

ASTUTO - UMA ARQUITETURA PARA O DESENVOLVIMENTO DE SISTEMAS BLACKBOARD EM PROLOG

GILBERTO SHIGUEO NAKAMITI

Universidade Estadual de Campinas

UNICAMP/FEE/DCA

Rua Marechal Deodoro, 906 apto 123 - Centro

13020 - Campinas - S.P. - Brasil

RESUMO

Blackboard é um paradigma de resolução de problemas em Inteligência Artificial, capaz de utilizar várias fontes de conhecimento de naturezas diferentes, que agem oportunisticamente para a resolução de um problema.

ASTUTO (Arquitetura para a Solução de problemas através do Uso de Técnicas de IA e Oportunismo) é uma arquitetura de desenvolvimento de Sistemas Blackboard, que dá ênfase às características relacionais entre as entidades do blackboard. Em ASTUTO, tentamos retomar a essência do modelo blackboard, apresentado inicialmente por Newell [07] e discutido em [04]. Para isso, combinamos uma modelagem do blackboard através do Modelo Entidade-Relacionamento (MER) [01] e uma abordagem dirigida por eventos.

PALAVRAS-CHAVE - blackboard, modelo entidade-relacionamento, oportunismo, prolog.

1. INTRODUÇÃO

Um Sistema Blackboard tradicional é constituído por fontes de conhecimento, por um mecanismo de controle, e pelo blackboard propriamente dito. O domínio do conhecimento necessário para a solução do problema é particionado entre as fontes de conhecimento, que são mantidas separadas e independentes, com o intuito de que levem à solução através da informação provida. O blackboard nada mais é que uma base de dados, que consiste de objetos do espaço de soluções. Seu objetivo é servir como meio através do qual as fontes de conhecimento interagem e conduzem à formação incremental e oportunística da solução do problema. O mecanismo de controle disciplina a execução das fontes de conhecimento, decidindo quais caminhos tomar, e monitora o acesso ao blackboard.

As fontes de conhecimento também têm papel muito importante no Modelo Blackboard. Elas devem agir oportunística e cooperativamente, acrescentando informação e conectando as ilhas de solução. Devem ainda ter acesso ao blackboard sob diferentes níveis de abstracção, para que esse processo possa ser realizado de forma mais eficaz. Desta forma, o modelamento do blackboard deve ser flexível e permitir representações diferentes da solução sob níveis diferentes de abstracção, encapsulando detalhes em entidades de maior poder semântico.

ASTUTO é um sistema que utiliza o Modelo Entidade-Relacionamento para modelar e implementar a estrutura do blackboard, e a linguagem Prolog para sua implementação. Na seção dois, é apresentada uma visão geral de ASTUTO e de seu funcionamento. A seção três apresenta o modelamento e implementação do blackboard. Na seção quatro, é mostrado um exemplo de como as Fontes de Conhecimento atuam sobre o blackboard.

2. VISÃO GERAL DO SISTEMA

O objetivo do sistema ASTUTO é se constituir em uma ferramenta através da qual se possa desenvolver e experimentar técnicas e teorias no domínio da aplicação. Pretende-se propiciar a obtenção de uma melhor compreensão do domínio através de seu modelamento, e facilitar o entendimento dos princípios de resolução de problemas nele aplicáveis. O desempenho do sistema não foi fator importante na concepção de ASTUTO, mas sua versatilidade quanto ao domínio da aplicação, modularidade e facilidade de manutenção e extensão do sistema.

A analogia do modelo blackboard com a montagem do quebra-cabeças, presente em [05], nos traz muito de sua essência e do oportunismo intrínseco. As pessoas devem ter visão global e detalhada do estado da solução para que melhor possam contribuir. Além disso, alguns acontecimentos podem alterar significativamente sua visão

do problema e caminhos de raciocínio. Isso pode ocorrer, por exemplo, quando duas ou mais porções do quebra-cabeças (ilhas de solução) se unem em uma única. É interessante que as Fontes de Conhecimento possam ser alertadas sobre fatos relevantes, para que possam se adequar mais rapidamente ao estado atual do problema. Um mecanismo de controle dirigido por eventos pode viabilizar a reprodução desse tipo de comportamento.

Para obter um comportamento dirigido por eventos, foi utilizada a metodologia SADT- Structured Analysis and Design Technique, apresentada em [11]. Através dela, obtivemos uma forte associação entre eventos e ações, permitindo que certos tipos de eventos possam alterar o raciocínio do sistema, e que outros tipos somente modifiquem o blackboard.

Foi também bastante utilizado o princípio de ocultamento de informação, bastante discutido na literatura (por exemplo, em [08,09]), e apresentado em [03] como recurso importante para se obter, por exemplo, características de flexibilidade em sistemas blackboard.

Na figura 1, observamos um esquema conceitual do funcionamento de sistemas desenvolvidos a partir de ASTUTO. Nele, as fontes de conhecimento atuam através de primitivas de vários tipos. O tipo de primitivas utilizado determina a estratégia de resolução de problemas empregado a cada passo. As estratégias podem ser, por exemplo, top-down, bottom-up, e inside-outside. As primitivas operam sobre entidades e relacionamentos, presentes no blackboard

a nível físico. Suas ações são vistas como eventos e fornecem base para o módulo de controle supervisionar a execução das fontes de conhecimento.

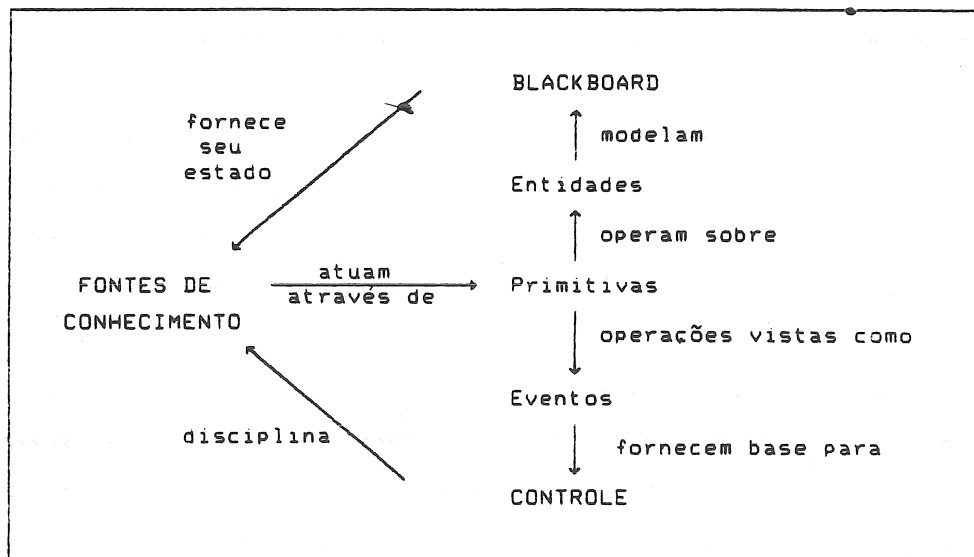


Figura 1 - Esquema Conceitual

3. O BLACKBOARD EM ASTUTO

Tanto flexibilidade quanto representação em diferentes níveis de abstração são facilmente obtidas através do Modelo Entidade-Relacionamento (MER), desenvolvido por Chen [01]. Desde a sua concepção inicial, vários pesquisadores vêm se preocupando em dar-lhe maior poder semântico, quer lhe acrescentando facilidades para especificação de restrições de integridade, quer lhe

possibilitando representar objetos em vários níveis de abstração. Como exemplo, temos o trabalho apresentado em [10], definindo facilidades para a representação de generalização, especialização e agregação de entidades, conduzindo a níveis diferentes de abstração. O conceito de "Transparência Parcial" proposto em [12] pode ser muito útil para a eliminação de redundâncias.

O Modelo Entidade-Relacionamento é bastante adequado para representar as entidades do blackboard, as relações entre elas, e também o nível de confiança das soluções parciais, expectativas de eventos e medidas de distância, normalmente manipuladas por Sistemas Blackboard. O MER foi utilizado inclusive para a representação física de todo o blackboard e para a construção da solução. A implementação do modelamento através da linguagem PROLOG foi também realizada com muita naturalidade, e os primeiros resultados têm encorajado muito a continuidade deste trabalho.

As entidades e os relacionamentos são representados por fatos Prolog. Embora as cláusulas apresentadas aqui possam ser entendidas sem conhecimento da linguagem, alguns leitores poder-
desejar consultar algum texto como [02].

Entidades para um restaurante e um clube poderiam ser:

```
estabelecimento(roma, restaurante, (rua_manaus, 935), (11, 22), 40).
```

```
estabelecimento(regatas, clube, (av_parana, 462), (8, 20), 90).
```

, onde "estabelecimento" seria o nome geral para todos os estabelecimentos; "restaurante" e "clube" representariam o tipo dos estabelecimentos; e em seguida teríamos seus endereços,

horários de funcionamento e expectativas de tempo de permanência, em minutos.

Abaixo, é dado um exemplo de relacionamento que mostra o trajeto entre o clube e o restaurante. Deixa-se o clube às 11:30, toma-se a avenida Paraná, segue-se pela rua Guanabara, e chega-se ao restaurante pela rua Manaus. O tempo de percurso é de dez minutos. A figura 2 ilustra este relacionamento.

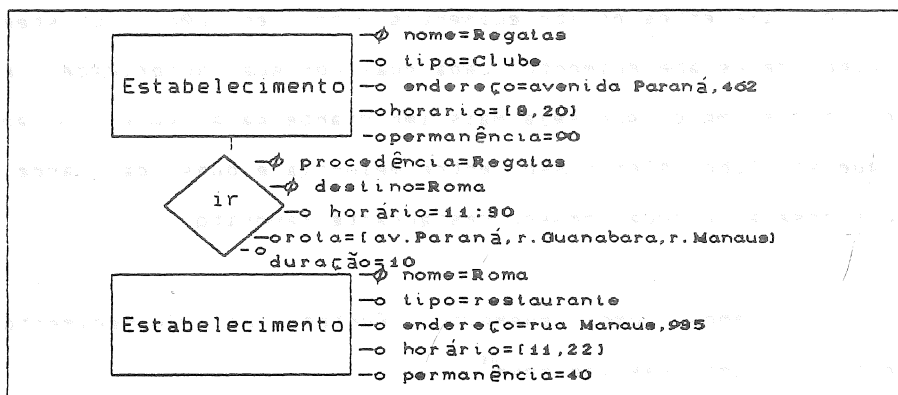


Figura 2 - Ilustração de Relacionamento

```
ir(regatas,roma,11.30,[av_parana,rua_guanabara,rua_manaus],10).
```

Os fatos (entidades e relacionamentos) presentes no blackboard são alterados pela ação das Fontes de Conhecimento, implementadas por regras Prolog.

4. OPERAÇÃO SOBRE O BLACKBOARD

Para solucionar o problema, as Fontes de Conhecimento agem modificando o blackboard. O procedimento das fontes é determinado por suas primitivas de ação, que por sua vez seguem uma estratégia de resolução de problemas.

Um exemplo é um sistema que determina uma rota em uma cidade, passando por diferentes estabelecimentos, como em [06]. Existem vários tipos de estabelecimento, cada qual com sua prioridade. É possível, por exemplo, que seja mais importante para você ir ao teatro que ao clube. Além disso, entre meio-dia e duas da tarde, você pode desejar almoçar em seu restaurante favorito.

Desta forma, temos como exemplo, Fontes de Conhecimento implementando primitivas top-down⁽¹⁾:

a) que geram soluções alternativas.

Pode haver mais de um caminho de resolução que nos leve a uma solução satisfatória para o problema. Muitas vezes é desejável manter as soluções alternativas, como ilustrado na figura 3. No problema da rota pela cidade, verificamos que existem muitos trajetos entre dois estabelecimentos. Mudando o trajeto, pode-se conseguir visitar outros estabelecimentos próximos ao novo

(1) Nem todos os atributos são mostrados por motivos de clareza.

caminho. Dependendo da estratégia de resolução, pode ser desejável manter ambos os percursos.

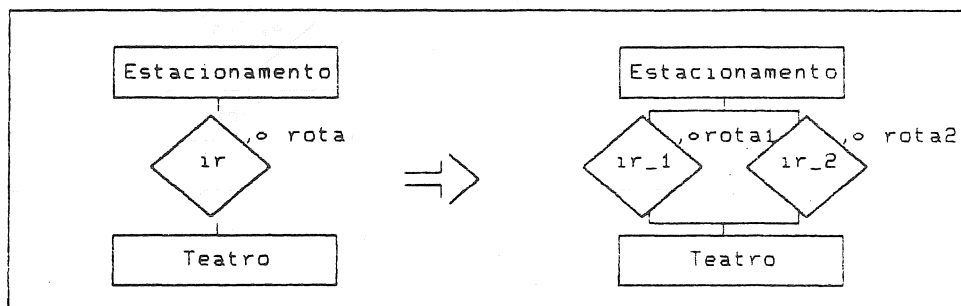


Figura 3 - Geração de soluções alternativas

b) que classificam as entidades.

Através da classificação de entidades, pode-se passar mais facilmente de um nível de abstração para outro, facilitando o raciocínio e a tomada de decisões (Figura 4). Classificar os estabelecimentos de acordo com sua localização pode, por exemplo, facilitar o trajeto na região. Isso viabiliza o planejamento das rotas por setores da cidade.

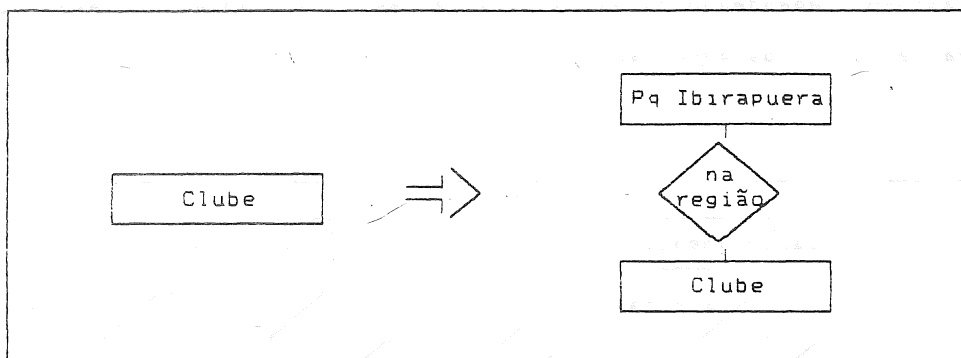


Figura 4 - Exemplo de Classificação

c) que dividem relacionamentos.

Podem existir entidades estreitamente relacionadas com outras. Muitas vezes, a representação dessa proximidade é desejável, e isto pode ser obtido através de divisão de relacionamentos (Figura 5). Na aplicação, pode ser proveitoso visitar certos estabelecimentos em sequência, por sua proximidade física, por exemplo. Além disso, se houver um relacionamento ir com horário=meio-dia, pode ser desejável ir a um restaurante antes de continuar o percurso.

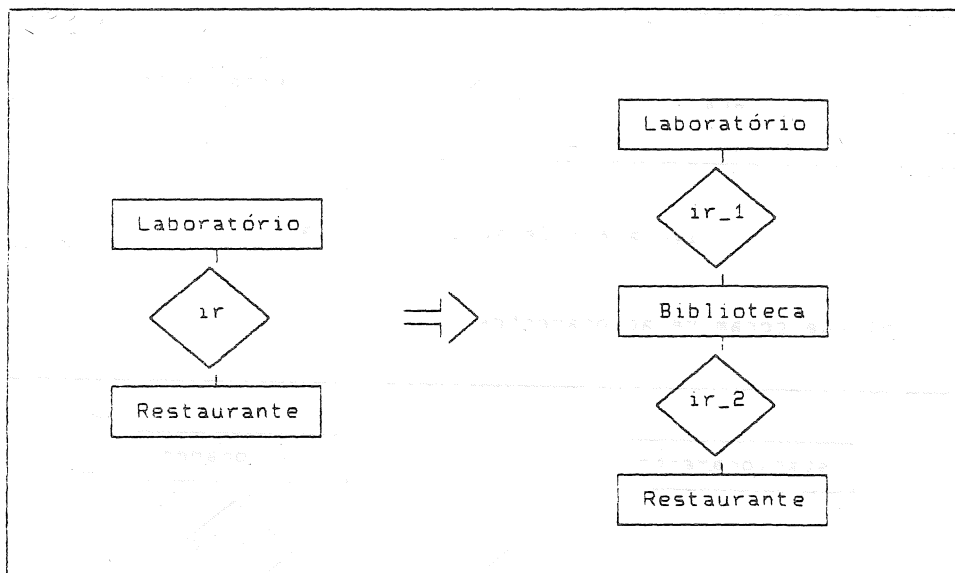


Figura 5 - Exemplo de Divisão de Relacionamento

Temos também as seguintes Fontes de Conhecimento implementando primitivas bottom-up:

a) que geram entidades.

Novas entidades passam a existir a partir da ocorrência de eventos externos, por exemplo. Primitivas de geração de entidades (Figura 6) inicializam sua existência no âmbito do sistema. Esse tipo de primitiva pode ser empregado, por exemplo, quando ao final do percurso houver tempo livre, inserindo-se um novo estabelecimento onde praticar esportes.

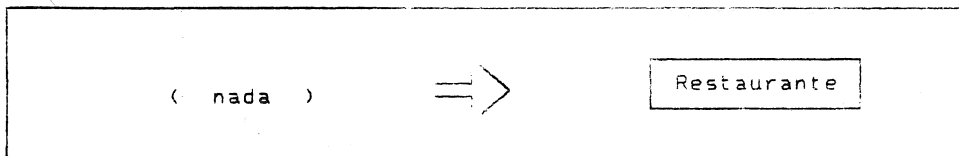


Figura 6 - Geração de entidade

b) que geram relacionamentos.

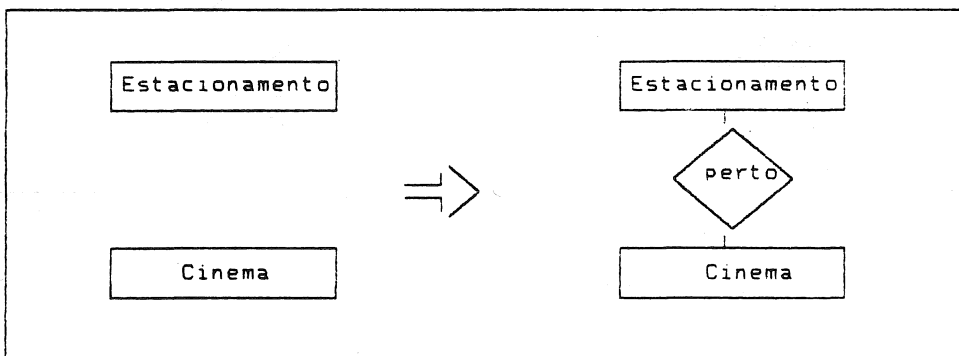


Figura 7 - Geração de relacionamentos

A geração de relacionamentos (Figura 7), além de contribuir para a construção da solução, é também uma maneira muito eficaz de melhorar o raciocínio das Fontes de Conhecimento, fornecendo a elas uma visão melhor do estado da solução.

c) que geram hierarquias.

A geração de hierarquias (Figura 8) permite que o sistema se abstraia de detalhes quando raciocina em níveis de abstração mais

altos. Este tipo de Fonte de Conhecimento pode também ser empregado de forma similar às de classificação, agrupando e privilegiando relacionamentos entre entidades afins.

Abaixo, temos um exemplo de classificação de entidades, que exemplifica como estas primitivas puderam ser implementadas:

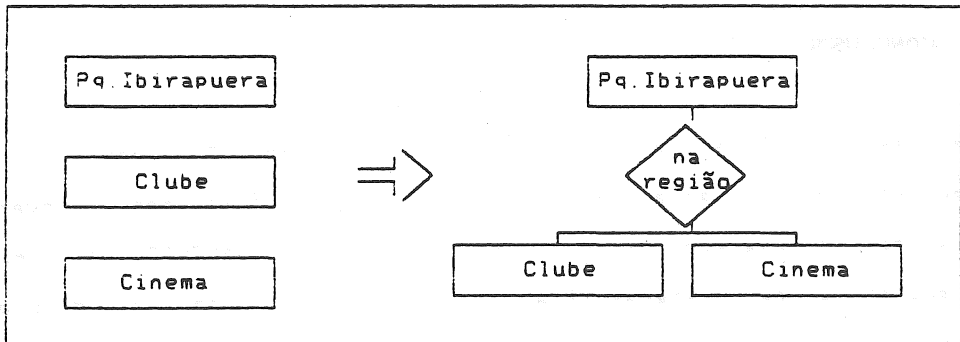


Figura 8 - Exemplo de geração de Hierarquia

```

ks_classificação_exemplo(região,Estabelecimento) :-
    na_região(Estabelecimento,Região),
    valida(Região),
    ins_blackboard(na_região,[Estabelecimento,Região]),
    ins_l_eventos(ger_classif,[na_região,Estabelecimento,Região]).
  
```

Nesse exemplo, ilustrado também pela figura 4, `na_região` é um predicado utilizado para determinar a região à qual o estabelecimento pertence; `valida` determina se a região determinada é de interesse e se o estabelecimento já não foi classificado. Os predicados `ins_blackboard` e `ins_l_eventos` introduzem o

relacionamento obtido no blackboard e na lista de eventos.

Todas as primitivas são tratadas como eventos pelo Módulo de Controle. Esse módulo simplesmente supervisiona a execução das Fontes de Conhecimento, auxiliando na escolha da estratégia mais apropriada.

5. CONCLUSÃO

Uma das características mais importantes do Modelo Blackboard é o oportunismo. Para alcançá-lo, as Fontes de Conhecimento devem enxergar o blackboard de forma apropriada. ASTUTO é uma Arquitetura de Sistemas Blackboard que utiliza o Modelo Entidade-Relacionamento para modelar e construir a solução, privilegiando a visão que as Fontes de Conhecimento têm do blackboard.

REFERÊNCIAS

- [01] CHEN, P.P., The Entity-Relationship Model: Toward a Unified View of Data, ACM Transactions on Database Systems 01, (1976).
- [02] CLOCKSIN, W.F. and MELLISH, C.S., Programming in Prolog, Springer-Verlag, 1984.
- [03] CORKILL, D., GALLAGHER, K., and JOHNSON, P., Achieving

Flexibility, Efficiency, and Generality in Blackboard Architectures, in Proceedings of the Sixth National Conference on Artificial Intelligence, (1987), 18-23.

- [04] ENGELMORE, R. and MORGAN, T., Blackboard Systems, Addison-Wesley Publ. Co., 1988.
- [05] ENGELMORE, R.S., and MORGAN, A.J., Introduction, in Blackboard Systems, Addison-Wesley Publ. Co., 1988.
- [06] HAYES-ROTH, B., and HEWETT, M., BB1: An Implementation of the Blackboard Control Architecture, in Blackboard Systems, Addison-Wesley Publ. Co., 1988.
- [07] NEWELL, A., Some Problems of the Basic Organization in Problem-Solving Programs, Proceedings of the Second Conference on Self-Organizing Systems, 1962, 393-323.
- [08] PARNAS, D.L., On Criteria to be Used in Decomposing Systems into Modules, CACM 15, (1972), 1053-1058.
- [09] PRESSMAN, R.S., Software Engineering - A Practitioner's Approach, McGraw-Hill Book Co., 1987.
- [10] SMITH, J.M., and SMITH, D.C., Database Abstractions: Aggregation and Generalization, ACM Transactions on Database Systems 02, 1977.
- [11] SofTech Inc., An Introduction to SADT, Manual, 1976.
- [12] WAGNER, C.F., and MEDEIROS, C.B., Um Modelo Binário Estendido para Entidade-Relacionamento, Anais do XIV Seminário Integrado de Software e Hardware, 1987, 164-174.