

SISTEMA DE HELP-DESK UTILIZANDO RACIOCÍNIO BASEADO EM CASOS

Fábio Ferreira¹, Marcello Thiry²

¹fabioferreira@iname.com,

Bolsista PIBIC/CNpq
Universidade do Vale do Itajaí
Campus São José, UNITEC

²thiry@sj.univali.br

Universidade do Vale do Itajaí
Campus São José, UNITEC.

RESUMO

Este artigo apresenta o desenvolvimento de um sistema para diagnosticar problemas na área de informática através da Internet, usando um banco de soluções existentes. O ambiente utiliza a técnica de Raciocínio Baseado em Casos (RBC) para a modelagem da base de conhecimento. Quando um técnico infere sobre um problema, ele utiliza suas experiências anteriores para solucionar a situação presente, entretanto, para uma corporação é altamente dispendioso manter uma equipe treinada à disposição dos usuários. A partir de um processo aquisição, análise e adaptação de diversos registros sobre problemas conhecidos, foi definida uma estrutura adequada para a representação de um caso. Além disso, procurou-se trabalhar com mecanismos de comparação eficientes para a comparação de textos livres. O objetivo foi desenvolver um sistema de *help-desk* que poderá ser utilizado diretamente pelo usuário final através da Internet.

Palavras-chave: Inteligência Artificial, Raciocínio Baseado em Casos, Sistemas de diagnóstico, Sistemas *help-desk*.

1. Introdução

O atual ambiente comercial, dinâmico e competitivo, tem exigido das corporações grande atenção no sentido de manter-se como empresas atraentes, diferenciadas e produtivas.

A melhoria do atendimento aos clientes como extensão do serviço prestado ou produto vendido, tem sido neste sentido um ponto ultimamente abordado com muita frequência. Tal preocupação é justificável considerando-se que um bom serviço de suporte é, não raramente, um grande diferenciador da empresa no mercado. É com esta mentalidade que muitas corporações estão procurando implantar centros de atendimento ou *help-desks*, onde profissionais treinados estão aptos a auxiliar no esclarecimento de eventuais problemas. Entretanto, para disponibilizarem estes serviços, muitas corporações têm se deparado com a questão custo. A implantação desta serviços exige um alto nível de investimento, pois são necessários investimentos em estrutura física, em *softwares* específicos como os de gerenciamento de chamadas e, principalmente, em capacitação dos profissionais envolvidos.

Com o intuito de minimizar estes problemas, estas empresas vêm na implantação de centros de atendimentos inteligentemente automatizados uma solução. Elas buscam, junto ao mercado de informática, aplicações que lhes proporcionem não só uma redução de custos, mas também uma elevação da qualidade destes serviços.

É neste contexto que gira o propósito deste trabalho: desenvolver um ambiente para um serviço de suporte de auto-atendimento, utilizando-se para isto recursos da Inteligência Artificial.

2. Raciocínio Baseado em Casos (RBC)

O princípio básico de funcionamento do Raciocínio Baseado em Casos é primeiramente procurar na sua base de casos por situações já resolvidas ocorridas anteriormente e posteriormente, aplicar a solução desta situação em uma nova situação, ajustando-a de acordo com as diferenças entre o problema atual e o problema anterior. Esta técnica esta fundamentada na característica que os seres humanos apresentam em modificar a estratégia de solução de um novo problema com base no conhecimento e experiência adquiridos na solução de situações similares.

O RBC é baseado em pesquisas da psicologia cognitiva, na qual o processo de reutilizar episódios passados é defendido como uma forma freqüente e poderosa para aprendizado humano. Fundamenta-se então na técnica de buscar uma solução para uma situação presente (caso de entrada) através da comparação deste caso, com situações já ocorridas (casos passados). Leake (1996) infere que problemas similares têm soluções similares e os tipos de problemas se repetem.

Segundo Riesbeck e Shank (1989), RBC propõe-se a resolver novos problemas adaptando soluções que resolveram problemas no passado.

A maioria dos autores como Watson (1997) e Aamodt e Plaza (1994) definem o ciclo RBC em quatro etapas genéricas, que são: Recuperar, Reutilizar, Revisar e Reter. Watson (1997) refere-se a essas etapas como os *quatro REs*:

1. **Recuperar** : consiste em buscar na base de casos, um ou mais casos similares ao caso de entrada. A critério da similaridade se dá por comparação.
2. **Reutilizar** : após a recuperação de um ou mais casos na fase anterior, tenta aplicar o conhecimento destes casos para a solução do problema atual.
3. **Revisar** : avalia a solução sugerida na fase anterior;
4. **Reter** : representa o processo de aprendizado de um sistema RBC. Nesta fase é armazenado o caso de entrada com sua respectiva solução já revisada.

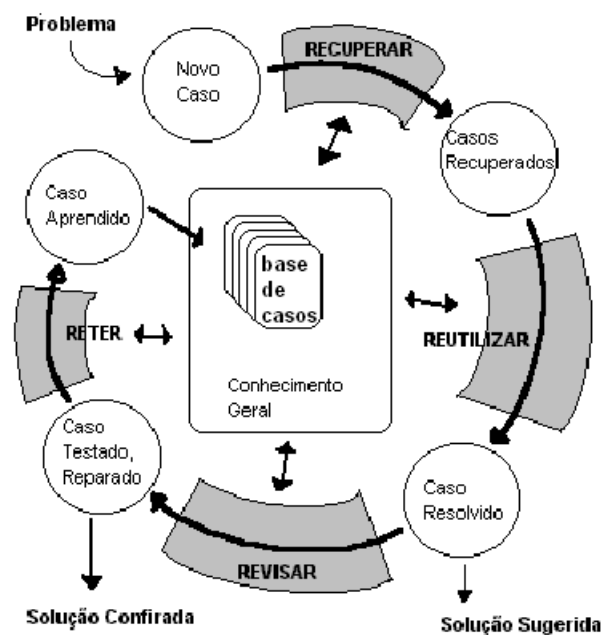


Figura 1 – O Ciclo RBC, fonte Aamodt e Plaza (1994).

Quanto às etapas de desenvolvimento de RBC, encontra-se uma divergência entre os autores, entretanto é mais comum seguir a seqüência de seis etapas que são:

1. Representação do caso
2. Indexação
3. Recuperação
4. Adaptação
5. Aprendizagem
6. Análise da situação

É importante ainda ressaltar que, dependendo do objetivo do sistema RBC a ser desenvolvido, nem todas as etapas acima necessitam ocorrer, como é o caso da fase de adaptação, que pode ser suprimida nos sistemas interpretativos onde se foca a interpretação do caso atual.

2.1 Representação do Caso

A representação dos casos no enfoque de sistemas RBC é a representação do conhecimento, pois é principalmente no caso que está o conhecimento que possibilitará a resolução de um problema.

A primeira tarefa da representação do caso é determinar o que se armazenará como um caso, ou seja, qual conhecimento será considerado, e como será esta estrutura de armazenagem.

Conforme Watson (1997), um caso é uma parte do conhecimento contextualizada que representa uma experiência. Ele contém experiência passada que é o conteúdo do caso e o contexto no qual a experiência pode ser aplicada. Ainda segundo Watson, um caso é representado por um registro e deve conter tipicamente:

- A descrição do problema, que descreve a situação onde o caso ocorreu e,
- A descrição da solução, que é o detalhamento da solução do problema e os resultados da aplicação desta solução ao problema. Ela deve ainda conter ainda a forma com que a solução foi obtida e se for o caso, de justificativas para a escolha desta solução.

2.2 Indexação

É a indexação, a essência do raciocínio baseado em casos, uma vez que, visando a recuperação, orienta o sistema na tarefa de avaliação da similaridade. Fazendo uma analogia a banco de dados, a indexação funciona como os índices, permitindo uma aceleração na recuperação dos casos.

Os índices de um caso são combinações de importantes indicadores de descrição, capazes de estabelecer distinções entre os casos existentes (Kolodner e Leake, 1996). No processo de recuperação o algoritmo empregado utilizará estes índices para encontrar e selecionar os casos similares.

2.3 Recuperação dos Casos

A recuperação dos casos ou também chamada de busca, é a fase onde se avalia, segundo algum critério e um determinado limiar, a similaridade de todos os casos da memória com o caso que representa o problema de entrada. O resultado desta busca é um conjunto de casos mais similares.

A tarefa de recuperação de casos possui três etapas: métrica de similaridade, recuperação de casos e seleção do melhor caso.

Métrica de Similaridade – é a medida de similaridade entre dois casos, ou seja, é a responsável pela síntese de similaridade ao nível de cada atributo através de uma medida da importância que é usada para modelar a relevância de cada atributo na avaliação sintética da similaridade..

Recuperação de Casos – é o momento em que se aciona uma função, que por utilizar-se de métricas ou por ser orientada a metas ou restrições, é capaz de recuperar os casos mais similares. Esta recuperação exige que se estabeleça um limiar para definir um limite de casos a serem recuperados e/ou um limite da medida de similaridade, para que se recupere apenas casos que superem esse limite.

Processo de Seleção – a importância desta fase reside no fato de que seu resultado é exatamente a saída do sistema. Para a realização desta tarefa utilizam-se de heurísticas, regras, perguntas ao usuário ou a combinação desses métodos. A participação de usuários pode ser diretamente na seleção do melhor caso ou apenas com contribuições ao sistema com informações adicionais que o auxiliem nesta seleção.

2.4 Adaptação

Embora designada apenas como Adaptação, esta fase relaciona-se com a revisão e adaptação do caso solução. Ela ocorre após o sistema ter selecionado e recuperado os casos destinados à solução do problema de entrada.

A etapa de adaptação consiste na avaliação das diferenças entre o problema escolhido e o problema de entrada.

Segundo Aamodt e Plaza (1994) há duas possibilidades: a solução proposta obteve sucesso, logo o sistema deve aprender com este caso, ou a solução proposta falhou, indicando que esta solução deve ser adaptada.

A adaptação deverá modificar um caso afim de torna-lo solução para o problema de entrada, utilizando-se para isto, de conhecimentos específicos do domínio do problema.

2.5 Aprendizagem

O processo de aprendizagem em sistemas RBC consiste na atualização da sua base de casos após um problema ter sido resolvido, ou seja, incorporar informações relevantes e de sucesso a sua base de casos com intuito de auxiliar na resolução de novos problemas.

A aprendizagem pode ocorrer ao nível da base de casos ou ao nível dos casos. Quando ao nível da base de casos, ocorre uma extensão da base, através de processos incrementais.

A aprendizagem ao nível da base permite que a partir de um caso semente obtenha-se novos casos, pois após o sistema ter interpretado ou solucionado um caso de entrada, estes serão submetidos a adaptação, gerando assim novos casos.

Já o aprendizado ao nível dos casos, refere-se a inclusão do desempenho de seu uso na resolução de um determinado problema de entrada, acrescentando desta forma o(s) seu(s) sucesso(s) ou fracasso(s). Este tipo de aprendizagem permite que o sistema evita a reutilização de sugestões de soluções menos favoráveis e ainda, possibilita ao usuário prever os resultados do uso de uma dada solução.

2.6 Ajuste da Situação

Esta fase se justifica sempre que o caso de entrada não estiver representado da mesma forma que os casos da base.

O objetivo desta fase é então, modelar o caso de entrada nos termos empregados nos casos da base, de forma a permitir que o caso de entrada possa:

- Ser recuperado quando adicionado a base de casos;
- Ser comparado aos casos candidatos ao nível das mesmas dimensões;

Este processo de ajuste da situação deve ser incremental, ou seja, deve ocorrer sucessivamente durante a recuperação refinando a indexação até o momento em que a recuperação de um caso similar for eficientemente concluída.

3. Sistemas de Help-Desk

Os *Help-Desks* (HD) são uma estrutura complexa, composta pela combinação de recursos humanos, processos e ferramentas. Estes elementos são requeridos da seguinte forma:

Recursos humanos – pessoal tecnicamente capacitado e em condições de solucionar um problema em um tempo válido. Deve ser ainda disponível para efetuar treinamentos que elevem a habilidade e os seus níveis de conhecimento.

Processos – identificar o problema e resolve-lo é o processo chave da operação do *help-desk*, assim, devem existir processos que reconheçam o tipo de chamado/problema e a encaminhe para um nível de pessoal mais ou menos técnico, conforme o caso.

Ferramentas – as centrais de *help-desk* devem possuir uma infra-estrutura tecnológica sofisticada, desde sistemas telefônicos, computadores, recursos de integração de telefonia-computador, recursos de voz até softwares especializados que disponibilizem ao operador : (1) informações sobre o equipamento do usuário como configuração do hardware e software; (2) recursos que permitam buscar problemas e soluções passadas.

Dentro das corporações tem-se constatado que os sistemas de HD têm apresentado pontos problemáticos. Os pontos mais levantados pelos gerentes dos *help-desks* são:

1. **Treinamento:** para serem efetivados, os operadores destes sistemas, necessitam de um conhecimento considerável sobre os produtos a serem suportados. Se uma companhia possui muitas linhas de produtos, ela necessitará de equipes para cada tipo de produto;
2. **Retenção:** os sistemas de atendimento possuem uma alta rotatividade de pessoal, fazendo com que a companhia tenha gastos consideráveis com os frequentes treinamentos;
3. **Novos produtos:** quando novos produtos são lançados, impossibilitam que os operadores de atendimento conheçam os novos problemas que surgirão. Conseqüentemente esses sistemas requerem um mecanismo que retenha o conhecimento dos operadores, para cada novo problema solucionado.

No intuito de solucionar estes problemas, muitos dos HDs estão sendo apoiados em softwares automatizados que provêm suporte vinte e quatro horas, reduzindo o custo de suporte e provendo um suporte mais eficiente. Estas novas ferramentas incluem bases de conhecimento e técnicas de análise que incorporam sistemas especialistas, raciocínio baseado em casos e outras técnicas.

Segundo Madeline Locke (1998), cada vez mais as corporações estão permitindo a seus usuários, maiores acessos a suas bases de dados de *help-desk* e os novos modelos de *help-desk* estão sendo baseados na *web*, desta forma, permite a usuários remotos o acesso a base de dados de problemas e soluções comuns, onde se pode realizar uma pesquisa inicial na tentativa de solucionar o problema, registrar um chamado ou verificar o status de problemas não resolvidos, tudo sem o envolvimento de recurso humano direto.

4. Modelo Desenvolvido

Durante a etapa de desenvolvimento optou-se em utilizar uma ferramenta que auxiliasse neste processo. Após um levantamento inicial, constatou-se que existe atualmente no mercado um grande número de ferramentas que facilitam a construção de aplicativos de RBC (Watson, 1997). Entretanto, adotou-se o ambiente CBR Works da Tecinno Inc. Esta ferramenta, além de ser utilizada em larga escala, oferece uma versão de demonstração funcional.

No ambiente CBR Works deve-se iniciar a construção de um sistema RBC com a modelagem da representação do domínio de trabalho. No contexto deste trabalho, este domínio é um sistema de *help-desk* para problemas de informática. Como pode ser observado na figura 2 que

representa o *Concept Manager* do CBR-Works, ficou interessante, para efeitos de futuras indexações e recuperações, dividir o domínio em dois sub-conceitos: Sistema Operacional e Aplicativo.

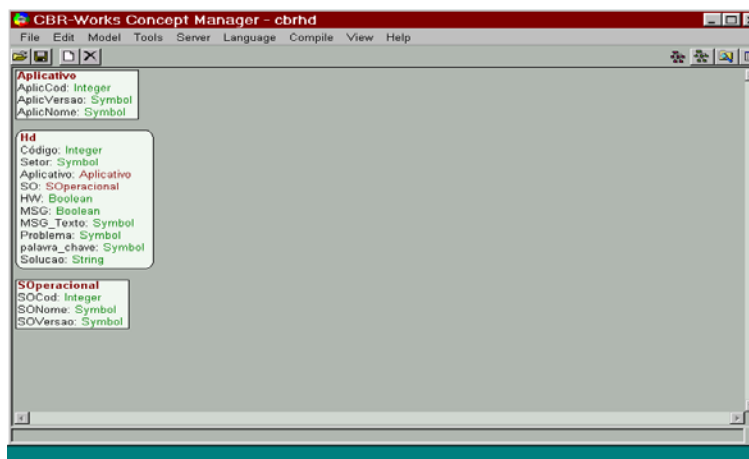


Figura 2 - Gerenciador de conceitos do CBR Works.

O conceitos *Aplicativo* e *SOperacional* apresentam atributos que definem o nome e a versão do Aplicativo/Sistema Operacional. Estes dois conceitos são chaves para a formulação do conceito maior, pois podem ter grande influência na determinação de similaridade de um caso com outro. Para criação dos conceitos acima foi necessária a especificação de tipos de dados, como é o caso do atributo SONome do sub-conceito SOperacional. Este foi definido como do tipo básico *symbol* para poder formar o tipo SOperacional do super-conceito HD (*help-desk*). Sendo do tipo *symbol*, o CBR Works permite que se enumere ocorrências para este atributo, onde também será possível imputar a similaridade entre estas ocorrências.

A figura 3 mostra exemplos de ocorrências pré-definidas para o atributo SO e os graus de similaridade existentes entre elas.

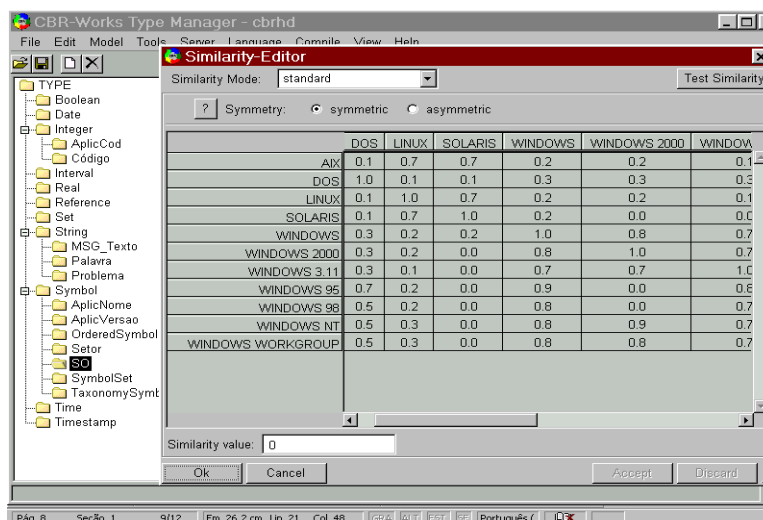


Figura 3 - Ocorrências e graus de similaridade

Outro ponto importante para a concretização do conceito HD é imputar os pesos para cada um de seus atributos, ou seja, especificar qual o grau de importância terá aquele atributo no momento da recuperação/comparação dos casos. Esta tarefa exige que se tenha uma base inicial de casos onde se possa realizar experimentações com o intuito de se especificar os melhores valores.

Desta forma a montagem de uma base de casos experimental passou a ser fundamental. Com entrevistas ao pessoal de suporte e buscas de FAQ's (*Frequently Asked Questions*) na Internet coletaram-se dados que posteriormente adaptados formaram a base inicial de casos (150 ocorrências). A figura 4 exemplifica dois casos modelados e com suas respectivas soluções.

Attributes	19	14
Aplicativo	7	7
AplicCod	7	7
AplicNome	INTERNET EXPLORER	INTERNET EXPLORER
AplicVersao	4.0	4.0
CodTq	19	14
HW	false	false
MSG	true	false
MSG.Texto	Internet Explorer não é corrente	?
palavra_chave	navegador	download
Problema	Internet Explorer não é corrente	O botão de radio. Salvar este aq
Setor	BIBLIOTECA	SOAE
SO	4	4
SOCod	4	4
SONome	WINDOWS 95	WINDOWS 95
SOVersao	?	?
Solucao	Há um outro navegador instalad	O IE 4.0 permite apenas 4 dow

Figura 4 - Exemplo de caso modelado

Tendo o conceito definido e com um número de casos considerável, realizou-se uma bateria de testes para avaliar o mecanismo de recuperação com relação a uma suposta situação de entrada. Verificou-se então a necessidade de se utilizar um método mais preciso para comparações de dados do tipo "string", como é o caso do atributo "Problema". Entretanto, o CBR Works não dispõe de um modo avançado de comparação para este tipo de dado. A solução encontrada foi o desenvolvimento de um método mais apurado no ambiente Delphi/ObjectPascal capaz de extrair destes atributos quais as palavras de maior relevância e compará-las em quantidade, grafia e em sinônimos. Desta forma, tornou-se também essencial a utilização de um dicionário de sinônimos para o contexto de *help-desk*, onde um termo como "computador" pode possuir um grau de similaridade com as palavras "micro", "pc" e "microcomputador".

Com esta abordagem o conceito HD ficou dividido em dois grupos, um englobando os atributos que definem um grau de similaridade através das matrizes de similaridades, e outro que aborda os atributos "string" onde a similaridade é obtida por uma métrica de similaridade de

atributo. Para este segundo grupo foi especificado um algoritmo que pode ser sintetizado pela equação: $SimilaridadeTexto = (NPCe \div NPCb) \times (\sum SPEnc \times NPEnc)^1$.

Com a aplicação deste algoritmo foi possível obter valores mais precisos para as similaridades dos atributos Problema, Descrição da Mensagem e Palavra Chave. Estes valores juntamente com as similaridades dos demais atributos são aplicados na métrica de similaridade, retornando deste modo o valor de similaridade do caso. A métrica de similaridade do caso é expressa pela equação: $SimilaridadeCaso = \sum [SimilaridadeAtributo(x) \times Peso(x)]$.

Migrando-se o modelo em CBR-Works para o ambiente Delphi e agregando-se estas implementações, obteve-se um protótipo funcional no qual pode-se realizar uma nova bateria de testes. Percebeu-se então uma grande evolução no sistema, representada pelas satisfatórias recuperações.

O projeto atualmente caminha no sentido de melhorar ainda mais o processo de comparação dos atributos “string”. Tais melhoramentos estão sendo propostos através da ampliação do dicionário de sinônimos para que este se torne sensível ao contexto, e através da especificação de índices para a base de casos via identificação do tipo de problema nestes atributos, incluindo aqui o atributo Solução.

5. Considerações Finais

Sabe-se que a modernização na forma de atuação das empresas é uma necessidade inevitável. Toda a corporação que se nega a assimilar as novas tendências tecnológicas acaba tornando-se uma empresa não competitiva. Dentre estas inovações, estão a implantação de serviços automatizados de atendimento a clientes.

Após o estudo do ambiente *Web* da Internet, comprovou-se que este pode ser perfeitamente empregado como plataforma de desenvolvimento para um sistema de *help-desk*. A *Web* não só permite a implementação de um aplicativo que supra as atuais necessidades das empresas, como também agrega a estes, princípios de agilidade, qualidade e economia.

Durante o desenvolvimento deste projeto verificou-se que para o processo de recuperação de casos onde a maior parte da caracterização do problema reside em atributos textuais, torna-se necessário utilizar métodos específicos para manipulação destes tipos de dados.

¹ NPCe = Número de Palavras do Caso de Entrada; NPCb = Número de Palavras do Caso da Base; SPEnc = Similaridade da Palavras Encontradas; NPEnc = Número de Palavras Encontradas

A eficácia do sistema pode ser verificada se colocado em um ambiente de produção, bastando para isto a adição de mais casos a sua base de conhecimento.

Um dos objetivos do projeto, que era a de aplicar a técnica de RBC no desenvolvimento de um sistema de *help-desk*, foi alcançada com sucesso mostrando-se muito condizente, e comprovando assim, as muitas aplicações de sucesso já implementadas com esta tecnologia.

A idéia de utilizar experiências passadas na resolução de novas situações é algo que se faz naturalmente, e muito mais intensamente ainda na área da resolução de problemas de informática. O RBC consegue representar esta habilidade com resultados muito interessantes.

6. Referências Bibliográficas

- [Aam94] AAMODT, A. e PLAZA, E., **Case-Based Reasoning: Foundational Issues, Methodological Variations, and Systems Approaches**. Artificial Intelligence Communications, Vol. 7, No. 1, 1994.
- [Kol93] KOLODNER, J. **Case-Based Reasoning**. Morgan Kaufmann, 1993.
- [Kol96] KOLODNER, J. e LEAKE, D. **A Tutorial Introduction to Case-Based Reasoning**. Case-Based Reasoning: Experiences, Lessons, & Future Directions. Ed. David B. Leake, AAAI Press, 1996.
- [Lea96] LEAKE, D. B. **Case-Based Reasoning: Experiences, Lessons, & Future Directions**. AAAI Press, 1996.
- [Loc99] LOCKE, Madeline, **Helping Yor Help Desk: Seven Consideration to Increase Effectiveness**. SD, [OnLine] Disponível na internet via www através do endereço URL: <http://www.tmcnet.com/ccs/>. Obtido em 1999.
- [Rie89] RIESBECK C.K., SCHANK R.C. **Inside Case-Based Reasoning**, Erlbaum, Hillsdale, NJ, 1989.
- [Wat97] WATSON, I. **Applying Case-Based Reasoning: Techniques for Enterprise Systems**. Morgan Kaufmann, 1997.