



FACULDADE DE INFORMÁTICA
PUCRS – Brazil
<http://www.inf.pucrs.br>

Manual do X-BDI

Alexandre de Oliveira Zamberlam
Rodrigo Rafael Villarreal Goulart
Michael da Costa Móra
Lucia Maria Martins Giraffa

TECHNICAL REPORT SERIES

Number 018
December, 2001

Contact:

michael@inf.pucrs.br

<http://www.inf.pucrs.br>

giraffa@inf.pucrs.br

<http://www.inf.pucrs.br>

Alexandre de Oliveira Zamberlam is a graduate student of UNIJUÍ – Universidade Regional do Noroeste do Estado do RS – Brazil. He is member of the GIE research group (Computer Science apply to Education Research Group) since 2000. He receives a federal graduate research grant from CAPES (Brazil) to support his research.

Rodrigo Rafael Villarreal Goulart is a graduate student of PUCRS – Pontifícia Universidade Católica do Rio Grande do Sul – Brazil. He is member of the GIE research group (Computer Science apply to Education Research Group) since 2000. He receives a federal graduate research grant from CAPES (Brazil) to support his research.

Michael da Costa Móra works at PUCRS/Brazil since 1994. He is an assistant teacher. He develops research in Multi-agent systems, Artificial Intelligence apply to Education, and E-Business. He got his Ph.D. in 2000 at UFRGS (Brazil).

Lucia Maria Martins Giraffa works at PUCRS/Brazil since 1986. She is a professor and leader of the GIE Gr. She develops research in Multi-agent systems, Artificial Intelligence apply to Education, and design of educational software. She got her Ph.D. in 1999 at UFRGS (Brazil).

Copyright © Faculdade de Informática – PUCRS

Published by– FACIN – PUCRS

Av. Ipiranga, 6681

90619-900 Porto Alegre – RS – Brazil

Manual do X-BDI

Relatório Técnico N° 018/2001

Alexandre de Oliveira Zamberlam* e Rodrigo Rafael Villarreal Goulart*

Michael da Costa Móra¹ Lucia Maria Martins Giraffa²

Introdução

Fruto do trabalho desenvolvido pelo professor Michael da Costa Móra em sua tese de doutorado no CPGCC/UFRGS - FCT/UNL, a ferramenta X-BDI foi desenvolvida com a intenção de diminuir a distância que existe entre teorias formais para especificação de agentes cognitivos e sua programação. Atualmente, o X-BDI vem sendo utilizado e aperfeiçoado pelo PPGCC da PUCRS e por outros grupos de pesquisa.

O X-BDI é um ambiente que permite a descrição formal de agentes cognitivos baseados modelo BDI, sendo ao mesmo tempo, uma linguagem para implementação desses agentes. Segundo Móra [MOR00], sua proposta disponibiliza um sistema formal cuja linguagem é adequada para a representação dos estados mentais de agentes. Esse sistema fornece suporte para diversos tipos de raciocínios, tratados de forma computacional, além de fornecer as ferramentas necessárias para se modelar os estados mentais da tríade BDI.

O modelo possui duas características: serve como ambiente de especificação de agentes, em que é possível definir formalmente um agente através da descrição dos seus estados mentais (crenças, desejos e intenções) e de suas respectivas propriedades. Este ambiente pode, também, fazer a verificação destas propriedades; serve como um ambiente de implementação de agentes. O X-BDI é um ambiente onde se formaliza o agente e se o executa, verificando se o comportamento desejado ocorreu ou não.

O X-BDI possui um ambiente de implementação com alto nível de abstração (os estados mentais), o que reduz a complexidade no desenvolvimento de sistemas baseados

* Mestrandos

¹ Professor Colaborador e autor do X-BDI

² Orientadora dos dois primeiros autores

em agentes. Isto significa que o projetista apenas tem de colocar suas heurísticas sobre a área de domínio a ser modelada, expressa pelo conjunto de estados mentais a serem considerados pelo agente, a fim de guiar o seu comportamento. Inicialmente, o agente possui uma base de crenças, um conjunto de desejos potencialmente factíveis e que podem ser dinamicamente atualizados ao longo das interações.

A linguagem do X-BDI possui baixa complexidade no que diz respeito à sintaxe, facilitando sobremaneira o trabalho de especificação e implementação de agentes baseados na arquitetura BDI. A arquitetura BDI passa a ser um paradigma de implementação de agentes cognitivos, devido à possibilidade de se poder executar o modelo formal.

O X-BDI, como modelo formal de agentes, reduz a distância entre especificação e implementação de agentes. Adotando o ELP (*Extended Logic Programming*) como formalismo de apoio, preserva-se tanto as principais características dos modelos formais (habilidade para definir e verificar formalmente agentes), como fornece mecanismo para agentes raciocinarem [MOR98]. Além do que, o modelo X-BDI tem a vantagem de modelar aspectos estáticos e dinâmicos de estados mentais pró-ativos.

Em particular, o que se deseja é possuir uma ferramenta que permita modelar agentes que exibam um comportamento autônomo e flexível. Em [MOR00] é descrito uma situação que exemplifica bem o que se espera poder modelar (e implementar) se existir uma ferramenta adequada. O texto a seguir foi extraído do original de Móra [MOR00].

“No edifício de gabinetes dos professores de uma Universidade, um robô autônomo tem por função transportar documentos entre a secretaria e os gabinetes. Este robô deve, ainda, conduzir os visitantes que chegam ao edifício até o gabinete do professor com quem este deseja falar.

Em um dado instante, o robô recebe um chamado do gabinete 310 para apanhar um envelope e entregá-lo à secretária, no andar térreo, onde o robô encontra-se naquele instante. O robô, então, planeja um curso de ações capaz de levá-lo a cumprir sua tarefa: ele deslocar-se-á até o elevador e, após acionar e aguardar a chegada do elevador, subirá ao terceiro andar e apanhará o envelope, levando-o de volta à secretaria.

Elaborado o plano, o robô passa a ação e começa a executá-lo. Sem nenhum percalço o robô chega ao gabinete 310, apanha o envelope e começa o caminho de volta. Ao chegar ao elevador, no entanto, o robô observa que há algum problema, pois ao pressionar o botão, a luz deste não se acende. Não podendo usar o elevador, o plano

previamente traçado por ele não pode mais ser executado. Assim, o robô reconsidera suas opções e decide tomar a rampa de acesso aos andares e, através dela, chegar a secretaria e entregar o envelope. Possuindo novamente um plano de ação, ele passa a executá-lo. Enquanto está a caminho, o robô recebe duas novas ordens: levar um envelope da secretaria para o gabinete 330 e conduzir um visitante ao gabinete 312 (ambos no terceiro andar). Analisando suas novas ordens, ele conclui ser mais adequado fazer a entrega do envelope que ele tem no momento antes de atender as duas outras ordens (pois ele terá que ir à secretária de qualquer forma para encontrar o visitante e apanhar o novo envelope).

Enquanto se dirige a secretaria, o robô constrói um plano para atender suas novas ordens: ele apanhará o envelope, encontrará o visitante e conduzi-lo-á ao gabinete 330, para depois fazer a entrega do envelope. Existiria outra opção, qual seja entregar o envelope primeiro. Mas ele tem como prioridade atender aos visitantes, para não fazê-los esperar ou caminhar desnecessariamente.

Após cumprir sua ordem corrente, o robô passa a executar seu novo plano. No entanto, enquanto se dirigia ao gabinete 312, após conduzir o visitante ao gabinete 330, o robô sensora uma baixa carga em suas baterias. Imediatamente, ele dirige-se ao ponto de recarga mais próximo. Feita a recarga da bateria, ele retoma o plano interrompido, entrega o envelope e retorna a secretaria.”

São agentes que exibam comportamentos com as características básicas do exemplo acima que se deseja modelar. Esse agente, para ser considerado inteligente deve possuir: autonomia (o robô decide por si que ordens priorizar, que ações tomar para satisfazer estas ordens), habilidade social (o robô recebe ordens dos professores e funcionários, comunica-se com os visitantes para conduzi-los ao seu destino), reatividade (o robô locomove-se sem esbarrar em obstáculos, imediatamente recarrega a bateria ao detectá-la com pouca carga), pró-atividade (o robô é capaz de construir planos para chegar nos gabinetes desejados, prioriza o atendimento dos visitantes para que estes não esperem ou caminhem em demasia) e adaptabilidade (o robô re-planeja quando constata que o elevador está com problemas, interrompe a entrega do envelope quando percebe que tem pouca carga na bateria e retoma a entrega depois de fazer a recarga).

A Figura 1 ilustra a situação acima descrita através de uma implementação reduzida.

```

Identity ( rbt ).
act(rbt, levar(visitante)) causes bel(rbt, -entrada(CARTA_VISITA)),
                                bel(rbt, acompanhando(CARTA_VISITA)).
current_time(NOW).
des(rbt, acompanhando(CARTA_VISITA), 0.3) if
                                bel(rbt, entrada(CARTA_VISITA), TEMPO).
act(rbt, usar(elevador)) causes bel(rbt, entregue(CARTA_VISITA)) if bel(rbt, elevador(ok)).
des(rbt, entregue(CARTA_VISITA)) if bel(rbt,acompanhando(CARTA_VISITA)).
act(rbt, usar(rampa)) causes bel(rbt, entregue(CARTA_VISITA)) if bel(rbt, -elevador(ok)).
act(rbt, voltar) causes bel(rbt, entrada(rbt)).
des(rbt, entrada(rbt)) if bel(rbt, entregue(CARTA_VISITA)).
act(rbt, carregar(bat)) causes bel(rbt, carregada(bat)).
des(rbt, carregada(bat), 0.5).
current_time(NOW).
before(TEMPO).
bel(rbt, -carregada(bat)) if bel(rbt, sinal_carga(bat), TEMPO).

```

Figura 1: Código X-BDI.

Para um melhor entendimento, este relatório técnico está dividido em 4 seções. A próxima seção descreve o Funcionamento do ambiente X-BDI. A seção Instalação, apresenta os procedimentos para a instalação e execução do X-BDI. Por último, os trabalhos relacionados e as Referências bibliográficas.

Funcionamento

A ferramenta X-BDI, como já mencionado, tem como função implementar a estrutura cognitiva do agente, e o restante do agente (sensores, atuadores, interface) pode ser implementado em qualquer linguagem. Conforme [GOU01, ZAM01], esta ferramenta faz parte de uma arquitetura distribuída em camadas (ver Figura 2), composta:

- ♦ pelo ambiente **SICStus Prolog**, que disponibiliza ferramentas para construção, depuração e manutenção de programas desenvolvidos com a linguagem Prolog;

- ◆ por um programa em Prolog que implementa a **ferramenta X-BDI** e que é executado no SICStus Prolog;
- ◆ por um arquivo **bdi.a**, contendo as especificações de crenças, desejos e ações do agente BDI, que é lido no momento da carga do X-BDI. No início deste arquivo deve ser feita a identificação do agente (`identity(Agente)`), sendo que esta informação não pode ser sensorada/percebida no ambiente;
- ◆ por uma camada que expressa o mecanismo de sensoriamento e atuação do kernel cognitivo de um agente. Esta interação pode ser realizada de duas maneiras:
 - **Normal (via rede)**: a conexão é via socket³, por onde a ferramenta X-BDI envia os planos a serem executados e recebe informações percebidas no ambiente. Assim, uma funcionalidade do X-BDI é a portabilidade, isto é, os sensores, atuadores e interface se mantêm integrados com a parte cognitiva. Esta capacidade foi herdada do ambiente SICStus Prolog, já que ele disponibiliza este mecanismo de comunicação com o ambiente.
 - **Coreografia (via coreo.a)**: onde a ferramenta lê o arquivo `coreo.a`, simula o sensoriamento do ambiente e gera os planos na tela. No arquivo `coreo.a` ficam informações do tipo: `[sense(propriedade1, tempo1), sense(propriedade2, tempo2), sense(...),...]`, onde `propriedade` é uma informação, como uma crença do agente sobre ele mesmo ou sobre o ambiente, que será percebida pelo agente em um tempo específico. Assim, é passado para o kernel cognitivo do agente uma estrutura contendo um conjunto de "senses", que dá condições de verificar/acompanhar o processo de deliberação do agente modelado/implementado.

³Um socket é um ponto final de uma comunicação bi-direcional entre dois programas executados na rede, o qual está ligado a um número de porta .)

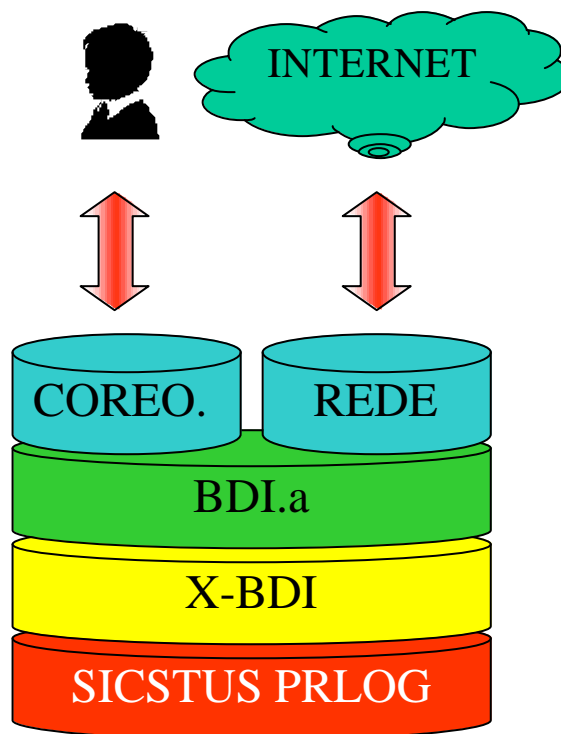


Figura 2: Visão geral da arquitetura X-BDI.

A seguir são descritos a sintaxe do arquivo "bdi.a", que contém as especificações de crenças, desejos e ações do agente BDI; e um exemplo que ilustra a utilização do X-BDI.

1.1 O Arquivo bdi.a

O arquivo bdi.a é composto pelos estados mentais: crenças, ações e desejos, representados pelas seguintes cláusulas:

Crenças

bel(AGENTE, PROPRIEDADE, TEMPO) [if PROPRIEDADE₁, ..., PROPRIEDADE_m].

Onde a especificação do tempo é opcional. A possibilidade de incluir tempo no formalismo do X-BDI, permite representar crenças no futuro - as expectativas do agente, isto é, acreditar em propriedades que se verificam em algum instante de tempo.

Exemplo:

bel(rbt, -carregada(bat)) if bel(rbt, sinal_carga(bat), TEMPO).

Tempo

current_time(TEMPO).

Indica o tempo corrente do sistema, sendo que "tempo" não pode ser uma operação aritmética do tipo `current_time(t1+t2)`. Note que `current_time(tempo)` é uma propriedade sensorada/percebida, ou seja, €

uma informação que vem do ambiente, onde o X-BDI não faz a correção do tempo.

"TEMPO" é um tempo antes ou depois do tempo corrente. Os predicados *before* e *after* podem ser representados de forma explícita e de forma implícita:

Exemplo de forma implícita:

current_time(t3).

bel(agente, propriedade,t1).

Exemplo de forma explícita:

current_time(t3).

before(t1).

bel(agente, propriedade,t1).

Isso significa que a crença do agente sobre a propriedade se verificará antes do tempo corrente t3. O predicado *after* funciona de forma idêntica.

after(TEMPO).

Veja a predicado *before*.

Ações

act(AGENTE, AÇÃO) causes PROPRIEDADE₁, ..., PROPRIEDADE_n [if PROPRIEDADE₁, ..., PROPRIEDADE_m].

A identificação do agente é opcional

Exemplos:

act(rbt, voltar) causes bel(rbt, entrada(rbt)).

act(rbt, usar(rampa)) causes bel(rbt, entregue(CARTA_VISITA)) if

bel(rbt, -elevador(ok)).

Desejos

des(AGENTE, PROPRIEDADE, PRIORIDADE) [if PROPRIEDADE].

A especificação de uma prioridade é facultativa e, se especificada deve representar valor entre um e zero inclusive.

Exemplos:

des(rbt, entrada(rbt)) if bel(rbt, entregue(CARTA_VISITA)).

des(rbt, carregada(bat), 0.5).

Negação

“-“ (hífen)

Exemplos:

bel(rbt, -carregada(bat)) if bel(rbt, sinal_carga(bat),TEMPO).

bel(rbt, -entrada(CARTA_VISITA))

Instalação

Instalando o X-BDI:

- Primeiro: instalar o ambiente Sicstus Prolog.
- Segundo: Certifique-se que o executável **sicstus.exe** esta no PATH.
- Terceiro: coloque os **arquivos do X-BDI** em um diretório (por exemplo c:\xbdi)

Executando o X-BDI:

- Primeiro: entre no diretório no qual foram colocados os **arquivos do X-BDI**.
- Segundo: existe um arquivo de lote (start.bat). Assim, digite "start" e ENTER.

Todos os detalhes e arquivos necessários para a instalação, execução e ilustração do funcionamento da ferramenta X-BDI podem ser encontrados em:

<http://www.inf.pucrs.br/~giraffa/x-bdi>

Trabalhos Relacionados

- MCOE - Multi Cooperative Environment

<http://www.inf.pucrs.br/~giraffa/mcoe/mcoe.html>

Protótipo relacionado ao trabalho de tese de doutorado de Lucia Maria Martins Giraffa (PPGCC/PUCRS) e Trabalho de Conclusão de Leandro Bernsmüller (UFRGS).

- MENSAGENS - Metodologia para modelagem de estados mentais em agentes.

<http://www.inf.pucrs.br/~mroza>

Dissertação de mestrado de Marcelo Roza (PPGCC/PUCRS).

- E-BDI - Editor BDI
<http://www.inf.pucrs.br/~giraffa/e-bdi>
 Parte do trabalho de mestrado de Alexandre de O. Zamberlam (PPGCC/PUCRS).
- E-MCOE
<http://www.inf.pucrs.br/~giraffa/e-mcoe>
 Parte do trabalho de mestrado de Rodrigo R. V. Goulart (PPGCC/PUCRS).
- Utilizando agentes inteligentes em jogos interativos
<http://www.inf.pucrs.br/~sewald>
 Trabalho de mestrado de Leonardo Sewald Cunha (PPGCC/PUCRS - em andamento).

Referências bibliográficas

- [GOU01] GOULART, R. R. V. Em direção a uma arquitetura multiagente para STI em ambientes de jogos educacionais. Porto Alegre: PPGCC/PUCRS, 2001. Dissertação de Mestrado.
- [MOR98] MÓRA, M.; LOPES, J.G.; COELHO, J.G; VICCARI, R. BDI models and systems: reducing the gap. In: AGENTS THEORY, ARCHITECTURE AND LANGUAGES WORKSHOP, 1998, Canarias. Proceedings... Springer-Verlag 1998.
- [MOR00] MÓRA, M. Um modelo de agente executável. Porto Alegre: PPGCC/UFRGS, 2000. Tese de Doutorado.
- [ZAM01] ZAMBERLAM, A.O. Em direção a uma técnica para programação orientada a agentes BDI. Porto Alegre: PPGCC/PUCRS, 2001. Dissertação de Mestrado.