

RACIOCÍNIO BASEADO EM CASOS - UMA VISÃO GERAL E MÉTRICAS DE SIMILARIDADE

DOI: <https://doi.org/10.5281/zenodo.14188200>

PESSÔA, SÁ, Fábio, Mestrando*
NAKAMITI, G., Dr.**

* Faculdade de Tecnologia da Baixada Santista – Campus Praia Grande
Departamento de Informática Gestão de Negócios – Praça 19 de Janeiro 144, 11700-100
Praia Grande/SP - Fone (13) 3591-1303
fabio.sa@iron.com.br

** UNIP - Universidade Paulista - Rua Dr. Bacelar, 1212 - São Paulo/SP
UNISANTOS - Universidade Católica de Santos - Rua Carvalho de Mendonça, 144
5º andar - Santos/SP
g_nakamiti@uol.com.br

RESUMO

Raciocínio Baseado em Casos trata-se de um modelo computacional bastante utilizado em diversos sistemas inteligentes. Essa abordagem tem motivado pesquisadores a criarem sistemas para diferentes domínios, assim como possibilidades de sistemas integrados com outras abordagens graças à flexibilidade e simplicidade desse modelo. A seção “Métricas de Similaridade” apresenta alguns modelos de métricas mais utilizados e a seção “Perspectivas para o Futuro” ilustra algumas aplicações e tendências dessa área.

PALAVRAS-CHAVE: Raciocínio Baseado em Casos, Aplicações Help Desk, Métricas de Similaridade

ABSTRACT

Reasoning based in cases deals with computational models very much used in several intelligent systems. This approach has motivated researchers to create systems for different dominion as well as possibilities of integrated systems with other approaches thanks to the flexibility and simplicity of this model. The section “Metrics of Similarity” presents some models of metrics more used and the section “Perspectives for the Future” shows some applications and tendencies in this field.

KEY-WORDS: *Reasoning based in cases, Help Desk Applications, Metrics of Similarity*

INTRODUÇÃO

Freqüentemente, o termo “raciocínio” é explicado como um processo de relacionamento de idéias, começando do nada. O dicionário Aurélio confirma essa definição como um “encadeamento, aparentemente lógico, de juízos ou pensamentos”. Raciocínio baseado em casos (RBC) tem uma concepção bastante diferente. Em RBC, a origem do conhecimento não é gerada por regras, mas por uma memória de *casos* guardados, registrados em episódios anteriores (LEAKE, 1996).

Vários exemplos podem ilustrar essa importante abordagem da Inteligência Artificial (IA): empresas procuram profissionais qualificados através de anúncios, onde se destaca a busca por candidatos com experiência prévia em determinados serviços. É bem coerente que empresas de suporte em informática procurem candidatos que já tenham enfrentado problemas e encontrado soluções, caracterizando sua experiência pela prática adquirida no exercício constante de sua profissão. Tais profissionais adquiriram habilidades em suas tarefas enfrentando eventuais problemas com as respectivas soluções, resultando em diversos *casos*.

O profissional de suporte, em seu dia-a-dia, encontra soluções para seus problemas novos através da lembrança ou *recuperação* de problemas similares anteriores, que pode ser através da sua memória ou consulta a documentações. Encontra também problemas diferentes, mas com soluções parecidas, cabendo então a *adaptação* de um caso que ele conhecia.

No processo de remoção de um vírus de computador, por exemplo, o profissional de suporte em informática pode lembrar de uma solução utilizada anteriormente, ainda que seja para a remoção de um vírus diferente. Se funcionar, a solução deve ser *revisada* e devidamente *armazenada*, para soluções de problemas futuros, caracterizando um aprendizado desse profissional.

Isso leva a crer que um profissional qualificado mantém diversos casos em sua memória. Logo, quanto mais casos adquiridos, mais experiência e melhor capacidade para executar sua função farão parte do currículo deste profissional. Esse é um dos motivos porque chamamos uma *base de casos* também de *banco de experiências*.

Podemos utilizar o RBC em diversas situações. De um modo geral pode-se dizer que esse sistema reutiliza experiência prévia para: (NAKAMITI, FREITAS, 1999).

- a) explicar novas situações (suporte em informática);
- b) encontrar novas demandas;
- c) interpretar novas situações (advogados);
- d) criticar novas soluções;
- e) criar uma solução justa para um novo problema (juízes).

Estes exemplos ilustram a simplicidade do sistema RBC, mas merecem algumas explicações dos pré-requisitos, os passos essenciais, e as diferenças em comparação com outras abordagens de construção de sistemas baseados em conhecimento.

O primeiro e mais importante pré-requisito é a coleção de experiências, denominados *casos*, armazenados na base de casos. Cada caso consiste ao menos em parte da descrição do problema, denominado *problema*, e parte da descrição da solução, denominada *solução*.

Estes dois ingredientes básicos são normalmente enriquecidos por uma parte administrativa que pode incluir, por exemplo, uma identificação do caso, uma parte que explica ou justifica e que oferece mais informação sobre os passos da solução do problema em direção à solução, uma parte da descrição do contexto, e uma parte de avaliação que contém informações sobre a qualidade e a reutilização do caso.

Pode-se descrever a solução de um problema com um sistema RBC pelo seguinte: Um novo problema aparece e é descrito como parte de um novo caso. Então, casos antigos contendo problemas similares são recuperados, e o mais adequado à sua solução é escolhido a tornar-se solução do novo problema. Essa solução é então testada e eventualmente guardada como um novo caso. Este último passo caracteriza uma das importantes etapas desse sistema que é o aprendizado (SPÖRL, LENZ, HÜBNER, 1999).

Além da base de casos composto por casos que representam situações anteriores, os sistemas de raciocínio baseado em casos requerem mecanismos para recuperá-los da base de casos, adaptá-los para o caso corrente, validar a solução proposta e armazenar o conhecimento obtido durante o processo.

Na figura 1 são mostradas as etapas de um Ciclo RBC clássico, que muitas vezes podem ter suas nomenclaturas diferenciadas:

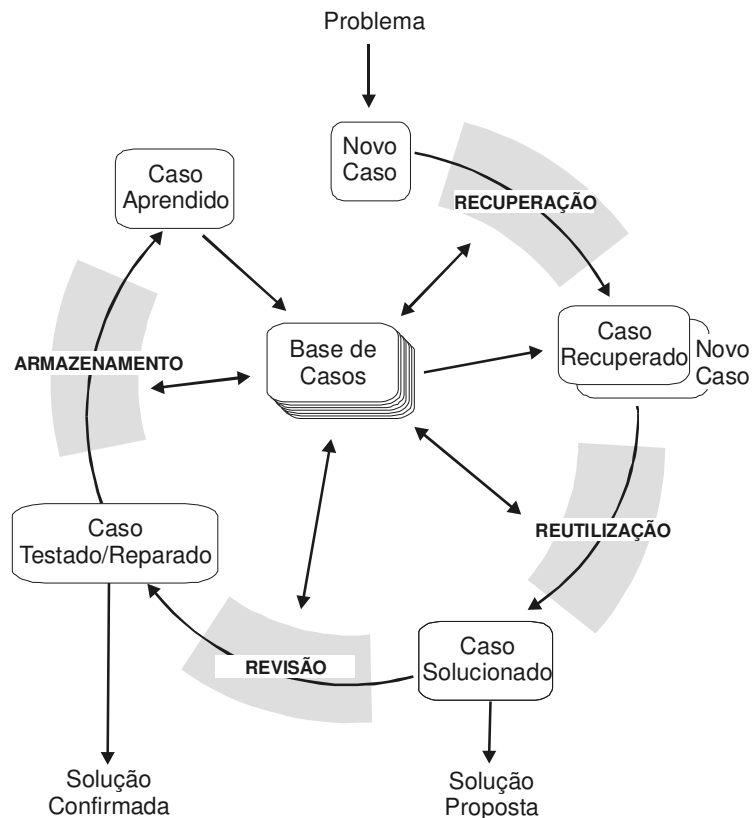


Figura 1. Ciclo RBC. Aamodt, Plaza (1994)

Pode-se citar os seguintes mecanismos encontrados no ciclo RBC:

Recuperação: Através de uma indexação adequada é que se pode recuperar casos de forma eficiente. A escolha dos itens que farão parte das chaves da indexação assim como sua organização na memória, determinará a eficiência da busca na base de casos.

Reutilização: Também chamado de Adaptação, este processo envolve a identificação das diferenças entre o caso recuperado e o caso corrente e a análise das partes do caso recuperado que podem ser transferidas para o caso corrente.

Revisão: Serve para saber se a solução gerada pelo sistema satisfaz as novas especificações. Podem ser feitas de formas diferentes, inclusive simulações.

Armazenamento: Uma vez armazenado, caracteriza-se de alguma forma um aprendizado, de modo a poder ser utilizado mais tarde. Destaca-se nessa etapa especificar o que será aprendido, separar o que é importante do desnecessário.

Na seção dois é apresentado um breve histórico dos sistemas de RBC. Na seção três são descritas formas de classificação desses sistemas, os tipos e suas principais características. Na seção quatro é apresentada a fundamentação teórica, onde se encontram entre outras definições, a do *contêiner do conhecimento*. A seção sete coloca uma das partes mais importantes de um sistema RBC, as métricas de similaridade, considerada por alguns autores como o núcleo do sistema. Em seguida, um destaque para aplicações *help-desk*, uma aplicação bastante natural para esses sistemas e, finalmente, uma seção sobre as perspectivas futuras onde pode-se encontrar aplicações para *web* e comércio eletrônico.

HISTÓRICO

Em 1977, Schank e Abelson propuseram que o conhecimento geral das pessoas sobre as situações está armazenado em *scripts*, permitindo que elas criem expectativas sobre o que ouvem, e assim, construam inferências sobre as relações entre as coisas das quais ouviram. Os *scripts* foram propostos como uma estrutura de memória conceitual, descrevendo informação sobre eventos estereotipados, como ir a um restaurante ou a uma consulta a um médico. Experimentos com *scripts* mostraram, entretanto, que eles não representam uma teoria completa de representação de memória, já que as pessoas confundem eventos que tem *scripts* similares. Os *scripts* fornecem apenas um tipo de conhecimento que as pessoas utilizam para o entendimento: elas se valem de também outros tipos de conhecimento, como o conhecimento sobre objetivos, planos, relações interpessoais e papéis efetuados pelas pessoas (KOLODNER, 1993).

Representações sobre estes tipos de conhecimento têm sido propostas e sistemas de computadores que usam estes tipos de conhecimento para o entendimento têm sido desenvolvidos.

Em 1982, Roger Schank propôs a teoria da memória dinâmica (*dynamic memory*) e o papel central que a lembrança de situações anteriores (episódios, casos) e modelos de situações (*scripts*, MOPs — pacotes de organização de memória) desempenham na resolução de problemas. Este trabalho, segundo Plaza (1994), apresenta as raízes do uso de raciocínio baseado em casos na Inteligência Artificial.

O primeiro sistema RBC, denominado CYRUS, foi desenvolvido por Janet Kolodner (1983). Este sistema, baseado no modelo de memória dinâmica de Schank e na teoria MOP de resolução de problemas e aprendizado, foi basicamente um sistema de questões e respostas com conhecimento de várias viagens e encontros do primeiro secretário de estado americano, Cyrus Vance. O modelo de memória de casos desenvolvido por este sistema serviu de base para outros diversos sistemas RBC, como MEDIATOR - (SIMPSON, 1985), PERSUADER - (SYCARA, 1988), CHEF - (HAMMOD, 1989), JULIA - (HINRICHS, 1992), CASEY - (KOTON, 1989).

Outra base de RBC e conjunto de modelos foi desenvolvida por Bruce Porter e seu grupo na Universidade do Texas, em Austin. Inicialmente trabalhando com o problema de aprendizado automático para classificação de tarefas, o grupo desenvolveu o sistema PROTOS - (PLAZA, 1994). Esse sistema enfatiza a integração do conhecimento geral do domínio e do conhecimento específico de casos em uma estrutura de representação unificada — um modelo de memória de casos. Além dessa, outra contribuição importante para a área foi o trabalho do grupo de Edwina Rissland na Universidade de Massachussetts, em Amherst, que desenvolveu o sistema HYPO, aplicado para o domínio do Direito.

Atualmente, as pesquisas em RBC têm estendido-se rapidamente, sendo percebido um crescente número de trabalhos envolvendo RBC em periódicos de IA e um aumento de aplicações comerciais de sucesso, além de pesquisas em desenvolvimento em diversos países. (WATSON, 1994).

CLASSIFICAÇÃO

Existem dois estilos básicos de raciocínio baseados em casos: solucionadores de problemas e interpretativos.

No estilo dos RBC solucionadores de problemas, soluções para o novo

problema são criadas usando soluções prévias. Soluções prévias podem oferecer soluções aproximadas para os novos problemas além de avisos de falhas ou erros potenciais. Podem ser usados em uma grande variedade de tarefas de solucionadores de problemas, incluindo:

- planejamentos - Ex. PLEXUS - (ALTERMAN, 1988), CHEF - (HAMMOND, 1989);
- diagnósticos - Ex. SHRINK - (KOLODNER, 1987), PROTOS - (BAREISS, 1989);
- projetos - Ex. JULIA - (HINRICHS, 1992).

Em cada um desses tipos de RBC solucionadores de problemas, os casos são úteis para sugerir soluções e avisar sobre possíveis problemas que possam surgir.

No estilo interpretativo de RBC, situações novas são avaliadas no contexto de situações prévias. Isso determina uma situação ou solução como uma entrada. Um advogado, por exemplo, faz uso de um raciocínio baseado em casos interpretativos quando ele utiliza uma série de casos prévios para justificar um argumento em um novo caso. Existem três tarefas onde o RBC interpretativo pode ser utilizado:

- justificação - Ex. HYPO;
- interpretação - Ex. HYPO - (ASHLEY, 1988);
- projeção - Ex. BATTLE PLANNER - (GOODMAN, 1989).

Esses tipos de RBC interpretativos dividem uma mesma linha de argumentação, alguns casos suportam somente uma interpretação ou efeito, outros suportarão várias interpretações ou efeitos. O sistema raciocinador deve comparar e diferenciar os casos entre eles para finalmente determinar uma solução.

FUNDAMENTAÇÕES TEÓRICAS

A seguir, serão apresentados alguns conceitos importantes para o estudo de sistemas de RBC.

Similaridade de Casos: Similaridade em RBC é uma tentativa de representar a utilidade de um caso prévio para resolver um problema de um

caso corrente. Deve-se notar que um caso pode ser considerado similar ao problema corrente se a solução deste caso pode ser facilmente reutilizada para o problema. Mas, em geral, isso pode ser determinado *depois* de tentar a reutilização dessa solução para o problema atual. Deve-se assumir também que, quanto mais o caso prévio for parecido com o problema atual, mais útil será a solução para o caso corrente.

Pode-se entender que o interesse sobre similaridade de casos se relaciona mais sobre a similaridade da descrição do problema prévio em relação ao problema atual.

Contêiner de Conhecimento em RBC: Cada um dos quatro mecanismos: recuperação, reutilização, revisão e armazenamento, denominados modelo R4 (figura 1) formam um tipo de “contêiner” de um número de técnicas refinadas que podem ser utilizadas para implementação de um sistema RBC. Recuperação, por exemplo, inclui a identificação das características relevantes, a pesquisa através da memória dos casos, o cálculo dos valores de similaridade e a seleção dos casos mais parecidos. A pesquisa pode se implementada por um número de técnicas alternativas como: indexação, pesquisa linear, pesquisa de estruturas especiais, etc.

O modelo de contêiner de conhecimento foi introduzido por Richter (1995). Ele é útil particularmente para descrever qual tipo de conhecimento pode ser incorporado em um sistema RBC e como tais componentes interagem. Um contêiner de conhecimento é usado para extrair conhecimento de uma determinada estrutura. Por exemplo, em sistemas baseados em regras tradicionais existem três contêineres: fatos, regras e o mecanismo de inferência.

Em RBC, quatro principais contêineres podem ser identificados:

- *vocabulário*, usado para descrever casos;
- *métrica de similaridade*, utilizada para comparar casos;
- *base de casos*, contendo todos os casos armazenados;
- *tecnologia de adaptação*, necessária para transferência de soluções.

Cada contêiner pode em princípio carregar e oferecer qualquer parte de conhecimento para resolver um problema corrente. Por exemplo, um sistema

RBC com uma base de casos completa se faz independente de outra parte de conhecimento, como a *métrica de similaridade* ou a *adaptação*. Ou seja, o sistema poderia ser implementado como um banco de dados tradicional.

Métodos para Recuperação de Casos: Nos últimos anos, um número de técnicas tem sido desenvolvidas de modo a implementar tarefas de recuperação. Ex.: pesquisa por casos similares a uma questão, incluindo:

– *kd-trees* combinando RBC e indução, visando deste modo, permitir uma recuperação baseada em similaridade e árvores de decisão. A vantagem desta abordagem é que métodos de RBC podem ser integrados com técnicas de aprendizado indutivo. Auriol, Wess, Manago, Althoff, Traphöner (1997). Isto parece ser particularmente útil para tarefas de diagnóstico. Lenz (1996).

– O modelo *Fish-and-Shrink*, desenvolvido com o projeto FABEL, parece ser particularmente útil para domínios onde há necessidade de casos altamente estruturados, como em projetos. Börner (1996).

– O modelo *Redes de Recuperação de Casos (RRC)*, por outro lado, é ideal para domínios com a representação de casos simples, e ainda assim, lidar com base de casos grandes. Lenz, Dburkhard (1996), o que é essencial em aplicações de comércio eletrônico. Wilke, Lenz, Wess (1996). A flexibilidade de modelos RRC permite sua utilização em casos de alta desestruturação, como em RBC's textuais.

Cada uma dessas técnicas tem suas propriedades específicas, vantagens e desvantagens. Ao desenvolver um sistema RBC em particular, o domínio deve ser analisado cuidadosamente de modo que se possa escolher qual método de recuperação mais apropriado.

Adaptação de Casos: Os casos similares recuperados por um sistema RBC muitas vezes não poderão ter uma aplicação direta em um problema corrente, dadas as diferenças entre as descrições dos problemas entre os casos. Conseqüentemente, as soluções devem ser ajustadas ao problema atual, caracterizando a adaptação. Em um caso geral, as adaptações se encaixam na mesma classe de problemas em uma solução baseada em conhecimento. Mas na prática existem situações onde adaptações podem ser bem mais fáceis de realizar, podendo ser percebidas e resolvidas.

RBC PARA APLICAÇÕES HELP-DESK

Nesta seção, consideramos RBC para *help-desk* como aplicações de diagnóstico para equipamentos técnicos fabricados em grande escala e utilizados por pessoas sem conhecimento na manutenção desses equipamentos. O fato desses equipamentos serem fabricados em grande escala aumenta a probabilidade de problemas parecidos ocorrerem com tais equipamentos, assim, a solução desses problemas pode ser reutilizada várias vezes. O fator de reutilização é muito importante para o cálculo do retorno do investimento nessa área.

Requisitos adicionais vêm do fato de que os usuários desses equipamentos não são especialistas e frequentemente podem fornecer informações incompletas e nem sempre corretas sobre o estado de seu equipamento e que o profissional de *help-desk* deve aparecer como um parceiro competente e experiente, mesmo sendo novo em seu emprego. O principal foco nessa situação recai no suporte à decisão - e nem tanto em encontrar causas precisas e explicações ao problema.

Existem ferramentas RBC disponíveis que são exatamente desenvolvidas para esse tipo de situação. Com estas ferramentas, um caso é usualmente descrito pela observação inicial de um sintoma e um diálogo com perguntas e respostas sobre fatos adicionais. A solução normalmente consiste em um pequeno texto que pode se facilmente comunicado, ou pode acompanhar um desenho, uma imagem ou mesmo uma animação. Spörl, Lenz, Hübner (1999).

Exemplos de aplicação: Existem diversos artigos sobre aplicações RBC *help-desk* de sucesso. Ex.: Compaq, IBM, HP, 3Com e diversas empresas de alta tecnologia. Na Alemanha existe uma lista enorme de clientes utilizando ferramentas RBC para inferência, telecomunicações e hardware de computadores, softwares e problemas de redes de computadores.

MÉTRICAS DE SIMILARIDADE

A similaridade é muito importante nas teorias que estudam o comportamento do homem ao solucionar problemas. Como consequência, nas teorias de Inteligência Artificial não podia ser diferente: elas têm grande importância no que diz respeito ao desenvolvimento de sistemas solucionadores de problemas.

Existem diversos estudos sobre a origem da similaridade. A pergunta “por que uma situação é considerada similar a outra?” não tem uma resposta exata. Ao consultar pesquisadores de IA, especialistas em recuperação, psicólogos, entre outros, percebe-se uma grande concordância de que o motivo que faz uma situação similar a outra é altamente dependente da tarefa que está sendo executada.

O caso recuperado é considerado relevante se suas características são similares às características mais importantes do novo caso. Bareiss (1989).

O quanto duas características são similares depende de como elas se correspondem e de como é organizada a estrutura da memória de casos.

Sabendo-se como medir a similaridade entre características, pode-se calcular o grau de similaridade entre os dois casos. Se há como computar valores numéricos para as similaridades entre as características pode-se utilizar esse método. Quando não há como computar um grau numérico, métodos heurísticos baseados em conhecimento são utilizados.

As métricas de similaridade podem ser definidas individualmente ou através de uma combinação de outras métricas, o que pode oferecer valores de tipos diferentes, possibilitando a criação de novas métricas de similaridade. Tais combinações podem oferecer um formalismo poderoso para construção de uma grande escala de métricas de similaridade aplicáveis em diversos domínios de problemas.

Vizinho-Mais-Próximo: Os métodos numéricos para medir similaridade mais usados têm sido o algoritmo “vizinho mais próximo”, ou adaptações dele. Pazzani (1991). Sua métrica de similaridade consiste no quociente dado pela soma dos graus de similaridades. O algoritmo original considera apenas dois valores para o grau de similaridade: 1 (um) para características iguais e o 0 (zero) para características diferentes. Os sistemas RBC geralmente usam adaptações deste algoritmo considerando o grau de similaridade variando no intervalo (0,1) e considerando a importância de cada característica.

$$\text{Grau_de_Similaridade} = \frac{\sum \{ \text{Similaridade}(\text{In}[i], \text{Ret}[i]) \times \text{weight}[i] \}}{\text{Soma_dos_Pesos}}$$

Figura 2. Equação do Vizinho-Mais-Próximo

Em Stottler (1992) há uma descrição de um sistema que utiliza o algoritmo vizinho-mais-próximo para computar o grau de similaridade entre o caso de entrada e cada um de seus casos armazenados. PROCASE - Yang, Lu & Lin, (1994), exemplifica um sistema que utiliza uma métrica de similaridade numérica híbrida. Inicialmente, ele computa um grau de similaridade parcial utilizando uma versão do algoritmo vizinho-mais-próximo apenas para as características geométricas dos planos (denominado avaliação estática). Em seguida, considerando as outras características do plano, ele usa um fator de ajustamento dinâmico, baseado em uma abordagem de pesos estatísticos, que pode incrementar ou decrementar esse grau de similaridade parcial.

Medidas de Contraste: A função de avaliação numérica, chamada *medida de contraste*, apresentada em Schild e Kerner (1994) para escolher os melhores PME (*multiple explanation pattern* - padrões de múltiplas explicações) no domínio de deliberações e decisões judiciais. Esta métrica de similaridade classifica os PE's (padrões de explicações) que compõem cada PME (caso da memória em relação aos fatos de um novo caso) como sendo: *diretamente relevantes, adaptivamente relevantes e irrelevantes*. Para cada classe de PE é considerado um somatório de seus pesos que é multiplicado por uma constante associada. No somatório dos PE's adaptivamente relevante, além dos pesos, são consideradas as diferenças entre cada PE e o seu correspondente no novo caso. São considerados, ainda, o número de fatos do novo caso sem correspondência com os PE's multiplicado por uma constante associada. O grau de similaridade é computado efetuando-se uma soma dos produtos, onde os produtos obtidos dos PE's diretamente e adaptivamente relevantes são parcelas positivas e os outros dois são parcelas negativas.

Métrica de Diferença de Valores: Definida por Stanfill e Waltz (1996), utiliza raciocínio baseado em memória para pronunciar palavras da língua inglesa. Esta métrica, como as variações do vizinho-mais-próximo, considera que as características têm pesos diferentes e que a similaridade de seus valores pode ser diferente da igualdade. O que diferencia é a forma como os pesos e a similaridade entre os valores são computados. O peso de cada característica muda com cada caso de entrada. Ele é a função do valor da característica no caso de entrada e da meta a ser alcançada pela resolução desse caso. Da mesma forma, a diferença entre o valor de uma característica do caso de entrada e o valor da mesma característica em um caso da memória

é a função desses próprios valores e das metas dos dois casos (da meta a ser realizada no caso de entrada e da meta já realizada pelo caso da memória). Essa métrica é bastante eficiente em termos de precisão na recuperação. O problema é que o cálculo dos pesos e das diferenças é grande consumidor de tempo, tornando a métrica inviável para ser implantada em máquinas seqüenciais, podendo ser aplicada em máquinas paralelas.

Métodos Heurísticos: Quando não há como computar um valor numérico para avaliar a similaridade entre as características do novo e do velho caso, métodos heurísticos são utilizados para analisar os dois casos e avaliar se a solução do velho caso não é incompatível com os elementos do novo caso. CASEY é um exemplo de um sistema que usa tais métodos para resolver as diferenças que ocorrerem nos valores das características de mesmo nome Kolodner, (1993). Amparado por um modelo causal e usando um conjunto de *regras de evidências* ele tem como avaliar se o diagnóstico do velho caso se aplica ao novo caso, mesmo quando a mesma característica se apresenta com valores opostos nos dois casos. Esse método não tem como atribuir um grau de similaridade, ele apenas diz que a solução do velho caso é compatível com os elementos do novo caso. Quando não há compatibilidade, o velho caso é descartado.

Modelo de recuperação com função de crença: Nesse modelo é suposto que cada caso é uma unidade independente armazenada em uma memória linearmente estruturada. Uma função de similaridade numérica, escolhida pelo implementador, é usada para computar o grau de similaridade parcial de cada caso em memória contra o caso de entrada. O conjunto de características a ser usado para representar cada caso, bem como a atribuição de importância para cada característica em relação a solução do caso, são de responsabilidade do implementador do sistema, juntamente com os especialistas do domínio considerado. Supõe-se que a função de similaridade escolhida atribui um valor numérico de similaridade parcial para todos os casos da memória. O grau de similaridade de cada caso da memória contra o caso de entrada é considerado como um valor de crença que favorece a solução do caso. Esse valor é considerado como sendo a crença fornecida pela função de crença definida pelas evidências do caso da memória contra o caso de entrada.

PERSPECTIVAS PARA O FUTURO

Embora nunca seja tão fácil fazer previsões para o futuro, podemos nos basear em algumas tendências ocorridas nessa área, principalmente na Web. São serviços que dispensam a interferência pessoal na resolução de problemas. Podemos imaginar uma FAQ (*Frequent Answered Questions*) mais dinâmica e funcional, espécies de serviços *self-services* na Web, onde os clientes podem resolver seus problemas com seu microcomputador. Já existem algumas aplicações desse tipo em uso. Bons exemplos de solucionadores de problemas *self-service* da empresa Lucas Art, para jogos de computadores ou Siemens *Automation and Drivers* para técnicos. No cenário do comércio eletrônico, podemos encontrar serviços de agências de turismo ou equipamentos de computadores, e ainda para vendas de cartões de créditos. As interfaces dessas aplicações *web* se tornam mais animadas e rápidas conforme a velocidade de transporte de dados aumenta. Isto permite desenvolvedores de sistemas RBC modelares e apresentarem soluções nas mais variadas formas do que texto puro, como imagens, vídeos ou mesmo em ambientes 3D (Spörl, Lenz, Hubner, 1999).

Outra tendência importante é o desejo de transmitir como solução, não apenas as informações que o cliente deve fazer para resolver seu problema em específico, mas enviar um agente reparador ativo para auxiliar o cliente na solução de seu problema em um caso extremo - ex. quando a ação reparadora consistir na ativação de um programa ou na instalação de um novo software - e deixar o agente tratar do problema (Spörl, Lenz, Hubner, 1999).

Uma abordagem bastante promissora é considerar a existência de casos como documentos. Perguntas do usuário podem ser aplicadas em um sistema RBC para encontrar o documento mais relevante em relação à pergunta. Um número considerável de técnicas alternativas para isso tem sido desenvolvida nos últimos anos (Lenz, 1998). *CBR-Answers* é um exemplo de uma ferramenta particularmente desenvolvida para suporte e aplicações *help-desk*. A idéia é analisar os documentos existentes, como FAQs, e encontrar o documento mais relevante. Para isso, um número de camadas de conhecimento são utilizadas, cada qual contendo um tipo específico de conhecimento sobre a aplicação do domínio em particular (Lenz, 1998).

Fica bastante clara a perspectiva otimista para todos problemas de RBC serem resolvidos, mas pode haver também a tendência de que essa tecnologia

fique menos identificável como parte de um sistema. Isto tem a ver com o fato de que raciocínio baseado em casos tem sido mais utilizado como um componente de um sistema integrado do que uma aplicação isolada.

CONCLUSÕES

RBC não assume que o mundo - de acordo com os modelos que fazemos do mundo - é ideal em termos de que ele é realmente estável, completo e consistente. De um ponto de vista teórico, fica difícil resolver problemas práticos em domínios de aplicação que não possam ser modelados com uma representação lógica formal baseada em conhecimento. Isso pode ser um dos motivos porque modelos híbridos têm surgido na tentativa de agregar o modelo RBC a outras abordagens menos parecidas com o mundo real.

Notamos que o interesse nesse sub-campo da IA fica por conta da própria essência do RBC: sua fácil compreensão é resultado dos termos utilizados pelo sistema que nos remetem à rotina do dia-a-dia, à maneira humana de aprender e coletar experiências, e mesmo descartar algumas, a cada instante de nossas vidas. Assim, o trabalho procurou contribuir para melhor compreensão dessa área, ou até mesmo para motivação no desenvolvimento de sistemas RBC.

REFERÊNCIAS BIBLIOGRÁFICAS

AAMODT, A., Plaza, E.: *Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches*. Published in AICom - Artificial Intelligence Communications, IOS Press, Vol. 7: 1, pp. 39-59. 1994

ALTERMAN, R.: *Adaptive planning*. Cognitive Science 12. 393-422. 1988

ASHLEY, K. D.: *Modeling Legal Argument: Reasoning with Cases and Hypotheticals*. Ph.D. Dissertation, COINS Technical Report No. 88-01, Department of Computer and Information Science, University of Massachusetts, Amherst. 1988

AURIOL, E., S. Wess, M. Manago, K. D. Althoff, and R. Traphöner.: *INRECA: A seamlessly integrated system based on inductive inference and case-based reasoning*. Em Aamodt and Veloso, pages 371-380. 1997

BAREISS, E. R.: *Exemplar-Based Knowledge Acquisition: A Unified Approach to Concept Representation, Classification, and Learning*. Academic Press, Boston, MA. 1989

BAREISS, R., and King J.: *Similarity Assessment in Case-Based Reasoning*. Proceedings of the 1989 DARPA workshop on case-based reasoning, Morgan Kaufman, Pensacola Beach, FL, pp. 67-71. 1989

BÖRNER, K.: *CBR for Design*. In *Case Based Reasoning Technology – From Foundation to Application*. Chapter 8, pages 201-234. 1996

GOODMAN, M.: *CBR in battle planning*. In Hammond, K. (Ed.), *Proceedings: Case-Based Reasoning Workshop (DARPA), II*, Morgan-Kaufmann. Pensacola Beach, FL. 1989.

HAMMOND, K. J.: *Case-Based Planning: Viewing Planning as a Memory Task*. Academic Press, Boston, MA. 1989.

Hinrichs, T. R.: *Problem solving in open worlds*. Lawrence Erlbaum Associates, 1992.

KOLODNER, J. L.: *Maintaining organization in a dynamic long-term memory*. *Cognitive Science*, Vol. 7, s.243-280. 1983

KOLODNER, J. L.: *Case-Based Reasoning*. San Mateo: Morgan Kaufmann, 1993.

KOTON, Phyllis: *Using experience in learning and problem solving*. Massachusetts Institute of Technology, Laboratory of Computer Science (Ph.D. diss, October 1988). MIT/LCS/TR-441. 1989.

LEAKE, D. B.: Appears in Leake, D., editor; *Case-Based Reasoning: Experiences, Lessons, and Future Directions*. AAAI Press/MIT Press, 1996.

LENZ, Mario and H. Dburkhard: *Case Retrieval Nets: Basic ideas and extensions*. Em Görz an Hölldobler. Pages 227-239. 1996.

LENZ, Mario: *CBR for Diagnosis and Decision Support*. Burkhard, P. Pirk, E. Auriol, and M. Manago.. *AI Communication*, 9(3): 138-146, 1996

NAKAMITI, G. e FREITAS, R.: *Raciocínio Baseado em Casos: Uma visão Geral*. 1999

PAZZANI, Cain T. M. J. & Silverstein, G.: *Using Domain Knowledge to Influence Similarity Judgments*. In: Proceedings Case-Based Reasoning Workshop, May/91. Washington, D.C., Morgan Kaufmann Publishers, 191-199, 1991

RICHTER, M. M.: *The knowledge contained in similarity measures*. Invited Talk at ICCBR-1995

SCHANK, R. C.: *Dynamic Memory, A Theory of Reminding and Learning in Computers and People*. New York, Cambridge University Press, 1982

SCHILD, U. J. & Kerner, Y.: *Multiple Explanation Patterns*. In: *Topics in Case Based Reasoning*. First European Workshop, EWCBR-93. Berlin, Germany, Springer-Verlag, 353, 364. 1994.

SIMPSON, Robert L.: *A computer model of case-based reasoning in problem solving: An investigation in the domain of dispute mediation*. Technical Report GIT-ICS-85/18, Georgia Institute of Technology. 1985.

SPÖRL, Lenz, Hübner: *Case-Based Reasoning - Survey and Future Directions*. To appear in Proc. XPS-99, Springer Verlag, LNAI

STANFILL, C. & Waltz, D.: *Towards Memory-Based Reasoning*. Communication of ACM, vol 29 no. 12;1213-1228. 1996.

STOTTLER, R. H.: *Case-Based Reasoning for Bid Preparation*. AI Experts, 44-49, March 1992.

SYCARA, K.: *Using case-based reasoning for plan adaptation and repair*. Proceedings Case-Based Reasoning Workshop, DARPA. Clearwater Beach, Florida. Morgan Kaufmann, pp. 425-434. 1988.

WATSON, Ian; Marir, Farhi.: *Case-based Reasoning: A review*. The Knowledge Engineering Review, London, v.9, n.4, p. 327-354, 1994.

WILKE, W., Lenz, Mario and Wess, S.: *Intelligent Sales Support with CBR. In Case Based Reasoning Technology – From Foundation to Application*. 1996.

YANG, H., Lu, W. F. & Lin, A. C. *PROCASE: A Case-Based Process Planning System for Matching of Rotational Parts*. Journal of Intelligent Manufacturing, vol. 5:430, 1994