistemas cada vez mais sofisticados com interfaces homem-computador mais amigáveis. Esta evolução nos sistemas de computação permitiu também a construção de aplicações distribuídas.

Dentro do contexto de interfaces e de distribuição, o Projeto Xchart está desenvolvendo um ambiente para especificação, implementação e execução de interfaces homem-computador: o ambiente Xchart. Este ambiente facilita o desenvolvimento de interfaces complexas, com vários de seus componentes concorrentes e potencialmente distribuídos. O ambiente Xchart pode ser classificado como um Sistema de Gerenciamento de Interfaces Distribuídas (SGID) por apoiar tanto a construção como a execução de interfaces distribuídas. O Sistema de Execução de Xchart é responsável por dar suporte, em tempo de execução, ao diálogo das interfaces desenvolvidas segundo a proposta do Projeto Xchart.

O Gerente de Distribuição (GD) é o componente do Sistema de Execução que fornece os recursos de distribuição do ambiente Xchart, auxiliando nas interações não locais (remotas) entre os elementos da interface. O foco do artigo é dado para o aspecto distribuído do ambiente, sendo assim, o GD vai merecer um destaque maior na apresentação.

Existe um grupo de aplicações distribuídas que poderia fazer uso das facilidades deste ambiente. Como existem atrasos envolvidos na comunicação via internet, a principal característica destas aplicações é a não exigência de desempenho em tempo real. Quando as aplicações se limitam à interação dentro de uma rede local, o ambiente pode contemplar um número maior de aplicações já que os atrasos na comunicação são menores e por conseguinte o seu desempenho é otimizado.

O resto do artigo é organizado da seguinte forma. Na Seção 2. deste documento será apresentado o Projeto Xchart com ênfase na descrição do Sistema de Execução do Ambiente, cabendo aqui contextualizar o GD dentro do nosso projeto. A Seção 3. apresenta trabalhos semelhantes realizados por outros grupos de pesquisa. Na Seção 4. é exemplificado o uso e o funcionamento do ambiente através de uma simples aplicação distribuída. A Seção 5. realiza um levantamento final do trabalho.

2. PROJETO XCHART

2.1. Visão Geral

A interface homem-computador (interface, por simplicidade) é o componente de um sistema interativo com o qual um usuário interage. Parte significativa dos custos de desenvolvimento de um sistema interativo está relacionada com o desenvolvimento deste componente. Existem ferramentas que são comumente empregadas para auxiliar o projeto de interfaces com o intuito de aumentar a produtividade e minimizar os custos. A maior parte das ferramentas contemplam apenas um subconjunto das funções existentes em uma interface.

Interfaces mais complexas, com aspectos de controle mais sofisticados onde o usuário conduz mais de uma tarefa ao mesmo tempo ou em sistemas de múltiplos usuários trabalhando concorrentemente, não têm suporte adequado pelas atuais ferramentas de desenvolvimento de interfaces do usuário. Nestas interfaces, o **gerenciador de diálogo** é o componente que apresenta maiores dificuldades de desenvolvimento. Entretanto, o compontente de interface mais enfocado pelas atuais ferramentas é o de **apresentação**.

O Projeto Xchart apresenta uma abordagem cuja ênfase é reduzir as dificuldades de desenvolvimento dos gerenciadores de diálogo de interfaces complexas. Esta abordagem dá apoio à especificação, implementação e execução destes componentes, e visa obter o mesmo sucesso que outras ferramentas conseguiram com o componente de apresentação.

O Projeto Xchart concentra seus esforços na definição de uma linguagem mais adequada para especificação dos gerenciadores de diálogos de interfaces (**a linguagem Xchart** [5]), e também na construção de um ambiente que apoie a implementação e a execução destes componentes (**o ambiente Xchart**).

2.2. A Linguagem Xchart

O projeto da linguagem Xchart decorre de experiências anteriores com a utilização da linguagem *Statechart* [7] para a especificação do controle de diálogo de interfaces. *Statechart* foi originalmente proposta para a descrição de sistemas reativos em geral. Dada as limitações que a linguagem *Statechart* apresenta neste domínio, foi proposta uma especialização dessa linguagem para a descrição de gerenciadores de diálogo, a linguagem Xchart.

Xchart é uma linguagem visual, formal, executável e devidamente projetada para descrever o controle de diálogo de interfaces homem-computador. Em especial, o controle de diálogo concorrente, possivelmente distribuído.

A especificação de um controle de diálogo, descrito na linguagem Xchart [8], pode ser composta por um conjunto de moldes (diagramas definidos na linguagem Xchart). Estes moldes servem de base para a formação de instâncias. Um diagrama (molde) definido na linguagem Xchart, ou simplesmente um **Xchart**, é a menor unidade utilizada para representar controle. Um Xchart, descrito em função da linguagem Xchart, é composto de vários elementos: estados, eventos, regras, transições e outros. A Figura 1 mostra uma simples estrutura de controle descrita na linguagem Xchart sob a forma de diagrama.

A partir de cada Xchart pode ser criado um número arbitrário de instâncias, cada uma com identidade própria, distinta das demais. Uma instância é um Xchart em execução que reproduz o controle descrito no Xchart a partir do qual foi criada. O Xchart é uma entidade estática, enquanto uma instância é a animação de um Xchart subjacente em tempo de execução. As instâncias interagem através de eventos e de memória compartilhada. Na execução distribuída de um diálogo, as instâncias dos Xcharts que o descrevem podem estar espalhadas sobre diferentes nós do ambiente distribuído.

2.3. O Ambiente Xchart

O ambiente Xchart denota um conjunto de ferramentas desenvolvidas para apoiar o emprego da linguagem Xchart. O usuário de Xchart é beneficiado pelo emprego destas ferramentas. Algumas ferramentas são utilizadas em tempo de especificação, enquanto outras são de uso exclusivo em tempo de execução.

As ferramentas do ambiente contemplam a edição de Xcharts, a geração de informações utilizadas em tempo de execução pelo sistema e o software responsável pela execução de instâncias de Xcharts.

Um projetista de software que tenha em mãos o projeto de interação de uma interface utiliza o ambiente para especificar o gerenciador de diálogo sob a forma de diagramas Xchart.

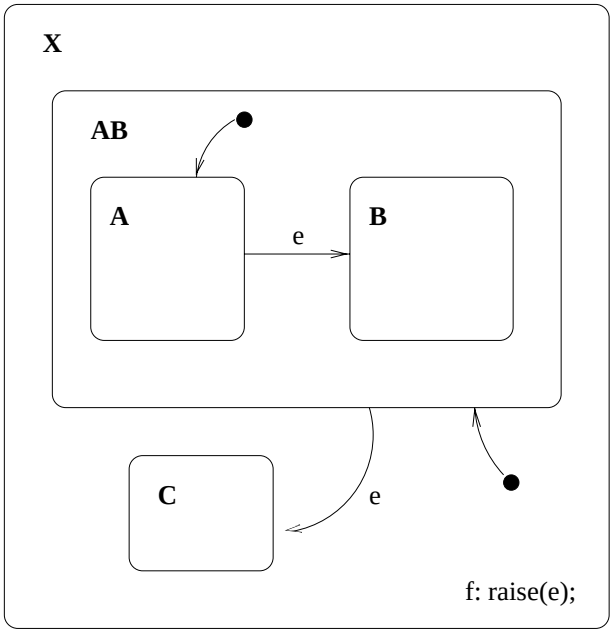

Figura 1: Controle descrito em Xchart

2.4. O Sistema de Execução (SE)

O Sistema de Execução de uma linguagem é o software responsável, em tempo de execução, pela execução de especificações nessa linguagem. O Sistema de Execução de Xchart, ou por simplicidade SE, implementa o componente deste ambiente que é responsável por dar suporte à execução de especificações feitas na linguagem Xchart. A Figura 2 descreve a arquitetura do SE.

O SE divide-se em dois grandes componentes: O Gerente de Distribuição (GD) e o Servidor Xchart (SX). Cliente é um componente que interage com instâncias de Xchart através de um Servidor Xchart. Em um cenário típico, um cliente gera um estímulo externo para uma dada instância de um Xchart, o SX recebe o estímulo e, caso necessário, aciona o GD com o intuito de obter e/ou fornecer informações necessárias à interpretação do estímulo sinalizado. A reação de uma instância é recuperada pelos clientes. Clientes em execução em um dado computador comunicam-se com um único SX em execução nesse computador.

2.4.1. Servidor Xchart (SX)

O SX é responsável pelas tarefas do SE que são executadas em um dado computador. Cada SX manipula um conjunto de instâncias. Se os clientes de um sistema interativo são executados em único computador, por exemplo, então as tarefas do SE resumem-se àquelas realizadas pelo SX.

O SX mantém dados pertinentes a instâncias em execução. Além disso, ele interpreta a ocorrência de um estímulo externo sinalizado para uma dada instância. Grande parte da semântica formal de Xchart é implementada por este componente do sistema. Dados um estímulo externo e a configuração de uma instância, o SX irá identificar a reação correspondente com o eventual apoio dos demais componentes do SE.

2.4.2. Gerente de Distribuição (GD)

O GD é o componente do SE responsável pelo suporte à distribuição. Controle de concorrência sobre os dados compartilhados pelas instâncias do sistema (tais como as variáveis globais), localização de recursos, comunicação em grupo e ponto a ponto, atomicidade de operações, são algumas das funções que este componente deve prover. Os serviços oferecidos pelo GD são utilizados por um SX.

Como exemplo, o SX poderia requisitar um *multicast* de um evento para outras instâncias do sistema distribuído. Esta execução seria de responsabilidade do GD através do seu sistema de comunicação em

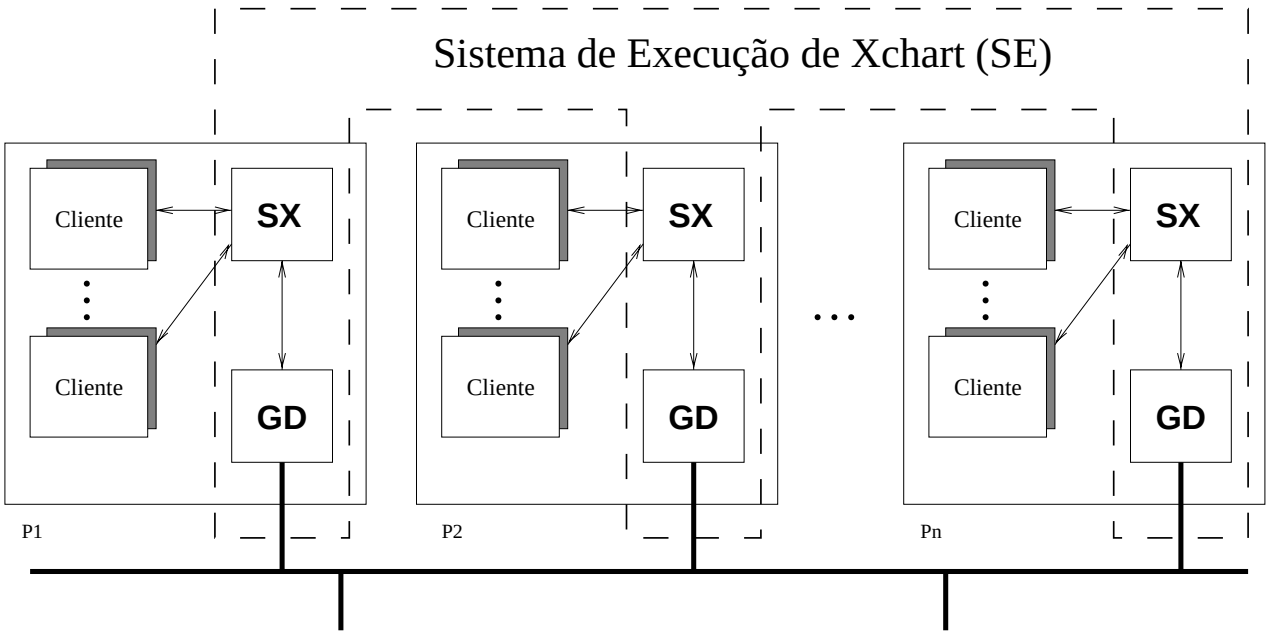

Figura 2: Arquitetura do sistema de execução

grupo. O GD também poderia informar se o estado de uma instância específica está ativo (mecanismo de localização de recursos) para que o SX possa avaliar uma particular expressão dependente de tal informação.

2.5. Maiores informações sobre o projeto

Maiores informações a respeito do projeto Xchart, tais como os trabalhos que estão sendo e os que já foram realizados, podem ser consultadas na página Web do projeto com o seguinte URL:

http://www.dcc.unicamp.br/proj-xchart/bem-vindo.html

3. TRABALHOS CORRELATOS

Nesta seção são apresentados trabalhos semelhantes realizados por outros grupos de pesquisa. Primeiro, é apresentado um ambiente que também se preocupa com especificação de controle, tal como é o diálogo de interfaces. Já o outro ambiente, trata com maior interesse da distribuição das interfaces.

3.1. BetterState

BetterState [1] é um produto comercial que implementa uma variante de *Statecharts*. BetterState automaticamente gera código pertinente aos diagramas fornecidos interativamente através de um editor. O código gerado por este ambiente pode ser destinado a diferentes linguagens de programação, tais como *Visual Basic*, *Delphi*, e *C*.

Em BetterState, o responsável pela especificação em *Statecharts* precisa conhecer mecanismos empregados pelo código gerado por essa ferramenta, ou seja, ele deve conhecer as linguagens de destino citadas para complementar o projeto do sistema. Em BetterState, uma única instância de um mesmo diagrama pode ser executada. Problemas relacionados com a concorrência das aplicações não são previstos pelo sistema de execução de BetterState e devem ser realizados manualmente pelo projetista.

3.2. Visual Obliq

Visual Obliq [3] é um ambiente para especificação, programação e execução distribuída de interfaces gráficas para aplicações multi-usuário. Este ambiente tem um editor para construção interativa da aplicação através de manipulação direta dos elementos da interface. O editor de interfaces foi projetado na linha de programação visual já existente, mas de forma específica para aplicações distribuídas. O projetista pode desenvolver e testar aplicações distribuídas de moderada complexidade de forma bem rápida neste ambiente.

O editor do Visual Obliq gera código de saída para uma linguagem interpretada chamada Obliq [4] que foi projetada especificamente para capturar semântica de computação distribuída. Quando uma aplicação projetada segundo esta filosofia é iniciada por um usuário em alguma máquina, uma *sessão* é criada. Outras máquinas (usuários) podem vir a participar da mesma sessão, fato este previsto pelo ambiente que dá suporte a execução distribuída destas aplicações.

O projetista de Visual Obliq não tem uma grande transparência com relação a distribuição das interfaces, pois aspectos distribuídos do sistema acabam presentes no código gerado. O ambiente não fornece suporte ao controle de diálogo das interfaces, focando a apresentação.

4. UMA APLICAÇÃO DISTRIBUÍDA

As instâncias com controle definido em termos de descrições feitas em Xchart podem estar potencialmente distribuídas sobre nós da internet. A questão relativa ao fato destas instâncias estarem ou não distribuídas em tempo de execução deve ser totalmente transparente ao projetista de sistemas no ambiente Xchart. Para poder prover este tipo de transparência, o Sistema de Execução do ambiente Xchart depende dos serviços do Gerente de Distribuição.

O nosso maior interesse é o de destacar o potencial deste ambiente para a distribuição das interfaces, portanto vamos enfocar o funcionamento do GD no cenário que segue. A aplicação escolhida como exemplo é uma versão distribuída do Jogo da Velha. Veja que a escolha atende ao fato levantado anteriormente, esta aplicação não exige resposta imediata das ações, um lance realizado por um dos jogadores pode ter seu resultado propagado para o seu adversário com um certo atraso, o que torna esta aplicação perfeitamente implementável conforme a nossa proposta.

4.1. Jogo da Velha distribuído: Um exemplo

Nesta seção temos um cenário para apresentação de alguns dos recursos providos pelo ambiente através do GD. O cenário escolhido foi a versão distribuída do Jogo da Velha, note que serão enfocados nesta apresentação apenas aspectos relativos ao controle da interface do Jogo. A Figura 3 descreve o controle de diálogo da interface de um jogador. Cada um dos dois jogadores tem comportamento conforme descrito nesta figura.

Veja que as duas instâncias relativas aos jogadores de uma partida podem estar em computadores diferentes (um jogador no Brasil e outro no Japão, por exemplo). Este detalhe é transparente para o projetista de controle em Xchart, basta notar que esta informação não é definida explicitamente pelo Xchart da figura e é determinada somente por ocasião da criação das instâncias.

Quando o Xchart **Jogador** é instanciado, ele fica com o estado de mesmo nome ativo (Jogador), aguardando a ocorrência de algum evento. A ocorrência do evento **vez** indica que o jogador pode realizar uma jogada, o estado **Jogada** modela esta situação do jogo. O evento **passa** indica que o seu adversário está com o controle do jogo e que este irá fazer a sua jogada.

O estado **FimJogo** é ativado quando o jogo se encerra através das seguintes situações: vitória ou derrota do jogador, ou ainda se o empate for identificado. Os eventos **venceu, perdeu, e empate** caracterizam estas situações, respectivamente.

As regras que irão acionar o uso do GD, estão destacadas na Figura 3 em itálico e negrito. Veja a regra presente no estado Jogador:

not(vez)[in(Jogada,@adversario)]: sync(beep);

Ela indica que a ocorrência de qualquer evento que não seja **vez** aliado ao fato do jogador adversário estar

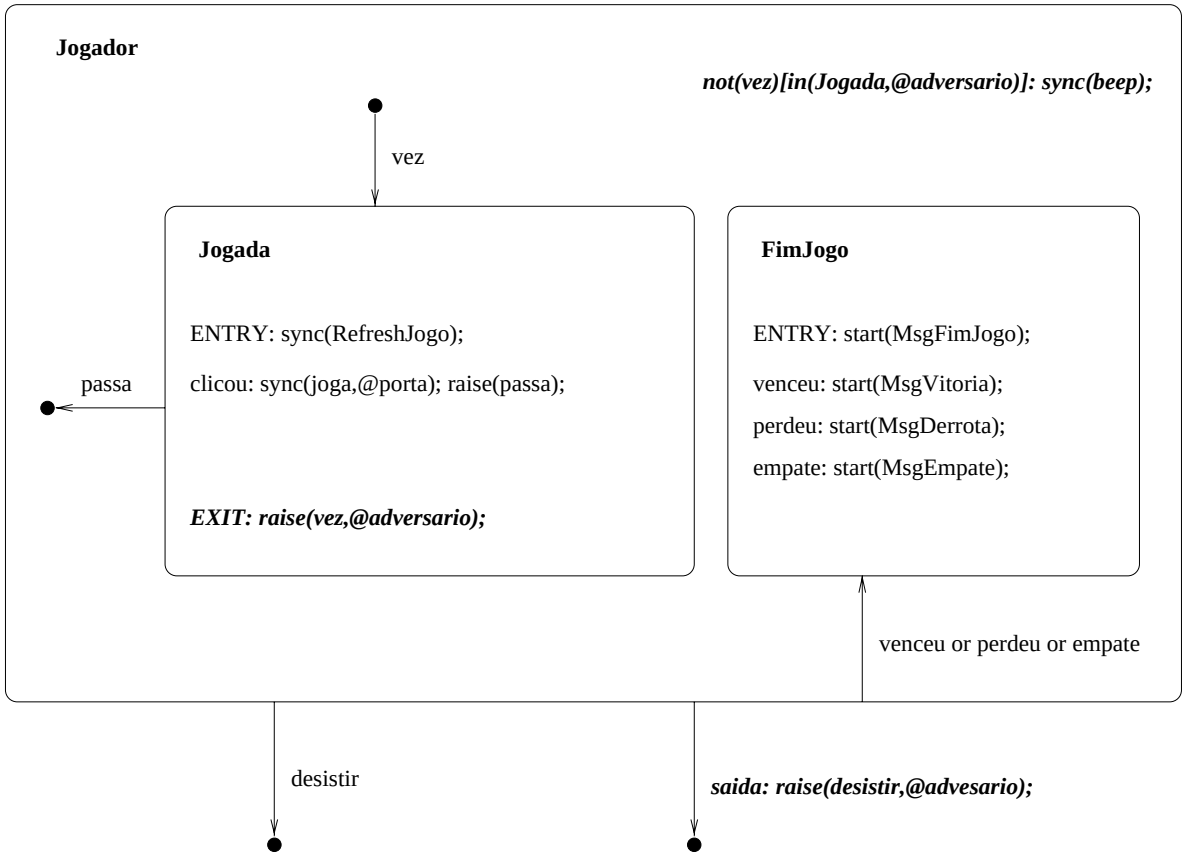

Figura 3: Jogo da Velha - Controle de diálogo de um jogador

realizando a sua jogada irá provocar um beep. Determinar se o jogador adversário está na sua vez de jogar significa descobrir se o seu estado **Jogada** está ativo. É exatamente isto que a função **in(Jogada,@adversario)** retorna quando avaliada. Se o jogador adversário estiver em outro computador, fica a cargo do GD buscar esta informação pela rede.

O GD é novamente exigido pela regra de desativação do estado **Jogada**:

EXIT: raise(vez,@adversario);

A ocorrência do evento **clicou** significa que o jogador acaba de escolher a sua jogada. Com isso, o estado **Jogada** será desativado e a vez deve ser passada para o seu adversário. Isto ocorre com a sinalização do evento **vez** para a instância do adversário. A função **raise(vez,@adversario)** indica esta tarefa. O GD é responsável por notificar à instância adversária sobre a ocorrência deste evento, caso esta execute em um outro nó da rede. Fato semelhante acontece com a regra de saída:

saida: raise(desistir,@adversario);

Uma explicação mais detalhada deste exemplo exigiria um aprofundamento nos recursos da linguagem Xchart. O objetivo desta seção é apenas mostrar um pequeno cenário da atuação do Sistema de Execução, e principalmente do Gerente de Distribuição.

5. CONCLUSÕES

O ambiente Xchart se apresenta como uma proposta muito interessante para o desenvolvimento de interfaces homem-computador distribuídas. A transparência de distribuição fornecida pelo ambiente possibilita que o projetista se concentre exclusivamente com o projeto de interação, deixando a cargo do sistema a resolução dos problemas de distribuição.

A tendência atual de cada vez mais surgirem aplicações cooperativas e distribuídas mostra que a abordagem do ambiente Xchart é emergente e muito interessante, considerando que existem poucas tentativas de realizar controle de interfaces homem-computador distribuído, menos ainda para rede de longa distância.

O trabalho continua em andamento com o intuito de aprimorar ainda mais as ferramentas do ambiente. Vem sendo feita uma reimplementação do ambiente Xchart na plataforma JAVA, para que as vantagens do uso desta linguagem, tais como independência de plataforma, possam ser agregadas ao projeto.

REFERÊNCIAS

[1] M. Ali. A Better Way to State It. *IEEE Computer*, 28(12):72–74, December 1995.

[2] Edilmar L. Alves. Port System: Sistema de Comunicação em Grupo para o Ambiente Xchart. Master's thesis, DCC/IMECC/UNICAMP, Campinas/SP, Fevereiro 1996.

[3] Krishna Bharat and Marc. H. Brown. Building Distributed Multi-User Applications By Direct Manipulation. In *Proceedings of ACM Symposium on User Interfaces Software and Tecnology*, pages 71–82, Marina Del Ray, CA, November 1994.

[4] Luca Cardelli. A Language with Distributed Scope. In *Proceedings of the 22nd Annual ACM Symposium on Principles of Programming Languages*, pages 286–297, January 1995.

[5] Fábio Nogueira de Lucena. *Xchart: Um Modelo de Especificação e Implementação de Gerenciadores de Diálogo*. PhD thesis, IC/UNICAMP, Campinas/SP, Dezembro 1997.

[6] Luciana de Paula Brito. Sistema de Ações Atômicas Distribuídas para Xchart. Master's thesis, DCC/IMECC/UNICAMP, Campinas/SP, Novembro 1996.

[7] D. Harel. A Visual Formalism for Complex Systems. In D. Draper, editor, *Science of Computer Programming, 8(3):231-274*, June 1987.

[8] Fábio N. Lucena, Carlos Neves Júnior, and Tallys H. Yunes. Descrição da Interface Homem-Computador do Spring usando Xchart, 1998. Em preparação, http://www.dcc.unicamp.br/proj-xchart/spring/.