

ISSN 1415-2061

REVISTA DO CCEI

Centro de Ciências da Economia e Informática

Volume 6 Número 9

MARÇO 2002

BAGÉ - RS

EDITORA DA URCAMP - EDIURCAMP

Rev. CCEI	BAGÉ - RS	V.6	N.9	Mar. 2002
-----------	-----------	-----	-----	-----------

REVISTA DO CCEI

ISSN 1415-2061

Revista do Centro de Ciências da Economia e Informática da Universidade da Região da Campanha (URCAMP), Bagé, RS, é uma publicação regular, de divulgação técnico-científica, editada pela Editora da URCAMP - EDIURCAMP.

URCAMP - Universidade da Região da Campanha

REITOR:

Prof. Morvan Meirelles Ferrugem

VICE-REITOR ACADÊMICO:

Prof. Francisco Arno Vaz da Cunha

VICE-REITOR DE ADMINISTRAÇÃO:

Prof. Roberto Nocchi Cachapuz

PRÓ-REITOR DE PÓS-GRADUAÇÃO, PESQUISA E EXTENSÃO:

Prof. Evaldo Rodrigues Soares

PRÓ-REITORA DE ASSUNTOS COMUNITÁRIOS

Prof^a. Angelina Feltrin Quintana

DIRETOR DO CENTRO DE CIÊNCIAS DA ECONOMIA E INFORMÁTICA:

Prof. Enio Del Geloso Nocchi

COORDENADOR DO CURSO DE INFORMÁTICA (*campus* Bagé):

Prof. Cristiano Cachapuz e Lima

COORDENADOR DOS CURSOS DE CIÊNCIAS CONTÁBEIS e TRANSAÇÕES

IMOBILIÁRIAS (*campus* Bagé):

Prof. Eduardo Roman Souza

COORDENADOR DO CURSO DE ADMINISTRAÇÃO, HABILITAÇÕES: EMPRESAS,

RURAL e GESTÃO DE SISTEMAS DE INFORMAÇÃO (*campus* Bagé):

Prof. Enio Del Geloso Nocchi

CAPA: Marsal Alves Branco

REVISÃO: Prof^a Lenice Barreto

COMPOSIÇÃO E IMPRESSÃO: CECOM - Centro de Comunicações URCAMP

Toda correspondência sobre assuntos ligados à Revista do CCEI deverá ser enviada para:

Universidade da Região da Campanha - URCAMP
Centro de Ciências da Economia e Informática
Av. General Osório, 2289
Cep 96400-101 - Bagé - RS - Brasil
revista@ccei.uncamp.tche.br

É permitida a reprodução com menção da fonte de artigos sem reserva de direitos autorais.

Aceita-se permuta.

Revista do CCEI / Universidade da Região da Campanha. v.1 n.1
(out.1997). - Bagé: URCAMP, 1997-
1415-2061
Semestral

2002. v.6 n.9

1. Economia - Periódicos. 2. Informática - Periódicos. 3. Administração de Empresas - Periódicos.

Catálogo Sistema de Bibliotecas/URCAMP

REVISTA DO CCEI

v.6, n.9, 2002

CONSELHO EDITORIAL:

Direção do CCEI:

Prof. Enio Del Geloso Nocchi, M.Sc.
Prof. Cristiano Cachapuz e Lima
Prof. Eduardo Roman Souza

Informática:

Prof. Luiz Cláudio Dalmolin, M.Sc.
Prof^a. Lóren Pinto Ferreira, M.Sc.

Ciências Contábeis:

Prof. Sérgio da Fonte Abreu
Prof. Flávio Garibaldi

Ciências Econômicas:

Prof. Carlos Storniollo
Prof^a. Marilene Silveira
Prof. Roberto Nocchi Cachapuz

Administração – Habil. Empresas:

Prof. Edar da Silva Añaña, M.Sc.
Prof^a. Nara Beatriz Pires da Luz, M.Sc.

Administração – Habil. Rural:

Prof. Cláudio Marques Ribeiro, M.Sc.

Administração – Habil. Gestão de Sistemas de Informação:

Prof. Cláudio Sonáglia Albano, M.Sc.
Prof. Léu Cardoso Carate, M.Sc.

Editor:

Prof. Luiz Cláudio Dalmolin, M.Sc.

Assessores Técnicos:

Prof^a. Ada M.M. Guimarães, M.Sc.
Prof. Cláudio Marques Ribeiro, M.Sc.
Prof. Cristiano Cachapuz e Lima
Prof. Edar da Silva Añaña, M.Sc.
Prof. Enio Del Geloso Nocchi, M.Sc.
Prof^a. Elza Maria Steinhorst
Prof^a. Jhansy Collares, M.Sc.
Prof. Léu Cardoso Carate, M.Sc.
Prof^a. Lenice Barreto
Prof. Luiz Cláudio Dalmolin, M.Sc.
Bibl. Maria Bartira N. Costa Taborda
Bibl. Nelci Maria Birk Jeismann
Prof. Salvador Loni Tadeu Camargo, M.Sc.

Colaboração (Abstracts):

Richard Grant Murray

EDITORIAL

Quando, em meados de 1997 idealizou-se a produção de uma publicação científica do Centro de Ciências da Economia e Informática, pensava-se em criar uma revista com periodicidade regular, semestral e com abrangência regional, principalmente no âmbito de atuação da Universidade da Região da Campanha - URCAMP, que possui 8 *campi* no sudoeste do Rio Grande do Sul.

Nas primeiras edições, houve alguns entraves, como: poucos artigos encaminhados, pouca participação do corpo docente e discente da URCAMP, dificuldades em relação à formatação de artigos científicos por parte dos autores, dificuldades em relação ao processo de seleção, entre outros. Mas, após a publicação de algumas edições, essas dificuldades foram sendo superadas e a Revista do CCEI, começou a ser reconhecida, também fora da região de abrangência da URCAMP, com artigos provenientes das mais diversas regiões do País e até do exterior.

Isso tudo não foi por acaso. Todos os envolvidos: URCAMP, Direção do CCEI, Coordenadores, Editores, Conselho Editorial, autores e demais colaboradores têm sua parcela de trabalho e merecem desfrutar esta conquista.

Tem-se, assim, com a publicação desta 9ª edição, mais uma fonte de pesquisa e estudos, que pretende contribuir na formação de profissionais das áreas de atuação do Centro de Ciências da Economia e Informática.

SUMÁRIO

Usando grupos de meta-objetos reconfiguráveis dinamicamente no suporte a atonicidade; FERNANDES, Acauan P.	7
Mineração de dados paralela; CAMARGO, Sandro da S.	16
Teste de software orientado a objeto baseado em estados; SILVEIRA, Fábio F.; PRICE, Ana M. de A	24
Uma ferramenta para auxiliar a avaliação de textos construídos colaborativamente em ambientes de ensino-aprendizagem; GUEDES, Gilleanes T. A.; VICCARI, Rosa M.; D'AMICO, Carmen B.	32
Clube virtual de ciências; BET, Sabrina; FROZZA, Ângelo A.	41
Os supermercados e a tecnologia de informação; GONÇALVES, Lóren P. F.; GONÇALVES, Everaldo L.	49
Arquitetura de aplicações voltadas para <i>Web</i> ; LACERDA, Guilherme S. de; CASTRO, Alexandre Ramires de; ZAVALIK, Claudimir	55
Robótica aplicada à assessoria de instrumentação em cirurgias médicas; CAMINHA, Flávia L.; GARSKE, Michele D'M.	62
Estudo da mortalidade infantil no sul do RS com o uso do Geoprocessamento; PEDROSO, Bibiana S.; VIEIRA, Lúcia A. S. LEON, Maria E. dos S.; GARSKE, Michele D'M.; GULARTE, Moisés	68
A insalubridade deve ser calculada sobre a remuneração e não mais sobre o mínimo; VICENZI, Benjamin.	75

USANDO GRUPOS DE META-OBJETOS RECONFIGURÁVEIS DINAMICAMENTE NO SUPORTE A ATOMICIDADE

Acauan Pereira Fernandes¹

RESUMO

Uma das características mais importantes que aplicações tolerantes a falhas devem apresentar é suporte a atomicidade. Os problemas que desenvolvedores de tais programas devem enfrentar são diversos e podem variar de acordo com o contexto do ambiente da aplicação. Este artigo emprega diversas técnicas na construção de um conjunto de meta-classes como uma camada adicional para monitorar os objetos da aplicação, no intuito de conferir suporte a atomicidade de forma transparente, dinamicamente reconfigurável e reutilizável. As técnicas utilizadas são discutidas, procurando-se definir o papel de cada uma das mesmas na introdução de atomicidade ao mesmo tempo em que omite-se os detalhes de implementação da aplicação. Busca-se independência das classes da aplicação e capacidade de auto-reconfiguração dinâmica para que a mesma possa lidar com as mudanças no ambiente sem que isto demande em alterações em seu código, ou mesmo que seus módulos percebam tais eventos.

Palavras-chave: atomicidade, reflexão computacional, middleware reflexivo, reutilização de software.

ABSTRACT

One of the most important characteristics that fault tolerant applications must demonstrate is support of atomicity. The problems that developers of such applications face are diverse and can vary according to the context of the application environment. This article employs various techniques in the construction of a package of meta-classes as an additional layer for monitoring the objects of the application in the intention of creating support for atomicity in a manner that is transparent, dynamically reconfigurable and reusable. The techniques utilized are discusses, attempting to define the purpose of each of them in the introduction of atomicity while omitting the implementation details of the application. Making the classes independent of any application and the capacity for dynamic auto-reconfiguration permits these classes to work with changes to the environment without requiring the alteration of code.

1. INTRODUÇÃO

Atomicidade é um dos requisitos não funcionais mais importantes que um desenvolvedor deve ter em mente quando se trata de implementar aplicações confiáveis. A diversidade de situações e contextos, com a conseqüente enorme quantidade de demandas que as mesmas introduzem, é um dos maiores problemas a serem enfrentados. É seguramente uma tarefa árdua tentar produzir código que cubra diferentes tipos de aplicações e suas demandas específicas usando métodos tradicionais. No intuito de construir soluções para um maior número de situações sem aumentar o grau de complexidade do desenvolvimento de aplicações, é necessário empregar novas técnicas que auxiliem o aumento da flexibilidade e adaptabilidade da aplicação às mudanças no contexto de seu ambiente. Este esforço também leva à construção de soluções mais reutilizáveis.

Deve-se prover uma aplicação com meios para que a mesma seja capaz de se auto-monitorar e, de acordo com o contexto corrente, realizar alterações para melhor lidar com

¹ Bacharel em Ciência da Computação (UCPel) e Agronomia (URCAMP). Mestre em Ciência da Computação pelo convênio URCAMP/UFRGS. Professor do Centro de Ciências da Economia e Informática e do Centro de Ciências da Saúde (URCAMP).

os novos requisitos apresentados. Uma questão a ser respondida quando da execução desta tarefa é como incluir em uma aplicação tais características de forma transparente. Isto não deve aumentar o grau de complexidade para o desenvolvedor. Introspecção, auto-reconfiguração dinâmica e transparência são conceitos chave na obtenção deste objetivo. Este artigo mostra uma combinação de diversas técnicas na construção de uma camada intermediária transparente, flexível, auto-reconfigurável e altamente adaptável para dar suporte à atômica e, objetos de uma aplicação, permitindo que a mesma seja reutilizada sem exigir qualquer alteração na aplicação, ou mesmo que a mesma tenha ciência de sua existência.

2. REFLEXÃO COMPUTACIONAL

O conceito de reflexão computacional foi primeiramente introduzido por Smith em 1982 [SMI 82] e difundido por Maes [MAE 87]. Corresponde à capacidade de um sistema de observar suas próprias estruturas internas e seus respectivos estados correntes, assim como uma representação de sua execução e, a partir deste conhecimento, tomar decisões sobre seu comportamento. Este pode ser alterado em tempo de execução pelo próprio sistema [BLA 00]. O sistema é conectado causalmente, ou seja, as alterações promovidas sobre sua representação refletem-se sobre seu estado corrente e comportamento, e vice-versa [COU 00]. De acordo com Maes, as estruturas internas e o domínio que representam são ligados de forma que se a alteração de uma delas afeta a outra.

O uso de reflexão pode aumentar a reutilização de uma aplicação, pois esconde os detalhes de implementação do programador. Isto faz com que algumas decisões devam ser tomadas no lugar da aplicação [BLA 99]. Para que isto seja possível, é necessário obter informações sobre a mesma em tempo de execução.

2.1. Reflexão computacional e reconfiguração dinâmica

Aplicações dinamicamente reconfiguráveis permitem modificações em sua arquitetura sem afetar seus módulos adjacentes. Isto não é possível em configurações estáticas, porque estas agrupam seus componentes em tempo de compilação. Configurações dinâmicas, entretanto, precisam de componentes extras responsáveis pela carga e destruição destes objetos adicionais. Esse é o preço a ser pago na obtenção de maior flexibilidade [ROM 0].

O emprego de reflexão permite a inspeção de mudanças no ambiente computacional com conseqüente adaptação da aplicação a estas através da modificação ou alteração de um método, ou até pela adição de novos [BLA 99]. Neste artigo, isto é exemplificado pelas alterações que as políticas de controle de concorrência e recuperação sofrem de acordo com o contexto do ambiente da aplicação. A arquitetura aqui introduzida foi desenvolvida tendo classes em Java [SUN 99] e o protocolo de meta-objetos Guaraná [OLI 99] em mente.

O conjunto de métodos de um objeto pode ser inspecionado e modificado dinamicamente. Até a classe de um objeto pode ser alterada em tempo de execução. Também é possível adicionar ou excluir não apenas métodos, mas também atributos de um objeto [COS 99]. Novos serviços podem ser adicionados a um objeto usando seu meta-objeto para inserir novos métodos [COS 99]. Este papel é desempenhado pela reflexão computacional quando empregada na construção de aplicações dinamicamente reconfiguradas.

Reflexão computacional é um caminho natural para obter-se reconfiguração dinâmica. Mais transparência e reuso podem ser logrados através da utilização de classes no meta-nível, que é o local mais apropriado para as mesmas.

2.2. Um *middleware* reflexivo

Middlewares são serviços que residem em uma camada intermediária entre a aplicação e o sistema operacional, no intuito de facilitar o desenvolvimento e gerenciamento de aplicações distribuídas [COU 00]. Estas aplicações são muitas vezes heterogêneas, de forma que um dos aspectos mais importantes a serem considerados na construção de *middlewares* é como mascarar tal heterogeneidade, fornecendo interfaces e métodos padrão de acesso.

Para executar tal tarefa de forma apropriada, um *middleware* deve ser capaz de reconfigurar seus componentes em tempo de execução, de acordo com os novos contextos nos quais estiver inserido. Há uma tendência de aplicação da tecnologia de componentes de processamento distribuído (CORBA, JavaRMI e DCOM). Estas tecnologias são usadas em nível de aplicação. Entretanto, seu uso pode ser estendido a *middlewares* [COU 00].

Middlewares devem ser capazes de adaptar-se de acordo com o contexto e as mudanças passíveis de surgimento em tempo de execução. Para possibilitar tal reconfigurabilidade e adaptabilidade, devem ser construídos utilizando-se engenharia aberta e permitindo inspeção e manipulação dinâmicas de seus componentes. Coulson [COU 99] sugere o uso de reflexão com uma forma de prover a necessária abertura exigida pelo *middleware* sem comprometer sua integridade. O uso de reflexão na construção de uma camada entre a aplicação e os módulos de suporte a seus requisitos não funcionais pode trazer novas possibilidades no que concerne a adição de flexibilidade e adaptabilidade.

Reflexão torna a aplicação mais adaptável a seu ambiente e mais capaz de lidar com mudanças inesperadas no mesmo. [BLA 99] considera a camada intermediária (um *middleware*) o local mais correto para situar a reflexão em um ambiente distribuído. Esta combinação também tem como conseqüência tornar tal camada mais apta a lidar com um maior número de demandas, especialmente quando se lida com requisitos funcionais, pois é muito difícil abranger tais requisitos com uma única implementação. O uso de reflexão computacional torna as aplicações menores e mais livres de restrições impostas por políticas estáticas de controle de concorrência e recuperação.

O desenvolvimento de *middlewares* é tarefa árdua para os programadores. Uma solução natural é a introdução de reflexão para ajudar a suportar a configuração de novos serviços para a aplicação [BLA 99]. *Middlewares* construídos sem tais considerações em mente apresentam limitações a sua flexibilidade, pois devem ser capazes de detectar alterações no ambiente e customizar a si próprios de forma a acomodar tais situações [ROM 00]. É extremamente difícil desenvolver um conjunto fixo de políticas e mecanismos para lidar com aspectos concernentes a atômica e suas demandas. Estratégias diferentes podem ser carregadas quando do início da execução da aplicação e trocadas mais adiante. Isto deve ser feito, entretanto, de modo a não tornar a aplicação grande ou complexa demais. Para que tal não ocorra, instâncias de classes podem ser criadas apenas quando necessárias e destruídas imediatamente após o término de sua utilidade.

O uso de configurações múltiplas permite conexões ponto-a-ponto entre meta-objetos e objetos no nível base, o que leva a um controle maior sobre o suporte provido pelo *middleware*, ou na forma de relacionamentos um-para-muitos, mais útil em situações onde uma única ação deve atingir vários objetos simultaneamente. *Middlewares* reflexivos permitem a interoperação de dispositivos heterogêneos, porém é preciso fornecer mecanismos para disparar as adaptações que eventualmente se façam necessárias.

2.3. Reflexão e o modelo orientado a objetos

O modelo orientado a objetos faz com o uso de reflexão computacional uma combinação particularmente produtiva. A decomposição de funcionalidades em módulos torna-os mais reutilizáveis, pois apenas a funcionalidade requerida está presente em cada módulo [ROM 00]. Este modelo é especialmente apropriado para organizar grupos de componentes [HAE 96], sendo uma das mais importantes características provenientes da junção destes conceitos. Kiczales [KIC 91] também explora uma combinação de reflexão computacional com o modelo orientado a objetos, sugerindo o uso do primeiro como forma de possibilitar alterações neste.

2.4. Agrupando meta-objetos

Grupos de objetos (Figura 1) são conjuntos de elementos computacionais semelhantes organizados para cooperar em certas tarefas ou para prover redundância. O grupo é externamente visto como um único objeto, atuando e sendo percebido como uma única entidade. Um de seus possíveis usos é no suporte a atomicidade. Objetos confiáveis podem ser obtidos via redundância, alcançada por diversidade de projeto ou replicação [HAE 96]. Desta forma, diferentes objetos implementando diversas políticas podem ser unidos para formar um grupo coordenado por um objeto que inclua algum mecanismo de decisão. Outra forma é replicar os objetos e usar um coordenador para conectá-los a seu grupo de réplicas. Operações que forem enviadas a um objeto são desviadas para suas réplicas e, através de um mecanismo de votação implementado no objeto coordenador, os resultados provenientes de cada réplica são testados e um escolhido.

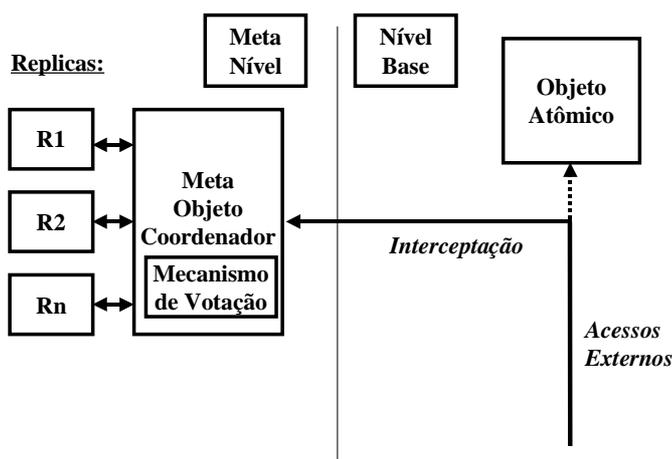


Figura 1 – Grupos de meta-objetos

A coordenação do grupo pode ser executada usando-se meta-objetos. No MOP Guaraná, uma instância da meta-classe Composer delega a execução de serviços a outros meta-objetos. Este se comporta como coordenador dos meta-objetos membros de uma meta-configuração múltipla. Pode haver mais de um meta-objeto com a mesma função. Cada um pode ter sido implementado de forma distinta, caracterizando assim diversidade de projeto. A instância de Composer pode enviar uma operação qualquer a todos os meta-objetos ligados a si e tomar decisões para aumentar a confiabilidade do resultado comparando as respostas obtidas. Este procedimento pode ser usado para gerar um histórico e, dinamicamente, excluir meta-objetos criados com problemas de projeto. Estes meta-objetos também podem ser construídos implementando diferentes políticas e algoritmos, permitindo a instância de Composer escolher os mais apropriados para cada situação.

Novos meta-objetos podem ser acrescentados dinamicamente ou ainda excluídos da meta-configuração (Figura 2).

O MOP Guaraná fornece algumas meta-classes que podem ser úteis na composição de configurações múltiplas, o que implica em aumento de sua efetividade e reuso, pois novas a adição e exclusão de objetos não exige alterações na aplicação no nível base. Mecanismos de decisão movidos para o meta-nível podem monitorar e selecionar as meta-classes disponíveis, liberando a aplicação de tal tarefa. Isto leva a soluções mais genéricas e reutilizáveis.

Diversos mecanismos de tolerância a falhas podem ser implementados utilizando-se o conceito de grupos de objetos, entretanto se estes forem colocados no meta-nível, seu reuso pode ser dramaticamente aumentado. Grupos de objetos podem ser usados para implementar diversas técnicas relativas a um único propósito, ou então compor um grupo de instâncias de uma única classe com um mecanismo de votação. A mesma operação é submetida a todas estas instâncias e o objeto implementando o mecanismo de votação recebe seus resultados e chega a uma conclusão sobre sua validade. Estes objetos podem estar distribuídos, o que significa que algum eventual problema de conexão apenas retirará um objeto de ação, mas o conjunto poderá continuar o trabalho sem maiores prejuízos. Qualquer falha pode ser percebida pelo resultado diferentes ou falta do mesmo. Esta configuração com instâncias da mesma classe não é suficiente para lidar com problemas de falhas de projeto. Neste caso, uma solução melhor é desenvolver classes diferentes com diferentes mecanismos. Uma classe que implemente um mecanismo de votação ainda será necessária.

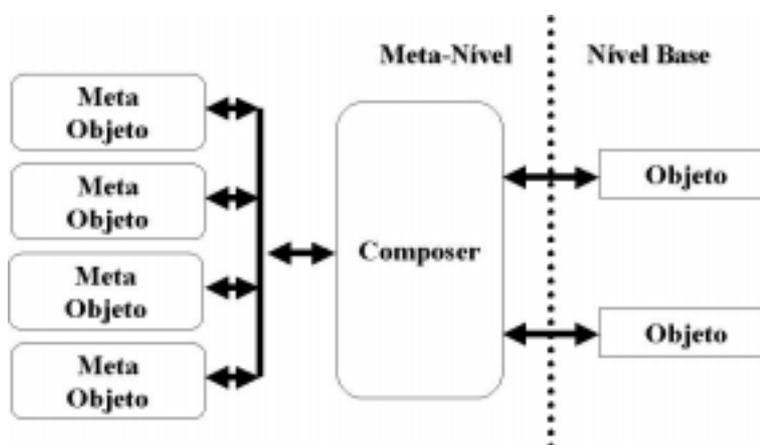


Figura 2 – Metaconfiguração múltipla

3. COMBINANDO OS CONCEITOS

3.1. A implementação

Meta-informações devem ser representadas e armazenadas de forma a serem facilmente acessadas pelos mecanismos reflexivos. Um modo natural é criar um repositório no meta-nível usado pelos meta-objetos quando da reificação dos objetos do nível base. Isto também pode garantir consistência entre meta-objetos relacionados [COS 00b].

A implementação sugerida neste artigo lida com os três principais problemas normalmente encontrados no desenvolvimento de *middlewares*: extensibilidade, adaptabilidade e reuso. É extensível porque novas meta-classes podem ser adicionadas a qualquer momento, sem que a aplicação necessite ficar ciente de tal fato. É adaptável porque usa reflexão para lidar com mudanças no ambiente da aplicação, ou ainda com novos requisitos da própria.

Finalmente, é reutilizável porque é mantida separada dos módulos da aplicação e pode ser empregada sem prévio conhecimento das classes desta. Este resultado é obtido utilizando-se reflexão no modelo orientado a objetos.

3.2. A visão do meta-nível

A arquitetura mostrada na Figura 3 exemplifica como os conceitos discutidos neste artigo podem trabalhar juntos. O objeto da aplicação (objeto de nível base) envia uma chamada a uma instância da classe MetaFront, que dispara uma série de operações que conectam esse objeto ao resto da arquitetura de suporte a atonicidade. Tal suporte é totalmente transparente ao objeto. A arquitetura pretende lograr expansibilidade, flexibilidade, adaptabilidade e reuso.

Este modelo é expansível, pois novas classes encapsulando novas políticas de controle de concorrência e recuperação podem ser incluídas ou excluídas a qualquer momento. É flexível, pois tais políticas podem ser mudadas dinamicamente, adaptando os objetos do nível base aos novos contextos da aplicação. A adaptabilidade é maior devido a possibilidade de alterar o comportamento da própria aplicação em tempo de execução através do uso de técnicas reflexivas. O reuso é obtido evitando-se a utilização de código mencionando explicitamente qualquer classe. Ao invés disso, reflexão é empregada para garantir introspecção em tempo de execução. Desta forma, não é necessário qualquer conhecimento prévio das classes da aplicação.

3.3. A arquitetura do middleware

Para lidar com diferentes políticas de controle de concorrência e recuperação, há um repositório de classes onde novas políticas podem ser inseridas e excluídas. Instâncias de uma classe especial chamada Loader são responsáveis pela manutenção destas políticas de forma organizada e passíveis de utilização pelo resto do mecanismo. As classes Policy Manager (Concurrency Policy Manager e Recovery Policy Manager) enviam solicitações às instâncias de Loader cada vez que uma nova política precisa ser inicializada. Cada Policy Manager, entretanto, deve trabalhar em sintonia com as classes Analyzer (Concurrency Analyzer e Recovery Analyzer). Estas implementam algoritmos especiais e possuem mecanismos de decisão para sugerir uma política a ser adotada. Elas são consultadas pelo Policy Manager cada vez que uma instância de Configurator solicitar uma nova política. A instância de Analyzer também informa ao Configurator que política será adotada. Instâncias de Concurrency Policy e Recovery Policies são então criadas para cada novo objeto no nível base que necessite-as. A instância de Configurator mantém registro de cada meta-objeto da classe MO e seu objeto base associado. De acordo com a política em uso, ativa a instância de Reifier, que armazena os objetos de nível base em uma instância de Repository, uma estrutura de dados responsável por manter os dados de cada objeto base reflexivo no meta-nível, para que seja possível recuperá-los caso necessário.

Para simplificar seu uso, há apenas uma “classe de entrada” nesta arquitetura: a MetaFront. Tudo que um objeto de uma aplicação (objeto de nível base) que precise de suporte a atonicidade deve fazer é chamá-lo. Esta classe contém métodos para criar instâncias de MO e então conectá-la ao objeto base solicitante. Deste momento em diante, a instância de MO associada ao objeto base interceptará e monitorará a estrutura e o comportamento deste objeto.

Outra característica que deve ser destacada é o fato de cada ação atômica poder implementar diferentes políticas de controle de concorrência e recuperação, de acordo com a situação corrente. Desta forma, a mesma execução de uma aplicação pode apresentar diferentes respostas ao mesmo problema, sendo que esta será sempre a melhor para o momento.

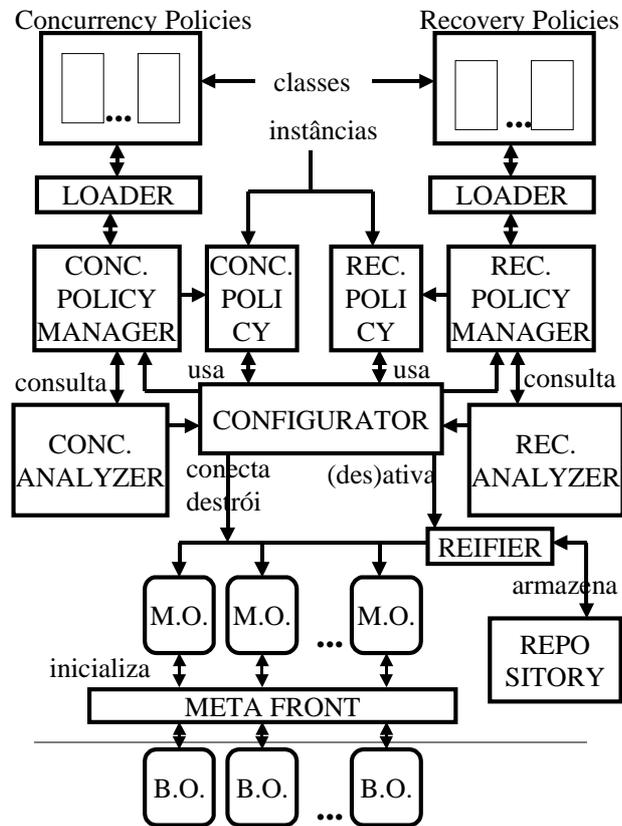


Figura 3 – Arquitetura do meta-nível

Tudo que o objeto base precisa fazer para usar a interface é conectar-se a um novo meta-objeto (tal fato é transparente a ele, pois uma nova instância de MO é automaticamente gerada em tempo de execução). O objeto obtém então uma política de controle de concorrência e uma de recuperação, que são escolhidas para ele no meta-nível. Quando a ação atômica chega a seu fim, a instância de Configurator termina a conexão inter-níveis e a instância de MO. Durante seu ciclo de vida, este meta-objeto é responsável pela monitoração e ligação entre o objeto base e o meta-nível. Dados podem ser reificados ou retornados do meta-nível de acordo com as políticas em uso. Tudo é transparente ao objeto base e pode ser usado com qualquer objeto de qualquer classe, pois não exige nenhum tipo de conhecimento prévio. A interface também apresenta métodos para iniciar e terminar uma ação atômica.

A Figura 4 mostra como implementar redundância ou diversidade de projeto usando um mecanismo de votação. O objeto base é monitorado por um meta-objeto (MO). Esta associação é controlada pela instância de Configurator. Quando uma chamada é enviada para o objeto base, seu meta-objeto a intercepta e envia para o Configurator, o qual, de acordo com as políticas previamente selecionadas, a passa para um grupo de objetos implementando ou redundância ou diversidade de projeto. A instância de Voter possui um mecanismo de decisão que descobre o correto processamento da chamada. Cada resultado é colocado no repositório Hits History para que instâncias com problemas possam ser excluídas da configuração.

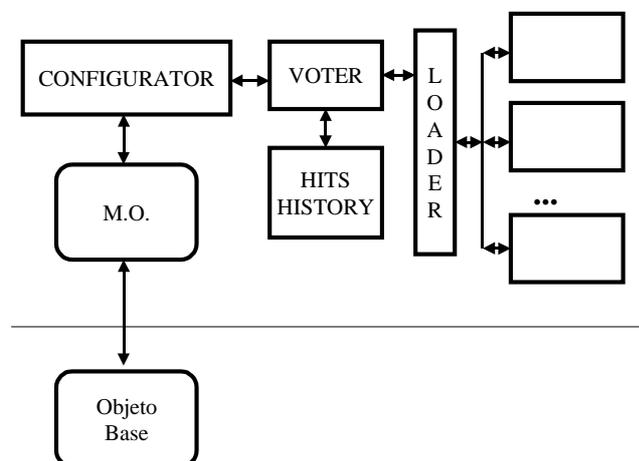


Figura 4 – Redundância ou diversidade de projeto usando um mecanismo de votação

CONCLUSÕES

Requisitos não funcionais, no intuito de obter-se maior grau de reuso e transparência, devem ser colocados à parte da funcionalidade da aplicação. Isto pode ser obtido através da combinação de várias técnicas. Reflexão computacional permite reconfiguração dinâmica e alteração comportamental. *Middlewares* e o modelo orientado a objetos permitem encapsulamento e aumento de reuso. A criação de um ambiente no meta-nível como um middleware torna a aplicação muito mais flexível e capaz de lidar com uma grande variedade de contextos e demandas.

Mais importante que tudo, esta combinação permite a alteração do comportamento de uma aplicação em tempo de execução. Este artigo focalizou a atomicidade como requisito não funcional desejado e mostrou a efetividade de algumas características reflexivas quando colocadas em um *middleware*, especialmente ao serem usadas na alteração do comportamento do *middleware*, em resposta às novas demandas provenientes de mudanças no ambiente computacional. Uma arquitetura independente, capaz de detectar alterações e responder às mesmas é útil no desenvolvimento de soluções que substituam o uso da estratégia de configuração estática. Reconfiguração dinâmica é um conceito chave para adaptabilidade e transparência. Ela não apenas ajuda a cobrir uma gama maior de situações, mas o faz de modo a incrementar seu reuso. Desde que não é possível prever o que uma aplicação terá de enfrentar durante sua execução, a melhor maneira de tornar soluções mais reutilizáveis é prover mecanismos que não dependam das classes presentes na aplicação e nem apresentem comportamento fixo. Pelo contrário, devem possuir mecanismos de decisão que permitam selecionar o próximo passo a ser dado.

REFERÊNCIAS BIBLIOGRÁFICAS

- [BLA 00] BLAIR, G.S.; COULSON, G.; COSTA, F.; DURAN, H. **On the design of reflective middleware platforms**. DCS, University of Illinois, USA, 2000.
- [BLA 99] BLAIR, G.; PPATHOMAS, M. **The case for reflective middleware**. Department of Computing, Lancaster University, UK, 1999.
- [COS 00] COSTA, F. **Reflective middleware platforms: design and implementation** DCS, University of Illinois, USA, 2000.
- [COS 00b] COSTA, F.; BLAIR, G. **Integrating meta-information and reflection in middleware** Department of Computing, Lancaster University, UK, 2000.

- [COS 99] COSTA, F.; BLAIR, G.; COULSON, G. **Experiments with reflective middleware**. Department of Computing, Lancaster University, UK, 1999.
- [COU 00] COULSON, G. **What is reflective middleware?** Distributed Multimedia Research Group, Department of Computing, Lancaster University, UK.
- [HAE 83] HAERDER, T.; REUTER, A. **Principles of transaction-oriented database recovery**, ACM Computing Surveys, 1983.
- [KIC 91] KICZALES, G.; des RIVIÉRES, J.; BOBROW, D.G. **The art of metaobject protocol**. MIT Press, USA, 1991.
- [MAE 87] MAES, P. **Concepts and experiments in computational reflection**. In Proceedings of OOPSLA'87, Vol. 22 of ACM SIGPLAN Notices, 1987.
- [OLI 98] OLIVA, A.; Garcia, I.C.; Buzato, L.E. – **The reflective architecture of guaraná**. Instituto de Computação, Unicamp, Campinas, SP, 1998.
- [ROM 00] ROMÁN, M.; KON, F.; CAMPBELL, R. **Reflective middleware: from your desk to your hand**. DCS, University of Illinois, USA, 2000.
- [SMI 82] SMITH, B.C. **Procedural reflection in programming languages**. PhD Thesis. MIT, Cambridge, Massachusetts, 1982.
- [SUN 99] **The Java tutorial – a practical guide for programmers**. Disponível via Internet em <http://java.sun.com/docs/books/tutorial/index.html>.

MINERAÇÃO DE DADOS PARALELA

Sandro da Silva Camargo¹

RESUMO

A mineração de bancos de dados tem se tornado inviável com algoritmos em série pelo tamanho crescente das bases de dados. A paralelização tem se constituído em uma alternativa viável e natural para a solução deste problema. Este artigo visa apresentar alguns princípios heurísticos para o projeto de algoritmos paralelos de mineração de dados.

Palavras chave: Mineração de dados, algoritmos paralelos.

ABSTRACT

Database mining becomes non-viable with sequential algorithms as database size increases. Algorithm parallelization is a viable option in the solution of this problem. This paper presents some heuristic principles of parallel mining algorithm design..

Keywords: mining, parallel algorithms.

1. INTRODUÇÃO

Segundo AGRAWAL (1996), atualmente as capacidades de geração e coleta de dados tem evoluído rapidamente, tanto a disponibilidade de meios de armazenamento a um baixo custo quanto o progresso nas tecnologias de aquisição dados tem tornado possível as organizações a formação de enormes bancos de dados sobre seus ramos de atividade de atividade.

De acordo com CHEN (1996), mineração de dados consiste no processo de extração de informação implícita, previamente desconhecida e potencialmente útil dos bancos de dados, sendo o resultado de tal processo utilizado para: suporte ao processo decisório, detecção de fraudes, análise de cesta de compras, entre outras finalidades. Um dos principais problemas encontrados na aplicação de mineração de dados é a escalabilidade, ou seja, o aumento do custo de processamento em consequência do incremento do tamanho do banco de dados.

O paralelismo se constitui em uma abordagem natural e com uma viável relação custo/benefício para enfrentar o problema de mineração em grandes bancos de dados, segundo FREITAS (1998). Entretanto o paralelismo não é a única abordagem para acelerar os algoritmos de mineração de dados, também são utilizadas com esta finalidade as técnicas de amostragem, seleção de atributos, discretização de atributos contínuos, restrição de espaço de pesquisa, otimização de algoritmo e código e mineração de dados distribuída.

Neste artigo é discutida a paralelização de algoritmos de mineração de dados de quatro paradigmas de descoberta de conhecimento: indução de regras, aprendizado baseado em instâncias, algoritmos genéticos e redes neurais, sendo também derivados um conjunto de princípios heurísticos para projeto de algoritmos eficientes para mineração de dados paralelos.

Na paralelização de algoritmos podem ser utilizadas duas soluções distintas: o paralelismo de dados e o paralelismo de controle. Tais soluções podem ser aplicadas separadamente ou em conjunto, influenciando decisivamente no desempenho do algoritmo.

¹ Mestre em Ciência da Computação PPGCC - UFRGS. Professor do CCEI - URCAMP. E-mail: scamargo@urcamp.tche.br.

2. PARALELISMO DE DADOS X PARALELISMO DE CONTROLE

Segundo FREITAS (1998), a diferença principal entre o paralelismo de dados e o paralelismo de controle é que o segundo refere-se a execução concorrente de diferentes operações ou instruções em vários subconjuntos de dados, enquanto o primeiro refere-se a execução da mesma operação ou instrução em vários subconjuntos de dados ao mesmo tempo, sendo que tais subconjuntos são paralelizados a partir de um conjunto inicial.

Sob a perspectiva do problema tratado neste artigo que é a mineração de dados, o paralelismo de dados têm três vantagens principais sobre o paralelismo de controle:

1. o fluxo de controle do programa com paralelismo de dados é essencialmente o mesmo que o fluxo de controle de um programa seqüencial, permitindo que grande parte do código de programa seqüencial possa ser reutilizado para aplicação sob um modo de paralelismo de dados. Tal reutilização simplifica a programação e reduz significativamente o tempo de desenvolvimento em relação a um programa similar que utiliza paralelismo de controle.
2. o paralelismo de dados tem um grau maior de independência de arquitetura de máquina em relação com paralelismo de controle. No paralelismo de dados as funções relativas a arquitetura de máquina são encapsuladas em uma camada inferior do software, não sendo acessíveis diretamente pela aplicação. O fluxo de controle do algoritmo é seqüencial e não há a necessidade de sua configuração para a arquitetura paralela. Já no paralelismo de controle o fluxo de controle deve estar explícito no algoritmo;
3. o paralelismo de dados possui uma melhor escalabilidade para bancos de dados maiores que o paralelismo de controle, pois à medida que a quantidade de dados é incrementada, é possível acrescentar ao sistema tantos de nodos processadores quantos forem necessários, sofrendo o tempo de resposta um incremento muito pequeno em razão do aumento das necessidades de comunicação entre os nodos.

Apesar das vantagens do paralelismo de dados no que cerne a mineração de dados, esta não é uma solução completa, a utilização de paralelismo de controle também se faz necessária em alguns casos tanto sozinha quanto em companhia da primeira. O paralelismo de controle é mais indicado para aplicação em espaços de regras muito grande enquanto o paralelismo de dados mais indicado para aplicação em bancos de dados com um conjunto muito grande de tuplas, e quando há um número muito grande de processadores disponíveis, ambos tipos de paralelismo podem ser explorados ao mesmo tempo, podendo ser acelerada enormemente a execução de algoritmos de mineração.

2. INDUÇÃO DE REGRAS EM PARALELO

De acordo com FREITAS (1998), em alto nível de abstração, um algoritmo de indução de regras (IR) pode ser visto como uma pesquisa heurística interativa, onde cada interação consiste de três passos:

- (1) selecionar a melhor **Regra Candidata (RC)**;
- (2) expandir a RC selecionada, gerando novas RC's;
- (3) avaliar as novas RC's.

Estes passos iterativos são repetidos até que um conjunto satisfatório de RC's seja encontrado. Os passos (1) e (3) estão baseados em uma função de avaliação de RC, tal função calcula a qualidade de uma RC através do acesso ao banco de dados, sendo que a maioria dos algoritmos de indução de regras está baseada na contagem do número de tuplas do banco de dados que satisfazem as RC's. O gargalo do algoritmo de mineração de IR em um banco de dados muito grande é o passo (3), onde o paralelismo de dados pode

ser aplicado com maior naturalidade fazendo com que uma única RC seja avaliada paralelamente por múltiplos processadores onde cada um irá contar tuplas que estão em sua memória local.

Já o paralelismo de controle é geralmente associado com os passos (1) e (2), podendo também ser associado com o passo (3), avaliando múltiplas RC's ao mesmo tempo, em diferentes processadores. Também é importante salientar um fenômeno relacionado com a escalabilidade deste tipo de paralelismo onde, em alguns casos, o incremento do domínio de pesquisa conduz a descoberta de conhecimento menos exato, quando o inverso deveria ocorrer, ou seja, com o aumento do número de tuplas analisadas pelo algoritmo a exatidão dos resultados não deveria ser menor.

2.1. Sistemas de indução de regras em paralelo sem recursos de SGBD

AGRAWAL (1996) faz experimentos de versões com paralelismo de dados e com paralelismo de controle do algoritmo de IR chamado *apriori* visando a descoberta de regras de associação. Com a utilização de paralelismo de dados, cada processador independentemente executa a contagem de suporte para todos os itens candidatos acessando apenas as tuplas do bancos de dados presentes na sua memória local. Já na versão com paralelismo de controle, cada processador realiza a contagem de suporte para seus conjuntos de itens candidatos locais acessando todas tuplas presentes na sua memória local e as presentes nas memórias locais dos demais processadores, o que além de fazer com que cada processador analise muito mais tuplas que a versão anterior, também aumenta a comunicação entre os processadores. O resultado obtido é que a versão com paralelismo de dados é executada significativamente mais rápida que versão com paralelismo de controle.

HAN (1997) propõe novos algoritmos para mineração de regras de associação. O algoritmo *Intelligent Data Distribution* que utiliza mais eficientemente a memória agregada de um computador paralelo através da utilização de um esquema de particionamento de candidatos inteligente e utiliza um eficiente mecanismo de comunicação para mover dados entre os processos. O outro algoritmo proposto denominado de *Hybrid Distribution* faz o particionamento dinâmico do conjunto candidato para manter um bom balanceamento. Através de experimentos realizados em um computador paralelo Cray T3D foi demonstrado que o algoritmo *Hybrid Distribution* aumenta linearmente e explorar a memória agregada melhor e pode gerar mais regras de associação com uma única passagem através do banco de dados.

HAN (1997) propõe duas novas versões do algoritmo *Apriori*. Estas novas formulações denominadas de *IDD* e *HD*, buscam resolver os defeitos dos algoritmos *CD* e *DD*, duas versões paralelas propostas anteriormente. *HD* é um algoritmo híbrido que combina as vantagens dos algoritmos *CD* e *DD*. São realizados experimentos em um sistema Cray T3E com 128 processadores onde é relatado que a escalabilidade do algoritmo *HD* é tão boa quanto a do algoritmo *CD* em relação ao número de transações, e a escalabilidade tão boa quanto *IDD* em relação ao incremento do tamanho do conjunto de itens candidatos.

2.2. Sistemas de indução de regras em paralelo com recursos de SGBD

Hoje em dia há uma quantidade de servidores de bancos de dados paralelos (*Parallel Database Servers - PDS*) comercialmente disponíveis de alta performance e boa relação custo/benefício. Estas máquinas oferecem o benefício de paralelização automática, as consultas de bancos de dados especificadas em um estilo declarativo são automaticamente paralelizadas e otimizadas pela máquina, reduzindo o tempo de processamento sem a complexidade da programação paralela. *PDS's* oferecem diversas facilidades de SGBD e

também são úteis para sistemas de mineração em larga escala, de acordo com FREITAS (1998).

O projeto *DAFS* inclui uma versão paralela do algoritmo *TDIDT*, de McLAREN (1997). *DAFS* é um *PDS* projetado para suportar o processo completo de descoberta de conhecimento (incluindo pré processamento e pós processamento para mineração de dados). Este projeto segue uma abordagem cliente/servidor, onde o cliente está baseado no conhecimento da ferramenta de mineração de dados Clementine, apresentada por SHERER (1996), e o servidor é um *PDS* não compartilhado. O algoritmo paralelo *TDIDT* implementado em *DAFS* essencialmente segue uma abordagem de paralelismo de dados.

HOLSHEIMER (1996) implementa uma versão com paralelismo de dados de um algoritmo de pesquisa de mineração de dados em *Monet PDS*, onde as regras candidatas (*RCs*) são avaliadas pelas consultas submetidas a *Monet*, tais consultas são processadas pela exploração de paralelismo. *Monet* usa particionamento vertical das relações do banco de dados, enquanto o *PDS* convencional usa particionamento horizontal.

De acordo com FREITAS (1998), uma motivação para usar *PDS* mais convencional opõe-se ao desejo de integrar mineração de dados e *data warehouses*. Sem dúvida a maioria das grandes *data warehouses* são implementadas no topo dos *PDS* comercialmente disponíveis para melhorar eficiência e escalabilidade. FREITAS (1996) propõe uma abordagem primitiva baseada na integração de mineração de dados e *data warehouses*. Mais precisamente, este projeto define uma indução de regra primitiva genérica que auxilia a avaliação das *RCs* em um número de algoritmos de indução de regras. Pela execução desta primitiva em um *PDS*, algoritmos de indução de regras são significativamente acelerados pela exploração do paralelismo de dados na execução de consultas de bancos de dados.

3. APRENDIZADO PARALELO BASEADO EM INSTÂNCIA

De acordo com FREITAS (1998), no contexto de uma tarefa de classificação, o paradigma de aprendizado baseado em instância consiste de dois passos básicos:

- (1) a comparação uma nova tupla, para ser classificada, junto com todas tuplas armazenadas, pelo cálculo de uma distância métrica entre uma nova tupla e cada tupla armazenada;
- (2) as k tuplas armazenadas mais próximas (k é um valor especificado pelo usuário) são selecionadas e suas classes são usadas para prever a classe de uma nova tupla de acordo com um dado esquema de resolução de conflito de classe.

O tempo de execução do passo (2) é muito pequeno sendo inteiramente dominado pelo passo (1) que deve ser o alvo do paralelismo e oferece uma oportunidade ideal para a exploração do paralelismo de dados. Primeiro, particiona-se os dados sendo minerados em p subconjuntos de dados mutuamente exclusivos e mutuamente exaustivos, onde p é um número de processadores e cada subconjunto é denotado para um processador distinto. Cada processador calcula a distância entre as tuplas em seu subconjunto de dados local e a nova tupla ser classificada de uma maneira completamente independente dos demais processadores, fazendo com que o *overhead* de comunicação entre os processadores não seja preocupação no primeiro passo de um algoritmo, e o segundo passo requer algum *overhead* de comunicação entre processadores, mas o tempo deste passo é muito mais curto que o do primeiro.

A exploração do paralelismo de dados em *IBL* também pode ser feito de um modo que integre mineração de dados e *data warehouses*, baseados na idéia de execução de primitivas de mineração de dados genéricas em um *PDS*, FREITAS (1998) define uma

primitiva genérica *IBL* que diversos algoritmos *IBL* e têm mostrado como usar esta primitiva para explorar o paralelismo de dados em um *PDS*, em ordem para reduzir significativamente o tempo de processamento de algoritmos *IBL*. A exploração do paralelismo de controle no aprendizado baseado em instância é também possível embora freqüentemente seja menos vantajoso que o paralelismo de dados, pois a cada processador é determinada a tarefa de classificação de um subconjunto de novas tuplas devendo os dados serem replicados através de todos os processadores.

Dependendo do algoritmo *IBL* utilizado, o cálculo da métrica de distância pode requerer alguma operação que ofereça potencial adicional para a exploração do paralelismo, como por exemplo o cálculo do peso de atributos, utilizado em muitos algoritmos *IBL* para determinar maior importância a atributos com maior força preditiva. Em alguns algoritmos *IBL* que paralelizam o cálculo de peso de atributos os pesos são dinamicamente calculados, uma vez que as tuplas estejam armazenadas são distribuídas através dos processadores, o cálculo de peso dos atributos é feito em paralelo por deixar que cada processador calculasse o valor do peso parcial dos atributos de suas tuplas locais e então ter os resultados parciais combinados de algum modo em pesos finais de atributos.

4. ALGORITMOS GENÉTICOS PARALELOS

Segundo FREITAS (1998), um Algoritmo Genético (AG) é um procedimento iterativo que mantém a população de “indivíduos”, que são strings de símbolos representando uma solução candidata para um dado problema. A cada iteração (ou “geração”) os indivíduos correntes são avaliados por uma função de adaptação, a qual mede a qualidade da solução candidata representada pelo indivíduo, e operadores genéticos são aplicados para adaptar indivíduos da geração corrente, modificando-os e criando uma nova geração de indivíduos. De acordo com o princípio da seleção natural de Darwin, a população tende a convergir para indivíduos altamente adaptados.

Sob o ponto de vista de mineração de dados, os indivíduos freqüentemente representam regras candidatas e a função de adaptação mede a qualidade destas regras, tal função tem um papel equivalente a função de avaliação da regra candidata no paradigma de indução de regras. Há duas fontes básicas de paralelismo nos algoritmos genéticos, uma pode explorar o paralelismo no cálculo da adaptação dos indivíduos e/ou na aplicação de operadores genéticos. A menos que os dados sendo minerados sejam pequenos, o tempo gasto com o cálculo da adaptação de indivíduos tendem a exceder grandemente o tempo gasto com a aplicação de operadores genéticos.

Há dois modos básicos de paralelização de cálculo de adaptação. O primeiro consiste da exploração do paralelismo de dados no cálculo da função de adaptação para cada indivíduo. Nesta abordagem os dados sendo minerados são particionados em p subconjuntos de dados mutuamente exclusivos e exaustivos, sendo cada subconjunto enviado para um processador distinto, onde p é o número de processadores. Então cada processador calcula um valor parcial para a adaptação de um indivíduo pelo acesso apenas a suas tuplas locais, e depois, os valores de adaptação parcial são combinados para produzir o valor final de adaptação do indivíduo corrente.

O segundo modo de paralelização de cálculo de adaptação é uma abordagem de paralelismo de controle e consiste do particionamento do conjunto de indivíduos da população corrente em p subconjuntos mutuamente exclusivos e exaustivos, sendo cada subconjunto enviado para um processador diferente. Então cada processador calcula a função de adaptação para todos seus indivíduos, independente do cálculo executado por outros processadores. Como desvantagem, a totalidade dos dados sendo minerados deve ser replicada através de todos os processadores o que reduz a escalabilidade do sistema.

Algoritmos genéticos com paralelismo de controle podem ser adicionalmente divididos em duas amplas abordagens, chamadas algoritmos de população única e população distribuída. No modelo todos indivíduos são respeitados como uma população única. Como resultado, em princípio este tipo de algoritmo genético com paralelismo de controle descobre o mesmo conhecimento que seu semelhante seqüencial. Por outro lado, um algoritmo genético de população distribuída trata cada uma das sub populações fisicamente distribuídas como uma população lógica distinta. Esta abordagem pode ser adicionalmente sub dividida em abordagens com grossa granularidade e fina granularidade. No modelo a população está dividida em um pequeno número de sub populações, cada uma delas com um grande número de indivíduos; enquanto nas populações mais recentes são divididas em um maior número de sub populações, cada um deles com um menor número de indivíduos. Em ambas abordagens há algum mecanismo que permite a troca de alguns indivíduos entre as sub populações de tempos em tempos.

A abordagem de população distribuída representa uma saída significativa de uma abordagem de população única, uma vez que estas duas abordagens conduzam a descoberta de diferentes peças de conhecimento. Atualmente, a modificação introduzida pela população distribuída de AG's é freqüentemente considerada como vantajosa principalmente pelo fato de que a população distribuída reduz a probabilidade de convergência prematura de soluções sub ótimas, em comparação com uma população única.

A exploração de paralelismo de dados em AG também pode ser feita de vários modos que integram a mineração de dados e *data warehouses*, baseados em uma idéia de execução de primitivas genéricas de mineração de dados em um PDS.

5. REDES NEURAIS PARALELAS

Essencialmente, uma Rede Neural (RN) consiste de muitos Elementos Processadores (EP's), imprecisamente chamados neurônios, e interconexões com pesos entre os EP's. Cada EP executa um cálculo muito simples, como calcular a soma com pesos de suas conexões de entrada, e calcular o sinal de saída que é enviado para outros EP's. A fase de aprendizado de uma rede neural consiste de um ajuste de pesos das interconexões, a fim de produzir uma saída desejada, segundo FREITAS (1998).

O conhecimento de um sistema é expresso em uma representação de baixo nível, distribuída através dos pesos das interconexões, sendo freqüentemente desejável converter os pesos das interconexões aprendidas para uma representação de conhecimento de alto nível tal como regras de "se – então", para fazer a descoberta de conhecimento compreensível para o usuário.

Há duas abordagens básicas para exploração do paralelismo em redes neurais. A primeira consiste da distribuição dos dados sendo minerados através de processadores, e utiliza o paralelismo de dados, também é chamada de aprendizado conjunto paralelo, na terminologia de redes neurais paralelas. Primeiro, os dados sendo minerados são particionados em p subconjuntos mutuamente exclusivos e exaustivos, sendo cada conjunto destinado para um processador distinto, onde p é o número de processadores. Cada processador tem uma cópia completa da rede neural, com todos seus neurônios (EP's) e todos os pesos de suas interconexões. Portanto, cada processador usa seu subconjunto de dados locais para calcular os pesos parciais atualizados para sua cópia local da rede neural inteira. Então estes pesos parciais atualizados são combinados para produzir o peso final atualizado para a rede neural inteira.

Esta abordagem geralmente envolve uma forma de atualização de peso *batch*, onde os pesos das interconexões estão atualizados apenas depois da combinação das atualizações

dos pesos parciais. Isto é um contraste com a natureza incremental dos procedimentos de atualização de peso convencionais, onde os pesos são geralmente atualizados depois que a rede neural processar cada tupla de aprendizado. O procedimento de atualização *batch* freqüentemente conduz a uma exatidão de previsão de algum modo diferente do executado com um procedimento de atualização incremental convencional.

A segunda abordagem para exploração do paralelismo em redes neurais consiste da distribuição da estrutura da rede neural entre os processadores. Esta abordagem é baseada no próprio controle de paralelismo associado com as redes neurais. A estratégia usada para distribuição da estrutura das redes neurais através dos processadores depende de diversos fatores, tais como o número de neurônios em cada camada da rede neural, o número de camadas, o número de processadores disponíveis e a topologia das interconexões, entre outros.

Nesta estratégia, cada processador é assinalado como um neurônio distinto e o conjunto de pesos de entrada no seu neurônio. Neste caso o processamento executado por cada neurônio é inteiramente dependente do processamento executado por outros neurônios, tanto que todos os neurônios podem ser operados em paralelo. É claro que, se o número de neurônios é consideravelmente maior que o número de processadores, esta estratégia pode ser modificada para que cada processador seja assinalado como um subconjunto de neurônios com suas interconexões de entrada correspondentes.

Deve-se evitar que muitos processadores permaneçam ociosos por um determinado tempo pela exploração de uma forma de paralelismo temporal através de sucessivas camadas da rede neural. Outra estratégia para exploração do paralelismo de controle em uma rede neural, trabalhando em um menor nível de granularidade que a estratégia superior, consiste da exploração do paralelismo sobre cada neurônio. Nesta estratégia, as interconexões de entrada de cada neurônio são distribuídas através de um conjunto de processadores. Devido a esta natureza de granularidade fina, esta estratégia é geralmente suscetível a sistema de processamento massivamente paralelo, particularmente quando o número de processadores é muito maior que o número de neurônios.

Embora haja uma extensiva pesquisa em redes neurais paralelas, há uma pequena pesquisa na área de redes neurais paralelas para aplicações de mineração de dados. Estas implementações essencialmente seguem uma abordagem de paralelismo de dados.

5. CONCLUSÕES

A distinção entre paralelismo de dados e paralelismo de controle é crucial para o projeto de algoritmos de mineração de dados em paralelo, independentemente do paradigma de descoberta de conhecimento que esteja sendo utilizado, seja por indução de regras, aprendizado baseado em instâncias, algoritmos genéticos ou redes neurais.

Através da análise, baseado tanto nos sistemas discutidos anteriormente, quanto nos conceitos fundamentais de processamento paralelo, é possível sugerir um conjunto de princípios heurísticos para projeto de algoritmos eficientes de mineração de dados paralela.

- (1) buscar a paralelização das partes do algoritmo que consomem maior parte de tempo;
- (2) buscar a paralelização de dados onde há utilização intensiva de dados, e paralelização de controle se há utilização intensiva de CPU;
- (3) buscar a utilização de um particionamento de dados inteligente quando utilizar o paralelismo de dados;
- (4) buscar a minimização da comunicação entre processadores quando utilizar o paralelismo de controle;
- (5) evitar a réplica de dados nos processadores quando utilizar o paralelismo de controle;

- (6) analisar a possibilidade da utilização de algoritmos que selecionam o paralelismo de dados ou o paralelismo de controle dinamicamente baseado nos parâmetros correntes do algoritmo e do espaço de pesquisa.

Tais princípios são apenas heurísticos e não garantem o melhor algoritmo paralelo de mineração de dados em todos os casos, mas eles trabalham razoavelmente bem na maioria dos casos.

REFERÊNCIAS BIBLIOGRÁFICAS

- AGRAWAL, Rakesh; SHAFER, John C. Parallel mining of association rules. In: IEEE Transactions on knowledge and data engineering, 1996. Vol. 8 N. 6, pag 962-969.
- CHEN, Ming-Syan, HAN, Jiawei, Yu, Phillip S. Data Mining: An overview from a database perspective. In: IEEE Transactions on knowledge and data engineering, 1996. Vol. 8 N. 6, pag 866-883.
- FREITAS, Alex A . A Survey of parallel data mining. In: 2nd Int. Conf. On The Practical Applications of Knowledge Discovery and Data Mining, 1998. Londres, Mar. 1998. P.287-300.
- HAN, Eui-Hong Sam; KARYPIS, George; KUMAR, Vipin. Scalable parallel data mining for association rules. In: IEEE Transactions on knowledge and data engineering. Vol. 8 N. 6, Dezembro, 1996.
- HOLSHEIMER M.; KERSTEN M.L. Architectural suport for data mining. Proc. AAAI-94 Workshop on Knowledge Discovery in Databases, 217-228. AAAI Press, 1994.
- McLAREN I.; BABB E.; BOCCA J.. DAFS: suporting the knowledge discovery process. Proc. 1st Int. Conf. Practical Applications of Knowledge Discovery, 179-190. The Practical Application Company, UK. Apr. 1997.
- SHERER C.; KHABAZA T.. Data mining for data owners: presenting advanced technology to non-technologists through Clementine integrated toolkit. Proc. Data Mining '96 Unicom Seminar, 236-242. London: Unicord, 1996.
- SRIVASTAVA, A.; SINGH, V.; HAN, Eui-Hong Sam; KUMAR, Vipin. An efficient, scalable, parallel classifier for data mining. Technical Report TR-97-010. University of Minnesota, Dept. of Comp. Sci., 1997.

TESTE DE SOFTWARE ORIENTADO A OBJETO BASEADO EM ESTADOS

*Fábio Fagundes Silveira*¹

*Dra. Ana Maria de Alencar Price*²

RESUMO

A atividade de teste constitui uma fase de grande importância no processo de desenvolvimento de *software*, tendo como objetivo garantir um alto grau de confiabilidade nos produtos desenvolvidos.

O paradigma da Orientação a Objetos (OO) surgiu com o objetivo de melhorar a qualidade bem como a produtividade no desenvolvimento de aplicações. Entretanto, apesar do aumento constante de aceitação do paradigma OO pela indústria de *software*, a presença de algumas de suas características torna a atividade de teste de programas neste paradigma mais complexa do que o teste de sistemas tradicionais. Entre estas características cita-se a herança, o encapsulamento, o polimorfismo e a ligação dinâmica [PRE 95] [UNG 97]. Este artigo aborda algumas técnicas de teste para sistemas orientados a objetos, detalhando uma destas, o teste baseado em estados, o qual avalia as mudanças de estados sofridos pelos objetos, baseando-se no modelo dinâmico da classe (máquinas de estados finitos) apresentado por algumas metodologias de análise OO.

Palavras-chaves: Teste de *Software* Orientado a Objetos, Orientação a Objetos, Teste Baseado em Estados.

ABSTRACT

Testing constitutes an important phase in the software development process, having as its objective, to guarantee a high degree of reliability in the products being developed.

The Object Oriented (OO) paradigm seeks to improve quality as well as productivity in the development of applications. However, in spite of constantly increasing acceptance of the OO paradigm by the software industry, some of its characteristics render the testing of programs written with this paradigm more complex than testing traditional systems. Among these characteristics are inheritance, encapsulation, polymorphism and dynamic binding [PRE 95][UNG 97]. This article discusses some testing techniques for object oriented systems, detailing one, a test based on states. This technique evaluates the state transitions undergone by the objects, based upon a dynamic class model (finite state machine) and demonstrated for some methods of OO analysis.

1. INTRODUÇÃO

O teste é uma atividade associada a qualquer processo, cujo objetivo seja produzir um produto [PER 00], sendo utilizado para determinar a qualidade deste produto durante seu desenvolvimento e, também, após sua construção. Desta forma, o teste é parte integrante do processo e não uma atividade auxiliar durante sua elaboração.

Na busca pela qualidade no desenvolvimento de produtos, com o objetivo de garantir um alto grau de confiabilidade, a atividade de teste tem se caracterizado como de grande importância. O processo de teste de *software* representa um conjunto de atividades realizadas com o objetivo de garantir a maior qualidade possível em produtos de *software*.

¹ Mestre em Ciência da Computação (UFRGS), Doutorando em Ciência da Computação (ITA). E-mail: ffs@comp.ita.br

² Dra. em Informática, professora do Instituto de Informática da Universidade Federal do Rio Grande do Sul (UFRGS). E-mail: anaprice@inf.ufrgs.br

O paradigma da Orientação a Objetos (OO) surgiu com o objetivo de melhorar a qualidade bem como a produtividade no desenvolvimento de *software*, desde que sejam bem exploradas suas características, possibilitando assim, maior reutilização, confiabilidade, modularização e rapidez de desenvolvimento [PRE 95]. Entretanto, apesar do aumento constante de aceitação do paradigma OO pela indústria do *software*, os novos conceitos presentes nesta abordagem introduziram um conjunto de problemas na atividade de teste de programas. Tais problemas tornam o processo de teste neste paradigma mais complexo do que em sistemas tradicionais, pois esconde informações e dificulta a identificação do código que gerou o erro [PRE 95], [UNG 97]. A seguir, são apresentadas técnicas de teste para sistemas orientados a objetos, detalhando-se uma destas, o teste baseado em estados.

2. TESTE DE SOFTWARE

Muitas são as publicações definindo teste de *software*, entretanto, todas recaem no mesmo princípio. Teste de *software* é o processo de execução de um programa de maneira controlada, com o objetivo de responder à questão: “O *software* se comporta segundo sua especificação?” Segundo Myers [MYE 79], o objetivo principal desta atividade é o de encontrar erros: “A atividade de teste não pode mostrar a ausência de *bugs*; ela só pode mostrar se defeitos de *software* estão presentes”.

Uma das razões pela qual o processo de teste tem ganhado significativa importância, é o fato deste consumir até 40% do esforço despendido no desenvolvimento de *software* [PRE 95]. Desta forma, esta atividade constitui um elemento crítico na busca pela garantia de qualidade de um produto de *software*, procurando torná-lo mais confiável.

Além dos aspectos já ressaltados, o fato do incrível aumento no número de usuários de *software*, além do aumento de empresas desenvolvedoras destes produtos enfatiza ainda mais a importância da fase de teste. Então, torna-se mister assegurar que a qualidade e confiabilidade nos produtos de *software* desenvolvidos não constituam um fator de risco e de preocupação.

As atividades geralmente associadas ao processo de teste são: análise estática e análise dinâmica [PRE 95]. A primeira investiga o código-fonte do programa, procurando por problemas e reunindo métricas sem efetivamente executar o código. Já a análise dinâmica, tem por objetivo inspecionar o comportamento do programa em teste durante sua execução. A atividade de teste não termina após finalizada suas conclusões. O *software* deve ser monitorado com o intuito de constatar novos problemas durante seu uso, adaptando-o a novas exigências. Ressalta-se ainda, que o teste de *software* deve ser repetido, modificado e estendido. Os esforços para revisar e repetir o teste, constituem o maior custo no processo de construção de um produto de *software*.

Basicamente, existem três abordagens de teste:

- teste estrutural: também conhecido como teste da “*caixa-branca*”, derivado da estrutura interna do programa, cuja estratégia concentra-se na tentativa de exercitar todo o código da aplicação, com base no fluxo de controle e/ou no fluxo de dados [PRE 95]. Entre os problemas desta abordagem, cita-se: (i) sendo a análise da estrutura interna realizada de maneira estática, programas com laços resultam numa quantidade infinita de caminhos; (ii) desperdícios de tempo e financeiros, na geração de casos de teste para caminhos não executáveis (*infeasible paths*) e (iii) o fato da execução de um caminho resultar em sucesso, não garante a correteza do mesmo, pois outros casos de teste podem invalidar o mesmo.
- teste funcional: também conhecido como teste da “*caixa-preta*”, preocupa-se com a verificação do atendimento das funcionalidades requeridas, baseando-se na especificação dos requisitos do sistema. Assim, não considera a estrutura interna do

programa, sendo este teste aplicado quando o sistema já encontra-se em fase final ou completamente construído.

- teste baseado em erros (Análise de Mutantes): é gerado um conjunto de programas semelhantes, denominados *mutantes*, a partir de um programa *P*, o qual assume-se como correto. Casos de teste são, então, gerados com o intuito de provocarem diferenças de comportamento em *P* e seus mutantes. Os chamados *operadores de mutação* são utilizados na geração destes mutantes, podendo ser associado a cada operador uma classe de erros.

Existe ainda, o teste de regressão, que alguns autores, como Herbert [HER 99], o descrevem separadamente, devido à importância deste nos projetos de desenvolvimento de *software*. Seu objetivo é assegurar que acréscimos de funcionalidade, aperfeiçoamentos de determinadas características, ou simplesmente correções de erros, não introduzam falhas, inexistentes até então.

3. TESTE DE SOFTWARE ORIENTADO A OBJETO

O paradigma da Orientação a Objetos (OO) surgiu trazendo consigo um novo enfoque, comparado aos métodos tradicionais de desenvolvimento de *software*. Entre as vantagens desta abordagem, pode-se citar a adoção de formas mais próximas dos mecanismos humanos com relação ao gerenciamento de complexidades inerentes ao desenvolvimento de produtos de *software*, buscando com isso um aumento de qualidade e maior produtividade, devido a uma de suas principais contribuições: a reutilização de código. Esta contribuição, entretanto, enfatiza que, assegurar que as classes desenvolvidas estejam corretas é essencial, o que deve ser feito o mais cedo possível, pois erros podem propagar-se durante sua reutilização em sub-classes.

O paradigma OO introduz novos conceitos e abstrações, como classes, métodos, mensagens, herança, polimorfismo e encapsulamento. Pode-se citar aqui, alguns benefícios da utilização deste paradigma: desenvolvimento mais rápido, maior qualidade, manutenção facilitada, estruturas de informação com melhor definição, bibliotecas de classes disponíveis, reutilização, entre outros.

Apesar desta abordagem apresentar várias vantagens em relação ao paradigma procedimental, a atividade de teste constitui um dos principais problemas no desenvolvimento de aplicações OO. Segundo Herbert [HER 99], existe uma carência de técnicas bem estabelecidas para o teste de aplicações desenvolvidas sobre este paradigma, constituindo-se numa área nova de pesquisa e aplicação. Do mesmo modo em que algumas das características encontradas em linguagens orientadas a objetos reduzem a probabilidade de determinados erros, outras favorecem o aparecimento de novas categorias dos mesmos [BIN 95]. Entre as características favorecedoras, cita-se o encapsulamento, o polimorfismo e ligação dinâmica.

Algumas facilidades do teste de *software* OO em relação ao procedimental são apresentadas por McGregor [MCG 96]: (i) métodos e *interfaces* de classes são explicitamente definidos; (ii) número menor de casos de testes para cobertura são resultantes, devido ao número reduzido de parâmetros e (iii) reutilização de casos de teste devido à presença da característica de herança.

McGregor aponta também algumas desvantagens que devem ser consideradas [MCG 96]: (i) a avaliação da correteza da classe é dificultada pela presença do encapsulamento de informações; (ii) o gerenciamento do teste é dificultado pelos múltiplos pontos de entrada (métodos) de uma classe e (iii) as interações entre os objetos são expandidas pelo polimorfismo e pela ligação dinâmica.

Os seguintes itens, a fim de verificarem a completeza de um sistema de *software* OO, devem estar presentes em sua especificação [MCG 96]: (i) a especificação de todas as classes; (ii) pré-condições, pós-condições e invariantes de um método: as pré-condições representam condições que devem ser verdadeiras para a execução do referido método. Já as pós-condições devem ser verdadeiras após a execução do método. A invariante de uma classe é uma condição que assume-se que esteja verdadeira em todos os momentos, para determinada classe e (iii) a especificação dos métodos de uma classe, bem como seu modelo de estados.

Uma classificação para os níveis de teste de código OO pode ser [SMI 92]:

- teste de classe: a menor unidade a ser testada é a classe [BIN 95];
- teste de cluster: conjunto de classes que interagem entre si, sendo este o principal fator a ser considerado neste tipo de teste;
- teste de sistema: neste nível, a funcionalidade é enfatizada.

Conforme descrito em [HER 99], as classes são executadas através da execução dos objetos. Abaixo, são descritas as categorias de erros que podem estar presentes em classes, mensagens e métodos:

- erros de classe: não implementação de alguns estados definidos na especificação; não implementação de alguns comportamentos definidos na especificação; a especificação da classe não reflete a descrição dos seus requisitos; violação do projeto e dos padrões de programação; implementação e documentação apresentam inconsistência.
- erros de mensagens: parâmetros incorretos; os receptores das mensagens retornam valores impróprios ao objeto que enviou a mensagem; exceções não são implementadas, como retornos de mensagens;
- erros de métodos: problemas de sintaxe na linguagem de programação; falhas para alcançar as pós-condições; problemas no fornecimento de valores apropriados como retornos; tempo de resposta insuficiente (aplicações *real-time*); código não executável.

A subseção seguinte descreve o teste de *software* OO baseado em estados, o qual avalia as mudanças de estados sofridos pelos objetos. Este teste é baseado no modelo dinâmico da classe [RUM 97] (diagrama / máquina de estados), a qual é formada por estados, transições, e pré e pós-condições associadas a transições, que são definidas como execução de métodos.

3.1 Teste de software orientado a objeto baseado em estados

De acordo com Binder [BIN 95], um sistema orientado a objeto pode ser visto como uma sociedade de agentes cooperantes, onde cada agente é responsável por seu estado. Binder define estado como um subconjunto de todas as combinações possíveis referentes aos valores dos atributos de uma classe. Define ainda que, comportamento é a seqüência de mensagens e respostas que uma classe envia e/ou aceita. Já o comportamento de um sistema é o resultado da interação de comportamentos individuais.

Neste tipo de teste, podem ser encontradas as seguintes falhas [BIN 95]: estados e transições não implementadas, estados e transições extras implementados, transições com entradas/saídas incorretas e interrupções de execuções.

O conjunto de valores encapsulados que a classe possui num determinado momento, determina o seu comportamento, sendo este controlado por esses valores encapsulados,

seqüências de mensagens ou ambos. O principal objetivo deste tipo de teste, é realizar um teste no sistema OO, não tentando realizar todas as combinações possíveis. A seleção destas combinações deve ser feita de forma que estas testem comportamentos representativos do conjunto original [HER 99].

McGregor [MCG 96], especifica alguns critérios para a cobertura do teste baseado em estados:

- todos-métodos: executar todos os métodos da classe, garantindo que todos os métodos necessários foram implementados. Todavia, não garante que todos os estados foram implementados também.
- todos-estados: “visitar” todos os estados, garantindo que todos os estados foram implementados, não garantindo, porém, que os métodos também foram;
- todas-transições: identificar métodos e estados não implementados, além da existência de transições extras;
- todas-N-transições: encontrar interrupções não necessárias, cobrindo combinações de transições;
- todos-caminhos: identificar todos os estados e métodos extras. Este critério despande grande esforço de teste, sendo praticamente impossível de ser utilizado [HER 99], como o critério “todos-caminhos” do teste procedimental estrutural [MYE 79].

É ainda sugerido em [MCG 96], uma lista que o desenvolvedor deverá seguir para certificar-se que o referido teste foi realizado em sua totalidade, sendo esta lista aplicada às demais estratégias de teste de *software* OO: (i) através de cada construtor da classe, testar os objetos criados; (ii) testar os métodos “get” e “set”; (iii) testar as pós-condições de métodos modificadores; (iv) testar as combinações de métodos que acessam mesmos atributos e (v) testar a liberação de memória alocada por cada objeto.

3.1.1 Diagrama de estados

A máquina de estados finitos, ou diagrama de estados, faz parte da composição do modelo dinâmico proposto por Rumbaugh [RUM 97]. Sua finalidade é descrever os possíveis padrões dos objetos, atributos e ligações que podem existir em um sistema, ou seja, apresentar o comportamento interno do programa. Neste diagrama, existe a descrição de como um objeto reage quando recebe um estímulo (uma mensagem, que consiste num evento do sistema), através da representação de estados anteriores e posteriores ao recebimento das mensagens, bem como das operações que são realizadas em transições (modificação do estado causada por um evento).

Um estado representa a reação de um objeto aos eventos de entrada. É uma abstração dos valores de atributos e ligações de um objeto [HER 99], sendo que, o estado corresponde ao intervalo entre dois eventos recebidos. Os eventos representam pontos no tempo e os estados intervalos de tempo [RUM 97]. Em uma máquina de estados finitos, os nodos representam estados e os arcos são transições, que são rotuladas com os nomes dos eventos. As condições para que a transição ocorra são apresentadas como expressões entre colchetes, podendo representar pré (quando a condição aparece antes do nome do método) ou pós-condições (quando a condição aparece depois do nome do método). A seqüência de transformações nos estados é descrita como conseqüência das seqüências de eventos.

Com o intuito de exemplificar os conceitos acima expostos, é apresentada na Fig. 3.1 o Diagrama de Estados da Classe *Quarto*, a qual é parte de um hipotético sistema de automação de hotéis.

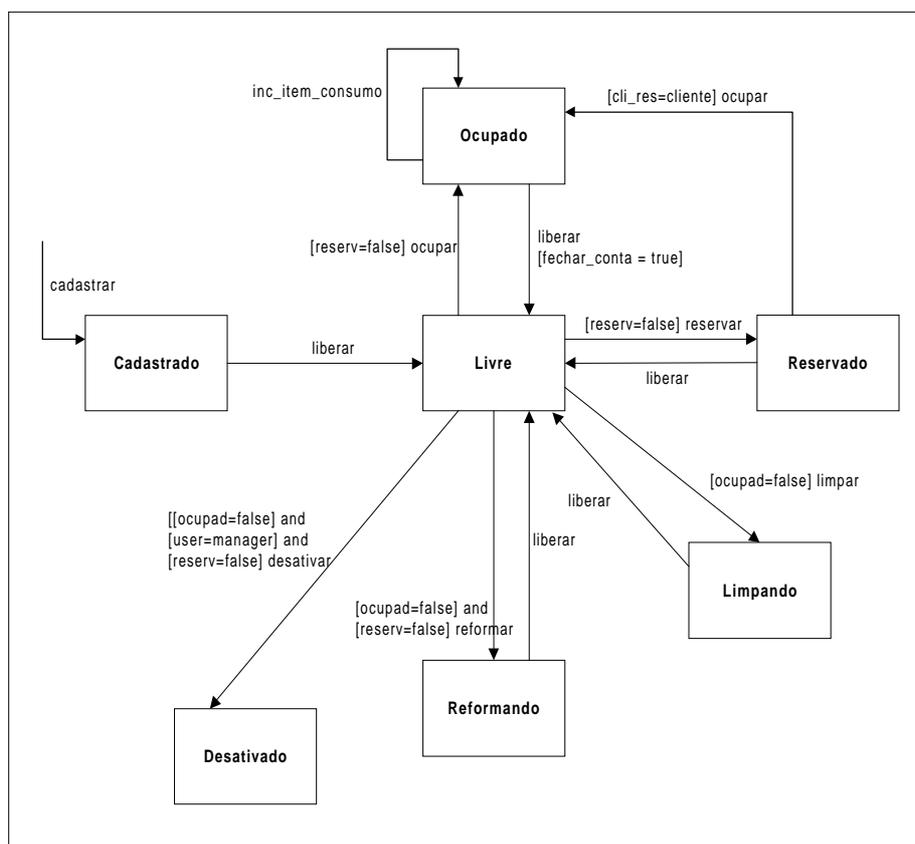


FIGURA 3.1 – Diagrama de Estados da Classe *Quarto*

Para a construção da máquina de estados finitos da classe *Quarto*, primeiramente identificou-se os estados possíveis dos objetos, baseada em sua especificação. São eles: cadastrado, livre, ocupado, reservado, limpando, reformando e desativado.

A próxima etapa consiste na criação de uma árvore de transição: a raiz da árvore origina-se no estado inicial da máquina de estados. Um arco é desenhado para cada transição fora do estado raiz, até um nodo que represente o estado resultante. Este procedimento repete-se para cada nodo do estado resultante, até que este já apareça na árvore na forma de um nodo anterior ou sendo este um estado final. De acordo com Rumbaugh [RUM 97], o próximo passo é a transcrição das seqüências de teste de transições a partir da árvore gerada. Um caso de teste corresponde a cada arco ou cada conjunto de arcos numa mesma direção. Para a execução do teste, inicializa-se o objeto com seu estado inicial, aplicando a seqüência e, logo após, compara-se o estado resultante com o estado esperado.

Conforme descrito por McGregor [MCG 96], os seguintes tipos de erros podem ser encontrados neste tipo de teste: transições perdidas, transições incorretas, ações de saída e estados incorretos.

3.1.2 O uso de asserções no teste de estados

O uso de asserções no teste de *software* OO baseado em estados constitui um importante mecanismo de auxílio a este tipo de teste, sendo utilizado por muitos desenvolvedores [VOA 97].

Segundo Binder [BIN 99], uma asserção é uma expressão *booleana* que define condições necessárias para uma correta execução. O uso das asserções geralmente inclui pré-condições, pós-condições e invariantes de classe. Quando uma dessas condições torna-se falsa, ocorre o que Binder [BIN 99] denomina violação de asserção, significando que uma condição necessária não foi validada. Ressalta ainda que esta situação é geralmente

chamada de falha da asserção, mas na realidade, a asserção não apresentou falha e sim sua especificação para um estado que determinaria válido.

O mesmo autor cita as seguintes vantagens na utilização de asserções: i) estados encapsulados, bem como variáveis podem ser diretamente verificados; ii) especificações incorretas de classes são evitadas, pois uma violação de uma asserção pode revelar um erro de especificação quando uma classe servidora, por exemplo, sinaliza uma mensagem legal com uma violação; iii) uma violação numa pré-condição num método, geralmente revela um erro no método que enviou a mensagem (cliente) e não no método detentor da pré-condição; iv) uma violação numa pós-condição num método, geralmente revela um erro no método detentor da pós-condição (servidor) e não no cliente; v) uma pós-condição pode revelar um estado corrompido resultante de um problema na especificação do algoritmo utilizado no método do objeto cliente, uso incorreto de um objeto servidor ou ainda um erro no próprio objeto servidor.

Várias metodologias de análise e projeto encorajam o uso de asserções, como por exemplo a OMT (*Object Modeling Technique*) [RUM 97] e a UML (*Unified Modeling Language*), possuindo diferentes notações gráficas para representar aplicações OO. Observa-se que, a utilização de asserções na especificação e modelagem de *software* OO é amplamente aceita, da mesma forma que sua utilização no processo de teste de aplicações OO.

4. CONCLUSÕES

O paradigma OO tem contribuído para a reutilização de soluções, evidenciando que o teste deve ser realizado com a finalidade de garantir que defeitos presentes nestas soluções sejam descobertos impossibilitando, assim, sua propagação em reutilizações de futuras aplicações. Apesar do paradigma OO surgir com o objetivo de melhorar a qualidade e produtividade no desenvolvimento de *software*, algumas de suas características tornam a atividade de teste mais complexa do que em outras abordagens. Entre essas características cita-se a herança, o encapsulamento, o polimorfismo e a ligação dinâmica [UNG 97].

A atividade de teste constitui um elemento crítico na busca pela garantia de qualidade de um produto de *software*, procurando torná-lo mais confiável. De acordo com Herbert [HER 99], existe uma carência de técnicas bem estabelecidas para o teste de aplicações desenvolvidas sobre este paradigma, constituindo-se numa área nova de pesquisa e aplicação.

Este trabalho apresentou uma visão sobre Teste de *Software*, seguido do Teste Orientado a Objeto, onde foram apresentadas as principais estratégias de teste sobre esse paradigma. Logo após foi abordado com um maior nível de detalhamento o Teste Baseado em Estados. Neste tipo de teste, o comportamento de uma classe é definido por um conjunto de valores encapsulados que a classe possui em determinado momento (estado), sendo este controlado pelos valores encapsulados, por seqüências de mensagens, ou ambos. Com o emprego do teste baseado em estados, torna-se possível a monitoração de interações entre os objetos, verificando a integridade destes, através da consulta aos valores de seus atributos, analisando se os estados dos objetos são válidos, de acordo com o emprego de asserções. Assim, alguns tipos específicos de erros podem ser detectados e conseqüentemente corrigidos, garantindo uma maior qualidade do produto de *software* desenvolvido.

REFERÊNCIAS BIBLIOGRÁFICAS

- [BIN 95] BINDER, R. V. **Testing object-oriented systems: A Status Report**. Chicago: RBSC Corporation., 1994. Disponível em: <http://www.rbsc.com/pages/site_map.html>. Acesso em: 12 dez. 1999.

- [BIN 99] BINDER, R. **Testing object-oriented systems: models, patterns, and tools**. New York: Addison Wesley, 1999.
- [CHO 78] CHOW, T. S. Testing Software Design Modeled by Finite-State Machines. **IEEE Transactions on software engineering**, New York, v.4, n.3, p.178-187, 1978.
- [HER 99] HERBERT, J. S.; PRICE, A. M. de A. Utilizando Ferramentas no Processo de Teste de Software OO: Gerenciamento e Distribuição de Tarefas. In: **Congresso nacional da Sociedade Brasileira de Computação**, 19., 1999. Rio de Janeiro. **Anais...** Rio de Janeiro: SBC, 1999.
- [MCG 96] MCGREGOR, J. D. Testing object-oriented components. In: **European Conference on object-oriented programming**, 10., 1996, Linz. **Tutorial Notes**. Berlin: Springer-Verlag, 1996.
- [MYE 79] MYERS, G. **The art of software testing**. New York: John Willey & Sons, 1979.
- [PER 00] PERRY, William E. **Effective methods for software testing**. New York: John Wiley & Sons, 2000.
- [PRE 95] PRESSMAN, R. S. **Engenharia de software**. São Paulo: Makron Books, 1995.
- [RUM 97] RUMBAUGH, J. Et al. **Modelagem e projetos baseados em objetos**. Rio de Janeiro: Campus, 1997.
- [SMI 92] SMITH, M. D.; ROBSON, D. J. A framework for testing object-oriented programs. **Journal of object-oriented programming**, [S.l.], v.5, n.3, p.45-53, June 1992.
- [UNG 97] UNGAR, D. et al. Debugging and the experience of immediacy. **Communications of the ACM**, Mountain View, CA, v.40, n.4, p.38-43, 1997.
- [VOA 97] VOAS, J. How Assertions Can Increase Test Effectiveness. **IEEE software**, Los Alamitos, v.14, n.2, p.118-120, Mar/Apr. 1997.

UMA FERRAMENTA PARA AUXILIAR A AVALIAÇÃO DE TEXTOS CONSTRUÍDOS COLABORATIVAMENTE EM AMBIENTES DE ENSINO-APRENDIZAGEM

Gilleanes Thorwald Araujo Guedes¹

Rosa Maria Viccar²

Carmen Barbosa D'Amico³

RESUMO

A ferramenta descrita neste artigo está relacionada ao Ambiente Multiagente de Ensino-Aprendizagem (AME-A), no qual os agentes que o compõem preocupam-se em ensinar e/ou aprender. Esta ferramenta baseia-se, em parte, na idéia do agente Promove_Interação, que tem por objetivo possibilitar que diversos aprendizes e professores se comuniquem, através da Internet e discutam assuntos determinados por um professor. Procurando auxiliar a tarefa do professor em determinar se os aprendizes estão realmente adquirindo conhecimento, desenvolveu-se uma ferramenta para analisar as interações dos aprendizes. O algoritmo desenvolvido utiliza um dicionário de palavras/frases-chaves relacionadas ao assunto em questão, referentes a tópicos que deveriam ser discutidos e/ou fazer parte das conclusões dos alunos. Ao ser ativado, o software identifica os aprendizes e suas respectivas interações e as armazena em uma base de dados; em seguida, avalia as interações de cada aprendiz, verificando a frequência com que este utiliza as palavras-chave através de dois métodos de avaliação, baseados em algumas características de técnicas de KDT. O *software* permite também a classificação de todas as palavras/frases empregadas durante a reunião.

ABSTRACT

The tool described in this paper is built upon the learning-teaching multiagent environment (Ambiente Multiagente de Ensino-Aprendizagem AME-A), in which individual agents are responsible for teaching and/or learning. In part the tool is based on the agent Promote Interaction that makes it possible for several students and teachers to communicate with each other through the Internet and to discuss issues as chosen by the teacher. In an attempt to assist the teacher in assessing the knowledge actually acquired by the students the authors developed a tool to analyze the students' interactions. The algorithm developed for the tool employs a key-sentence/key-word related to the chosen subject that refers to issues that should be discussed by the students. When activated, the software identifies the learners and their interactions and stores these into a database. Following this, it evaluates each learner's interactions, assessing the frequency with which they use the key-words by two different methods, based on KDT techniques. The software also allows for the classification of all words and sentences employed during the meeting.

Keywords: Long-distance Learning, KDT, AME-A, Collaboratively-Built Text Analysis.

INTRODUÇÃO

Este trabalho está baseado na proposta de D'Amico, em [DAM 99], onde é sugerido um ambiente multiagente de ensino-aprendizagem, no qual os agentes que o compõem preocupam-se em ensinar e/ou aprender. A ferramenta descrita neste trabalho baseia-se, em parte, na idéia do agente Promove_Interação, que tem por objetivo possibilitar a diversos

¹ Mestre em Ciências da Computação – convênio URCAMP / UFRGS – uid628@obinonline.com.br

² Professora Doutora do Instituto de Informática da UFRGS - rosa@inf.ufrgs.br

³ Professora Doutora do Instituto de Informática da UFRGS - carmenbd@inf.ufrgs.br

aprendizes e professores se comunicarem, através da Internet, e discutirem assuntos determinados por um professor, construindo textos de maneira colaborativa. Os aprendizes se reuniriam nas datas estabelecidas e tentariam chegar a conclusões por meio de discussões sobre os assuntos em questão, coordenados por um professor. O agente Promove_Interação é representado neste trabalho por um *software* de “chat”, que, embora não implemente ainda muitas das características atribuídas ao agente, permite a reunião virtual de diversos aprendizes e a gravação do conteúdo produzido durante a discussão.

Procurando auxiliar a tarefa do professor em determinar se os aprendizes estão realmente adquirindo conhecimento, desenvolveu-se uma ferramenta para analisar as interações dos aprendizes contidas nos textos por eles construídos durante suas reuniões. Esta ferramenta será também de grande utilidade para o agente Analisa_Aprendizagem, descrito em [DAM 99]. O algoritmo desta ferramenta baseia-se em um dicionário de palavras/frases-chave relacionadas ao assunto em questão, referentes a tópicos que deveriam ser discutidos e/ou fazer parte das conclusões dos alunos. Ao ser ativado, o software aqui descrito identifica os aprendizes e suas respectivas interações e as armazena em uma base de dados; em seguida avalia as interações de cada aprendiz, verificando a frequência com que este utiliza as palavras/frases-chave através de dois métodos de avaliação, baseados em algumas técnicas de KDT (Knowledge Discovery in Text – Descoberta de Conhecimento em Texto); além de permitir também a classificação de todas as palavras/frases empregadas durante a reunião. A ferramenta procura, dessa forma, auxiliar o professor a determinar se seus aprendizes estão ou não adquirindo conhecimento e se não estão desperdiçando tempo em trivialidades.

Pretende-se, com o desenvolvimento desta ferramenta, associada ao software de “chat” acima descrito, enriquecer o ambiente AME-A, fornecendo-lhe, mesmo que muitas das características do agente Promove_Interação não tenham ainda sido implementadas no software de chat, um ambiente em que os diversos aprendizes que utilizam o ambiente AME-A possam virtualmente se reunir, trocar idéias através da discussão de determinados assuntos e produzir textos de maneira colaborativa. Além disso, a ferramenta descrita neste trabalho fornece uma maneira de auxiliar o professor a determinar o avanço de seus aprendizes na aquisição de conhecimentos por meio das discussões, o que permite ao professor verificar os termos mais frequentemente utilizados, bem como determinar se os aprendizes estão discutindo todos os tópicos propostos e não se estão desviando do assunto em questão.

Este artigo primeiramente descreverá brevemente o ambiente AME-A, passando em seguida, para uma pequena revisão de literatura a respeito de KDT e, finalmente, discorrerá sobre a ferramenta desenvolvida.

1. AME-A – AMBIENTE MULTIAGENTE DE ENSINO-APRENDIZAGEM

A ferramenta em questão baseou-se no projeto AME-A, proposto por [DAM 99], mais precisamente no agente Promove_Interação. Este ambiente é composto por uma sociedade de agentes, onde cada agente é responsável por parte do processo de ensino ou por parte do processo de aprendizagem. Os agentes desta sociedade comunicam-se por mensagens, atuam como processos concorrentes e cooperam entre si com o objetivo de ensinar e/ou aprender. O AME-A é formado por um agente humano (que pode ser tanto o aprendiz como o professor) e oito agentes artificiais, além de um banco de dados. Este ambiente almeja fornecer um acompanhamento sobre o assunto desejado ao aprendiz, maximizando seu desempenho.

O agente Promove_Interação é responsável por promover o contato entre os agentes humanos. Este agente deve utilizar inteligência para analisar as interações dos aprendizes

no grupo em relação ao tema proposto, servindo como uma ferramenta de apoio à avaliação pelo professor [DAM 99]. Dessa maneira, utilizou-se um software de “chat” já existente para representar o agente Promove_Interação e efetuaram-se algumas modificações sobre o mesmo, como identificação dos usuários e gravação das interações. Em seguida desenvolveu-se uma ferramenta para analisar as já citadas interações produzidas durante as reuniões dos aprendizes.

2. KDT (KNOWLEDGE DISCOVERY IN TEXT – DESCOBERTA DE CONHECIMENTO EM TEXTO)

KDT também conhecida como *Text Mining* (Mineração de Texto), é um ramo recente de pesquisa da área de Data Mining [GOT 97], mais especificamente da área de KDD (Knowledge Discovery in Databases) [LOH 00]. Data Mining é um processo que busca extrair informações compreensivas, válidas e anteriormente desconhecidas de bancos de dados, de outras fontes de dados da empresa e através da rede para auxiliar na tomada de decisões empresariais [FEL 95], [GOT 97], [RAJ 98].

As aplicações de data mining costumam descobrir de forma semi-automatizada tendências e padrões dentro de enormes grupos de dados [HEA 99]. Nos últimos anos, o uso da mineração de dados tem crescido muito no meio empresarial; no entanto, data mining costuma trabalhar somente com dados estruturados, deixando de lado uma enorme gama de informações textuais que poderiam ser extremamente úteis [GOT 97], [FEL 98], [RAJ 98], [LOH 00].

Além disso, com o advento e imensa utilização da Internet, enormes quantidades de informações foram colocadas à disposição de qualquer pessoa que saiba encontrá-las (como e-mails, chats, home-pages, etc. [WIV 99]). Informações preciosas estão agora disponíveis em centenas de milhares de documentos sem nenhuma ou com muito pouca estruturação [GOT 97] [HEA 99] [WIV 99]. No entanto, devido à falta de estruturação destas informações, torna-se muito difícil decifrar as mesmas de forma automática [HEA 99] [LOH 00]. Wives em [WIV 99] refere-se à essa imensa quantidade desorganizada de informação como “Sobrecarga de Informação”. Recentemente, alguns pesquisadores têm procurado desenvolver ferramentas que possibilitem a extração dessas informações de uma forma ordenada, de maneira que elas possam ser unidas e utilizadas para gerar novos conhecimentos e, assim auxiliarem na tomada de decisões e a chegada de conclusões [GOT 97].

3. A FERRAMENTA DESENVOLVIDA

A ferramenta aqui descrita está associada à idéia do agente Promove_Interação, descrito no ambiente AME-A em [DAM 99]; este agente é representado por um módulo de “chat” independente, ativado em períodos determinados pelo professor. As interações produzidas pelo módulo de “chat” são gravadas em um diretório específico. Tomou-se um software de “chat” já pronto e realizaram-se algumas modificações, como identificação dos aprendizes e gravação das interações produzidas.

Esta ferramenta tem por objetivo auxiliar o professor a determinar se seus aprendizes estão conseguindo produzir conhecimento por meio de textos construídos colaborativamente durante discussões sobre um determinado tema. A ferramenta analisa os comentários dos aprendizes, produzidos durante uma ou mais reuniões através do módulo de “chat”, realizando tarefas como classificação das palavras e frases utilizadas durante a discussão, além de prover dois módulos de avaliação baseados em conjuntos de palavras/frases-chave consideradas relevantes dentro do tema escolhido pelo professor. Os algoritmos utilizados por este software baseiam-se em algumas características de técnicas de KDT. A Figura 3.1

demonstra como interagem o módulo de “chat” e a ferramenta para avaliação das interações produzidas.

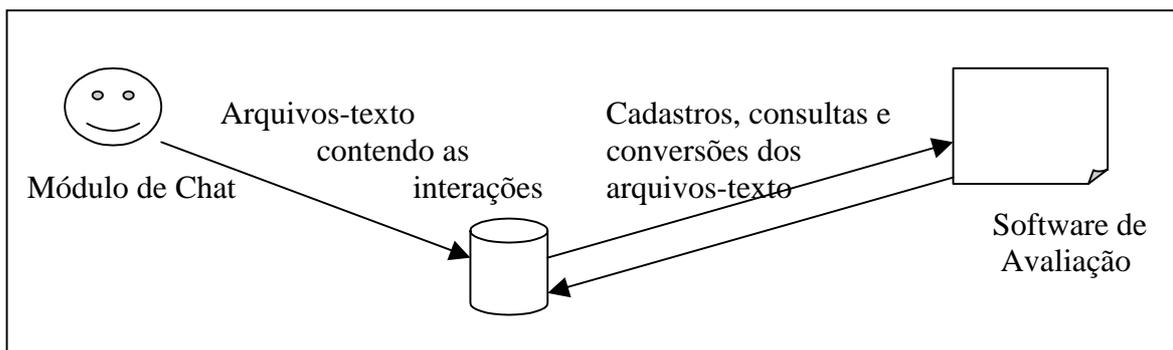


Figura 3.1 – Interação entre o módulo de chat e a ferramenta

A ferramenta de análise textual dispõe de uma subrotina que identifica qualquer novo arquivo proveniente de uma discussão virtual em um diretório específico, embora o *software* permita também a seleção manual de um arquivo pelo professor. Após a seleção manual ou a identificação automática de um novo arquivo, a ferramenta o examina, identifica todos os aprendizes com suas respectivas interações e os grava em tabelas estruturadas. Caso a identificação tenha sido realizada de forma automática, uma mensagem informando o nome do arquivo e a data em que a análise foi realizada é colocada na tela.

A etapa de identificação de aprendizes/interações também “limpa” as interações dos aprendizes, procurando excluir caracteres que atrapalhem a avaliação das interações dos alunos, como pontos de exclamação, interrogação, etc. A etapa de limpeza dos dados é descrita em [WIV 99] como parte do processo de descoberta, além disso, [FEL 98] afirma que a maioria dos sistemas de text mining operam sobre documentos previamente categorizados e classificados e, segundo [LOH 00], diversas técnicas de KDT, primeiro extraem as informações dos seus repositórios textuais e em seguida as gravam em depósitos de dados estruturados, para depois aplicarem técnicas de KDD. Isso também está de acordo com a técnica de Descoberta Tradicional após Extração descrita em [WIV 99], onde os dados não-estruturados são primeiro convertidos em bases estruturadas para depois serem minerados por técnicas convencionais.

Após identificar os aprendizes/interações, o professor pode realizar uma pré-classificação das palavras/frases empregadas durante a discussão, o que já pode ser identificado como uma forma de avaliação, ela permite ao professor observar as palavras utilizadas por um aprendiz específico ou por todos os aprendizes, além de poder fazer esta visualização dividida por cada interação individual ou por todas as interações em conjunto do aprendiz. O professor pode também escolher visualizar somente palavras-chave utilizadas durante a discussão, bem como somente “palavras ferramenta” (também conhecidas como stop words, palavras sem grande significado dentro do contexto, como conetivos, artigos, etc.), ou desconhecidas.

Esta operação está de acordo com [GOT 97] que afirma que um dos objetivos das ferramentas de KDT é encontrar temas predominantes em conjuntos de documentos, além do que pode auxiliar o professor na etapa de Identificação da linguagem, por meio das palavras mais frequentemente utilizadas e nas etapas de Extração de nomes e Extração de termos de domínio, também descritas em [GOT 97]. Os termos mais comumente utilizados

podem ser usados como características [GOT 97], [LOH 00], para dar uma idéia sobre o que os aprendizes estão falando. Este módulo está de acordo com a técnica de Descoberta por Listas de Conceitos-Chave, descrita em [WIV 99], que apresenta uma lista dos principais conceitos do texto, afirmando que o significado do texto pode ser determinado pela análise dos elementos léxicos mais frequentes. Feldman em [FEL 98] destaca também a necessidade de uma taxonomia de termos, para realização de mineração de textos a este nível. Loh, em [LOH 00], também afirma que uma classificação prévia é necessária para criar definições de conceitos. Além disso, este módulo também identifica os substantivos utilizados durante a discussão, essa característica é semelhante a um dos algoritmos descritos no trabalho de [JAQ 99].

Neste módulo, o professor deve selecionar o tema discutido durante a reunião (necessário para identificar as palavras-chave). Em seguida o professor deve escolher se deseja classificar as palavras contidas nas interações de todos os aprendizes ou de um aprendiz específico. O professor deve também determinar se quer identificar todas as palavras contidas no chat ou somente palavras-chave, palavras ferramenta, substantivos ou palavras desconhecidas. Isto está parcialmente de acordo com um dos métodos pré-processamento descritos por [NET 00], antes da aplicação de técnicas de KDT, que afirma ser útil retirar ou ignorar as palavras ferramentas para evitar o excesso de entradas em vetores e deixar a análise mais rápida.

Finalmente, o professor pode definir se deseja uma classificação dividida por interação ou geral, que classificará as palavras contidas em todas as interações do aprendiz. Este módulo classifica todas as palavras contidas no chat: elas são apresentadas por ordem de maior utilização. O módulo informa o número de vezes que a palavra foi utilizada, seu percentual em relação ao total de palavras utilizadas pelo aprendiz e sua classificação (palavra-chave, palavra ferramenta, substantivo ou palavra desconhecida). Isto também está de acordo com a abordagem de Léxico Resumido descrito por Moscarola, citado em [FRE 98], que afirma que se pode ter uma idéia do conteúdo de um texto identificando as palavras que ocorrem mais frequentemente. Este módulo também apresenta características da abordagem de Léxico Seletivo, descrita por Moscarola, já que permite ao professor identificar somente um tipo de palavra (palavras-chave, palavras ferramenta, substantivos ou palavras desconhecidas). É preciso destacar que o termo “palavra-chave” não se refere unicamente a palavras individuais, podendo se referir a um conjunto de palavras. Este módulo, após a classificação, permite uma visualização por meio de gráficos que demonstram o nível de utilização das palavras/frases-chave mais frequentemente utilizadas.

O professor também pode aplicar um dos dois tipos de avaliação disponíveis: padrão e por queries. Na primeira, o professor deve selecionar o tema ao qual se refere o arquivo de chat que vai ser avaliado; se deseja avaliar um único aprendiz ou todos; se deseja pesquisar por palavras/frase-chaves de um nível específico ou de todos; qual o percentual mínimo de interações válidas por aluno em relação ao total de interações geradas pelo mesmo (interações válidas são as interações que contêm um número mínimo de palavras/frases-chave, especificado no campo seguinte da tela e que não ultrapassem o máximo de palavras desconhecidas contidas na interação); o número mínimo de palavras/frases-chave por interação e o percentual máximo de palavras desconhecidas na interação.

Além disso, o professor pode ainda determinar se deseja levar em conta a raiz das palavras analisadas (o número máximo de letras a ser retirado do final ou do início da palavra é determinado no módulo de configurações; esta opção é útil no caso do aluno utilizar palavras/frases consideradas chave, mas mal escritas ou em tempos de conjugação diferentes ou com algum prefixo, por exemplo; no entanto, esta opção pode incidir no erro

de lemas, ou seja, palavras com significados diferentes mas com raízes idênticas [LOH 00]). O professor pode informar ainda se deseja mostrar uma pontuação relacionada aos níveis das palavras encontradas (o valor da pontuação deve ser informado pelo professor ao cadastrar o nível).

Este módulo de avaliação retorna todas as interações consideradas válidas organizadas alfabeticamente por aluno, trazendo o percentual de interações válidas em relação ao total de interações que o aluno gerou, e, se o professor escolheu esta opção, a pontuação total do aluno. Deve-se destacar que esta pontuação é definida por meio das palavras/frases-chave encontradas e o seu respectivo peso, definido pelo professor no momento de seu cadastramento.

Este método de avaliação está de acordo com a técnica de Descoberta por Extração de Passagens, descrito em [WIV 99], que tenta descobrir informações através da utilização de regras que procurem por passagens do interesse do usuário no texto; as passagens no caso são representadas pelas palavras/frases-chave cadastradas pelo professor. Segundo [WIV 99], nesta técnica o usuário não precisa ler todo o texto, mas deve interpretar os trechos retornados. Este método de avaliação também está de acordo com a técnica de Descoberta por Recuperação de Informações, também descrita em [WIV 99], que retorna informações contendo determinados trechos do texto. Além disso, este módulo retorna também a utilização das palavras/frases-chave pelos alunos, em ordem alfabética, e as palavras desconhecidas mais utilizadas, o que também apresenta características da técnica de Descoberta por Listas de Conceitos-Chave, descrita em [WIV 99]. Após os resultados da avaliação, este módulo fornece também uma visualização gráfica de todas as interações consideradas válidas em relação a todas as interações identificadas, de todos os aprendizes ou de um aprendiz específico.

Já a avaliação por queries permite ao professor selecionar quais palavras/frases-chave deseja pesquisar dentro do conjunto de interações de um aprendiz ou de todo o grupo. O professor pode escolher as palavras/frases-chave em um combo contendo todas as palavras-chave relacionadas ao tema escolhido ou, se quiser, pode digitar qualquer palavra ou frase dentro do combo e pressionar a tecla <ENTER>. Todas as palavras/frases-chave selecionadas ou digitadas são adicionadas a uma lista de pesquisa.

Este módulo está baseado na afirmação de [GOT 97] que afirma ser um dos objetivos de KDT pesquisar por documentos relevantes através do uso de queries flexíveis e poderosas. Neste módulo, o professor deve também informar se deseja pesquisar somente interações contendo todas as palavras/frases-chave informadas ou se deseja pesquisar qualquer interação contendo alguma das palavras selecionadas, caso em que deverá informar ainda se quer que só sejam selecionadas interações contendo as palavras selecionadas em ordem ou se aceita qualquer interação que contenha alguma das palavras informadas em ordem ou não. O professor também pode pesquisar palavras/frases por meio de sua raiz, como no módulo de avaliação padrão; o número de letras a serem retiradas de cada palavra para isolar sua raiz é definido no módulo de configurações. Este módulo de avaliação também está de acordo com a técnica de Descoberta por Extração de Passagens descrita em [WIV 99], onde, como já foi descrito, procura-se por passagens do interesse do usuário; e na técnica de Descoberta por Recuperação de Informações, também descrita em [WIV 99]. Neste caso, as passagens são representados pelas palavras/frases escolhidas ou digitadas pelo professor.

Esta ferramenta disponibiliza ao professor os cadastros de temas, com suas respectivas palavras/frases-chave, palavras ferramentas (stop words), substantivos e professores relacionados aos temas que coordenam. Finalmente, existe o módulo de configurações, onde é permitido ao professor estabelecer algumas regras a serem utilizadas nos módulos

de avaliação e classificação de palavras e serve também para informar qual o último arquivo de chat classificado automaticamente. Neste módulo, o professor precisa determinar quantas palavras podem compor uma frase (este valor é utilizado nos módulos de avaliação e classificação); O tamanho mínimo que uma palavra deve possuir, caso o professor queira comparar as raízes das palavras, para impedir que a palavra fique pequena demais e possa se igualar a palavras que possuam uma ou duas letras iniciais iguais, mas que tenham significados totalmente diferentes (erro de lemas [LOH 00]); e o máximo de letras que podem ser retiradas do início ou do fim de uma palavra para compará-la com palavras-chave. Logicamente, a comparação entre uma determinada palavra e uma palavra-chave deve ser realizada de forma que ambas as palavras (ou conjunto de palavras em caso de frases), possuam o mesmo tamanho, assim, esta quantidade é diminuída também da palavra-chave.

4. CONCLUSÃO E TRABALHOS FUTUROS

A avaliação de um texto produzido de maneira colaborativa por meio de uma discussão ocorrida durante uma sessão de “chat” apresenta algumas dificuldades particulares, devido principalmente ao fato de que a conversação é muito coloquial e a discussão é muito dinâmica, sendo muito semelhantes às “tempestades cerebrais” utilizadas em algumas escolas. Além disso, as discussões são por vezes competitivas, onde alguns alunos querem se sobrepor a outros, o que acarreta uma digitação rápida, causando freqüentes erros de português.

A ferramenta desenvolvida, descrita neste trabalho, procura auxiliar o professor a determinar se os aprendizes estão realmente discutindo os temas em questão, se não estão perdendo tempo com trivialidades e se estão realmente aprendendo e produzindo conhecimento. Esta ferramenta então realiza algumas tarefas para este fim, como identificar os aprendizes, bem como suas respectivas interações, apresentar e classificar as palavras/frases utilizadas com seu nível de utilização e disponibilizar algumas formas de avaliação já descritas anteriormente, que basicamente procuram por ocorrências de palavras/frases, consideradas relevantes dentro do contexto discutido, nas interações produzidas pelos aprendizes. Obviamente, é preciso que o professor examine os resultados produzidos pela ferramenta, pois é possível que esta verifique a utilização de termos coerentes, mas não considerados relevantes inicialmente e deva fazer ajustes nas suas opções de avaliação, após os exames preliminares.

Acredita-se estar ajudando a enriquecer o ambiente AME-A, com o desenvolvimento desta ferramenta, associada ao software de “chat” já descrito, fornecendo-lhe, ainda que muitas das características do agente Promove_Interação não tenham ainda sido implementadas no software de “chat”, um ambiente em que os diversos aprendizes que utilizam o ambiente AME-A possam virtualmente se reunir, trocar idéias através da discussão de determinados assuntos e produzir textos de maneira colaborativa. Além disso, a ferramenta descrita neste trabalho fornece uma maneira de auxiliar o professor a determinar o avanço de seus aprendizes na aquisição de conhecimentos por meio das discussões, provendo maneiras do professor verificar os termos mais freqüentemente utilizados, bem como determinar se os aprendizes estão discutindo todos os tópicos propostos e não se estão desviando do assunto em questão.

Como um trabalho futuro, pretende-se melhorar o *software* de “chat” de maneira que ele apresente mais características semelhantes ao agente Promove_Interação. Além disso, pretende-se acrescentar introdutores de sentença, descritos em [SOL 01], ao mesmo, para facilitar a identificação das intenções dos aprendizes durante as discussões.

Seria útil também a adição de um corretor ortográfico que eliminasse erros gramaticais e de digitação dos arquivos de chat, antes destes serem analisados, o que evitaria que muitos termos promissores fossem desprezados pela ferramenta. Outra adição útil seria o acréscimo de um léxico mais completo de português, onde estivessem discriminadas as principais palavras do idioma, identificando sua classificação como substantivo, verbo, etc.

Pretende-se também melhorar os métodos de avaliação da ferramenta, como por exemplo, possibilitar que a ferramenta retorne a média de frequência de uso de uma palavra-chave, o que não é exatamente a quantidade de vezes em que ela foi citada, mas sim, o intervalo de tempo que um aprendiz demora para citá-la. Além disso, pretende-se que a ferramenta possa classificar as interações dos aprendizes de acordo com os introdutores de sentença utilizados por eles durante as discussões.

REFERÊNCIAS BIBLIOGRÁFICAS

- [DAM 99] D'AMICO, Carmen B. Aprendizagem estática e dinâmica em ambientes multiagentes de ensino-aprendizagem. Porto Alegre: CPGCC – UFRGS, tese de doutorado, 1999.
- [FEL 95] FELDMAN, Ronen & DAGAN, Ido. Knowledge discovery in textual databases (KDT). 1st. International Conference on Knowledge Discovery (KDD-95). Montreal, Agosto de 1995. Disponível na Internet (WWW) no endereço (junho de 2001): <http://citeseer.nj.nec.com/feldman98text.html>.
- [FEL 98] FELDMAN, Ronen *et alii*. Text mining at the term level. Nantes: 2nd European Symposium on Principles of Data Mining and Knowledge Discovery (PKDD'98). Setembro de 1998. Disponível na Internet (WWW) no endereço (junho de 2001): <http://citeseer.nj.nec.com/feldman98text.html>.
- [FRE 98] FREITAS, Henrique, MOSCAROLA, Jean, JENKIN, Milton. Content and lexical analysis: A Qualitative Practical Application. Information Systems Research Group (ISRC), University of Baltimore, WP ISRC No. 070498, 1998.
- [GOT 97] GOTTHARD, Willi, MARWICK, Alan & SEIFFERT, Roland. Mining Text Data. DB2 Magazine, Volume 2, Número 3, inverno de 1997. Disponível na Internet (WWW) no endereço (junho de 2001): http://www.db2mag.com/db_area/archives/1997/q4/
- [HEA 99] HEARST, Marti A. Untangling text data mining. Berkeley: School of Information Management & Systems, University of California, 1999. Disponível na Internet (WWW) no endereço (julho de 2000): <http://www.sims.berkeley.edu/~hearst/papers/acl99/acl99-tdm.html>.
- [JAQ 99] JAQUES, Patrícia Augustin. Agentes de software para monitoramento da colaboração em ambientes telemáticos de ensino. Porto Alegre: Pontifícia Universidade Católica do Rio Grande do Sul – PUCRS, 1999.
- [LOH 00] LOH, Stanley; WIVES, Leandro Krug & OLIVEIRA, José Palazzo M. de. Concept-Based Knowledge Discovery in Texts Extracted from the Web. SIGKDD Explorations, Volume 2, Número 1, páginas 29-39, junho de 2000. Disponível na Internet (WWW) no endereço (junho de 2001): <http://www.acm.org/sigs/sigkdd/explorations/issue2-1/contents.htm#Loh>

- [NET 00] NETO, Joel Larocca *et alii*. Document clustering and text summarization. Proceedings: 4th International Conference on Practical Applications of Knowledge Discovery and Data Mining (PADD – 2000), Páginas 41-55. Londres: The Practical Application Company, 2000. Disponível na Internet (WWW) no endereço (setembro de 2001): <http://citeseer.nj.nec.com/297292.html>.
- [RAJ 98] RAJMAN, Martin & BESANÇON, Romaric. Text mining – knowledge extraction from unstructured textual data. Nantes: 2nd European Symposium on Principles of Data Mining and Knowledge Discovery (PKDD'98). páginas 23-26, Setembro de 1998. Disponível na Internet (WWW) no endereço (junho de 2001): <http://liawww.epfl.ch/~inmain/old/tln-gb-public.html>.
- [SOL 01] SOLLER, Amy L., Supporting social interaction in an intelligent collaborative learning system. International Journal of Artificial Intelligence in Education IJAIED, 2001. Disponível na Internet (WWW) no endereço (setembro de 2001): www.cbl.leeds.ac.uk/ijaied.
- [WIV 99] WIVES, Leandro Krug & LOH, Stanley. Tecnologias de descoberta de conhecimento em informações textuais. Porto Alegre: PPGC-UFRGS, 1999.

CLUBE VIRTUAL DE CIÊNCIAS

Sabrina Bet¹

Angelo Augusto Frozza, Esp.²

RESUMO

Este trabalho tem como objetivo apresentar uma proposta para o desenvolvimento de um Clube Virtual de Ciências, que se caracteriza como um ambiente de auxílio à aprendizagem, a ser utilizado por alunos e professores do Ensino Fundamental e Ensino Médio. Para alcançar esse objetivo foram realizados estudos sobre temas relacionados com o uso do computador na educação, a Internet como meio de comunicação e sua aplicação na educação, o ensino de ciências e o uso de laboratórios de ciências. Como consequência desses estudos, conseguiu-se desenvolver o conceito do Clube Virtual de Ciências e o projeto básico para implementação do mesmo, onde foram destacados como principais recursos: a Biblioteca Virtual, que servirá como um repositório de trabalhos; o Catálogo de Projetos, que incentiva o desenvolvimento de trabalhos cooperativos; os Laboratórios Virtuais, que são uma série de aplicativos para simulação de experimentos; e o Banco de Desafios, que compõe-se de testes para avaliação e acompanhamento da aprendizagem. Até o momento, foram implementados os módulos “Biblioteca Virtual” e “Catálogo de Projeto”, possibilitando que o trabalho já tenha condições de ser amplamente divulgado e utilizado.

Palavras-chave: Clube de Ciências, Internet, Biblioteca Virtual, Educação.

ABSTRACT

This work presents a proposal for the development of a Virtual Science Club to assist the teaching of science at the elementary and junior high school levels. To achieve this goal, studies were conducted on themes such as the use of computers in education, the Internet as a means of communication in education, science laboratory usage and inferences of laboratory experiences in the teaching of science. After these studies, it was possible to understand and develop the concept of the Virtual Science Club and to propose its basic functions. Principal resources include the Virtual Library to keep works, the Project Catalog to develop cooperative works, the Virtual Laboratory to simulate experiments and the Test Bank to collect tests that evaluate teaching and learning. At the moment, the Virtual Library and the Project Catalog have been implemented, opening the possibility for more widespread usage of the Virtual Science Club.

Key-words: Club of Science, Internet, Virtual Library, Education.

1. INTRODUÇÃO

Os Clubes de Ciências caracterizam-se como grupos de alunos e professores que se reúnem com o objetivo de transformar as teorias estudadas na sala de aula em experimentos práticos, nos quais podem ser utilizados desde os mais sofisticados equipamentos até materiais cotidianos como garrafas plásticas, caixas de sapatos, entre outros. (SKY VIEW, 2001). Eles estimulam o aprendizado dos alunos, pois os mesmos estão vivenciando a teoria e interagindo com outros alunos. A interação entre pessoas é muito importante para o processo de aprendizagem, pois o compartilhamento de informações possibilita que a pessoa abra seus horizontes e amplie seus conhecimentos e sua compreensão sobre um determinado assunto. (FONSECA JUNIOR, 2001).

Um outro recurso que a educação vem explorando cada vez mais para auxiliar a interação

¹ Acadêmica do curso de Informática, Univ. do Planalto Catarinense – UNIPLAC (sabrina@uniplac.rct-sc.br)

² Professor do curso de Informática, Univ. do Planalto Catarinense – UNIPLAC (frozza@uniplac.rct-sc.br)

entre professores, alunos e o próprio conteúdo didático são os ambientes computacionais, em particular a Internet, por causa das inúmeras facilidades para acesso às informações.

Novas tecnologias, como a Internet, estão permitindo uma interação instantânea com conteúdos informativos, desafios ao pensamento e desenvolvimento do entendimento, possibilitando a troca dessas informações com outras pessoas e/ou grupos. Sua utilização na educação como fonte de pesquisa, em grupos ou individuais, permite a troca de experiências entre alunos, tornando assim, as aulas dinâmicas e interativas, proporcionando um ambiente colaborativo, ressaltando características de união e participação. Com base nisso, desenvolveu-se a idéia do Clube Virtual de Ciências, para permitir que usuários de diversas localidades e instituições pudessem ter acesso às mais variadas informações na área de Ciências, além de que, através de recursos de interatividade, pudessem praticar e aprender um pouco mais sobre os conteúdos ensinados em sala de aula.

O Ministério da Ciências e Tecnologia, através do Livro Verde (TAKAHASHI, 2000), sugere ações para a ampliação do uso dos meios eletrônicos de informação que levem à inclusão social. Uma dessas ações é estimular e capacitar as comunidades a gerar seus próprios conteúdos na Internet, com ênfase em formação para cidadania. Outra ação é a criação de serviços de informação na Internet que disseminem conteúdos para as comunidades especializadas das áreas de ciência, tecnologia, arte e cultura, com *sites* nacionais e estrangeiros, de acordo com sua área de atuação.

2. UMA REFLEXÃO SOBRE O ENSINO DE CIÊNCIAS

O ensino de Ciências compreende o ensino do mundo físico e natural. No Ensino Fundamental essa disciplina é normalmente designada *Ciências*, enquanto que no Ensino Médio divide-se em três outras disciplinas: *Biologia*, *Física* e *Química*. Existem ainda as disciplinas correlacionadas com a área de Ciências, como Matemática, História e Geografia.

Atualmente, se vê uma grande evolução no ensino de Ciências, onde professores e alunos estão cada vez mais estimulados para a pesquisa e o pensamento crítico, isso tudo podendo ser feito através de diferentes modalidades didáticas: aulas de laboratório, de demonstração e de discussão.

Para KRASILCHICK (1987), existem vários pontos que influem negativamente no ensino de Ciências, dentre os principais pode-se citar: falta de laboratório nas escolas; falta de equipamento ou de material para as aulas práticas; falta de auxílio técnico para reparação e conservação de material; memorização de muitos fatos; e, falta de coordenação com as outras disciplinas. Também cita este autor algumas condições para o melhoramento do ensino das Ciências: construção de laboratórios e aquisição de equipamentos adequados; desenvolvimento de programas de aperfeiçoamento para professores; aumento do número de horas-atividade; e aumento de aulas seguidas.

3. PROPOSTA DE UM CLUBE VIRTUAL DE CIÊNCIAS

O Clube Virtual de Ciências pretende ser uma ferramenta auxiliar no processo ensino-aprendizagem, levando para a Internet um pouco do que acontece dentro de um Clube de Ciências do mundo real. Ele caracteriza-se pelo fato de permitir a interação de professores e alunos de uma mesma turma ou de várias turmas de uma mesma escola, bem como, propiciar a interação de alunos e professores de escolas diferentes, sem a necessidade da presença física das pessoas para a troca de experiências.

A criação do Clube Virtual de Ciências visa fornecer facilidades para que alunos e professores de instituições diversas pesquisem materiais, publiquem seus trabalhos e

troquem experiências, auxiliando principalmente as disciplinas relacionadas com a área de ciências naturais.

O projeto está dividido em diversos módulos: a Biblioteca Virtual, o Catálogo de Projetos, os Laboratórios Virtuais, os Banco de Desafios e a gerência do ambiente, sendo que os usuários poderão interagir com o mesmo através de interfaces baseadas em navegadores Web.

3.1. Biblioteca virtual

A necessidade da criação de uma Biblioteca Virtual dentro do Clube Virtual de Ciências surgiu por dois fatores: 1) o fato da existência de uma grande quantidade de dados na Internet que contêm muito “lixo” e são pouco confiáveis; e, 2) a possibilidade de se registrar e divulgar as pesquisas feitas nas escolas.

A Biblioteca Virtual servirá como um repositório de trabalhos, que passarão por uma seleção realizada pelos professores colaboradores do Clube Virtual de Ciências. Assim, os usuários poderão fazer consultas com a mesma confiança que se tem em uma biblioteca real.

Será organizada em disciplinas por áreas: Ciências (Ciências, Biologia, Física, Química) e Matemática, podendo ser consultada por qualquer usuário que possua um navegador Web e acesso à Internet. O conteúdo poderá ser composto por arquivos de texto, sons e imagens, sendo que, no momento de sua inclusão, serão associados a uma disciplina específica e a um conjunto de palavras-chave, a fim de facilitar a consulta.

3.2. Catálogo de projetos

É um ambiente para incentivar o desenvolvimento cooperativo de pesquisas entre diversas turmas e escolas. Deverá conter recursos como fórum e *chat*, para garantir a interação e uma melhor comunicação durante as atividades.

Os projetos serão cadastrados por professores, podendo ser classificados como: aberto ou fechado. Os projetos abertos poderão ter a participação de qualquer visitante do ambiente, ficando livre a participação no fórum e no *chat*. Já, os projetos fechados, serão utilizados apenas pelos usuários que possuem a senha fornecida pelo professor para a entrada no módulo de Projetos. Os projetos fechados normalmente serão criados quando os professores desejarem trabalhar apenas com suas próprias turmas.

Ao cadastrar um projeto, o professor deverá informar: o título, a disciplina, o objetivo do trabalho, a metodologia de desenvolvimento, a data de entrega do trabalho, o tipo (aberto ou fechado) e uma senha (para acesso a projetos fechados).

Os usuários poderão enviar suas contribuições durante todo o período de validade do projeto e após esse período, o professor responsável pelo mesmo deverá fazer uma seleção dos trabalhos recebidos e então disponibilizá-los no ambiente, sendo que esses trabalhos também farão parte do acervo da Biblioteca Virtual.

3.3. Laboratórios virtuais

Um dos mais importantes serviços oferecidos pelo Clube Virtual de Ciências são os Laboratórios Virtuais, que se caracterizam como uma série de aplicativos onde os alunos poderão simular os mais variados tipos de experimentos relacionados com as disciplinas que fazem parte do Clube, interagindo com os mesmos e reforçando, assim, os conceitos estudados em sala de aula. Por se tratar de aplicativos, os experimentos contarão com um tutorial para que os usuários, sejam eles alunos ou professores, não tenham dificuldade na utilização do ambiente de simulação.

A implementação dos Laboratórios Virtuais exigirá conhecimentos avançados na área de programação, por esse motivo serão criados pela equipe de administração do Clube Virtual, ou então por colaboradores, sendo sua publicação e manutenção realizada exclusivamente pelo Administrador do ambiente.

3.4. Banco de desafios

Os Desafios são testes de conhecimentos desenvolvidos pelos professores e disponibilizados no ambiente, para avaliar a aprendizagem dos assuntos estudados. Cada professor, em sua área de conhecimento, poderá criar quantas questões desejar, associando-as a disciplinas e palavras-chave. As questões poderão ser objetivas para facilitar o mecanismo de correção automática.

Os usuários escolherão o tema para o qual deseja responder o desafio e o Banco de Desafios irá montar um questionário com base nas questões cadastradas. O número de questões que aparecem na tela pode ser configurado no ambiente. Aqueles que responderem aos desafios, receberão o resultado logo em seguida. Além disso, será informado para o aluno a quantidade de respostas certas e erradas, bem como, serão indicados quais conceitos ele precisará estudar mais (com base nos seus erros). Também será fornecida uma relação de conteúdos sobre o assunto, existentes na Biblioteca Virtual. Os usuários terão a opção de responder novamente o desafio ou iniciar um novo desafio.

3.5. Gerência do ambiente

O Clube Virtual de Ciências prevê a figura de um superusuário, o Administrador do Sistema, com a função de gerenciar todo o ambiente, através do módulo de Gerência do Ambiente. O Administrador será o responsável pela integridade do ambiente, podendo criar, autorizar e excluir usuários, conforme houver necessidade. Deverá solucionar eventuais problemas com o banco de dados, bem como, será responsável por incluir novas funcionalidades ao ambiente, como por exemplo, os experimentos dos Laboratórios Virtuais. Sua principal ferramenta será o relatório de estatísticas, que deverá conter informações sobre: visitas diárias, visitas mensais, conteúdos mais procurados, projetos mais visitados, desafios mais procurados, laboratórios mais utilizados, entre outros.

4. ANÁLISE E PROJETO DO CLUBE VIRTUAL DE CIÊNCIAS

A descrição do ambiente foi feita por meio da análise dos requisitos funcionais, os quais foram definidos levando-se em consideração a observação das características básicas que um sistema para Web deve ter, como também, por dados obtidos em entrevista com usuários potenciais do ambiente. Além disso, em função do tempo hábil para realizar o presente trabalho, foi definido como objetivo específico apenas a implementação dos módulos Biblioteca Virtual e Catálogo de Projetos, ficando os módulos restantes como sugestão para trabalhos futuros.

Através de análises da proposta do Clube Virtual de Ciências e de outros ambientes relacionados com a área de ciências já existentes na própria Internet, verificou-se a necessidade da separação dos requisitos em três grupos distintos, de acordo com o tipo de usuário do ambiente: Professor, Visitante e Administrador.

O usuário Professor tem um papel fundamental na operacionalização do Clube Virtual de Ciências, uma vez que ele é o responsável pela inclusão do conteúdo na Biblioteca Virtual e pelo cadastro de novos projetos no Catálogo de Projetos (Figura 1).

O usuário Visitante representa o conjunto de usuários que utilizam o ambiente para fazer suas pesquisas.

O usuário Administrador, por sua vez, é responsável pela gerência do ambiente, liberando

o uso do sistema para os usuários novos, assim como, pela inclusão de novos recursos.

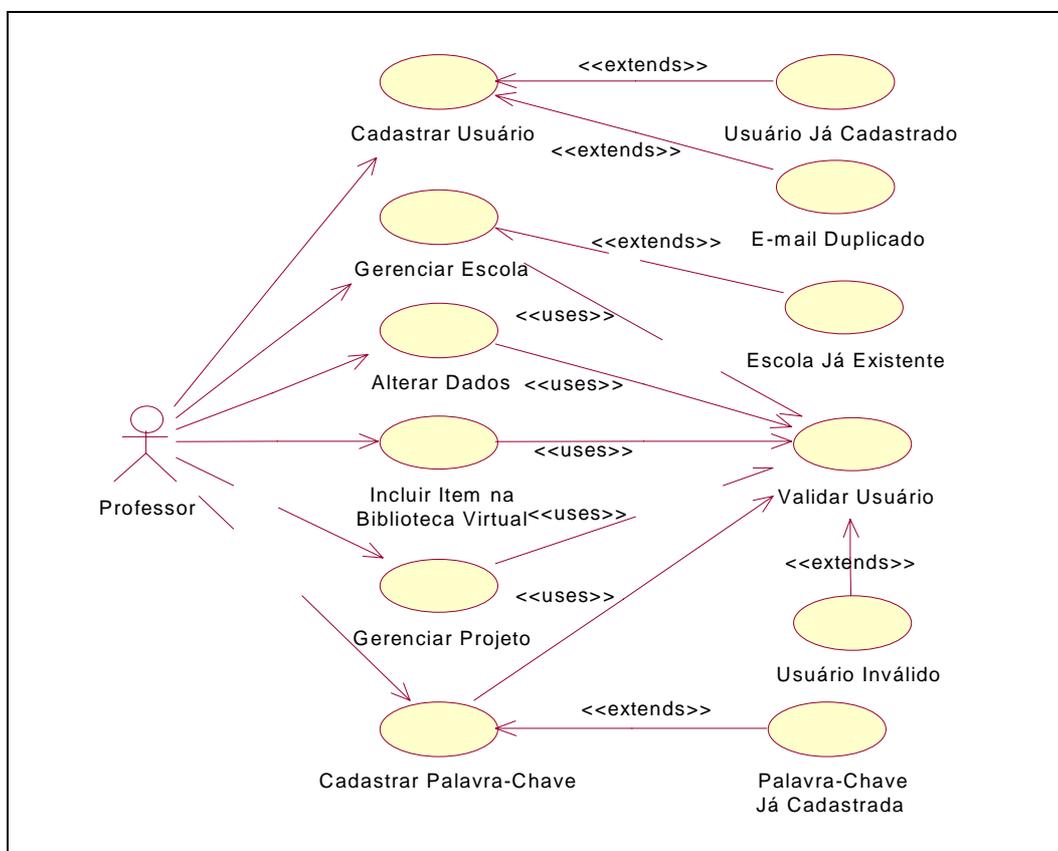


Figura 1 - Especificação dos Requisitos Funcionais do usuário Professor

5. IMPLEMENTAÇÃO DO CLUBE VIRTUAL DE CIÊNCIAS

Para a implementação do projeto, foram selecionadas as seguintes ferramentas: plataforma de programação ASP associada à linguagem *VBScript*, servidor Web IIS (Windows NT), banco de dados MySQL. Além dessas, durante o desenvolvimento do trabalho houve a necessidade da utilização de outras ferramentas como: *Front Page 2000*, *CorelDRAW 9*, e ferramentas de domínio público encontradas na Internet (<http://www.persits.com/>), como o *AspEmail 4.5*, que é um componente de auxílio para o envio de e-mail; e o *AspUpload 2.1*, que é um componente para a transferência de arquivos do computador do usuário para o servidor (*Upload*).

Atualmente, o Clube Virtual de Ciências está disponível no endereço <http://ead.uniplac.rct-sc.br/clubevirtual/>.

Através da página inicial do Clube Virtual de Ciências tem-se acesso ao *Cadastro de Usuários*, à entrada (*Logon*) no sistema, e às consultas na *Biblioteca Virtual* e no *Catálogo de Projetos* (Figura 2).

O *Cadastro de Usuários* é destinado aos professores, que devem fornecer uma justificativa da necessidade de utilização do ambiente, para que o Administrador possa autorizar posteriormente seu uso. Esse procedimento tem por objetivo, limitar o cadastramento somente a professores, garantindo um mínimo de confiabilidade para os conteúdos cadastrados.

Quando um professor conclui seu cadastro, o sistema envia por e-mail uma senha de acesso. No entanto, o acesso aos recursos de *Cadastro de Itens* na *Biblioteca Virtual* e no

Catálogo de Projetos, só estará disponível após o professor receber a autorização do Administrador. Ainda, no caso de esquecer a sua senha, um usuário poderá solicitar que esta seja enviada para o *e-mail* informado no cadastro.



Figura 2 – Página Inicial do Clube Virtual de Ciências

Após informar o Usuário e Senha, o sistema reconhece o tipo de usuário (Professor ou Administrador) e encaminha o mesmo para a interface apropriada. Tanto a interface do Administrador quanto a interface do Professor possuem a opção *Meus Dados*, onde o usuário poderá alterar seus dados pessoais, como: nome, senha, e-mail e instituição de ensino.

As opções disponíveis na interface do Administrador são: *Autorização de Usuários* e *Exclusão de Usuários*. Na *Autorização*, um usuário pode estar em dois estados: “*não*” e “*and*”. O “*não*” refere-se aos usuários que já tentaram acessar o sistema, porém que ainda não estão autorizados a utilizar o ambiente. O “*and*” refere-se a “*andamento*” e identifica os usuários que se cadastraram no ambiente mas ainda não confirmaram seu cadastro, conforme as instruções enviadas por e-mail:

“Você tem o prazo de 30 dias para fazer a confirmação do seu cadastro, para isso, basta acessar o endereço a seguir e entrar no sistema (Entrada), digitando seu *login* e senha. <http://ead.uniplac.rct-sc.br/clubevirtual/>”.

5.1. A interface da Biblioteca Virtual

Existem duas interfaces para a interação com a *Biblioteca Virtual*: a interface de *Cadastro* (Figura 3) e a interface de *Consulta de Itens*. A ação de *Cadastrar Itens* é dividida em 3 passos: a) o usuário inclui um título, uma descrição e especifica a disciplina na qual o item a ser incluído melhor se identifica; b) é solicitada a inclusão de palavras-chave e o tipo de documento que está sendo cadastrado: Imagem, Som, *Site* e Documento (texto); e, c) é feita inclusão do item na Biblioteca Virtual através de *upload* (no caso de arquivo) ou pelo endereço de um *Site*.

A ação de *Consultar* pode ser acessada por qualquer usuário, através da própria página inicial. A consulta inicia pela escolha de uma disciplina: Ciências, Matemática, Física, Química ou Biologia e, então, digita-se uma palavra-chave. O sistema irá fazer a procura no banco de dados, nos campos: palavra-chave, título e descrição. Ao final, uma lista de itens encontrados é mostrada ao usuário, ou uma mensagem avisando que nenhum item foi encontrado.

CLUBE VIRTUAL DE CIÊNCIAS

CADASTRO DE ITEM NA BIBLIOTECA VIRTUAL

Forneça os dados para inclusão do item na biblioteca virtual

Título:

Breve descrição do Trabalho:

Disciplina: Ciências

Avançar

Você está no PASSO 1

> Inclusão de Título

> Palavras-chaves / Tipo do Item

> Indicação Item

CONCLUÍDO

Figura 3 – Interface de Cadastro de Itens na Biblioteca Virtual.

5.2. A interface do Catálogo de Projetos

O Catálogo de Projetos possui as opções de *Inclusão*, *Edição*, *Consulta* e *Participação*. A opção de *Inclusão*, mostrada na Figura 4, possui uma estrutura de formulário para o cadastro do projeto. Esta interface também é usada pela opção de *Edição*, que serve para alteração dos dados dos projetos.

Na opção de *Consulta*, o sistema exibe todos os projetos cadastrados, organizados por disciplina e título, mostrando ao lado de cada projeto seu tipo: aberto ou fechado. Ao selecionar um projeto, o sistema irá apresentar uma nova tela com os detalhes do projeto e três botões: “Entrar no Chat”, “Participar do Fórum” e “Contribuir” e, se o projeto for fechado, aparecerá também um campo para informar a senha.

CLUBE VIRTUAL DE CIÊNCIAS

CADASTRO DE PROJETOS

Forneça os dados para inclusão do projeto:

Título:

Disciplina: Ciências

Objetivo:

(O que se pretende com o Projeto)

Metodologia:

(Como deverá ser desenvolvido)

Data de Entrega: Dia: 01 | Mês: Janeiro | Ano: 2001

Tipo: Aberto Fechado Senha:

Observações:

Cadastrar Projeto

Figura 4 – Interface de Cadastro de Projetos

Quanto ao *Chat* e ao *Fórum*, optou-se por usar programas de terceiros, em vez de implementar um código novo. No endereço <http://www.aspbrasil.com.br>, foram encontrados os dois programas, desenvolvidos por Mário César Mancinelli de Araújo, em ASP e utilizando banco de dados Access.

Procurou-se desenvolver uma interface limpa, usando apenas recursos básicos, a fim de tornar o sistema rápido e, sobre tudo, fácil de utilizar por usuários menos experientes. Além disso, deverão ser desenvolvidos mecanismos para tirar possíveis dúvidas sobre a utilização do ambiente, como: ajuda, FAQ e assistentes (*wizards*).

6. CONCLUSÃO

A idéia de se criar o Clube Virtual de Ciências surgiu, inicialmente, da necessidade de organizar a grande quantidade de informações que se propagam na Internet em grande velocidade. Procurou-se fazer visitas às escolas da rede pública e privada da região, apresentando a idéia inicial do projeto e tentando conhecer um pouco mais a estrutura do ensino de Ciências. Essas visitas proporcionaram um embasamento inicial e auxiliaram na elaboração de novas idéias para o Clube Virtual de Ciências.

Pode-se verificar, através dos estudos, que o computador é um ótimo instrumento para o ensino e destaca-se principalmente pelos seus recursos audiovisuais e interativos, além de ser uma das tecnologias mais utilizadas no mercado de trabalho. A Internet é uma tecnologia que traz um novo conceito de comunicação, e vem sendo utilizada na educação como fonte de pesquisa. Por outro lado, muitos autores questionam sua utilização, afirmando que os alunos usam apenas o recurso “copiar e colar”, não produzindo nada de novo.

Com base nisso, definiu-se o conceito de Clube Virtual de Ciências, bem como, os principais módulos que o compõe: Biblioteca Virtual, Laboratórios Virtuais, Banco de Desafios e Catálogo de Projetos. Por fim, foram implementados os mecanismos de controle da Biblioteca Virtual e do Catálogo de Projetos, juntamente com seus canais de interação (*chat* e fórum).

Pretende-se dar continuidade no desenvolvimento do Clube Virtual de Ciências, fazendo a implementação dos Laboratórios Virtuais e do Banco de Desafios. Outras ações, que visam a continuidade do trabalho são: criar projetos de uso pedagógicos do ambiente, em parceria com os cursos de graduação na área de educação; estabelecer convênios de uso e divulgação com Secretarias de Educação, em nível municipal e estadual; promover oficinas de treinamento no uso operacional e pedagógico do ambiente; além de disponibilizar o código para que possam ser criados Clubes Virtuais de Ciências regionais.

REFERÊNCIAS BIBLIOGRÁFICAS

BRASIL. Lei nº 9.394, de 20 de dezembro de 1996. Lei de Diretrizes e Bases da Educação Nacional, Lei nº 9394/96. **Ministério da Educação e Cultura**. Brasília: DF, 1996.

Disponível em: <<http://www.mec.gov.br/home/legislacao/default.shtm>>. Acesso em: 11/05/2001.

BRASIL. Parâmetros Curriculares Nacionais. **Ministério da Educação e Cultura**.

Brasília, DF, 1995. Disponível em: <<http://www.mec.gov.br/sef/ensfund/paramnac.shtm>>. Acesso em: 11/05/2001.

FONSECA JÚNIOR, Fernando Moraes. Tecnologias em educação a distância. **TV Escola**.

Disponível em: <<http://www.tvebrasil.com.br/salto/distancia/default.htm>>. Acesso em: 10/04/2001.

KRASILCHIK, Myriam. **O Professor e o currículo das ciências**. São Paulo: EPU – Editora da Universidade de São Paulo, 1987.

SKY VIEW. **Clube de Ciências**. Apresenta textos e experiências de clubes de ciências.

Disponível em: <<http://www.skyview.com.br>>. Acesso em: 06/03/2001.

TAKAHASHI, Tadao (Org.). **Sociedade da informação no Brasil: livro verde**. Brasília: Ministério da Ciência e Tecnologia, 2000.

OS SUPERMERCADOS E A TECNOLOGIA DE INFORMAÇÃO

Lóren Pinto Ferreira Gonçalves¹

Everaldo Lima Gonçalves²

RESUMO:

Este artigo trata sobre a utilização da tecnologia da informação em supermercados, evidenciando os objetivos buscados com a implantação destas tecnologias em dois momentos distintos: no início da sua utilização (década de 80) e nos dias atuais. É dada ênfase à tecnologia de mineração de dados, mais especificamente às regras de associação, que vêm sendo utilizadas neste tipo de dados.

Palavras-chave: Supermercados, varejo, tecnologia de informação, automação, mineração de dados, regras de associação, análise de cesta de supermercado.

ABSTRACT:

The use of information technology in retail supermarkets continues to evolve. This paper describes the use of this technology at its inception in the 1980's and compares this to current. It practices in the industry with particular emphasis on the technology of data mining. Specifically, the authors discuss the association rules and their use in modern data mining techniques.

Key-words: Supermarkets, retail, information technology, automation, data mining, association rules, market basket analysis.

1. INTRODUÇÃO

Não há outro setor da atividade econômica no País que tenha crescido tanto, do zero ao estágio atual, em prazo tão curto quanto o varejo (ABRAS, 1999). Os supermercados conquistaram a condição de maiores abastecedores de alimentos e artigos de higiene e limpeza, longe de quaisquer subsídios ou favores oficiais, graças a uma vocação para o crescimento impulsionada por investimentos contínuos e crença no desenvolvimento do Brasil.

Um dos grandes desafios a serem enfrentados pelo setor de supermercados no Brasil será, sem dúvida, o de como aglutinar e dominar a informação.

O varejo, juntamente com os bancos é um dos ramos de negócios que mais investiu em novas tecnologias de informação nos últimos tempos (MARCOVITCH, 1996). A tecnologia de informação surgiu como uma ferramenta de redução de custos e agilizadora do processo de troca de informações (GONÇALVES e GONÇALVES FILHO, 1995 *apud* MARCOVITCH, 1996). Porém, as novas tecnologias deixam de ser apenas uma forma de melhorar a eficiência interna ao ponto de venda e passam a intermediar as relações do varejo com fornecedores e clientes finais.

Uma das tecnologias que pode ser utilizada na busca de informações que possam estreitar a relação entre supermercados e seus clientes é a mineração de dados.

1 Mestre em Administração – PPGA/EA/UFRGS - Bacharel em Informática - URCAMP - Professora do CCEI e CCS - URCAMP - loren@altnet.com.br

2 Bacharel em Informática - URCAMP - Diretor do CPD da Prefeitura Municipal de Candiota - elg@altnet.com.br

2. SUPERMERCADOS

Segundo ROJO (1998), os supermercados são lojas com o método de auto-serviço no varejo de alimentos. Os produtos oferecidos incluem uma ampla variedade como: hortifrutigranjeiros, mercearia, frios e laticínios, carnes frescas e não-alimentos básicos (perfumaria e limpeza).

2.1. OS SUPERMERCADOS NO BRASIL

De acordo com a ABRAS (Associação Brasileira de Supermercados, 1999), não há outro setor da atividade econômica no País que tenha crescido tanto, em prazo tão curto (Quadros 1 e 2). Em mais de quatro décadas, o auto-serviço brasileiro impôs-se como a forma mais moderna, econômica e racional de se adquirir uma infinidade de produtos.

Os supermercados conquistaram a condição de maiores abastecedores de alimentos e artigos de higiene e limpeza, longe de quaisquer subsídios ou favores oficiais, graças a uma vocação para o crescimento, impulsionada por investimentos contínuos e crença no desenvolvimento do Brasil (ABRAS, 1999). O setor é o responsável pela distribuição de mais de 82% dos gêneros de primeira necessidade.

Quadro 1 - Evolução das vendas de supermercados (ROJO, 1998)

Ano	Vendas (bilhões de dólares)
1990	28.7
1991	25.7
1992	26.9
1993	28.1
1994	37.5
1995	43.7
1996	46.5
1997	46.6

Quadro 2 - Números gerais do setor (AGAS, 1998)

Item	Valor
Faturamento em 1997	US\$ 46,6 bilhões
Percentual sobre o PIB	6,2%
Número de lojas	47.847
Número de empregados	655.000

Uma análise do setor desenvolvida pela AGAS (ROJO, 1998) reforça a percepção de que, após a estabilização da economia, ocorreu um acirramento da concorrência, o volume de vendas cresceu, enquanto as margens de lucro foram pressionadas para baixo, levando as empresas a perseguir vantagens competitivas por meio de serviços melhores e da busca incessante da eficiência administrativa. A AGAS, segundo ROJO (1998), afirma que o segmento supermercadista está sendo obrigado a repensar seu negócio, necessitando de muitos ajustes em busca de um novo modelo de operação baseado no controle mais eficiente dos negócios e na satisfação dos clientes.

Um dos grandes desafios a serem enfrentados pelo setor de supermercados no Brasil será, sem dúvida, o de como aglutinar e dominar a informação (GONÇALVES, 2001). Diante de uma concorrência, tanto interna como internacional que se prenuncia cada vez mais acirrada e competente, conhecer o próprio negócio e ter acesso a diferentes fontes de informação serão condições vitais para a sobrevivência no mercado.

O varejo, juntamente com os bancos é um dos ramos de negócios que mais investiu em novas tecnologias de informação nos últimos tempos (MARCOVITCH, 1996).

2.2. AUTOMAÇÃO NOS SUPERMERCADOS BRASILEIROS

Desde a década de 80, o varejo brasileiro tem incorporado novas tecnologias de forma crescente.

Num primeiro plano buscou-se a melhoria da eficiência interna: a partir da venda de um produto ao consumidor final, dava-se baixa no estoque e acionava-se o setor de compras quando os níveis de estoque baixavam, para que fosse providenciada a reposição. A idéia era controlar e melhorar a eficiência interna da empresa, principalmente através de um giro mais rápido de estoques. Em seguida implantou-se a leitora óptica e o código de barras e as máquinas de preenchimento de cheques; estas tinham como intuito aumentar a velocidade de passagem do cliente pelo 'check-out' e, portanto, reduzir filas (GONÇALVES, 2001). Trata-se de uma ação com dois benefícios claros: um do lado do cliente ao oferecer maior rapidez e menor espera; outro do lado do varejo, ao permitir o atendimento de maior número de clientes com o mesmo número de *check-outs* (MARCOVITCH, 1996).

Em 1980 o Pão de Açúcar em São Paulo apresentou o *scanner* na frente da loja da Rua Maranhão. Em 1986, o Supermercados Real, do Rio Grande do Sul, automatizou uma loja Kastelão de 10.000 m². Em 1989, o Bompreço adotou o sistema HiperFarol, de Maceió-AL. Um importante marco neste aspecto aconteceu em 1987, com a implantação e padronização do código de barras, realizada pela Associação Brasileira de Automação Comercial (ABAC), hoje EAN Brasil.

A partir dos anos 90, a automação dos supermercados ganha impulso, com o fim da reserva de mercado de informática. Nasceu uma nova era para os supermercados, que agora podem oferecer aos seus consumidores o que há de melhor no mundo em tecnologia.

De acordo com o Super Censo da Abras (ALBUQUERQUE, 2000), 41,4% das lojas, em 2000, tinham leitor óptico no caixa. Os campeões em automação eram, em primeiro lugar o Espírito Santo, com 84,2% e em segundo o Rio Grande do Sul, com 52,2%.

A Figura 1 mostra a evolução do número de lojas automatizadas, no Brasil, até 1997.

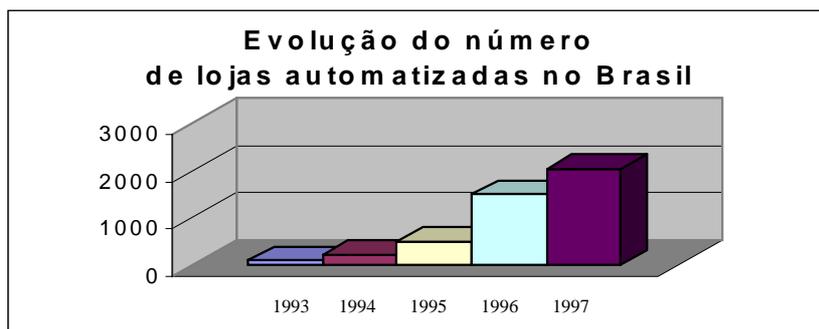


Figura 1 - Número de lojas automatizadas (AGAS, 1998)

Na segunda etapa deste processo, na metade da década de 90, a inovação tecnológica passou a ser utilizada como uma forma de agregar valor ao cliente, através de informação, serviços e facilidades (GONÇALVES e GONÇALVES FILHO, 1995 *apud* MARCOVITCH, 1996).

Com a estabilidade monetária, o setor de supermercados foi obrigado a se profissionalizar, porque as margens de comercialização diminuíram. Algumas empresas não se adaptaram aos novos tempos e fecharam, outras venceram os desafios e cresceram ainda mais. Iniciou-se um período de fusões e aquisições no setor. A abertura econômica, ampliada pelo governo de Fernando Henrique Cardoso, tornou o mercado brasileiro mais atraente às grandes redes de varejo (ABRAS, 1998).

A tecnologia de informação surgiu como uma ferramenta de redução de custos e agilizadora do processo de troca de informações (GONÇALVES e GONÇALVES FILHO, 1995 *apud* MARCOVITCH, 1996).

As novas tecnologias deixam de ser apenas uma forma de melhorar a eficiência interna ao ponto de venda e passam a intermediar as relações do varejo com fornecedores e clientes finais (Marcovitch, 1996).

Uma das tecnologias que pode ser utilizada na busca de informações, sobre o comportamento do consumidor, que possa estreitar a relação entre os mesmos e os supermercados, é a mineração de dados.

2.3. MINERAÇÃO DE DADOS EM SUPERMERCADOS

A mineração de regras de associação é particularmente importante quando se tenta encontrar associação relevante entre itens dentro de uma transação. Um exemplo do resultado de tal mineração seria a declaração de que 80% das transações onde foram adquiridas fraldas também foi comprado leite.

A descoberta de regras de associação tem por objetivo encontrar relacionamentos ou padrões freqüentes em conjuntos de dados (GONÇALVES, 2001). O interesse nesta busca de informações ocorre devido, principalmente ao progresso feito na tecnologia de código de barras, que tornou possível para as organizações de varejo coletar e armazenar grandes quantidades de dados referentes às vendas efetuadas, conhecidos como dados da cesta. Um registro destes dados tipicamente consiste da data da transação e dos itens comprados. Organizações de sucesso atualmente vêem tais bancos de dados como importantes peças da sua infra-estrutura de *marketing*, pois permitem que o processo de *marketing* seja dirigido, além de auxiliar em programas e estratégias customizadas como reorganização do *layout* das lojas e projeto de catálogos. Como exemplo de uma regra que poderia ser encontrada em um banco de dados de um supermercado seria o fato de que 90% dos clientes que compram o produto A, também adquirem, na mesma ocasião, o produto B (análise do comportamento do consumidor no comércio varejista).

A análise de uma cesta de supermercado nos fala de um consumidor, mas a análise de todas as compras feitas pelos consumidores fornece muito mais informações (BERRY e LINOFF, 1997). Consumidores não são iguais, cada um compra uma combinação de produtos diferentes, em diferentes quantidades, em horários e dias diferentes durante a semana. A análise de cesta de supermercado utiliza a informação sobre o que os consumidores adquirem para dar a compreensão de quem são eles e por quê eles fazem certas compras. Esta análise dá o discernimento para a comercialização mencionando quais produtos tendem a ser comprados em conjunto e quais são mais adequados a promoções. Suas raízes estão em analisar transações feitas nos pontos de vendas.

Segundo os mesmos autores, esta análise também é realizada como um ponto de partida quando os dados das transações estão disponíveis e não se sabe que padrões específicos devem ser procurados. Este é um exemplo de mineração de dados indireta, porém a análise de cesta de supermercado pode ser utilizada para mineração de dados direta e indireta.

As técnicas que estão por baixo da análise de cesta de supermercado são oriundas da probabilidade e da estatística. O apelo da análise de cesta de supermercado vem da clareza e utilidade dos seus resultados, os quais são apresentados em forma de regras de associação.

Segundo BERRY & LINOFF (1997), os três tipos de regras produzidas por esta análise são: as regras úteis, as triviais e as inexplicáveis.

As regras úteis contêm alta qualidade, informação para ação. Uma vez que o padrão é encontrado, não há dificuldade em justificá-lo. A conhecida regra sobre fraldas e cervejas nas quintas-feiras sugere que ao anoitecer de quinta-feira, jovens casais se preparam para o final de semana estocando fraldas para os bebês e cerveja para o pai (o qual irá assistir futebol na sexta-feira tomando cervejas). Mais importante do que sugerir as causas é o fato de que os gerentes agora podem agir. Localizando a estante de fraldas perto do corredor onde encontram-se as cervejas, eles podem aumentar as vendas de outros produtos. Devido à facilidade de entendimento da regra outras ações podem ser tomadas, tais como: colocar outros produtos para bebês e/ou para homens entre as cervejas e as fraldas, colocar outros produtos, tais como salgadinhos e aperitivos perto dos dois produtos (BERRY E LINOFF, 1997).

Os resultados inexplicáveis não sugerem um curso de ação.

A análise de cesta de supermercado começa com transações contendo uma ou mais ofertas de produtos ou serviços e alguma informação pouco desenvolvida sobre a transação.

Os dados usados para análise de cesta de supermercado são tipicamente os dados detalhados das transações capturadas nos pontos de vendas.

A análise de cesta de supermercado tem utilidade para a área de varejo, tal como supermercados, lojas de conveniências, drogarias e redes de lancherias, onde muitas das compras são efetuadas à vista. Transações à vista são anônimas, significando que a loja não tem conhecimento algum sobre os clientes porque não há informação alguma o identificando na transação. Para transações anônimas, geralmente a única informação conhecida sobre a compra é a data e o tempo, a localização da loja, o caixa, os itens comprados, e o troco. Para a análise de cesta de supermercado mesmo estes dados limitados fornecem resultados interessantes e que podem gerar ação (BERRY & LINOFF, 1997).

3. CONSIDERAÇÕES FINAIS

Os supermercados encontram-se em uma época de concorrência acirrada, por isso a busca pela eficiência administrativa é essencial. Nesta busca, a informação acionável sobre o cliente é de extrema importância. A tecnologia da informação é uma das armas utilizadas, pelas empresas, nesta guerra pela conquista de seu mercado consumidor.

O conhecimento das preferências, ou seja, dos padrões de compras dos clientes é uma informação valiosa que pode trazer vantagem competitiva aos supermercados. Uma das tecnologias que vem sendo utilizada para este fim é a mineração de dados, mais especificamente as regras de associação (análise de cesta de supermercado) que visa encontrar regras e padrões de compras que estão escondidas nas montanhas de dados armazenadas por estas empresas.

Esta análise dá o discernimento para a comercialização dizendo quais produtos tendem a ser comprados em conjunto e quais são mais adequados a promoções. Esta informação pode sugerir novos *layouts* para o interior das lojas e para catálogos, determinar quais produtos podem formar pacotes promocionais, etc.

4. REFERÊNCIAS BIBLIOGRÁFICAS

ABRAS (Associação Brasileira de Supermercados) [25 de novembro de 1999] Disponível na World Wide Web <<http://www.abrasnet.com.br>>.

AGAS (Associação Gaúcha de Supermercados) [25 de novembro de 1999] Disponível na World Wide Web <<http://www.agas.com.br>>

ALBUQUERQUE, Eliete. **Pequenos Supermercados têm espaço**. Zero Hora. 24 de setembro de 2000. Caderno Economia.

BERRY, Michael J. A., LINOFF, Gordon. **Mineração de dados techniques: for marketing, sales and customer support**. USA: Wiley Computer Publishing, 1997.

GONÇALVES, Lóren P. F. **Avaliação de ferramentas de mineração de dados como fonte de dados relevantes para a tomada de decisão – aplicação na Rede Unidão de Supermercados, São Leopoldo-RS**. Dissertação de Mestrado. PPGA/EA/UFRGS. Porto Alegre, 2001.

MARCOVITCH, Jacques (Org.). **Tecnologia da informação e estratégia empresarial**. São Paulo: FEA/USP, 1996.

ROJO, Francisco J. G. **Supermercados no Brasil: qualidade total, marketing de serviços, comportamento do consumidor**. São Paulo: Atlas, 1998.

ARQUITETURA DE APLICAÇÕES VOLTADAS PARA WEB

*Guilherme Silva de Lacerda*¹

*Alexandre Ramires de Castro*²

*Claudimir Zavalik*³

RESUMO

Na arquitetura de aplicações para *Web*, destaca a importância do desenvolvimento e da distribuição das aplicações no ambiente Internet, citando ferramentas e tecnologias.

Palavras-chave: Desenvolvimento, Web, Banco de Dados, Arquitetura Internet.

ABSTRACT

We describe the architecture of Web applications, in addition to showing the importance of development and distribution of applications in the Internet environment.

Keywords: Development, Web, Database, Internet Architecture

1. INTRODUÇÃO

Quando se fala em sistemas de informações, estes devem ser fundamentalmente caracterizados por alguns aspectos: integridade dos dados, segurança, disponibilidade de informações e desempenho de transações [GHE 1991]. Agregando-se a estas características, tem-se a *Web*, atualmente tão comentada e que está inserida cada vez mais dentro da área dos sistemas de informações [ZAV 2000]. O aumento significativo das tecnologias voltadas para a *Web*, junto com a necessidade das empresas disponibilizarem informações e serviços aos seus clientes passam a ser requisitos mínimos para uma melhor dinâmica de mercado.

Mais importante do que armazenar informações de forma organizada e segura em um banco de dados, é disponibilizá-las a quem de fato as necessita. Uma aplicação utilizando a *Web* como meio de comunicação entre o usuário e o sistema passa a ser indispensável para a obtenção de resultados eficientes.

As pesquisas na área de *Web* e Banco de Dados vêm aumentando ao longo do tempo, abrangendo vários tópicos distintos. Entretanto, pode-se definir, em linhas gerais, três níveis de pesquisa: a *Web* vista como um enorme Banco de Dados distribuído, o estudo de técnicas e ferramentas de desenvolvimento de aplicações *Web* integradas a Banco de Dados e o acesso a Banco de Dados via *Web*.

Atualmente, existem várias ferramentas que permitem o desenvolvimento de aplicações para a *Web*, com características peculiares a cada plataforma operacional. Neste contexto, busca-se descrever a experiência de desenvolvimento de aplicações *Web* na elaboração de níveis de abstração relevantes para o processo [LAC 2001].

¹ Bel. em Informática (URCAMP), Mestrando em Informática (UFRGS). É Diretor de Tecnologia da Apoena Software Livre. Instrutor de Programação Java, Orientação a Objetos e Padrões de Codificação e Qualidade em Desenvolvimento. E-mail guilherme_lacerda@bol.com.br

² Bel. em Informática (URCAMP), Pós-Graduando na Especialização em Banco de Dados (ULBRA). É Analista de Sistemas pela InfoSaúde - Gestão Informatizada em Saúde. E-mail xandre@matrix.com.br

³ Bel. em Informática (URCAMP) e Mestrando em Informática (UFRGS). É Sócio-gerente da Apoena Software Livre. E-mail zavalik@matrix.com.br

2. ACESSO A BANCO DE DADOS VIA WEB

A grande necessidade de se armazenar informações e, mais importante que isso, distribuí-las para quem de fato as necessita, fez com que desenvolvedores se preocupassem com a utilização de mecanismos para armazenamento e gerenciamento de dados na *Web* [ROW 1998].

Muitas aplicações *Web* armazenam dados em arquivos convencionais, controladas por linguagens como C, C++ e *Perl*. Estas aplicações não são adequadas para o tratamento de um grande volume de informações [LIM 1997].

Através da integração de Banco de Dados e a *Web* se torna possível criar mecanismos de gerenciamento e recuperação de informações de maneira mais correta.

Segundo [LIM 1998], algumas características que se destacam dos Sistemas Gerenciadores de Banco de Dados (SGBDs) são:

- suportar várias organizações de arquivos e estrutura de acesso auxiliares;
- processar e otimizar consultas;
- controlar autorizações de acesso aos dados;
- possuir suporte a transações em ambientes multiusuário;
- utilizar linguagens específicas de consulta que independem da aplicação.

Para demonstrar esta interação do acesso a Banco de Dados via *Web*, pode-se dividir em três níveis de abstração [ZAV 2000]: interação cliente-servidor, interação do *browser* com o Banco de Dados e estrutura interna dos programas.

2.1. Interação Cliente/Servidor

Para que ocorra a interação, deve-se relacionar alguns itens relevantes, que podem ser visualizados conforme Figura 1 [BUT 1997], [ROW 1998], [ZAV 2000].

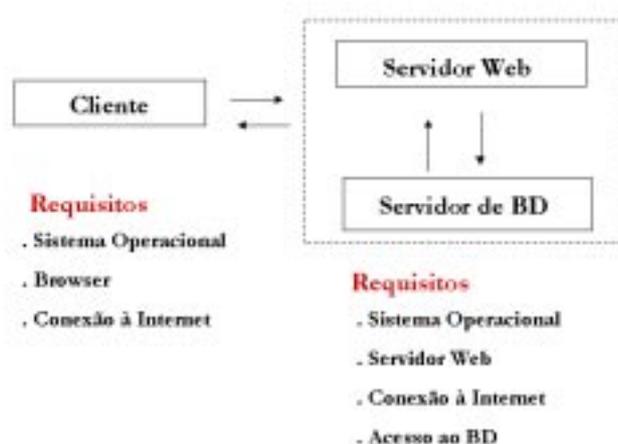


Figura 1: Interação Cliente-Servidor

2.2. Interação Browser/BD

A interação do *browser* com o Banco de Dados pode se dar de duas formas [BUT 1997] e em dois aspectos: aplicações no cliente interpretadas, através de aplicações *JavaScript*, *VBScript* e compiladas como as *applets* [SUN 2001] ou através de aplicações armazenadas no servidor compiladas como CGIs e *servlets* [LIM 1997], [ROW 1998], [ZAV 2000] e aplicações interpretadas como PHP, ASP e JSP.

Conforme Quadro 1, tem-se as principais formas de conexão a Banco de Dados.

Quadro 1 – Tecnologias de Conexão de com BDs via Web

Forma	O que é...
API	<i>Application Programming Interface</i> são funções, normalmente fornecidas com o Banco de Dados, nativas de algumas linguagens de programação como C, por exemplo, contendo as principais funcionalidades necessárias para realizar operações com o Banco de Dados. Podem estar incorporadas aos CGIs.
SSI	Os <i>Server-Side Includes</i> existem para estender as funcionalidades dos servidores <i>Web</i> . São mais simples que os CGIs. São <i>tags</i> que são substituídos dinamicamente em páginas HTML.
CGI	Segundo [BRU 1999], o CGI (<i>Common Gateway Interface</i>) é a definição de interface entre o servidor <i>Web</i> e os demais recursos do computador servidor. Pode ser interpretado ou compilado. O acesso de CGIs a Banco de Dados pode ser feito com ou sem o auxílio de APIs.
JDBC	O JDBC (<i>Java DataBase Connectivity</i>) é uma versão escrita em <i>Java</i> da ODBC. É um protocolo para as aplicações e <i>applets Java</i> trocam dados com o Banco de Dados [BUT 1997], [SUN 2001]. A implementação da JDBC consiste em duas partes: (1) um <i>driver</i> gerenciador que controla o carregamento dos <i>drivers</i> do Banco de Dados; e (2) um <i>driver</i> que transmite os dados entre a aplicação e o Banco de Dados.
<i>Applets</i>	Programas escritos em <i>Java</i> executados na máquina cliente.
<i>Servlets</i>	Programas escritos em <i>Java</i> , executados no servidor. São semelhantes aos CGIs, diferenciando-se pelo fato de serem <i>threads</i> dentro do servidor <i>Web</i> .
<i>Scripts (PHP, ASP e JSP)</i>	São aplicações interpretadas pelo servidor <i>Web</i> , embutidas no código HTML.

As formas de acesso a Banco de Dados via *Web* são muitas, porém deve-se analisar a melhor forma para a sua implementação. Deve-se levar em conta aspectos como desempenho do servidor, tempo de resposta da aplicação, integridade dos dados em tempo de execução cliente-servidor, entre outros. O acesso a Banco de Dados através de aplicações executadas no servidor como CGIs usando APIs e aplicações *scripts*, pode ser melhor representada pela Figura 2 [ZAV 2000].

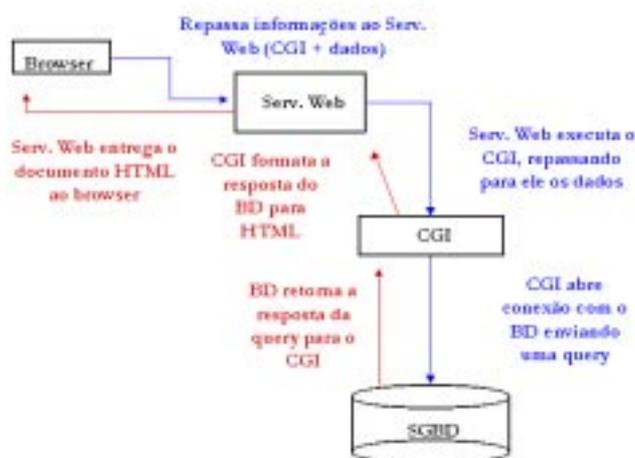


Figura 2: Interação do *browser* com o Banco de Dados, através de CGIs e *Scripts*

No caso de aplicações *Java*, como *servlets* e *scripts JSP* ainda se pode ter um protocolo de comunicação denominado *JDBC* (ver quadro). Na figura 3, é melhor representado o acesso de aplicações *Java* a Banco de Dados através de *JDBC* [LAC 2001].

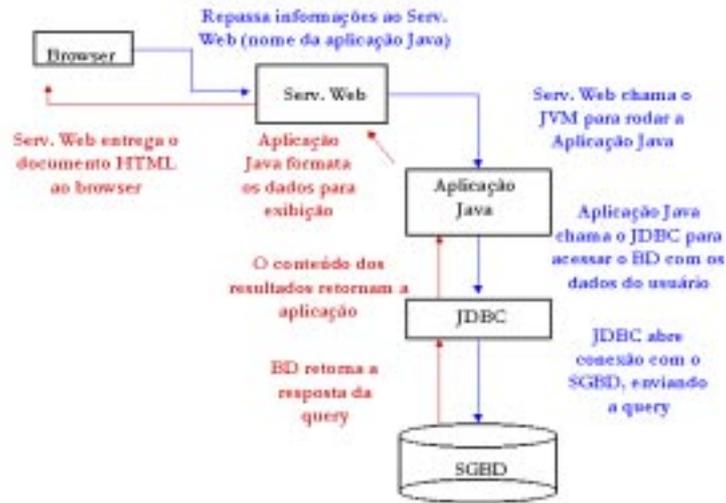


Figura 3: Interação do *browser* com o Banco de Dados, através de aplicações Java e JDBC.

2.2. Estrutura interna dos Programas

Em linhas gerais, um programa que necessite realizar uma conexão com um Banco de Dados via *Web*, deve obrigatoriamente, ter a seguinte estrutura interna, proposta na figura 4 [ZAV 2000].



Figura 4: Estrutura interna de um programa

Cabe ressaltar que esta estrutura interna independe de sua estrutura de implementação, devendo manter estas características de qualquer forma.

2.3. Vantagens da Interação

A interação de um Banco de Dados com a *Web* proporciona uma série de vantagens. Dentre estas, se pode relacionar [LIM 1997], [ROW 1998]:

- os SGBDs permitem disponibilizar na *Web* mecanismos muito eficientes de armazenamento e gerenciamento de dados, além de usarem linguagens específicas de consulta que independem da aplicação, como o SQL;
- há uma grande massa de dados residindo e sendo gerenciada por SGBDs. A *Web* veio para abrir a possibilidade de dispor estas informações a um número ilimitado de usuários;
- a *Web* proporciona uma interface gráfica relativamente amigável para aplicações em Banco de Dados multi-plataforma;

- a *Web* possibilita o desenvolvimento e expansão de novas aplicações baseadas em SGBDs. São citados como exemplos o *e-business*, *e-commerce* e *Sistemas Web*.

2.4. Desvantagens da Interação

Apesar das inúmeras vantagens proporcionadas pela Interação da *Web* com Banco de Dados [BUT 1997], pode-se relacionar alguns problemas funcionais que surgem da interação destas tecnologias [LIM 1997]:

- **Problemas Transacionais:** o protocolo HTTP usado na comunicação Cliente *Web* e Servidor *Web* é baseado na solicitação de pedido/resposta. Os atuais SGBDs foram projetados para atenderem pedidos de usuários que se conectem com ao Banco de Dados em sessões contínuas. O *PostgreSQL* versão 6.5.2, usado para a implementação do CAFE, já possui recurso de controle de transações;
- **Segurança:** questões sobre segurança no ambiente *Web* são sempre foco de discussão, principalmente quando se necessita de transmissão de informações confidenciais, que são armazenadas em SGBDs;
- **Restrições de Integridade:** as restrições de integridade presentes na maioria dos SGBDs têm o objetivo de assegurar que possíveis alterações sigam as regras e semânticas definidas para o Banco de Dados. Em um ambiente Cliente/Servidor, como a *Web*, se faz necessário para fins de desempenho, que algumas restrições sejam feitas no Cliente *Web*, minimizando o tráfego na rede. Para isso, usa-se linguagens como *Java* e *JavaScript*, que são entendidas pelo Cliente *Web*;
- **Desempenho:** o problema de desempenho pertencente ao ambiente *Web* está relacionado com o tempo de resposta a uma requisição do usuário, dependendo do desempenho de outros componentes como o Cliente *Web*, tráfego na rede e o Servidor *Web*;
- **Linguagens de Programação:** algumas linguagens têm se destacado para o desenvolvimento de aplicações *Web*, com conexões a Banco de Dados. Toda essa variedade de ferramentas pode gerar problemas de portabilidade e complexidade para as aplicações.

3. FERRAMENTAS

Para o desenvolvimento *Web*, não só as metodologias tiveram de se adaptar [BAR 2001], como também as ferramentas, além de que, muitas foram criadas voltando o seu foco diretamente para o ambiente distribuído da *Internet*. Muitas são estas, podendo ser citadas as Ferramentas mais utilizadas:

- Servidores de Rede: Windows NT/2000, Linux, Unix, FreeBSD, etc.;
- Servidores Web: Apache, IIS, Xitami, etc.;
- SGBDs: Oracle, SQL Server, SYBASE, MySQL, PostgreSQL, Informix, DB2, etc.;
- Linguagens de Programação para Desenvolvimento Web: JAVA, ASP, PHP, Perl, Python, Plataforma .NET etc; e suas respectivas ferramentas de edição (Visual Interdev, Drumbeat, JBuilder, FORTE, etc.);
- Ferramentas de Programação Visual Basic, C/C++, Delphi, Java entre outras;
- Editores de Sites/HTML: HomeSite, Dreamweaver, FrontPage, GoLive!, Composer, etc.;
- Editores de Imagem: Photoshop, ,Fireworks etc.;
- Editores de Multimídia: Flash, Director, Realplayer, etc.

Ainda existem tecnologias emergentes, tais como XML(XSL,DTD,etc) , plataforma .net, EJB, JSP, tecnologia Microsoft *Distributed Internet Applications* (DCOM, MTS, ISS2, CORBA, entre outras), que por si só serviriam de tema para o artigo, mas que no momento só serão citadas.

4. CONSIDERAÇÕES FINAIS

A integração da *Web* com Banco de Dados visando o desenvolvimento de aplicações para a Internet é um tema que está em grande evidência.

Para a implementação desta integração, faz-se necessário um amplo domínio de conhecimento em diversas áreas da computação, como Arquitetura de Sistemas Operacionais onde é importante conhecimento dos recursos, ferramentas disponíveis e limitações do Sistema Operacional a ser utilizado. Na área de Banco de Dados, é necessário avaliar características relevantes a aplicação, sendo fatores determinantes para a escolha do Banco de Dados que atendesse às exigências. Como se trata de uma aplicação que funciona via *Web*, o conhecimento sobre a área de Redes de Computadores se faz muito necessário. Outra área muito envolvida é a Engenharia de *Software*, onde é relevante a utilização de métodos e técnicas, necessárias para assegurar o que o processo de desenvolvimento do *software* esteja sendo realizado com qualidade. Devido a complexidade do desenvolvimento de aplicações para *Web*, relacionadas a utilização de várias linguagens de programação, de suma importância para o entendimento e funcionamento das principais técnicas de programação.

A união de todas estas áreas relacionadas mostra que, mais que um benefício, é uma necessidade dos sistemas de informação, procurando otimizar caminhos e resolver problemas. Mais importante do que armazenar informações de forma organizada e segura, em um banco de dados, é disponibilizá-las a quem de fato as necessita. Nesse contexto, uma aplicação utilizando a *Web* como meio de comunicação entre o usuário e o sistema passa a ser indispensável para a obtenção de resultados eficientes.

Como esta área de pesquisa é relativamente nova, pode-se relacionar inúmeros trabalhos para o aperfeiçoamento desta área como questões de segurança direcionados a aplicações *Web* com Banco de Dados, estudo de formas de modelagens de aplicações *Web* e Banco de Dados, estudos comparativos de modelos Cliente/Servidor entre aplicações *Web* e aplicações com Banco de Dados tradicional, estudos sobre performance de servidores *Web*, suportando aplicações com Banco de Dados integrados, novas ferramentas de desenvolvimento de aplicações *Web*, estudo comparativo entre Banco de Dados com recursos para desenvolvimento de aplicações *Web*, entre outros.

REFERÊNCIAS BIBLIOGRÁFICAS

- [BAR 2001] BAREINBOIM, Elias. *Desenvolvimento Web*. Disponível em <<http://www.olinix.com.br/artigos/292/1.html>>. Acesso em 27 set 2001.
- [BRU 1999] BRUSSO, Marcos. *Programação CGI*. Disponível em <<http://vitoria.upf.tche.br/~brusso/prog2/cgi.html>> Acesso em 16 nov 1999.
- [BUT 1997] BUTZEN, Fred. FORBES, Dorothy. *Linux banco de dados – como projetar e gerenciar*. Rio de Janeiro: Ciência Moderna, 1997.
- [GHE 1991] GHEZZI, Carlo. JAZAYERI, Mehdi. MANDRIOLI, Dino. *Fundamentals of software engeneering*. Englewood Cliffs: Prentice Hall, 1991.

- [LAC 2001] LACERDA, Guilherme. ZAVALIK, Claudimir. *O Uso de software livre no desenvolvimento de aplicações orientadas a objetos para Web*. II Fórum Internacional Software Livre - Anais do II Workshop sobre Software Livre. Porto Alegre: UFRGS, 2001.
- [LIM 1997] LIMA, Iremar N. *O Ambiente Web banco de dados: funcionalidades e arquiteturas de integração*. PUC-Rio: Rio de Janeiro, 2000. Dissertação de Mestrado. Disponível em:
<<http://www.cecom.ufmg.br/~iremar/dissertacao/dissert.htm>>. Acesso em 19 out 2000.
- [LIM 1998] LIMA, Iremar. LIFSSCHITZ, Sérgio. *Arquiteturas de integração Web SGBD: um estudo do ponto de vista de sistema de banco de dados*. XXV Seminário Integrado de Software e Hardware SEMISH98, Belo Horizonte, 1998. Disponível em:
<<http://www.cecom.ufmg.br/~iremar/publicacoes/semish98html/semish98.htm>>. Acesso em 19 out 2000.
- [ROW 1998] ROWE, Jeff. *Construindo servidores de banco de dados Internet com CGI*. São Paulo: Makron Books, 1998.
- [SUN 2001] SUN MICROSYSTEMS. *The source for Java™ Technology*. Agosto de 2001. Disponível por WWW em <<http://java.sun.com>>.
- [ZAV 2000] ZAVALIK, Claudimir. LACERDA, Guilherme S. *O Uso de Software Livre no acesso a Banco de Dados via Web*. I Fórum Internacional Software Livre 2000 - Anais do I Workshop de Software Livre WSL2000. Porto Alegre: UFRGS, 2000.

ROBÓTICA APLICADA À ASSESSORIA DE INSTRUMENTAÇÃO EM CIRURGIAS MÉDICAS

Flávia Luiz Caminha¹

Michele D'Mutti Garske²

RESUMO

Este trabalho tem por objetivo a aplicação prática dos princípios fundamentais de robótica, principalmente no que se refere à robótica médica. A literatura pertinente foi buscada a título de fundamentação do trabalho e, através dos conceitos obtidos, foi construído o protótipo de um robô e seu sistema de controle. O trabalho apresenta a construção e funcionamento do referido robô, bem como as características primárias do software gerenciador.

ABSTRACT

This work describes the practical application of basic principles of robotics in the design and construction of a medical robotic system. The authors consulted the available literature and based on the concepts obtained from this search, they created a robot and its control system. This article details its construction and performance and the characteristics of the controlling software.

1. INTRODUÇÃO

O homem, desde a invenção da roda, vem tentando tornar cada vez mais fácil e prático seu modo de vida. A partir deste fato, com o passar do tempo, foram surgindo novas invenções. O homem está dispondo, cada vez menos, de sua força física e ajuda de outros companheiros.

No começo, animais eram usados como tração de arados, carruagens e outros; depois foram inventados motores a vapor, e logo em seguida, motores movidos a combustível, os quais existem até os dias de hoje.

Posteriormente, apareceram os circuitos eletrônicos e assim nasceram os microcomputadores que, a cada dia, vêm se tornando algo muito comum, tanto nas empresas como nos lares.

Com o avanço tecnológico e a disseminação dessa tecnologia, o microcomputador tornou-se uma ferramenta muito útil em residências e empresas dos mais diversos portes, pois além de facilitar a execução das tarefas no dia a dia, acabou tornando-se quase que indispensável.

Em grandes empresas, hodiernamente, o uso da mecânica unida aos microcomputadores, vem tornando-se algo muito comum, pois, além de facilitar ainda mais os trabalhos que antes eram manuais, os lucros obtidos com estas maravilhas são bem maiores, devido ao trabalho contínuo e pouca mão-de-obra humana. Em residências tem-se destacado muito, pois, em qualquer documentário ou mesmo reportagens de casas futuristas, são destacadas algumas funções de como receber comandos de voz para que uma luz se acenda, ou mesmo, olhar em um *scanner* ocular para que a pessoa seja identificada, autorizando que uma porta se abra.

¹ Bacharel em Informática da URCAMP - flavia@alternet.com.br

² Aluna do Curso de Informática da URCAMP - garske@engenhocoradini.com.br

Tudo isso se torna possível quando há a união de um *software*, um *hardware* e uma mecânica se comunicando. No entanto, se depender da crescente evolução da tecnologia, em alguns anos, robôs inteligentes estarão andando livremente por aí. Para Hans Moravec [MOR 2001], fundador do laboratório de robótica da Universidade Carnegie Mellon, já em 2010 existirão autômatos que cuidarão da limpeza doméstica, como Rose, a empregada cibernética da família Jetson, do desenho animado futurista.

Alguns autores acreditam, ainda, em inovações mais ousadas. “Em 2040, acredito que finalmente conseguiremos construir uma máquina que se moverá sozinha e terá capacidade intelectual de um ser humano”, diz Moravec [MOR 2001], que não está sozinho. Ray Kurzweil [SHI 2001], pioneiro da tecnologia de reconhecimento de voz, vai mais longe e calcula que em 2029 os computadores domésticos terão a capacidade de processamento de mil cérebros.

Este trabalho se insere no contexto da pesquisa em robótica, pois apresenta os conceitos fundamentais da área, principalmente de robótica médica, e propõe um robô experimental que auxilie na instrumentação cirúrgica.

1.1. Robótica

Para [MIC 98], robótica é a ciência dos sistemas que interagem com o mundo real com pouca, ou mesmo nenhuma, intervenção humana. Pertencente ao grupo das ciências informáticas, a robótica é hoje uma área científica em expansão e altamente multidisciplinar. Para engendrar os vários dispositivos robóticos e seus movimentos, a robótica exige dosagens de conhecimento de microeletrônica, também de matemática, de inteligência artificial e de outras disciplinas informáticas que provêm recursos para expressar e viabilizar a inteligência dos robôs, enquanto a engenharia industrial e a economia interagem com a robótica no que tange a processos industriais e a impactos sociais.

1.2. Robô

A palavra robô vem da palavra tcheca *robota*, que significa trabalho árduo ou trabalho escravo. Ela foi usada pela primeira vez em uma peça de ficção do autor Karel Capek, intitulada “Os robôs Universais de Rossum” em 1920.

De acordo com [MIC 98], robô é a máquina capaz de perceber e reagir à entrada, e produzir alterações em seu ambiente com algum grau de inteligência, se possível sem a intervenção humana. Raramente os robôs têm uma aparência humanóide, embora muitas vezes sejam projetados para imitar movimentos humanos na realização de seu trabalho.

De uma maneira mais genérica, pode-se definir um robô como uma máquina reprogramável que é capaz de imitar as ações ou a aparência de uma criatura inteligente, geralmente um ser humano.

Para se caracterizar como um robô, a máquina deve ser capaz de, no mínimo, duas coisas [HEI 2001]:

- obter informações sobre seu próprio ambiente;
- atuar de alguma forma física neste ambiente, movendo-se ou manipulando objetos.

2. RHYS

A visão de construir um braço mecânico sempre foi um desafio, pois é necessário ter um conhecimento em várias áreas, não somente em informática. De acordo com [LEM 2000], o mundo entrou no ano 2000 com um milhão de autômatos, sendo que quase 99% destes são braços mecânicos usados na indústria automobilística.

Após definir o propósito do trabalho, foram feitas entrevistas com médicos da cidade de Dom Pedrito para detectar a aceitação de um braço mecânico a ser utilizado em procedimentos cirúrgicos como auxiliar de instrumentador. A seguir, foram acompanhadas algumas intervenções cirúrgicas para verificar quais são as tarefas desempenhadas pelos membros da equipe cirúrgica, principalmente as funções exatas do instrumentador.

Atualmente, as intervenções cirúrgicas analisadas são realizadas por um grande número de pessoas: um cirurgião, um auxiliar, dois enfermeiros, um instrumentador e um anestesista. Portanto, a possibilidade de erro ou de contaminação adquire grandes proporções. Com o uso da robótica, pode-se diminuir relativamente esta possibilidade.

Com base no estudo dos capítulos anteriores e no acompanhamento realizado, foi desenvolvido um protótipo de um braço mecânico, sendo formado por vínculos em alumínio, motores, manipulador, placas controladoras e software gerenciador.

2.1. Vínculos

Segundo [FER 86], vínculo, em um sistema mecânico, é a relação entre suas coordenadas generalizadas, que traduz a existência de uma restrição material que limita o movimento de um sistema.

Para a construção dos vínculos, foi utilizado o alumínio, por ser um material leve e de fácil manuseio. Suas medidas foram dimensionadas através de várias pesquisas e acompanhamento de procedimentos cirúrgicos, para que as mesmas atendessem às necessidades do protótipo, conforme apresentado nas Figuras 1 e 2.

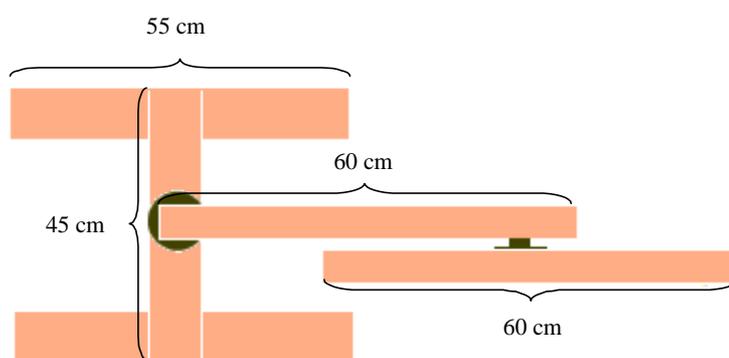


Figura 5 - Vista superior dos vínculos

Os vínculos podem ser melhor visualizados na Figura 1, onde é apresentada uma vista superior do protótipo e, onde podem ser verificadas as medidas de largura e comprimento de cada um deles. Já na figura 2, os vínculos podem ser visualizados lateralmente, podendo-se identificar suas medidas em relação à altura.

2.2. Motores

Neste projeto, estão sendo utilizados três motores do tipo servo, que apresentam um certo grau de inteligência. Os motores *alternating current* (AC) ou *direct current* (DC) são os mais comuns e têm o seu funcionamento mais conhecido. Quando é aplicada a alimentação, este tipo de motor começa a rotacionar em determinada direção até chegar à velocidade de operação, transferindo ao eixo um torque baseado no tamanho do motor e na alimentação fornecida.

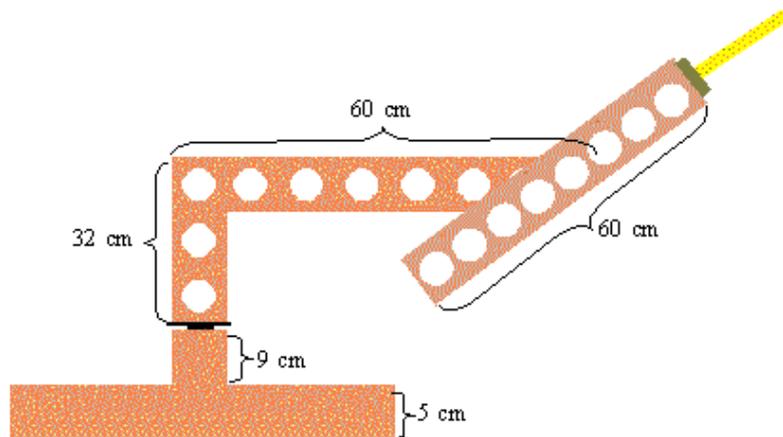


Figura 6 - Vista lateral dos vínculos

Os servomotores utilizam um pequeno motor DC, que é controlado por um circuito eletrônico e um dispositivo de realimentação constituído por um potenciômetro conectado mecanicamente ao eixo do motor DC e eletricamente ao circuito eletrônico. A variação da resistência do potenciômetro em função da rotação informa ao circuito eletrônico. A variação da resistência do potenciômetro em função da rotação informa ao circuito eletrônico a rotação e direção do motor.

Ao movimentar-se em determinada direção, o motor DC move uma série de engrenagens que amplificam e transferem o torque do motor ao eixo externo onde são os controles mecânicos. Se o motor fosse conectado diretamente ao eixo de controle, o tamanho físico do motor teria que ser 10 vezes maior que o apresentado pelos servomotores utilizados. Também não seria possível obter baixas rotações, como se pode ver na figura 3.



Figura 7 - Engrenagens internas do servomotor

A Tabela 1 apresenta as especificações técnicas de um servomotor.

Tabela 3 - Especificações do servomotor FUTABA. Adaptado de [CES 2001]

Modelo	Dimensões	Peso	Torque (oz./in.)	Velocidade Seg./60°
S3003 Standard	0.78 x 1.59 x 1.42 in.	1.31 oz.	44.4	0.23

Levando-se em conta a maneira em que estão dispostos os motores no braço mecânico, pode-se dizer que o mesmo possui dois graus de liberdade. Para união dos vínculos, foram utilizadas juntas de rotação, deixando-o com uma articulação vertical; para o acionamento dos motores, utilizam-se *drivers* elétricos com conexão indireta.

2.3. Manipulador

Para o manipulador, foi adaptada uma garra em acrílico de dois dedos com movimento paralelo, sendo a mesma acionada através de dois ganchos e os mesmos ligados ao servomotor através de um fio de aço. Este movimento é obtido através das articulações, que se dá pelo encaixe das partes em acrílico uma sobre a outra deixando uma discreta folga entre elas para que ocorra o procedimento de abertura e fechamento, conforme apresentado na Figura 4.



Figura 8 - Manipulador do RHYS

2.4. Placa Controladora

Adquiriu-se uma placa SERVO, que controla, no máximo, oito servomotores do tipo FUTABA ou compatíveis através do computador, utilizando para a comunicação de dados a porta serial.

Na Tabela 2 são apresentadas algumas de suas características, como alimentação de energia necessária, consumo de energia, sua velocidade e a pinagem compatível.

Tabela 4 - Especificação da Placa Servo. Adaptado de [IOR 2001]

Alimentação	Consumo	Velocidade	Pinagem Compatível
12 volts	$\cong 550$ mA	2400 bps	Servomotor FUTABA

2.5. Software Controlador

Utilizou-se Delphi e Paradox para desenvolvimento de um software para acompanhamento de procedimentos cirúrgicos, com funções para cadastro de pacientes, médicos e cirurgias, de onde será possível extrair informações importantes sobre ocorrência de enfermidades e controle de acesso ao bloco.

Neste sistema, ao agendar uma cirurgia, é possível habilitar o uso do protótipo. Quando este está habilitado, interage da seguinte maneira: recebe um comando e processa uma função que passa para a placa as diretrizes para a movimentação correta do braço mecânico.

3. CONCLUSÃO

O presente trabalho apresentou a implementação de um protótipo de um braço mecânico e seu software de controle, proporcionando a oportunidade de aplicação e do aprimoramento dos conhecimentos específicos sobre a área de robótica, a qual se apresenta como um nicho de mercado significativo para os profissionais que nela estão investindo.

Com a construção do RHYS, foi possível demonstrar que a robótica médica é uma área de pesquisa, dentro da Inteligência Artificial, que permite a implementação de uma infinidade de projetos, os quais poderão contribuir para estimular, ainda mais, o avanço científico e tecnológico, capaz de proporcionar melhor qualidade de vida às pessoas.

BIBLIOGRAFIA

- [CES 2001] CESARIN, Flademir. **Servos**. 2001. Disponível em: <<http://www.flademir.hpg.com.br/>>. Acesso em: 24 ago. 2001.
- [FER 86] FERREIRA, Aurélio Buarque de Holanda. **Novo dicionário aurélio da língua portuguesa**. Rio de Janeiro: Nova Fronteira, 1986.
- [FER 97] FERRARI, D. G. & ETO, R. M. **Mundo da robótica**. 1997. Disponível em: <<http://www.din.uem.br/~ia/robotica/index.htm>>. Acesso em: 05 out. 2001
- [HEI 2001] HEINEN, Farlei J. **O que é um robô ?** 2001. Disponível em: <<http://ncg.unisinos.br/robotica/>>. Acesso em: 18 mai. 2001.
- [HEU 99] HEUSER, Carlos Alberto. **Projeto de banco de dados**. Porto Alegre: Sagra Luzzatto, 1999.
- [IOR 2001] I/O ROBOTICS. **Instalação e uso da placa SERVO**. 2001. Disponível em: <<http://www.irobotics.com.br/servo.htm>>. Acesso em: 10 mai. 2001
- [LEM 2000] LEMOS, José Augusto; MELLO, Mariana; RENTSCHLER, Mônica; WUESTHOF, Roberto; JOBIN, Nelson. **Seu novo amigo, o robô**. Revista Super Interessante. Editora Abril: Edição Junho/ 2000.
- [MAR 93] MARTINS, A. **O que é robótica**. São Paulo: Brasiliense, 1993.
- [MIC 98] MICROSOFT PRESS. **Dicionário de informática**. Trad. 3. ed. americana. Gilberto Castro e Valéria Chamon (trad.). Rio de Janeiro: Campus, 1998.
- [MOR 2001] MORAVEC, Hans. **Hans Moravec Home Page**. 2001. Disponível em: <<http://www.frc.ri.cmu.edu/users/hpm/>>. Acesso em: 01 out. 2001.
- [SAB 2001] SABATINI, Renato M. E. **Robôs cirúrgicos**. 2001. Disponível em: <<http://www.epub.org.br/correio/cp000922.html>>. Acesso em: 10 ago. 2001.

ESTUDO DA MORTALIDADE INFANTIL NO SUL DO RS COM O USO DO GEOPROCESSAMENTO

Bibiana Santos Pedroso¹

Lúcia Azambuja Saraiva Vieira²

Maria Elaine dos Santos Leon³

Michele D'Mutti Garske⁴

Moisés Gularte⁵

RESUMO

Este artigo tem o objetivo socializar alguns conhecimentos sobre a análise da distribuição espacial da mortalidade infantil no município de Dom Pedrito, relacionando-a com os serviços de saúde existentes, bem como a organização urbana e social encontrada. Foi utilizado um Sistema de Informação Georreferenciada - SIG, através da disponibilidade do mapa cartográfico digitalizado de Dom Pedrito, com os bancos de dados da 7ª Coordenadoria Regional de Saúde, dados estes secundários (já existentes) da mortalidade infantil do município, ocorridos na zona urbana, nos anos de 1998, 1999 e 2000. A possibilidade da análise espacial desse indicador pela ferramenta do geoprocessamento é de extrema importância pois serve de apoio para a tomada de decisões, reforçando a idéia de que a mortalidade infantil está ligada às questões sanitárias e sociais relacionadas com as condições de vida das pessoas, pois a distribuição geográfica das ocorrências se situa basicamente na periferia da cidade, onde as condições são precárias em relação a infra-estrutura urbana, onde os serviços oferecidos são mais escassos e assim torna as pessoas mais vulneráveis.

Palavras-chave: Mortalidade infantil, Sistema de Informação Georreferenciada (SIG), serviços de saúde, condições de vida

ABSTRACT

This paper describes the spatial distribution of the incidence of infant mortality in Dom Pedrito city, relating this mortality to existing health services and social organization. A Geographic Information System (GIS) was used with a digital city map and annual database information from the 7th Regional Health Coordination (secondary data) from 1998 to 2000. This spatial analysis with a geoprocessing tool is important because it aids decision making. The incidence of premature death in children is linked to social and living conditions with increased vulnerability for those people living far from the city center who have less access to health care.

Keywords: *Children Death, Geo-referenced Information Systems (GIS), Health Services, Living Condition.*

1. INTRODUÇÃO

Este trabalho é fruto do Projeto sobre o Diagnóstico de Saúde com o Uso do Geoprocessamento em Dom Pedrito, no sul do RS, desenvolvido por um grupo de acadêmicos e professores dos cursos de Informática, Arquitetura e Enfermagem da

¹ Acadêmica de Enfermagem CCS - URCAMP- Bagé, RS. E-mail: bibipedroso@bol.com.br

² Professora do CCS – URCAMP e Coordenadora Adjunta da 7ª Coordenadoria Regional de Saúde Bagé, RS. E-mail: luvieira@matrix.com.br

³ Acadêmica de Informática CCEI - URCAMP- Bagé, RS. E-mail: meleon@provesul.com.br

⁴ Acadêmica de Informática CCEI-URCAMP- Bagé, RS. E-mail: garske@coradini.com.br

⁵ Acadêmico de Arquitetura e Urbanismo CCEX - URCAMP- Bagé, RS. E-mail: arq.gulart@bol.com.br

Universidade da Região da Campanha - URCAMP/Bagé-RS em parceria com a 7ª Coordenadoria Regional de Saúde e Prefeitura Municipal de Dom Pedrito.

A interação, epidemiologia e geografia é bem antiga. Há mais de um século, epidemiologistas e outros cientistas da medicina, começaram a explorar o potencial das informações veiculadas pelos mapas e buscar entendimento da ocorrência espacial das doenças. Um dos mais antigos e importante estudo, foi de John Snow, no século XIX, que mapeou os casos de cólera com as bombas d'água em Londres, durante a epidemia nesse país, mostrando o papel da contaminação da água no aparecimento da patologia.

O uso de geoprocessamento na saúde tem facilitado pelo o amplo acesso a base de dados epidemiológicos. Os SIGs permitem a interação e combinação de dados cartográficos oriundo de diferentes fontes.

E a análise das medições dos processos sobre a mortalidade infantil tem despontado como tema de interesse crescente em saúde coletiva, com esse trabalho pretende-se aliar essas duas ferramentas, o uso do geoprocessamento com o indicador da Mortalidade Infantil.

2. A MORTALIDADE INFANTIL

O Fundo das Nações Unidas para a Infância – UNICEF, também aponta a mortalidade infantil, como um ótimo indicador das condições de vida e de saúde de uma população. Portanto, o mesmo não tem sido apenas utilizado como um indicador clássico de saúde, mas também como um dos índices de desenvolvimento social.

Oliveira e Mendes (1995:57), escrevem que, *“de uma forma bem mais intensa do que a mortalidade adulta, os óbitos infantis estão sujeitos a vários condicionantes que atuam a partir da vida social. O organismo infantil, por ser um complexo psicobiológico em formação, tem a capacidade de defesa das agressões externas naturalmente reduzidas, com frequência expõe a criança a um conjunto de doenças e complicações que potencializam o risco de morte.”*

Na perspectiva de ser um indicador social, tem sido estudada com uma certa evidência, pois sua persistente queda nos últimos anos, também nos países em crise econômica ou sob condições recessivas, traz a sua dinamicidade e relação com as políticas públicas apontando o papel do Estado como de fundamental importância no comportamento desse indicador (Duarte, 1992).

É um coeficiente que tem sido utilizado internacionalmente como um dos principais indicadores de qualidade de vida de uma população, podendo ser determinado por múltiplas variáveis, tais como os fatores sócio-econômicos, políticos, ambientais e relativos à assistência à saúde.

No Brasil, historicamente, os níveis de mortalidade apresentam diminuição de seus indicadores. Pode-se dizer que a redefinição do perfil do setor saúde, a partir da década de 70, com a expansão ambulatorial, com a descentralização e a ampliação da oferta de serviços básicos, programas para grupos de risco, especificamente, voltados à saúde materno-infantil com o apoio de UNICEF, contribuíram para esta queda.

Esse índice passou de 117% em 1960, para 40% em 1994 e 36% em 1998. Apesar desse decréscimo em todas as regiões, o ritmo de tal descenso foi muito diferente entre elas, aprofundando ainda mais as desigualdades dentro do país. As regiões mais desenvolvidas, situam-se três a quatro vezes abaixo das mais carentes.

Vê-se que, apesar da desigualdade social e da crise prolongada, a redução da MI seguiu uma trajetória bem definida da década de 80 e início dos anos 90 pela ação dos serviços médicos-sanitários, sobre tudo no campo preventivo (Oliveira e Mendes, 1995)

No Rio Grande do Sul nas últimas décadas, percebe-se uma redução do n.º de óbitos menores de um ano, e desde 1992 mantém a posição de menor índice do país.

E como no Brasil, o RS também apresenta diferenças marcantes nas suas diversas regiões. Diferenças não só sanitárias, mas econômicas e sociais importantes.

Dom Pedrito é um município de 40000 hab. (IBGE, 2000), localizado na região da Campanha do RS, pertence a 7ª Coordenadoria Regional de Saúde - CRS, a principal atividade econômica é a agropecuária, numa estrutura fundiária com o predomínio de propriedades com mais de 500 ha (Gov. RS, 1999).

O indicador MI, nesse município, tem um comportamento irregular, nos últimos anos, principalmente a partir de 1997, como o do RS em declínio, mas mantendo-se superior a da região e do Estado como um todo, como mostra o gráfico (Figura1) a seguir.

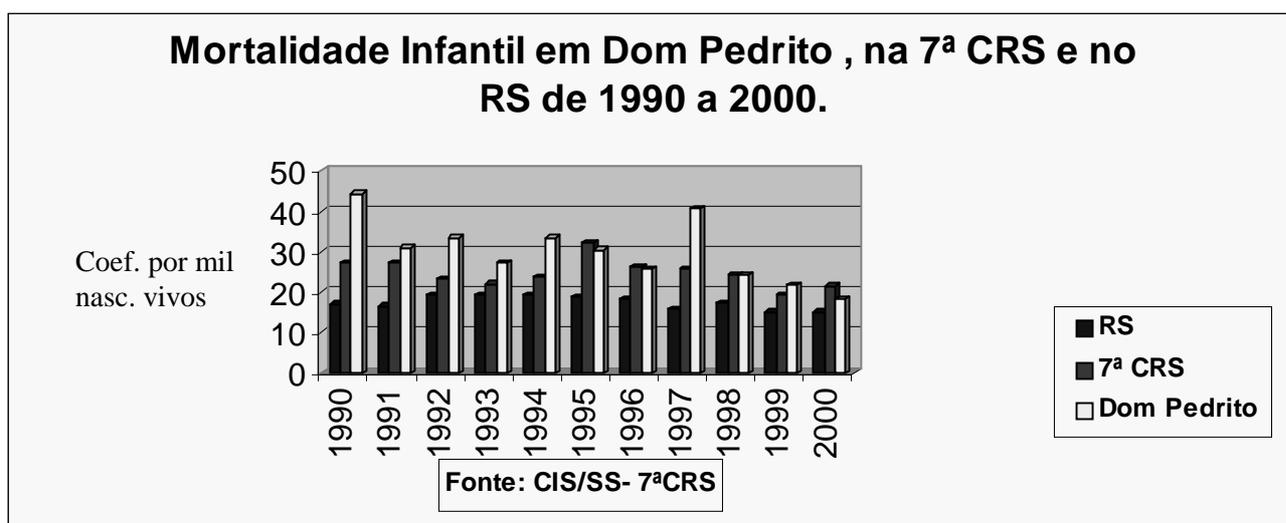


Figura 1 - Gráfico da Mortalidade Infantil em Dom Pedrito

3. O GEOPROCESSAMENTO

O geoprocessamento é um termo muito amplo que engloba diversas tecnologias de tratamento e manipulação de dados geográficos através de programas computacionais (ver Figura2).

É um conjunto de procedimentos analíticos que opera sobre uma base de dados georreferenciada, e os Sistemas de Informação Geográfica, os SIGs, são estruturas de processamento eletrônico de dados que permitem a captura, armazenamento, manipulação, análise, demonstração e relato dos dados referenciados geograficamente, eles são a base física e lógica onde opera o geoprocessamento (Medronho, 1995).

Os SIGs surgiram na saúde pública para melhorar as possibilidades da descrição e análise espacial das doenças em grandes conjuntos georreferenciados, estes sistemas são úteis na conexão da avaliação das relações de algum agravo com variáveis ambientais. A utilização de técnicas de geoprocessamento aplicadas à área da saúde vem se disseminando rapidamente em todo o país, em diversos e variados contextos.

São sistemas computacionais usados para o entendimento dos fatos e fenômenos que ocorrem no espaço geográfico.

A importância da utilização dos SIGs na epidemiologia está no sentido de contribuir para o planejamento das ações de saúde, na prevenção e controle de doenças e agravos relacionados com saúde. Levando-se em conta o papel importante das informações nas tomadas de decisões.

Outra característica de se trabalhar com um sistema de informação georreferenciada é a articulação intersetorial, pois não será tarefa só de um setor, nos municípios onde serão implantados, e sim de uma ação conjunta de vários setores que servirá para um geral, que é a população, seja no que diz respeito a arrecadação municipal, educação, abastecimento de água, luz, telefonia ou saúde (REPSA, 2000).

O propósito de sua utilização é proporcionar o desenvolvimento de análises onde se consideram as implicações de localização geográfica na determinação da situação de saúde. Os objetivos, dessa técnica, são, entre outros a análise ambiental, estudos ecológicos, delimitação de áreas homogêneas segundo o gradiente de risco, etc. Uma das mais importantes, diz respeito à relação com o ambiente e saúde (GeoSaúde, 2000).

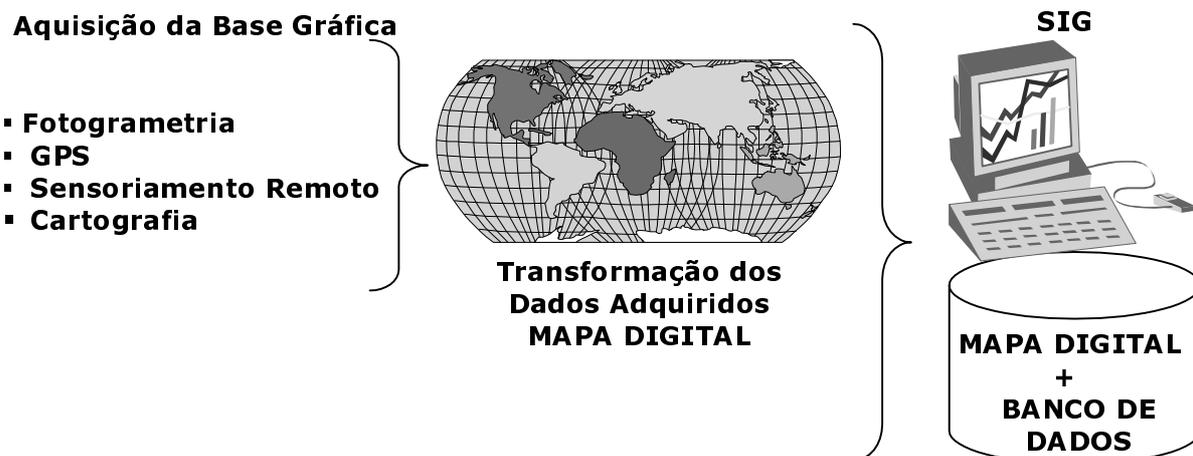


Figura 2 - A aquisição e Transformação de um SIG

3.1. Benefícios de um SIG:

- Melhor armazenamento e atualização dos dados;
- Recuperação de informações mais precisas;
- Rapidez na análise de alternativas;
- A vantagem de decisões mais acertadas.

3.2. Projeto de um SIG (Sistema de informações georreferenciadas)

A saúde pública e ambiente estão intrinsecamente influenciadas pelos padrões de ocupação do espaço: não basta descrever as características das populações, mas é necessário localizar o mais precisamente onde estão acontecendo os agravos, que serviços a população está procurando, o local de potencial risco ambiental (Figura 3). Onde o planejamento, monitoração e avaliação de programas e o estudo do contexto socioeconômico, vigilância em saúde, são ações essenciais à reorientação das atitudes do setor de saúde são beneficiadas por uma visão incorporando a distribuição espacial (REPSA,2000).

Daí a surge a necessidade de se conceber um projeto em geoprocessamento conciso e livre de redundâncias que se utilize de todas as geotecnologias disponíveis nas ferramentas de SIG, disponíveis atualmente no mercado. Tendo em vista a possibilidade de adaptar a ferramenta bem como o mapa digital e a base de dados às reais necessidades do projeto. Para que o projeto seja realmente bem planejado, o mesmo deve obedecer as seguintes fases de desenvolvimento:

- especificação do problema - deve-se definir os motivos do desenvolvimento e que tipo

de informações precisam ser geradas através do sistema para que seja possível solucionar o problema;

- definição da base de dados - é importante listar o tipo de dado que é necessário para atender aos objetivos expostos e a forma de obtenção dos mesmos;
- especificação do sistema - é necessário definir qual o equipamento e quais programas serão necessários para alcançar os objetivos;
- pré-processamento dos dados - todos os dados obtidos de diversas fontes, em escalas diferentes, sistemas de projeção diferentes, necessitam igualmente de um pré-processamento de maneira que se tornem compatíveis;
- análise dos dados - após a limpeza e compatibilização dos dados é possível realizar as análises, gerando informações úteis;
- gerenciamento dos dados - a organização a atualização das bases já existentes e a incorporação de novas tecnologias. A medida que os usuários vão obtendo informações, novas indagações são elaboradas, sendo natural a necessidade de aprofundar as análises e ampliar o acesso a novas informações;
- Saídas Gráficas - as informações obtidas podem ser divulgadas de diversas formas, como: mapas, gráficos, tabelas, relatórios, etc.;
- Avaliação dos Erros - é importante verificar os resultados para ter certeza de que fazem sentido. Um computador executa tarefas, mas a análise crítica dos resultados, avaliando a coerência e a qualidade das informações, depende da equipe técnica;
- Fase Operacional - o sistema atingiu esta fase quando os usuários finais estão fazendo uso efetivo do Sistema. A conversão de dados pode ainda não estar completa, mas já existem rotinas padronizadas de trabalho. É importante, também, determinar procedimentos que garantam a atualização do Sistema, satisfazendo a necessidade de informações;
- Tomada de Decisões - quando as fases anteriores forem cumpridas, a informação gerada torna-se um importante instrumento de auxílio à tomada de decisão.

4. METODOLOGIA

Foram utilizados neste projeto dados secundários (já existentes) sobre a MI nos anos de 1998, 1999 e 2000 no município de Dom Pedrito/RS, bem como os dados de infraestrutura urbana, baseando-se no mapa digitalizado existente do ano de 1995 e de domínio do município, como mostra a Figura 3.

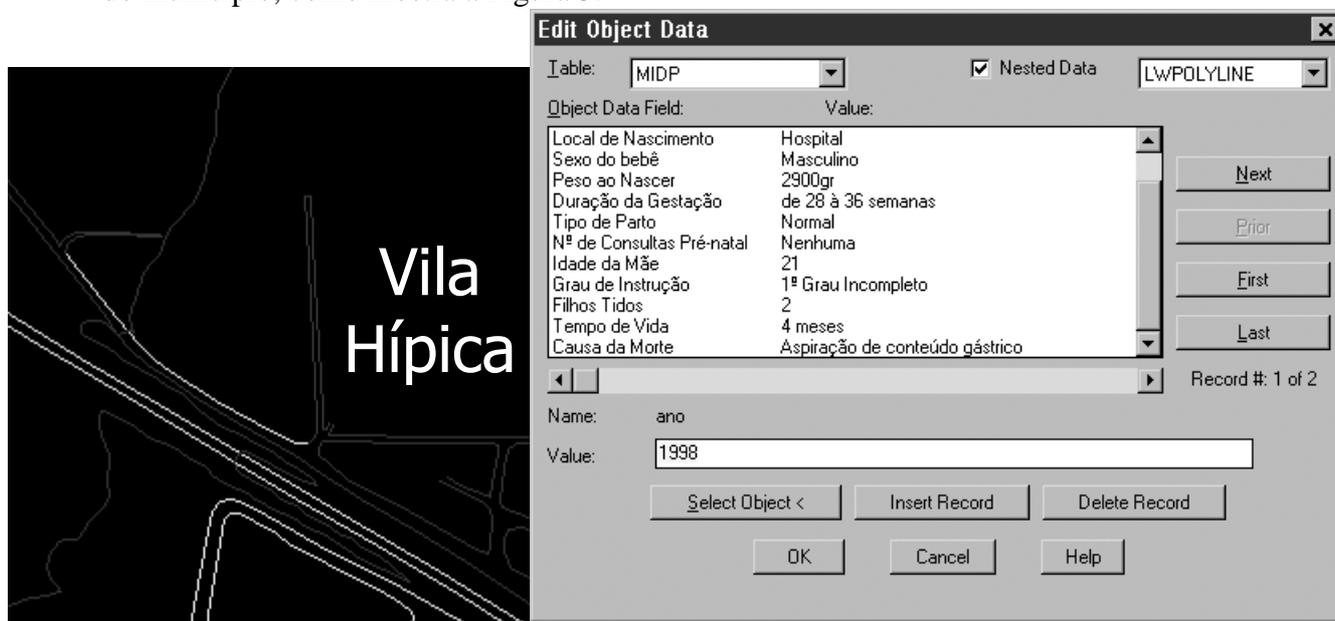


Figura 3 - A Interação de um SIG com uma Base Gráfica vinculada a uma Base de Dados

Como objetivo principal desta etapa do trabalho, relaciona-se o coeficiente MI e sua distribuição espacial no município, com ênfase na zona urbana, identificando assim áreas de risco.

5. CONSIDERAÇÕES FINAIS

Conhecer as condições de vida e de saúde dos diversos grupos populacionais é uma etapa indispensável do processo de planejamento da oferta de serviços e avaliação do impacto das ações de saúde. Além disso, o enfoque epidemiológico atende ao compromisso da integralidade da atenção, ao incorporar, como objeto das ações, a pessoa, o meio ambiente e os comportamentos interpessoais (MS, 1997).

Sabendo que a MI foi conceituada por VÍCTORA (1988:78) como: “*a proporção de crianças que morrem antes de completar um ano de vida, é tida como um ótimo indicador das condições de vida e de saúde de uma população*”.

Fica, então, aqui fortalecido o convencimento de que para resolver-se estes dilemas é preciso um projeto político que combata a pobreza e as diferenças sociais e que através da utilização das tecnologias disponíveis se possa chegar mais rapidamente ao foco de incidência, podendo então dispendir um maior espaço de tempo para a realização das ações de melhoria, que se fazem necessárias, com relação à saúde nesta região.

6. REFERÊNCIAS BIBLIOGRÁFICAS

ANDRADE, J. Bittencourt. **Fotogrametria**. SBEE, 2001. 246p.

CARVALHO, Marília Sá, PINA Maria de Fátima de e SANTOS, Simone Maria dos. **Conceitos de sistemas de informação geográfica e cartografia aplicados à saúde**. Brasília/DF- Ministério da Saúde, 2000. 124 p.

DUARTE, Cristina Maria R. Qualidade de Vida e Indicadores de Saúde: aspectos da mortalidade infantil no Estado do Rio de Janeiro e suas regiões. **Cadernos de saúde pública** / Rio de Janeiro: FIOCRUZ-ENSP, v.8,n414-427,out./dez. 1992.

GeoSaúde: **Geoprocessamento & saúde – levantamentos de projetos e demandas**[s.d.], [capturado 2000, abril] Disponível em <http://www.proc.fiocruz.Br/~geo>

GOVERNO DO ESTADO DO RIO GRANDE DO SUL. Secretaria de Coordenação e Planejamento. **Seminários de desenvolvimento regional**. Porto Alegre, 6 de nov. a 19 de dez de 1999.

FITZ, Paulo Roberto. **Cartografia básica**. LaSalle Centro Universitário, 2001.171p.

HEUSER, Carlos Alberto. **Projeto de banco de dados**. Instituto de Informática da UFRGS, Editora Sagra, 2001. 246p.

LAMPARELLI, Rubens A.C., ROCHA Jansle Vieira e BORGHI, Elaine. **Geoprocessamento e agricultura de precisão - fundamentos e aplicações**. Guaíba/RS - Editora Agropecuária, 2001. 118p.

LIMA, Cláudia Campos N. Alves. **Trabalhando com AUTOCAD LT**. Editora Érica, 1999. 228p.

OLIVEIRA, Luiz Antônio e MENDES, Márcia Martins. Mortalidade Infantil no Brasil: tendências recentes In: MINAYO, Maria Cecília de Souza (org.). **Os muitos brasis – saúde e população na década de 80**. São Paulo/Rio de Janeiro: HUCITEC-ABRASCO, p.291-303, 1995.

VÍCTORA, César, BARROS, Fernando e VAUGHAN, Patrick. **Epidemiologia da desigualdade**. São Paulo: HUCITEC, 1988. 187 p.

VIEIRA, Lúcia Azambuja Saraiva Vieira. **Mortalidade infantil: desigualdades sociais e em saúde. Estudo da 7ª Coordenadoria Regional de Saúde/RS**. Florianópolis/SC-PEN-UFSC set. 2000.121p. (Dissertação de Mestrado em Enfermagem)

ROCHA, José Antônio M.R. **GPS - Uma abordagem prática**. Editora CATAU, 2001. 151p.

SANTOS, Simone M. e NORONHA, Cláudio P. Padrões espaciais de mortalidade e diferenciais sócio-econômicos na cidade do Rio de Janeiro. **Cadernos de saúde pública** / Rio de Janeiro: FIOCRUZ-ENSP, v.17, n5: 1099-1121, set/out. 2001.

A INSALUBRIDADE DEVE SER CALCULADA SOBRE A REMUNERAÇÃO E NÃO MAIS SOBRE O MÍNIMO

Benjamin Vicenzi¹

RESUMO

Hoje, o adicional de insalubridade deve ser calculado sobre a remuneração ou salário contratual do empregado e não mais sobre o salário mínimo como, historicamente, aconteceu. Isso se deve, fundamentalmente, em decorrência dos princípios insculpidos nos incisos IV, XXII e XXIII, DO ART. 7º da Constituição Federal, somados às recentes decisões de nossa Suprema Corte, que interpretou ditos incisos em consonância com o desejo do Constituinte de 1987/1988, sistematicamente, no conjunto das normas ápices.

ABSTRACT

This work proposes the fundamental position that additional hazard compensation should be calculated on the remuneration or the contractual salary of the position and not on the minimum wage as has been the case historically. This should be fundamentally as a consequence of the principles enshrined in clauses IV, XXII and XXIII of the 7th Article of the Federal Constitution added by the recent decision of the Supreme Court, that interpreted these clauses in consonance with the intent of the Constitution of 1987/1988 systematically in the group of superior norms.

1. INTRODUÇÃO

Insalubridade, que em termos laborais significa o ambiente de trabalho hostil à saúde, pela presença de agentes agressivos ao organismo do trabalhador, acima do limite de tolerância permitidos pelas normas técnicas e suportável pelo organismo da pessoa, em condições normais, foi instituída, em nosso sistema jurídico, há muito tempo (década de trinta) quando, ainda, os direitos trabalhistas eram quase inexistentes. Naquela época havia a preocupação com as condições de trabalho, isto porque, o governo que assumira comprometeu-se com o aspecto social do trabalhador. Isso ficou patente nos temas acidentário², previdenciário³ e, mais tarde, com o laboral⁴, os quais exigiam locais de trabalho salubres que, apesar de todo o aparato legal, lamentavelmente, o que sempre se viu foi o descaso para com a prevenção onde, as empresas, historicamente preferiram monetarizar a saúde do obreiro a tomar atitude de conservar o ambiente de trabalho de acordo com as normas de segurança e saúde ocupacional, isto porque, o valor que se pagava, a título de insalubridade, sempre foi irrisório, vil.

2. INTENÇÃO DO LEGISLADOR CONSTITUINTE/88

Os ANAIS, Emendas e demais documentos da Assembléia Nacional Constituinte/88, são provas da intenção do legislador constituinte de banir, de vez, os casos de insalubridade, até mesmo para atender aos princípios e normas ditadas pelos organismos internacionais, especialmente as Convenções da Organização Internacional do Trabalho (OIT), referentes ao tema⁵ e da Organização Mundial da Saúde (OMS), sob o ponto de vista da saúde⁶,

¹ Advogado, Professor de Direito do Trabalho da URCAMP e Doutorando em Direito do Trabalho pela Universidade de León/Espanha). E-mail: bvicenzi@alternet.com.br

² Decreto-lei n. 3.700, de 09/10/1941 e seu Decreto Regulamentador n. 10.569, de 05/10/1942.

³ Decreto-lei 7.036, de 10/11/1944, Capítulo XII.

⁴ Refletido no vetusto art.187, da CLT, na versão original de 1943.

⁵ Convenções OIT ns. 148, 155 e 161. Todas aprovadas e promulgadas pelo Brasil.

ratificando-as⁷, procurando oferecer, desde o plano constitucional, mesmo que de forma detalhista, para que as normas lá insculpidas tivessem eficácia, a fim de que o Brasil pudesse justificar, inclusive, sua condição de membro fundador e efetivo da OIT⁸ e tudo fazer para que os ambientes ou condições de trabalho fossem compatíveis com os ditames da saúde e dignidade humana, na condição de elementos fundamentais⁹, que representam.

O constituinte/88, coerente com a evolução do conceito humanitário de saúde e bem-estar do trabalhador no ambiente de trabalho, já que saúde “não é um ponto fixo que se atinge, mas a direção em que se empreende a caminhada, uma constante melhoria que deve ser perseguida, sempre”¹⁰, desejou continuar obrigando a empresa a *eliminar* a insalubridade, muito embora a redação final da norma (CF, art. 7º, XXII) pareça não conter a cogência da *eliminação* e apenas a flexível opção da *redução* dos riscos. Mas, nenhuma dúvida paira quando forem compulsadas as várias Emendas do projeto constituinte, onde se verifica que foi isso mesmo que desejou o legislador maior, pautando pela preocupação em atender aos princípios da OIT, externados principalmente nas Convenções n. 155 e 161, ratificadas pelo Brasil, as quais ordena que se empregue todos os recursos e tecnologia necessários para eliminar os riscos ambientais, como acontece, aliás, nos países adiantados¹¹, que seguem as normas e princípios dos organismos internacionais pertinentes.

É mister que se registre, que a obrigação de *eliminar* o risco veio desde o início do processo constituinte, atravessando todas as fases até chegar à redação final da Carta Magna. Reconhece, aquele constituinte, todavia, a despeito de sua vocação prevencionista e obstinação à elisão do agente agressivo, que existem casos que não é possível fazê-lo, seja pela natureza da atividade ou pela condição ambiental. Por isso optou pelo termo *redução* (XXII) mas não com sentido de permitir que a empresa desleixasse o cumprimento das normas prevencionistas, até porque “as leis fundamentais devem ser mais rigorosamente obrigatórias do que as ordinárias, visto pertencerem, em geral, à classe das imperativas e de ordem pública”¹².

Percebe-se claramente a cogência da norma (7º, XXII) quando *ordena* a redução dos riscos ambientais e a *sanção econômica* (7º, XXIII) esta, como elemento secundário e condicional *enquanto não eliminar ou reduzir* os agentes agressivos, pois nenhuma dúvida surge quanto à intenção do constituinte em preferir que a matéria fosse elevada ao patamar constitucional, para que a empresa mais facilmente cumprisse com a norma prevencionista, como bem se depreende da redação do Parecer exarado pelo Relator, à Emenda n. 07647/87, apresentada na Comissão de Sistematização, quando assevera que:

“O imperativo de que sejam reduzidos os riscos à saúde e segurança do trabalhador, parece-nos um dos pontos fundamentais dos direitos que lhe devem ser assegurados pela Constituição. Não

⁶ Onde, para a OMS ‘saúde’ é “o completo bem-estar físico, mental e social, e não somente a ausência de afecções ou enfermidades”. Já para a OIT (Convenção n. 155, art. 3, alínea ‘e’ define saúde como sendo a que “...abrange não só a ausência de afecções ou de doenças, mas também os elementos físicos e mentais que afetam a saúde e estão diretamente relacionados com a segurança e higiene do trabalho”

⁷ Pelas regras da OIT, o país membro quando ratifica suas Convenções, o conteúdo destas passa a equiparar-se às leis internas do país. No Brasil, às normas celetistas (SÜSSEKIND, Armando. Convenções da OIT. SP. Editora LTr.- 2ª ed. – 1998, pg. 36/37)

⁸ SÜSSEKIND, Arnaldo. Convenções... Ob. Cit. pg. 02 (matéria de pé de página)

⁹ CF/88, art. 1º, III e Art. 5º “caput”

¹⁰ OLIVEIRA, S. Geraldo. Proteção Jurídica à Saúde do Trabalhador. SP. Editora LTr.- 3ª edição, 2001, pg. 118

¹¹ A Espanha, p.ex., que ratificou a Convenção 155 da OIT, a exemplo do Brasil, determina, em sua normativa prevencionista (Ley de Prevención de Riesgos Laborales: 31/95, de 08 de Novembro, art. 14.2 c/c art. 19.1 del Estatuto de los Trabajadores), que a empresa tome tantas medidas quantas sejam necessárias para a completa e eficaz proteção do trabalhador nas áreas de segurança e saúde ocupacional.

¹² MAXIMILIANO, Carlos. Hermenêutica e Aplicação do Direito. Forense/RJ/92. 12ª ed. pg. 305

basta a determinação do pagamento dos adicionais de insalubridade e periculosidade, pois o objetivo maior é a preservação da integridade psicossomática do trabalhador. Por isso, não podemos concordar que a matéria figure apenas na lei ordinária”.

Qual teria sido a intenção do legislador constituinte ao tomar atitude extrema de inserir, no corpo permanente da Carta Magna, tema pertinente às vias ordinárias ou regulamentadoras como fez com o inciso XXII do artigo 7º, quando grafou que a empresa seria obrigada a *reduzir* (até os limites de tolerância: NR-15, itens 15.1.1 e 15.1.5.) os riscos *por meio da normas técnicas*? Além da matéria segurança e saúde do trabalho ser do rol dos direitos fundamentais, (cláusulas pétreas), certamente o constituinte estava receoso, pelo eterno descumprimento das normas prevencionistas, de que se não inserisse no texto magno, tal descumprimento haveria de continuar com as facilidades de sempre, já que as Constituições anteriores¹³ sempre previram apenas nas formas de princípios e, por isso, as empresas optaram por monetarizar a saúde do trabalhador ao cumprir com o dever de prevenir e elidir o risco ambiental, como recomenda a OIT.

3. O TERMO ‘REMUNERAÇÃO’, DO INCISO XXIII, ART. 7º, DA C.F.

Com o propósito de fazer as empresas cumprirem e ordenar que seja cumpridas as normas de segurança e saúde do trabalho (7º, XXII da CF c/c art. 157, I da CLT), sem dúvida, que o constituinte desejou estabelecer uma punição mais severa para a empresa inadimplente com suas obrigações prevencionistas. Assim o fez, dizendo que o adicional de insalubridade (7º, XXIII) deveria ser sobre a *remuneração* e não mais sobre o salário mínimo, não só porque aviltante e um convite ao desprezo e desconsideração das questões de saúde e integridade física do obreiro mas, também, pelo fato de que, igualmente desejou, isolar o mínimo de qualquer vinculação indexatória (CF: 7º, IV), para que tivesse seu poder aquisitivo permanente. A intenção de majorar o adicional, numa espécie de sanção econômica, pode ser percebida no Parecer do Relator da Sistematização, exarado à Emenda n. 15397/87, onde ficou grafado que **“Os adicionais salariais pelo trabalho em condições insalubres ou perigosos são muito menos uma compensação pecuniária para o empregado que uma forma de induzir o empregador a eliminar ou restringir esses riscos”**, para ter a certeza do cumprimento das normas prevencionistas. Ou, como consignado em outro Parecer, desta feita, à Emenda n. 01302/87, parcialmente aceita pela Relatoria da Sistematização, que continha a seguinte proposição: “Modifique-se a redação do inciso XXI do art. 13º¹⁴ (...) que além dos controles tecnológicos visando à eliminação do risco, promova a redução da jornada e **um adicional de remuneração incidente sobre o salário contratual**” (grifo nosso). Ou no Parecer do Relator da Sistematização, exarado em outra emenda, a de n. 20298, que propugnava pelo seguinte: “Dê-se a seguinte redação ao item mencionado:¹⁵ - Adicional de insalubridade sobre o salário mínimo pela prestação de trabalho em atividade insalubre, penosa ou perigosa, além de proteção através de controles e equipamentos que visem a reduzir o grau de risco da atividade”.

Parecer do Relator: ...”Quanto à incidência do adicional sobre o salário mínimo, trata-se de medida completamente inócua, pois o empregador preferirá pagar o adicional do que introduzir melhorias para reduzir a insalubridade. É por isso que no texto consta o termo **“REMUNERAÇÃO”** (o grifo é do original, portanto, do Relator).

Como se vê, nenhuma dúvida paira sobre a real intenção do legislador máximo, quando da elaboração da atual Carta Magna. Tinha, como se viu, o propósito deliberado de majorar o adicional, pois “para reduzir os riscos e melhorar as condições de trabalho, agravamento

¹³ Onde previam apenas de forma enunciativa e como princípio: “Higiene e Segurança do Trabalho”.

¹⁴ Artigo este que se referia aos direitos dos trabalhadores nas fase anteriores da Constituinte .

¹⁵ Item e Emenda se referiam aos direitos dos trabalhadores, naquela fase da Constituinte.

dos adicionais compensatórios é o meio mais eficaz”¹⁶. E o fez dizendo ser *sobre a remuneração* do empregado, como, aliás, é o adicional de periculosidade. É de importância capital, para conclusão da real intenção do legislador constituinte, o grifo aposto na expressão “de remuneração”, hoje, inclusive, constante do texto final da Carta Magna. Dizia o Relator da Sistematização” *já constava do texto*, numa clara demonstração que assim quiseram os que elaboraram a Carta Republicana, desde as primeiras fases de sua elaboração.

O intérprete não pode descurar da real intenção do legislador, a fim de se manter fiel, o mais proximamente possível, da vocação normativa, até porque “não é possível extrair da norma toda sua potencialidade sem conhecer sua gênese, características, atributos, virtudes, finalidade e eficácia...” até mesmo para não “...empobrecer o rendimento do preceito legal”¹⁷. O juiz ou intérprete deve se valer do critério interpretativo sistemático (combinativo) e não apenas se apegar a um item do texto. Deve preferir o conjunto ao dispositivo isolado¹⁸. Interpretar o conteúdo do inciso XXIII a partir da própria

Constituição, sem buscar sua gênese, é interpretar e concluir com prejuízo ao beneficiário da norma: o trabalhador¹⁹. Também o Enunciado n. 47, do TST, pode dar margem a essa interpretação. Senão vejamos:

“ADICIONAL DE INSALUBRIDADE – TRABALHO EXECUTADO EM CARÁTER INTERMITENTE. O trabalho executado, em caráter intermitente, em condições insalubres, não afasta, só por essa circunstância, o direito à percepção respectivo adicional”.

Ora, se o adicional fora instituído para remunerar trabalhador que estivesse em contato permanente com o risco, não há razão jurídica para remunerá-lo quando ausente ou afastado deste. Na interpretação do TST, mesmo que o empregado não esteja em contato com o risco, este continua existindo. Por isso o Enunciado manda pagar *enquanto* existir o risco ou *condições insalubres*, desde que nocivas à saúde ou integridade física do obreiro. A lógica seria pagar o adicional, quando o ambiente hostil à saúde ou integridade do trabalhador. Como faz, por exemplo, a lei previdenciária, no concernente à aposentadoria especial, a qual exige prova da permanência permanente e habitual²⁰ do obreiro ao risco para obtenção do benefício²¹, que não considera, entretanto, intermitente, por ficção jurídica, o tempo que o trabalhador empregado estiver afastado do risco motivado pelas necessidades fisiológicas, inclusive férias e descanso remunerado²². Portanto, o contato com o risco deve ser permanente e habitual, inexistindo nas exposições eventuais²³.

4. ÓBICE DO SALÁRIO MÍNIMO COMO INDEXADOR

Conceitualmente, o salário mínimo serve para atender as necessidades básicas do trabalhador e sua família (CF: 7º, IV)²⁴. As necessidades são representadas pelas utilidades que compõe o mínimo, as quais foram exaustivamente debatidas ao longo do processo legislativo da constituinte e propositadamente aumentadas, para fazer frente às crescentes

¹⁶ OLIVEIRA, S. Geraldo de: CIPA – Ed. CIPA - SP – edição n. 168 – ano XIV – 1993 – pg. 56

¹⁷ OLIVEIRA, Sebastião G. de: Proteção... Ob. Cit. pg. 438

¹⁸ Idem, pg. 345.

¹⁹ Ibidem, pg. 345

²⁰ Decreto n. 93.412/86, art. 2º, I. ou Orientações Jurisprudenciais da SDI-I (TST) n. 174.

²¹ BRASIL. Lei 8.213/91, art. 57, § 3º e Decreto n. 3048/99, art. 64, §§ 3º e 4º

²² MARTINEZ, Wladimir Novaes. Comentários á Lei Básica da Previdência Social. SP. Ed. LTr. - 5ª edição – pg. 374

²³ CARRION, Valentin. Comentários à CLT. SP. Saraiva - 26ª edição. 2001. pg. 174, ao se referir do adicional de periculosidade, que deve ter, pelo que vimos, tratamento igual à insalubridade.

²⁴ Ou como prefere dizer o Min. Arnaldo Süssekind, tem por finalidade precípua a de atender as necessidades biológicas e sociais não só do trabalhador como também de sua família. In Instituições de Direito do Trabalho. SP. Ed. LTr. - 4ª edição. 1994. pg. 381.

necessidades vitais do trabalhador e sua prole, “uma vez que é difícil admitir que o trabalhador que ganha salário mínimo não deva ter o direito teórico à educação, saúde, lazer”²⁵. Além do mais, a tendência será a elevação real do salário mínimo e com aumentos periódicos, para manter o poder aquisitivo²⁶ e isso não seria possível se continuasse a ser fator de indexação da economia. Em outras vezes se tentou desindexá-lo²⁷ sem muito sucesso, pois o mercado, mais forte, tratou de disciplinar, com prevalência, as relações sobre a norma jurídica. Se a lei não resolveu o fenômeno indexatório, era preciso, então, um tratamento mais radical, no sentido de expurgar, de vez, não só a norma indexatória como a cultura de tudo indexar ao mínimo. Foi o que se fez. Elevou-se à Constituição a proibição de indexar ao mínimo (7, IV), *para todo qualquer fim, sem exceção*²⁸, já que a indexação “criaria dificuldades para os aumentos efetivos e reais do valor do salário para atender as necessidades”²⁹.

Mesmo estando previsto na Constituição de que o salário mínimo não poderia ser parâmetro indexatório, a jurisprudência dos pretórios trabalhistas continuou aceitando o cálculo da insalubridade sobre o mesmo até que a questão fosse enfrentada pela Suprema Corte Brasileira, a qual deu um basta à reiterada indexação, como se pode depreender nos seguintes arestos:

“ *EMENTA: Adicional de Insalubridade: vinculação ao salário mínimo, estabelecida pelas instâncias ordinárias, que contraria o preceito no art. 7º, IV, da Constituição*” (STF, RE-236396 - DJ 20/11/98 – 1ª Turma (unânime) - Rel. Min. Sepúlveda Pertence ou

“ *EMENTA: Adicional de insalubridade (LC 432/85). Vinculação ao salário mínimo. Vedação constitucional, (art. 7º, IV). Precedente do STF. Fundamento do despacho agravado não afastado. Regimental não provido.*” (STF, AGRRE – 271752 – DJ 20/10/00 – 2ª Turma (unânime) – Rel. Min. Nelson Jobim.

“ *EMENTA: Salário mínimo – Vinculação. A teor do disposto no inciso IV do artigo 7º da Constituição Federal, descabe tomar o salário mínimo como fator relativo a cálculo de parcela ainda que de natureza trabalhista*”. (STF – 2ª Turma, RE n. 221234-4 – Rel. Min. Marco Aurélio, DJ de 05/05/2000)

Indubitavelmente, “à medida foi adotada ao se concluir que a indexação do salário mínimo vinha dificultando a sua elevação na medida em que provocava efeitos sobre a economia, em desacordo com os propósitos de contenção da elevação dos preços como forma de combate à inflação”³⁰.

²⁵ NASCIMENTO, Amauri Mascaro. Teoria Jurídica do Salário. Ed. LTr. - SP – 1994 – pg. 299

²⁶ Idem. Ob. Cit. pg. 299

²⁷ Lei 6.205, de 29/04/75 e Decreto Lei n. 2.351, de 07/08/87

²⁸ O Min. Nelson Jobim, em uma das passagens de sua decisão (STF, AR no RE n. 271752-5/SP, 2ª T,- DJ 20/10/00) 20/10/2000 no STF, quando enfrentou a questão diz que “O art. 7º, IV, da Constituição de 1988 dispõe que é vedada a vinculação do salário mínimo para qualquer fim. Essa norma tem, evidentemente, **caráter de vedação absoluta** (o grifo é nosso), tendo em que, sendo ele utilizado como parâmetro indexador de obrigações de **qualquer natureza** (grifo nosso), se criassem dificuldades para os aumentos efetivos do valor deste pela extensão de seu reflexo ocasionado por essa utilização”. Não se pode esquecer que o Min. Jobim, agora no STF, interpreta e julga a Constituição que ajudou a elaborar. Decide, portanto, com autoridade e conhecimento como poucos tem a oportunidade de fazer. Experiência que não se pode desprezar.

²⁹ STF, AGRRE n. 271752-5/SP, interposto em Recurso Extraordinário, 2ª T – Rel. Min. Nelson Jobim julgamento em 20/10/00 – DJ de 20/10/00

³⁰ NASCIMENTO, Amauri Mascaro. Teoria... Ob. Cit. pg. 121

5. BASE DE CÁLCULO DA INSALUBRIDADE

Sendo vedado calcular o adicional sobre o salário mínimo, qual é, então, a base de cálculo? Valemo-nos da própria decisão da Suprema Corte, que de forma definitiva, aponta a solução para o caso, quando ordena que “as instâncias ordinárias que estabelecer vinculação com o salário mínimo, o processo a ela deve retornar, a fim de que **decida** (grifo nosso) qual critério legal substitutivo do adotado e aplicável”³¹. Portanto, aos Regionais Trabalhistas e até mesmo ao TST cabem **decidir** qual base de cálculo adotar. Não quer dizer, com isso, que o juiz passa a ser legislador, criando ou legislando a base de cálculo. Deve saber ou aplicar o parâmetro ou parâmetros que o sistema lhe apresenta. Tanto o constituinte como o STF apontou o caminho a seguir pois, “Quanto melhor souber a jurisprudência adaptar o Direito vigente às circunstâncias mutáveis da vida, tanto menos necessário se tornará pôr em movimento a máquina de legislar. Até mesmo a norma defeituosa (em comento, /art. 192 da CLT) pode atingir seus fins, (os da Constituição) desde que seja inteligentemente aplicada”³².

O constituinte ao colocar os dois agentes na mesma norma (XXIII), quis dizer: dêem o mesmo tratamento e paguem na mesma forma insalubridade e periculosidade “pois, o objetivo maior é a preservação da integridade psicossomática do trabalhador”³³ e deve ficar na Constituição porque matéria relevante e de ordem pública³⁴. Além do mis, esta é a recomendação dos órgãos internacionais: OIT/OMS, sempre preocupados com o sentido amplo de saúde como sendo a promoção e manutenção, no mais alto grau, do bem estar físico, mental e social...”³⁵.

Não destoia dessa orientação e pensamento, a mais respeitável doutrina³⁶, concordando com quem elaborou a norma ápice, no sentido de dar o mesmo tratamento entre a insalubridade e periculosidade³⁷. E, depois, “todos os adicionais são calculados sobre o salário contratual, não havendo, portanto, razão lógica tampouco jurídica para dar tratamento diverso para o adicional de insalubridade”^{38 39}. Muito menos em termos de pagamento, uma vez que os “motivos de ordem humana ou econômica só sugerem a igualdade no cálculo dos dois adicionais”⁴⁰. Seria até desumano tratamento desigual para algo que produz e causa dor e sofrimento e que não se pode mensurar tais elementos.

5.1. “NA FORMA DA LEI”, DO INCISO XXIII, DO ART. 7º DA C.F.

Se o salário mínimo não pode mais ser a base de cálculo para o adicional de insalubridade, pelo que foi visto alhures, não tem-se dúvida que deverá ser aplicado sobre o salário contratual ou a remuneração do empregado, como acontece com a periculosidade e outros adicionais. Na ‘forma da lei’ que se refere a Constituição (7º, XXIII) são os elementos que o sistema jurídico oferece. (analogia, pelo art. 8º da CLT). Este, na falta de um lei ou norma específica, diz ser perfeitamente possível tomar, como parâmetro, o salário contratual ou a remuneração (como o da periculosidade, CLT, art. 193, § 3º). Não se está dizendo, tampouco querendo, que o julgador ou intérprete, em função do critério da analogia, passe a ter liberdade para criar direito novo e sim apenas aplicar o que já existe,

³¹ STF, RE-236396-5/MG, 1ª T - Rel. Min. Sepúlveda Pertence – DJ 20/11/98

³² DÜRINGER, *apud* MAXIMILIANO, Carlos. *Hermenêutica...* Ob. Cit. pg. 61

³³ Cfe. Parecer do Relator da Sistematização da Constituinte à Emenda n. 07647, em 05/08/87

³⁴ Cfe. Parecer do Relator da Sistematização, à Emenda n. 12940/87.

³⁵ VIEIRA, Sebastião I. *Manual de Saúde e Segurança do Trabalho*. Vol. I. Mestra. 2000. pg. 51

³⁶ OLIVEIRA, Sebastião G. de: Ob. Cit. pg. 344, que alinha os fundamentos aqui traçadas.

³⁷ OLIVEIRA, Francisco A de. *Comentários aos Enunciados TST - SP- RT- 5ª ed.* 2001. pg. 289

³⁸ OLIVEIRA, Sebastião G. de: Ob. Cit. pg. 345

³⁹ SAAD, Eduardo G. *In CLT Comentada – Editora LTr. – SP – 1993 – 27ª edição – pg. 145/146*

⁴⁰ *Idem*, Ob. Cit. pg. 145/146

adaptando, patenteado pela vida social”⁴¹. O juiz deve agir mais como investigador que criador, sua argúcia revela-se em não se apegar a um texto incompleto e sim a recorrer a uma combinação, preferindo o conjunto, ao dispositivo isolado⁴². Aplicar os percentuais do art. 192 (insalubridade), na base de cálculo da do art. 193 (periculosidade), por analogia (CLT, art. 8º), não é nenhum despropósito ou ‘fora da lei’, isto porque, o inciso XXIII do art. 7º da Lei Máxima não só permite como ordena que assim seja. De mais a mais, a norma em comento não permite que se interprete apenas na visão estreita da literalidade e sim no conjunto sistemático constitucional⁴³, por ser, este, o método mais adequado à interpretação⁴⁴. E, depois, não “se aplica uma norma jurídica, mas sim o Direito, não se interpretam normas constitucionais, isoladamente, mas sim no seu todo, conforme a própria Constituição,”⁴⁵ pois, “se o objeto é idêntico, parece natural que as palavras, embora diversas tenham significado semelhante”⁴⁶.

6. CONCLUSÃO

Por tudo o que foi alhures, não se tem receio em afirmar que o adicional de insalubridade *dever* ser calculado sobre a remuneração que perceber o empregado ou o contratado, a exemplo da periculosidade, *não mais*, sobre o salário mínimo como historicamente sempre aconteceu. Caso contrário, estar-se-á ofendendo a Constituição Federal e o próprio Supremo Tribunal Federal, guardião da Carta Magna, quando enfrentou a questão ao julgar e decidir que o salário mínimo não pode ser fator de indexação, exatamente como quis o legislador constituinte de 87/88, ao grafar os preceitos dos incisos IV, XXII e XIII do artigo 7º da Leis das Leis. Assim sendo, não há mais diferença ou desigualdade de tratamento entre os empregados em contato com agentes insalubres ou perigosos, nem no tratamento prevencionista, nem no trato da remuneração. Todos são iguais perante a lei e isonomicamente considerados perante a lei trabalhista. Nem mesmo a recente proposta de alteração da legislação obreira (projeto de lei n.5.483/01, do Executivo Federal), já aprovado na Câmara Federal, tem o condão de alterar o que aqui se viu. O máximo que as partes (sindicatos) poderão negociar é o percentual que incidirá, jamais a base de cálculo já decidida pela Suprema Corte Brasileira.

BIBLIOGRAFIA E FONTES UTILIZADAS OU CONSULTADAS

ASSEMBLÉIA NACIONAL CONSTITUINTE. ANAIS. Vol. 7 e 8. 1987.

CARRION, Valentin. Comentários à CLT. São Paulo. Ed. Saraiva. 26ª edição. 2001.

CIPA. Revista CIPA. Ed. CIPA. São Paulo. Edição n. 168. Ano 1993.

FIGUEIREDO, Guilherme José P. de. Direito Ambiental e as Saúde dos Trabalhadores. São Paulo. Ed. LTr. 2000.

GOMES, Orlando *et alii*. (atualizado por José Augusto R. Pinto) Curso de Direito do Trabalho. Rio de Janeiro. Ed. Forense. 16ª ed. 2001.

GRAU, E. R., “Ordem Econômica na Constituição de 1988 (interpretação e crítica)”. São Paulo. Ed. Revista dos Tribunais, 1990.

⁴¹ MAXIMILIANO, Carlos. Hermenêutica...Ob. Cit. pg. 214

⁴² Idem. Ob. Cit. pg. 49

⁴³ LEDUR, José Felipe. A REALIZAÇÃO DO DIREITO AO TRABALHO. Sergio Antonio Fabris Editor – Porto Alegre – 1998 – pg. 109

⁴⁴ Idem, pg. 109

⁴⁵ GRAU, E. R., “Ordem Econômica na Constituição de 1988 (interpretação e crítica)”. SP – Ed. Revista dos Tribunais, 1990 – pg. 181, *apud* José Felipe Ledur, Ob. Cit. pg. 109, nota de pé de página

⁴⁶ MAXIMILIANO, Carlos. Hermenêutica... Cb. Cit. pg. 110

- LEDUR, José Felipe. A Realização do Direito ao Trabalho. Porto Alegre/RS. Sergio Antonio Fabris Editor. 1998.
- MARTINEZ, Wladimir Novaes. Comentários à Lei Básica da Previdência Social. São Paulo. Ed. LTr. 5ª edição. 2001.
- MARTINS FILHO, Ives G. da Silva. “Os direitos Fundamentais e os Direitos Sociais na Constituição de 1988 e sua Defesa”. Revista Jurídica Virtual, nº4, Ago/99: <http://www.planalto.gov.br>. capturado em: 15/01/02
- MARTINS, Sergio Pinto. Comentários à CLT. São Paulo. Atlas. 3ª ed. 2000
- MAXIMILIANO, Carlos. Hermenêutica e Aplicação do Direito. RJ. Ed. Forense. 12ª edição. 1992.
- MORAES, Alexandre de. Direito Constitucional. São Paulo. Ed. Atlas. 7ª ed. 2000
- NASCIMENTO, Amauri Mascaro. Teoria Jurídica do Salário. São Paulo. Ed. LTr. 1994.
- _____, Curso de Direito do Trabalho. São Paulo. Saraiva. 17ª ed. 2001.
- OLIVEIRA, Francisco A de. CLT Comentada. São Paulo. Ed. RT. 2ª edição. 2000.
- _____, Comentários aos Enunciados do TST. São Paulo. Ed. RT. 5ª ed. 2001.
- OLIVEIRA, Sebastião G. de. Proteção Jurídica à Saúde do Trabalhador. São Paulo. Ed. LTr. 3ª edição. 2001.
- RUSSOMANO, Mozar V. Comentários à CLT. Rio de Janeiro. Forense. 11ª ed. 1985
- SAAD, Eduardo G. CLT Comentada. São Paulo. Ed. LTr. 27ª edição. 1993.
- SALIBA, Tuffi Messias *et alii*. Insalubridade e Periculosidade – Aspectos Técnicos e Práticos – São Paulo. Ed. LTr. 5ª edição. 2000.
- SENADO FEDERAL. Home Page <http://www.senado.gov.br>.
- SUSSEKIND, Arnaldo *et alii*. Instituições de Direito do Trabalho. São Paulo. Editora LTr. 4ª edição. 1994
- SUSSEKIND, Arnaldo. Convenções da OIT. São Paulo. Ed. LTr. 2ª ed. 1998.
- VIEIRA, Sebastião Ivone *et alii*. Manual de Saúde e Segurança do Trabalho. Vol. I. Florianópolis/SC. Mestra Editora. 2000

NORMAS PARA PUBLICAÇÃO

Artigos submetidos podem ser escritos em português, inglês ou espanhol, utilizando espaçamento 1,0 linha, em apenas uma face do papel, formato A4, fonte "Times New Roman", em geral, tamanho 12, texto "justificado", com margens de, no mínimo, 1 polegada (2,5 cm) em todos os lados. Cada artigo deve ter, no máximo, 08 páginas, incluindo todo o texto, figuras e referências bibliográficas.

A primeira página deve conter o título do artigo, nomes dos autores, um resumo, seguido por palavras-chave. Na mesma página, deve estar o "abstract" seguido das "keywords" do artigo.

O título do artigo deve ser conciso e completo, para facilitar sua indexação futura, e deve ter no máximo 15 palavras (fonte "Arial", tamanho 16, maiúsculo, negrito, centralizado). O nome dos autores (abreviados quando necessário) deve ser escrito em fonte tamanho 10, itálico, centralizado. Afiliação e endereço (postal e/ou eletrônico), devem estar como notas de rodapé (fonte 10).

O resumo e seu "abstract" correspondente devem ter no máximo 200 palavras, sendo vedadas citações bibliográficas, fórmulas e equações. Sempre que possível, deve ter 1/3 sobre material e métodos, e 2/3 sobre resultados, devendo transmitir a idéia de seu conteúdo de forma clara e completa. O resumo deve ser seguido por no máximo seis palavras-chave identificadoras do artigo, e o "abstract" também deve ser seguido pelas mesmas palavras-chave em inglês ("keywords"). O texto do resumo e do "abstract" devem ser em fonte tamanho 10, "justificado".

Todos os subtítulos devem ser escritos na fonte "Arial", tamanho 12, alinhados à esquerda e numerados (iniciando na introdução com número 1). Deixar uma linha em branco antes de cada subtítulo.

Figuras, imagens e tabelas devem estar incluídas no próprio texto, já em sua localização definitiva, numeradas com algarismos arábicos. As ilustrações não devem ser coloridas.

Equações e fórmulas devem ser numeradas seqüencialmente no texto, usando algarismos arábicos.

As referências no texto, sua citação no final do artigo, e todo tipo de notas adicionais devem seguir as normas estabelecidas pela ABNT - Associação Brasileira de Normas Técnicas ou ISO - *International Standards Organization*.

A formatação do artigo deve obedecer o formato disponível no *site* da Revista do CCEI (arquivo: *template.zip* - para *Microsoft Word*).

A **submissão de artigos deve ser via e-mail** para o endereço revista@ccei.urcamp.tche.br.

O resultado da seleção é comunicado via e-mail, quando então, os artigos selecionados deverão ser novamente enviados em arquivo eletrônico, formato *.DOC* (*MS-Word for Windows*).

Próxima edição

Data limite para submissão: 15/06/2002

Publicação: Agosto/2002

Endereço para correspondência:

URCAMP-Universidade da Região da Campanha
CCEI-Centro de Ciências da Economia e Informática
Av. General Osório, nº 2289
Cep 96400-101 - Bagé - RS
E-mail: revista@ccei.urcamp.tche.br
<http://www.urcamp.tche.br/ccei/revista/revista.html>