

ISSN 1415-2061

REVISTA DO CCEI

Centro de Ciências da Economia e Informática

Volume 5 Número 7

MARÇO 2001

BAGÉ - RS

EDITORA DA URCAMP - EDIURCAMP

Rev. CCEI	BAGÉ - RS	V.5	N.7	Mar. 2001
-----------	-----------	-----	-----	-----------

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 irregular, de divulgação técnico-científica, editada pela Editora da URCAMP - EDIURCAMP.

As opiniões e conceitos emitidos nos artigos são de exclusiva responsabilidade de seus autores.

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:

Prof. Cristiano Cachapuz e Lima

COORDENADOR DO CURSO DE CIÊNCIAS CONTÁBEIS:

Prof. Eduardo Roman Sonza

COORDENADOR DO CURSO DE ADMINISTRAÇÃO, HABILITAÇÕES: EMPRESAS,

RURAL e ANÁLISE DE SISTEMAS:

Prof. Enio Del Geloso Nocchi

CAPA: Marsal Alves Branco

REVISÃO: Prof^a Elza Maria d' Athayde

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.urcamp.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

Irregular

2001. v.5 n.7

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

Catálogo Sistema de Bibliotecas/URCAMP

REVISTA DO CCEI

v.5, n.7, 2000

CONSELHO EDITORIAL:

Direção do CCEI:

Prof. Enio Del Geloso Nocchi
Prof. Cristiano Cachapuz e Lima
Prof. Eduardo Roman Sonza

Informática:

Prof. Luiz Cláudio Dalmolin, M.Sc.
Prof. Salvador L.T.Camargo, M.Sc.

Ciências Contábeis:

Prof. Augusto Pinheiro Grande
Prof. Flávio Garibaldi

Ciências Econômicas:

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

Administração:

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

Administração Rural:

Prof. Cláudio Marques Ribeiro, M.Sc.
Prof. João Antônio Gomes Martins da Silva, M.Sc.

Análise de Sistemas:

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

Editor:

Prof. Luiz Cláudio Dalmolin, M.Sc.

Assessores Técnicos:

Prof^a. Ada M.M.Guimarães, M.Sc.
Prof^a Elza Maria d' Athayde
Prof^a. Elza Maria Steinhorst
Prof^a. Jhansy Collares, M.Sc.
Bibl. Maria Bartira N. Costa Taborda
Bibl. Nelci Maria Birk Jeismann

Comissão Avaliadora:

Prof. Cláudio Marques Ribeiro, M.Sc.
Prof. Cristiano Cachapuz e Lima
Prof. Edar da Silva Añaña
Prof. Enio Del Geloso Nocchi
Prof. Gilleanes Thorwald Araújo Guedes
Prof. Henry Gomes de Carvalho
Prof. João Antônio Gomes Martins da Silva, M.Sc.
Prof. Léu Cardoso Carate, M.Sc.
Prof. Luiz Cláudio Dalmolin, M.Sc.
Prof. Salvador Loni Tadeu Camargo, M.Sc.

EDITORIAL

Na sua sétima edição e quinto ano de circulação, a Revista do CCEI tornou-se conhecida dentro e fora dos limites da URCAMP.

É dever de toda Universidade ser um centro que oferece ensino, pesquisa e extensão. Um dos veículos mais importantes para o registro e divulgação de pesquisas é o periódico científico. Estes, ainda são poucos no Brasil e de circulação quase que exclusiva nos meios acadêmicos. A Revista do CCEI insere-se nesse cenário e serve como fonte de informação para estudantes, pesquisadores e demais profissionais das áreas compreendidas pelo Centro de Ciências da Economia e Informática.

Nesta edição, os autores abordam temas variados, como: solução para problemas de roteamento de veículos, gerador de programas para a simulação de autômatos, formalismos e linguagens de programação, especificação de hiperdocumentos, reflexão computacional, visualização de música, aplicações multimídia na Internet, gerenciamento de transações distribuídas, mecanismos de buscas, XML, marketing, comunicação organizacional, tamanho de organizações, cooperativismo e solução para sistemas de equações lineares.

SUMÁRIO

Resolução do problema de roteamento de veículos utilizando a heurística de <i>Savings</i> ; SANTOS, Willian B. dos; CORDENONSI, Andre Z.	7
Ambiente gerador de programas Java para a simulação de Autômatos RS; RUBIN, Rodrigo R.; LIBRELOTTO, Giovani R.; TOSCANI, Simão S.	15
Aspectos formais sobre linguagens de programação; MELLO, Fabrício de R.; LACERDA, Guilherme S. de	23
Um estudo sobre a solução paralela de sistemas de equações lineares utilizando a técnica “ <i>Bulk Synchronous Parallel</i> ”; CHAGAS, Elza M. P. de F.	29
Abordagens sobre especificação de hiperdocumentos; SEVERO, Carlos E. P.	36
Tamanho da organização e o indivíduo; LUNELLI, João P.	43
A importância do <i>marketing</i> de relacionamento e a implementação do planejamento de <i>marketing</i> ; COUGO, Ricardo L.	52
Reflexão computacional; FERNANDES, Acauan P.	61
Sistemas de visualização de música; POLONSKII, Mikhail M.	67
Marketing: da segmentação ao posicionamento, uma estratégia para enfrentar momentos de crise; MARTINS, Anderson V.	76
Comunicação organizacional: perspectivas e modelos, ANDRADE, Jeanice J. de	84
Exportação de aplicações multimídia para o ambiente Web utilizando Toolbook; CARDOSO, Rodrigo F.	91
Técnicas de gerenciamento de transações distribuídas; SILVEIRA, Fábio F.	98
Idéias para o desenvolvimento de um mecanismo de buscas para a Embrapa; MARGARITES, Marcus V. F.	107
A criação de cooperativas autênticas como alternativa de geração de emprego e renda; SILVA, Laudes M. da	115
Desenvolvimento de um esquema XML para banco de dados sobre ovinos; COSTA, João A. M.; COSTA, Antônio Carlos da R.	120

RESOLUÇÃO DO PROBLEMA DE ROTEAMENTO DE VEÍCULOS UTILIZANDO A HEURÍSTICA DE SAVINGS

Willian Bolzan dos Santos¹

Msc. Andre Zanki Cordenonsi²

RESUMO

Este artigo tem por objetivo apresentar um estudo sobre a resolução do Problema de Roteamento de Veículos (PRV) utilizando técnicas de Inteligência Artificial e Pesquisa Operacional. O PRV é um clássico problema de Otimização Combinatorial sendo um dos mais difíceis de serem resolvidos por exigir grande esforço computacional. O problema consiste, basicamente, em estabelecer rotas eficientes para veículos entregarem mercadorias. Utilizou-se a Heurística de *Saving* como método de resolução para o problema, já que sua implementação é simples e fornece bons resultados. Problemas desse tipo estão presentes em praticamente todas as empresas que trabalham com transporte, já que a escolha de rotas mais eficientes representa ganho de tempo e redução no número de veículos, proporcionando menor consumo de combustível, menor desgaste dos veículos, etc.

Palavras-chave: Roteamento, Otimização Combinatorial, Inteligência Artificial, Heurística.

ABSTRACT

The objective of this paper is present a study about the Vehicle Rotering Problem, using techniques of Artificial Intelligence and Operational Research. The VRP is classical problem of Combinatorial Optimization and it is one of the most difficult problem to be solved. The problem is find a efficiently rout for a set of vehicles to delivery merchandise. This kind of problem, usually, requires a lot of computational effort to find the best rout. The Savings Heuristic was the algorithm choosed to solve this problem, because it is simple and supply good results. This kind of problems can be viewed in a lot of real problems, like the transport companies. The benefits reached by the use of heuristics can be summarized in three points: global time reduction, reduction of the number of vehicles and reduction of the global path of the all vehicles.

Keywords: *Routing, Combinatorial Optimization, Artificial Intelligence, Operational Research, Heuristic.*

1 INTRODUÇÃO

Pode-se definir um PRV de maneira simples. Para o problema que será abordado neste trabalho, dispõe-se de uma frota de veículos e deseja-se atender um conjunto de clientes, cada um com uma demanda específica. Os veículos devem partir e retornar ao depósito, visitando os clientes uma única vez. O objetivo principal é minimizar os "custos" de transporte no atendimento aos clientes.

A idéia principal é criar rotas para os veículos percorrerem todos os pontos de entrega de modo que a capacidade do veículo não exceda e que o percurso inicie e termine no depósito. Devido a sua complexidade, torna-se difícil a obtenção de soluções ótimas se o número de clientes for grande. Para tanto, existem procedimentos ou critérios chamados heurísticas que exploram de maneira inteligente o problema, buscando soluções de boa qualidade, mas sem garantir a obtenção da solução ideal.

¹ Aluno do Curso de Informática – URCAMP / Campus São Gabriel. *E-mail: wbolzan@hotmail.com*

² Prof. Curso Informática – URCAMP / Campus São Gabriel. *E-mail: andrezc@uol.com.br*

2 OTIMIZAÇÃO COMBINATORIAL

Otimização Combinatorial pode ser definida como sendo o estudo matemático para encontrar um arranjo, agrupamento, ordenação ou seleção ótima de objetos discretos (Lawler – 1976). A maioria dos Problemas de Otimização Combinatoriais – (POCs) são bem definidos, isto é, uma solução ótima sempre existirá se um conjunto de soluções factíveis não for vazio.

Mas esta solução ótima não é o principal objetivo dos pesquisadores e sim uma forma de encontrar uma boa solução (próxima da solução ótima) em um tempo computacional razoável. POC são considerados fáceis e difíceis e podem ser classificados em quatro classes de problemas: *P*, *NP*, *NP-Árduo* e *NP-Completo*.

Os POC da classe *P* podem ser resolvidos por algoritmos de tempo polinomial; *NP* tem dificuldade crescente, sendo os problemas *NP-Completo*, os tipos mais difíceis de serem resolvidos. Dentre esses problemas, podemos citar o Problema do Caixeiro Viajante e o Problema de Roteamento de Veículos.

Uma forma de resolver tais problemas seria simplesmente enumerar todas as soluções possíveis e guardar aquela de menor custo. Entretanto, para qualquer problema de um tamanho minimamente interessante (e útil), este método torna-se impraticável já que o número de soluções possíveis é imenso. Portanto, técnicas de Inteligência Artificial são necessárias.

O objetivo principal em trabalharmos com técnicas de Inteligência Artificial e Otimização é estudar como minimizar (ou maximizar, dependendo do problema) variáveis ligadas a determinadas tarefas.

3 O PROBLEMA DE ROTEAMENTO DE VEÍCULOS

3.1 Introdução

Um PRV consiste basicamente em estabelecer e organizar rotas ou itinerários eficientes para veículos realizarem entregas de mercadorias.

Em outras palavras, dispomos de uma frota de K veículos idênticos ou não e desejamos atender um conjunto de n clientes, cada um com uma demanda específica. Todos os veículos devem partir e retornar a uma mesma origem (depósito) e cada cliente deve ser visitado uma única vez.

O objetivo geral será minimizar o "custo total" de transporte no atendimento aos clientes, isto é, minimizar custos fixos, custos operacionais e o número de veículos envolvidos no transporte.

3.2 Fatores Importantes

Alguns fatores importantes que podem ser considerados e que dificultam a construção das rotas são:

- a) Frota heterogênea: veículos com diferentes capacidades de carga.
- b) Janelas de Tempo: Janelas de Tempo em um PRV representam o intervalo de tempo no qual a entrega da mercadoria pode ser feita. Um exemplo típico desse problema que podemos citar ocorre na distribuição de bebidas para restaurantes. Por exemplo, não seria conveniente para o restaurante receber suas mercadorias no horário do almoço. Portanto, as Janelas de Tempo em um PRV possuem valores limites para iniciar a entrega.
- c) Viagens longas: a circulação entre cidades geralmente força os veículos a rodarem durante mais de um dia. Devem então ser invocadas regras especiais na formação da rota que tornam o problema muito complexo.

- d) Horários de início e fim das atividades: alguns dos problemas requerem que todos os veículos devam começar e acabar na mesma hora. Outros permitem diferentes especificações de horário de início e final para as rotas de cada um dos veículos, o que os torna mais realistas, pois na prática não seria sempre possível que todos os veículos deixem o depósito simultaneamente.
- e) Questão de Assimetria: em um PRV, a Questão de Assimetria é um fator muito importante. Um problema assimétrico é o que possui o caminho para ir de i até j diferente de j até i ($ij \neq ji$). Um exemplo prático disso são ruas de mão única e vértices separados por uma serra, se o custo for representado pelo tempo.

3.3 PRINCIPAIS APLICAÇÕES

O PRV possui um número extraordinário de aplicações práticas, pois implica tipicamente em uma série de situações reais que afetam principalmente a indústria, o comércio, o setor de serviços, a segurança, a saúde pública e o lazer. Dentre muitas outras, podemos destacar:

- Distribuição de jornais;
- Distribuição de bebidas;
- Distribuição de produtos químicos;
- Transporte escolar;
- Recolhimento de lixo;
- Distribuição de gás;
- Entrega de correspondência;
- Roteamento de linhas aéreas;
- Sistema de transportes coletivos;
- Serviços de emergência..

4 ALGORITMOS HEURÍSTICOS

Heurísticas são critérios ou métodos utilizados para decidir, entre várias alternativas, a melhor forma de chegar a um objetivo. A maioria dos problemas de OC exigem a avaliação de um grande número de alternativas para determinar a sua solução exata. Heurísticas aumentam a eficiência do processo em detrimento da exploração de todas as alternativas.

A utilização de heurísticas em problemas práticos de grande dimensão e de solução inviável por métodos exatos de solução visam encontrar soluções de boa qualidade de maneira rápida, isto é, com baixo esforço computacional.

Para entendermos melhor, tomemos um exemplo do Problema do Caixeiro Viajante - PCV: O PCV é uma variante do PRV, o qual consiste em encontrar uma rota em um grupo de cidades tal que o caminho percorrido entre a primeira e a última cidade seja o menor possível.

Suponhamos que temos N clientes para atendermos, temos:

- $(N-1)!$ Caminhos
- N para examinar um caminho
- $N!$ para realizar a busca

Agora veja a comparação entre o uso de um algoritmo exaustivo (processa até encontrar a melhor solução: $(N!)$) com a heurística do vizinho mais próximo (partindo de uma cidade, a próxima cidade a ser visitada será aquela que estiver mais próxima da cidade atual, quando não houver mais cidades a inserir na rota, retorna a cidade de partida fechando a rota: (N^2)). Considerando-se um computador capaz de examinar cada solução em 10^{-9} segundos (1 nanosegundo) tem-se:

N	Algoritmo Exaustivo	Heurística
20	80 anos	0,4 ms
21	1.680 anos	0,44 ms

Tabela 1: Comparação entre metodologias

5 O ALGORÍTMO DE SAVINGS

O Algoritmo de Saving foi escolhido para a implementação do PRV por fornecer soluções de qualidade razoável, por ser simples e de implementação fácil.

Conforme (Goldbarg 00), o algoritmo de *saving* realiza a progressão da uma configuração para outra segundo o critério de minimização da função-objetivo, também chamado de *saving* (economia). Arcos de menor custo devem substituir arcos mais caros dentro da rota que vai sendo melhorada nesses termos. No procedimento de economia e inserção não existe a obrigatoriedade de que a rota seja viável ao longo do processo de melhoria. Se alguma solução alcançada for viável, então caracteriza-se a obtenção de um limite superior para o problema.

Primeiramente, forma-se uma solução inicial para n pontos de entrega com n rotas, todas contendo o depósito e um ponto de entrega. Após essa fase, tenta-se unir duas rotas em uma rota factível a cada iteração. Sendo i e j dois pontos de entrega, o critério utilizado para eliminar o maior custo é dado por:

$$S_{ij} = c_{i0} + c_{0j} - c_{ij} \quad (1)$$

onde c_{i0} é a distância entre o ponto de entrega i e o depósito (aqui representado por 0), c_{0j} é a distância entre o depósito 0 e o ponto de entrega j e c_{ij} é a distância entre os dois pontos de entrega.

A cada iteração as rotas são organizadas em conjuntos de pares. Um ponto i e um ponto j podem formar o par (i,j) se:

- os pontos de entrega i e j não estão na mesma rota;
- a capacidade de carga do veículo não é violada;
- nem i nem j são *pontos interiores* em uma rota.

Um ponto interior em uma rota significa que seu ponto anterior e seu ponto sucessor não podem ser o depósito .

6 RESULTADOS

Para a realização dos testes foram estabelecidos, primeiramente, alguns parâmetros que estão listados a seguir:

- Número de Clientes: 10, 25, 50, 100, 150 e 200;
- Capacidade do Veículo: 2000 e 4000 kg;
- Limite da Demanda: 1000 , 2000 e 4000 kg. Esse parâmetro significa que o cliente terá um valor limite para sua encomenda não podendo, portanto, excedê-la.

Com os parâmetros estabelecidos, podemos formar as configurações. Para cada cliente temos 2 valores para a capacidade e 3 valores para o limite da demanda. Portanto, para 6 conjuntos de clientes temos 36 tipos de configurações.

De posse dos parâmetros e das configurações, é necessário um critério para iniciar os testes. Esse critério refere-se ao fato de lidarmos com dados de entrada gerados aleatoriamente, o que causa pequenas variações nos dados finais.

O critério estabelecido é executar 5 testes para cada configuração e retirar a média. Agora, se temos 36 tipos de configurações e devemos executar 5 testes para cada configuração, temos 180 execuções do algoritmo.

6.1 Análise dos Resultados

A seguir, a Tabela 2 mostra a média dos resultados obtidos com a execução dos testes iniciais., a qual contém as seguintes informações:

- Nº de Rotas: a quantidade de rotas geradas. Podemos deduzir que esta informação também representa o número de veículos necessários para atender todos os pontos de entrega;
- Nº de Clientes: a quantidade de clientes ou pontos de entrega envolvidos no roteamento;
- Capacidade (kg): o valor estabelecido para a capacidade de carga dos veículos;
- Economia (%): representa o valor da economia conquistada pelo roteamento. O cálculo da economia baseou-se na distância inicial antes do roteamento menos a distância final percorrida após o roteamento;
- Taxa de Ocupação (%): esta taxa refere-se à ocupação do veículo, o quanto da sua capacidade de carga está sendo utilizada;
- Limite da Demanda (kg): refere-se ao valor estipulado para o limite de produtos encomendados pelos clientes;
- Tempo (s): representa o tempo computacional gasto para a execução desta configuração;

Nº de Rotas	Nº de Clientes	Capacidade (kg)	Economia (%)	Taxa de Ocupação (%)	Limite da Demanda (kg)	Tempo (seg)
6,4	10	4000	24,17%	75,26%	4000	1,4
2,8	10	4000	31,45%	74,63%	2000	1
1,8	10	4000	27,48%	65,40%	1000	1,4
6,2	10	2000	24,99%	82,16%	4000	1,2
2,8	10	2000	36,77%	77,26%	2000	1,2
1,6	10	2000	19,11%	73,10%	1000	1
15,4	25	4000	43,47%	80,08%	4000	1,4
7,2	25	4000	48,81%	79,18%	2000	1,6
4,4	25	4000	30,02%	77,39%	1000	1,6
14,2	25	2000	43,20%	81,60%	4000	1,2
7,4	25	2000	47,45%	82,20%	2000	1,6
4	25	2000	13,55%	78,05%	1000	1,4
28,4	50	4000	50,60%	83,26%	4000	1,8
15,4	50	4000	48,19%	85,00%	2000	1,6
7,6	50	4000	11,84%	82,76%	1000	2
27,4	50	2000	49,74%	83,93%	4000	1,8
13,8	50	2000	38,38%	82,89%	2000	1,8
7,6	50	2000	15,34%	81,98%	1000	2
60,6	100	4000	49,37%	84,86%	4000	5,2
28	100	4000	43,51%	85,75%	2000	4,8
15,4	100	4000	13,02%	83,30%	1000	4,6
57	100	2000	52,53%	85,48%	4000	5,2
27,6	100	2000	32,87%	87,11%	2000	4,8
15,6	100	2000	11,80%	81,43%	1000	5
87,2	150	4000	50,44%	86,15%	4000	13
42,8	150	4000	41,94%	85,62%	2000	12
22,2	150	4000	7,36%	85,41%	1000	11,4
22,6	150	2000	8,31%	84,94%	4000	11,4
43,4	150	2000	43,55%	85,69%	2000	12,2
21,8	150	2000	8,92%	84,05%	1000	11,4
111,2	200	4000	55,68%	85,45%	4000	27
56	200	4000	37,80%	86,18%	2000	25
28,2	200	4000	7,82%	87,08%	1000	24,2
116,2	200	2000	53,68%	84,03%	4000	26,6
56	200	2000	45,63%	86,69%	2000	25
28	200	2000	7,74%	85,43%	1000	24,2

Tabela 2: Resultados

Em uma primeira análise, podemos verificar que a economia se manteve, aproximadamente, na faixa dos 25% a 30%. Para ser mais preciso, levando-se em conta todas as configurações analisadas neste problema, a média da economia conquistada é de 32,68%. Podemos dizer que, na média, a economia com a implementação desse algoritmo simples de PRV foi muito alta.

De posse das informações contidas na tabela anterior, podemos começar a analisar aspectos específicos desses resultados.

Começamos a análise da economia obtida, em relação à capacidade do veículo. O Gráfico 1 mostra essa relação. Com o auxílio da Linha de Tendência (linha pontilhada), podemos verificar facilmente que a economia se manteve entre os 30% e 35%.

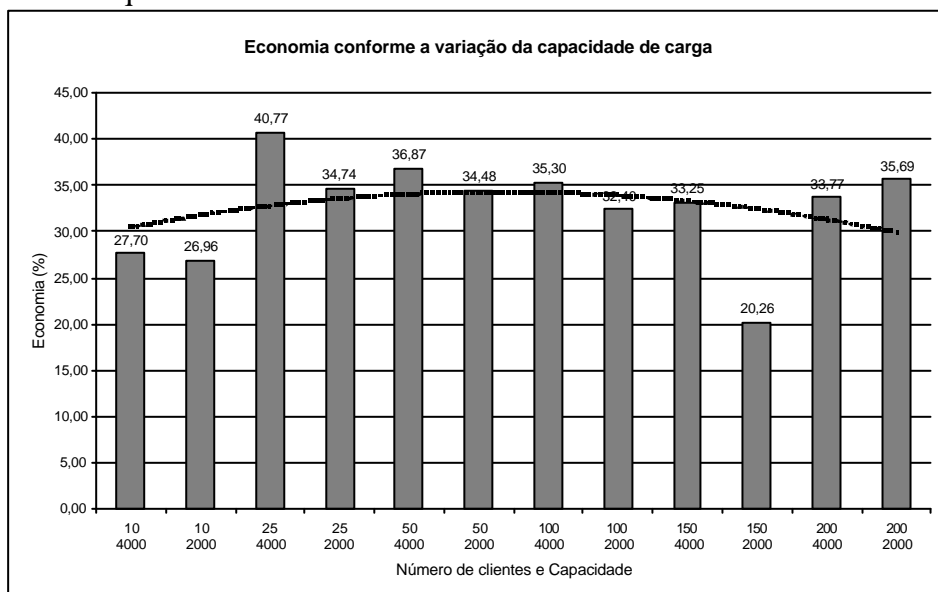


Gráfico 1: Economia conforme a variação da Capacidade

O Gráfico 2 analisa a taxa de ocupação do veículo. Para todos os dados a taxa se manteve na faixa dos 80%. Um veículo com uma taxa de ocupação de 80% significa que sua capacidade está 80% ocupada.

Por exemplo, se o veículo possui uma capacidade de 4000 kg e sua taxa de ocupação é de 80%, neste caso ele estaria com uma carga de 3200 kg.

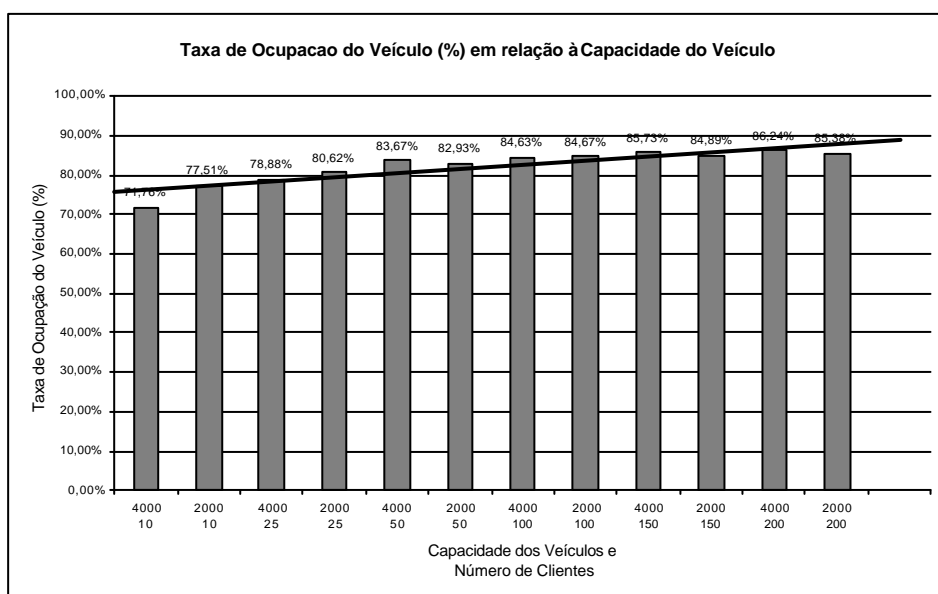


Gráfico 2: Taxa de Ocupação em Relação à capacidade e ao número de clientes

A variação do número de rotas em relação à capacidade do veículo é muito pequena (Gráfico 3), cerca de 1 rota de diferença, ou seja, quando a capacidade está fixada em 2000 kg temos um número x de rotas, mas quando a capacidade está fixada em 4000 kg, o número de rotas passa a ser $x+1$.

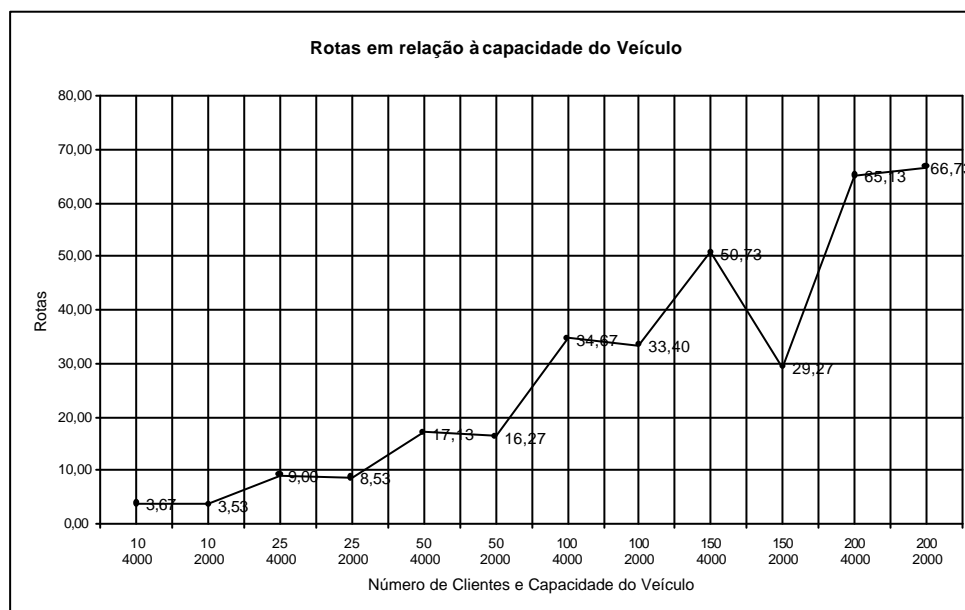


Gráfico 3: Rotas em relação à capacidade

7 CONCLUSÕES

No Brasil, ainda é pequeno o número de empresas que adotam sistemas de otimização. É comum que as empresas não tenham consciência de que certas tarefas podem ser otimizadas para tornarem-se mais eficientes e lucrativas.

Em empresas que têm suas atividades ligadas ao transporte, o seu sistema de roteamento de veículos é de fundamental importância para um bom desempenho. Portanto, esse trabalho busca analisar um método simples de otimização para auxiliar as empresas em suas atividades.

O método de otimização utilizado neste estudo possibilitou uma solução de boa qualidade, obtendo uma economia considerável, variando em torno dos 30%. No entanto, ao trabalharmos com algoritmos heurísticos não obtemos, ao final, uma solução definitiva. Assim, é importante salientar que alguns métodos de melhoramento são necessários.

Ao analisarmos a firma como as rotas são geradas, notamos a importância de se trabalhar com rotas não factíveis. No nosso algoritmo, se uma rota resultasse em uma rota não factível, essa troca não podia ser realizada.

Essa característica impede que sejam realizadas algumas trocas que poderiam provocar redução no seu custo. Algumas técnicas, como Busca Tabu, permitem que sejam geradas rotas infactíveis para permitir trocas mais lucrativas.

Outro aspecto importante nesse trabalho está relacionado à utilização de arquivos para os dados de entrada e saída, ocasionado certa dificuldade de utilização. Para tanto, faz-se necessária uma futura construção de uma interface amigável.

8 REFERÊNCIAS BIBLIOGRÁFICAS

BITTENCOURT, G. Inteligência Artificial – Ferramentas e Teorias. Livro Texto - X Escola de Computação, Instituto de Computação, Campinas, 08-13 jul., 1996.

- CHAN, Patrick. The Java Class Libraries - An Annotated Reference. Addison Wesley. Massachusetts, 1996.
- CORNELL, Gary. Core Java - Guia Autorizado. Makron Books. 1998
- GOLDBARG, Marco Cesar, et al.. Otimização Combinatória e Programação Linear - Modelos e Algoritmos. Editora Campus. Campus Rio de Janeiro, 2000.
- MULLER, Felipe Martins. Algoritmos heurísticos e exatos para resolução de problema de sequenciamento em processadores paralelos. Tese de Doutorado, Faculdade de Engenharia Elétrica, Universidade Estadual de Campinas. Campinas, 1993.
- PASIN, Márcia, Um estudo de problemas de roteamento de veículos. Trabalho de Graduação, Curso de Bacharelado em Informática, Universidade Federal de Santa Maria. Santa Maria, 1995.
- TANEMBAUM, Aaron et at. Estruturas de Dados usando C. Editora Makron. São Paulo, 1995.
- V ESCOLA REGIONAL DE INFORMÁTICA DA SBC REGIONAL SUL, 5., 1997, Florianópolis - SC, Maringá - PR, Santa Maria - RS. Mini-curso: Heurísticas e Metaheurísticas. Santa Maria, 1997. 209p .
- VELOSO, Paulo et al. Estruturas de Dados. Editora Campus. Campus - Rio de Janeiro, 1996.

AMBIENTE GERADOR DE PROGRAMAS JAVA PARA A SIMULAÇÃO DE AUTÔMATOS RS

Rodrigo Rubert Rubin¹, Giovani Rubert Librelotto², Simão Sirineo Toscani³

RESUMO

O objetivo do trabalho é construir um ambiente para a geração de programas Java, que terá como finalidade a simulação de autômatos gerados pelo Compilador RS 5.0. O ambiente gerador criará códigos fonte em Java a partir do autômato RS fornecido. Os autômatos gerados pelo compilador RS estão no formato *autômato-regras*, que é o padrão para representação dos autômatos RS. O modelo dos programas Java a serem criados tenta ser o mais genérico possível, possibilitando o fácil entendimento por programadores que tenham alterações a serem feitas no código gerado pelo ambiente de geração de programas Java para autômatos RS.

ABSTRACT

The goal of this paper is to build an environment for the generation of Java programs, that will have as purpose the simulation of the automaton generated by the RS Compiler 5.0. The generating environment will create source codes in Java starting from the RS automaton supplied. The automaton generated by RS Compiler are in the format automaton-rules, that is the pattern for the representation of the RS automaton. The model of the Java programs tries to be the most generic possible, facilitating the understanding for programmers that have alterations be done in the generated code by the environment of generation of Java programs for RS automaton.

1 INTRODUÇÃO

Sistemas reativos são sistemas que interagem com um ambiente externo, mantendo um relacionamento dinâmico com esse. Tais sistemas devem responder a estímulos provenientes do ambiente externo, alternando em dois tipos de períodos: espera o estímulo dos sinais e reage a eles. As linguagens reativas síncronas constituem uma ferramenta de programação de sistemas reativos. Essas linguagens adotam a hipótese de sincronismo, onde se presume que toda reação seja instantânea e, portanto, atômica. Os sistemas reativos são normalmente divididos em três camadas: uma camada de interface com o ambiente, um núcleo reativo e uma camada de manipulação de dados.

A linguagem RS destina-se à programação de núcleos reativos, que constituem a parte central e mais difícil de um sistema reativo. Como as demais linguagens síncronas, o código objeto gerado pelo compilador RS também é um autômato finito cuja execução pode ser realizada de forma muito eficiente.

O trabalho tem por objetivo principal criar um ambiente que, a partir dos autômatos gerados pelo compilador RS, gere código na linguagem Java. Esses programas gerados pelo ambiente tem como principal função a de simular a execução desses autômatos. O ambiente gerará código Java, que será responsável pela simulação do autômato RS fornecido. Os sinais teriam entrada via teclado e as saídas, via monitor.

¹ Aluno da Faculdade de Ciência da Computação da UNICRUZ – guigo@dinf.unicruz.tche.br

² Aluno M.Sc. do Programa de Pós-Graduação em Computação da UFRGS – giovanir@inf.ufrgs.br

³ Professor e pesquisador da UNICRUZ – Universidade de Cruz Alta, RS – simao@unicruz.tche.br

Os autômatos gerados pelo compilador RS estão no formato *autômato-regras*, que é o padrão para representação dos autômatos RS. Esse Compilador RS 5.0 está em desenvolvimento na UFRGS, como tema de uma dissertação de mestrado.

2 A LINGUAGEM RS

Sistemas reativos são sistemas que interagem com um ambiente externo, mantendo um relacionamento dinâmico com esse. Tais sistemas devem responder a estímulos provenientes do ambiente externo, alternando em dois tipos de períodos: espera o estímulo dos sinais e reage a eles. As linguagens reativas síncronas (como Lustre, ESTEREL e SR) constituem uma ferramenta de programação de sistemas reativos. Essas linguagens adotam a hipótese de sincronismo, onde se presume que toda reação seja instantânea e, portanto, atômica. Isso corresponde a considerar sistemas ideais, que reagem instantaneamente a cada estímulo externo com uma transformação de estado interno e com uma emissão de sinais. Os sistemas reativos são normalmente divididos em três camadas: uma camada de interface com o ambiente, um núcleo reativo e uma camada de manipulação de dados.

A linguagem Reativa Síncrona (RS) destina-se à programação de núcleos reativos, que constituem a parte central e mais difícil de um sistema reativo. Tais núcleos são responsáveis por toda a lógica de um sistema reativo, manipulando os sinais de entrada, realizando as reações e gerando os sinais de saídas. Essa linguagem permite, a partir de um programa escrito em RS, gerar um autômato finito correspondente.

A linguagem RS constitui uma notação adequada para representar o comportamento de núcleos reativos, pois um programa é uma especificação quase direta das transformações internas e das emissões de sinais que devem acontecer para cada estímulo possível.

A *hipótese do sincronismo*, adotada pela linguagem RS, supõe que toda reação se desenvolva em tempo zero. Como consequência, os sinais de saída são síncronos com os sinais de entrada e o tempo só passa durante a atividade do ambiente externo. Esta suposição simplifica a semântica da linguagem e permite que os programas sejam compilados para autômatos finitos. Esta hipótese resulta em vantagens para o programador tais como: reconciliar concorrência e determinismo, escrever programas simples e mais rigorosos e desassociar a lógica de um sistema das características dependentes de implementação, tais como tempos de reação.

3 O COMPILADOR RS 5.0

Inicialmente, foram feitos estudos, tanto da Linguagem Reativa Síncrona (RS) quanto do Compilador RS, ambos na sua versão mais atual, a versão 5.0. Os conhecimentos na linguagem RS serão de grande valia, pois os autômatos a serem simulados neste ambiente serão gerados pelo compilador que está em fase de testes na UFRGS. As observações e algumas conclusões tiradas destes estudos estão citadas neste artigo.

Na figura 1, mostramos o funcionamento do Compilador RS 5.0. Esse compilador trouxe várias alterações na Linguagem RS, em relação à sua versão anterior (RS 4.0), como por exemplo o acréscimo de mais de um sinal em uma regra de disparo, a possibilidade de ter-se sinais inibidores no gatilho de disparo de uma regra de reação, uma simplificação na escrita dos programas, com a eliminação de colchetes para a delimitação de listas e o fim da declaração de listas vazias.

O Compilador RS 5.0 recebe como entrada o código fonte RS nesta nova versão e após a sua compilação, retorna 2 arquivos, contendo a descrição do autômato RS. No primeiro arquivo, o arquivo *autômato*, encontra-se o autômato propriamente dito, com todo o seu comportamento relacionado a cada sinal proveniente do ambiente externo. No segundo arquivo, o arquivo *regras*, encontram-se as regras de execução referenciadas pelo arquivo anterior. Portanto, ambos os arquivos (*autômato-regras*) se completam.

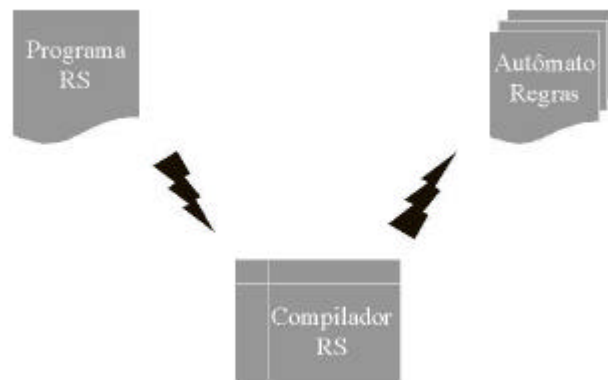


FIGURA 1 – O funcionamento do Compilador RS 5.0

4 MODELO DOS AUTÔMATOS RS

Em geral, as linguagens reativas síncronas geram um código objeto que é um autômato finito. A linguagem RS também apresenta a mesma característica. O comportamento de qualquer programa RS pode ser representado por um autômato finito que simula a sua execução em tempo mínimo. Na realidade, esse autômato que representado um programa RS é um autômato estendido, similar à máquina de Mealy, o qual inclui uma memória e ações para serem realizadas em tempo de execução.

Com um programa fonte RS, pode-se realizar a compilação deste autômato. O código gerado pelo Compilador RS 5.0 é composto por duas partes principais: o autômato e as regras ligadas a esse autômato. As duas partes são interdependentes e devem ser estudadas em conjunto.

A seguir, temos um exemplo que demonstra o funcionamento de uma roleta de cinema. Os sinais que vêm do ambiente externo são *open* (significa que a roleta foi aberta), *close* (significa que a roleta foi fechada), e *people* (significa que passou uma pessoa pela roleta). As saídas do sistema são: *openRoleta* (significa que a roleta está aberta), *closeRoleta* (significa que a roleta está fechada), *people(X)* (significa que passaram X pessoas, quando a roleta foi fechada), e *ring* (que é um aviso de erro). Dois sinais internos permanentes são declarados, *a* e *b*, que significam que a roleta está aberta (*b*) e que ela está fechada (*a*). E a variável *peopleCount* é o contador de pessoas que passaram pela roleta enquanto ela esteve aberta.

Este sistema reativo tem seis regras de execução, as quais possuem um funcionamento semelhante, como descrito: no caso da regra onde os sinais disparadores têm o formato *open+[a] ==> ...*, a regra será executada somente quando o sinal *open* vier do ambiente externo no mesmo instante em que o sinal permanente *a* estiver ligado. O mesmo serve para todas as regras de disparo do exemplo a seguir:

```

module roleta:
  input : open, close, people;
  output : openRoleta, closeRoleta, people(X), ring;
  p_signal : a, b;
  var : peopleCount;
  initially : up(a), emit(closeRoleta);
  open+[a] ==> peopleCount:=0, emit(openRoleta), up(b);
  close+[a] ==> emit(closeRoleta), up(a);
  people+[a] ==> emit(ring), up(a);
  open+[b] ==> emit(ring), up(b);
  close+[b] ==> emit(people(peopleCount)), emit(closeRoleta), up(a);
  people+[b] ==> peopleCount:=peopleCount+1, up(b);
end module.

```

Após o programa acima ser compilado pelo Compilador RS 5.0, serão gerados os códigos abaixo, que representam o funcionamento de uma roleta de cinema:

```

AUTOMATON
init      -      [1, *, go_to(1)]
1        open   [2, 3, *, go_to(2)]
1        close  [1, *, go_to(1)]
1        people [4, *, go_to(1)]
2        open   [4, *, go_to(2)]
2        close  [5, 1, *, go_to(1)]
2        people [6, *, go_to(2)]

RULES
Module roleta:
1. [] ==> [emit(closeRoleta)]
2. [] ==> [peopleCount:=0]
3. [] ==> [emit(openRoleta)]
4. [] ==> [emit(ring)]
5. [] ==> [emit(people(peopleCount))]
6. [] ==> [peopleCount:=peopleCount +1]

```

A palavra *AUTOMATON* inicia a definição de um autômato, podendo ser visto como a descrição de uma máquina de estados. O número à esquerda representa um estado do autômato, sendo que *init* é o estado inicial. Cada linha possui três componentes identificados *n, s, a*, onde *n* é um número de estado, *s* é um nome de sinal externo e *a* é a ação a ser executada quando, no estado *n*, o autômato é estimulado pelo sinal *s*. Os asteriscos separam regras que podem ser executadas em paralelo. A Figura 2 sintetiza o formato de uma linha do autômato.

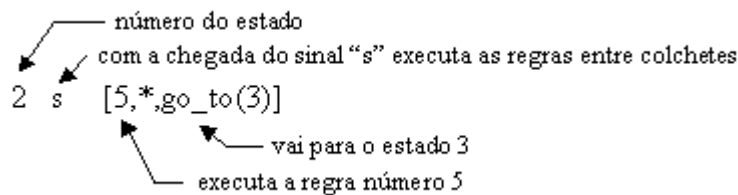


FIGURA 2 - Linha de um autômato gerado por RS

A máquina de estados correspondente ao autômato está representada na Figura 3.

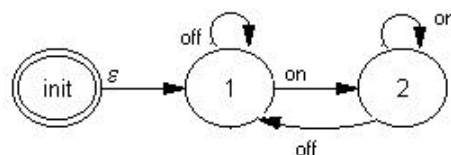


FIGURA 3 - Máquina de estados correspondente ao programa RS

A segunda parte inicia com a palavra "RULES". Nesta parte, são descritas as ações a serem executadas por um autômato. O formato geral de uma regra é a seguinte:

n. [condição] ==> [ação]

O funcionamento de uma regra pode ser descrita assim: quando uma regra do arquivo do autômato, como *1 open [2, *, go_to(2)]* é executada, isto é, quando o autômato estiver no estado *1* e o sinal de entrada for *open*, o autômato executará a regra *2* do arquivo de regras, e após isso, passará automaticamente para o estado *2*. A regra *2* é a seguinte:

2. [] ==> [peopleCount := 0, emit(openRoleta)]

que nos mostra que ao ser executada, atribuirá para a variável *peopleCount* o valor 0, e emitirá para o ambiente externo o sinal *openRoleta*. O autômato referente a esta especificação de roleta, pode ser visualizado na figura 4.

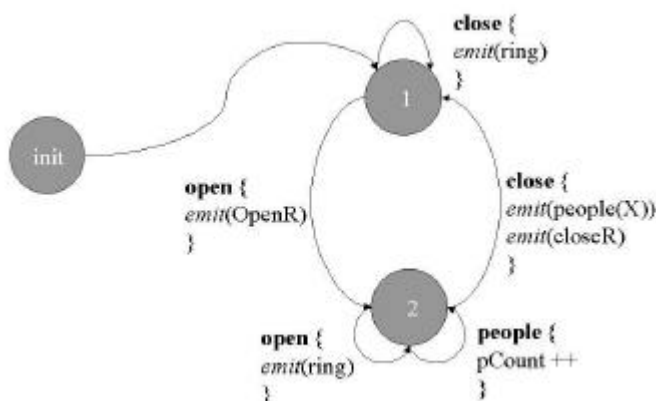


FIGURA 4 – Autômato que descreve o funcionamento da roleta

4.3 GERADOR DE CÓDIGO

O gerador de programas para a simulação de autômatos RS terá como entrada, justamente os arquivos de autômato e regras de autômato, gerados pelo Compilador RS 5.0. A partir desses 2 arquivos, o gerador de código criará um programa Java, que poderá ser compilado e executado em qualquer sistema operacional que tiver um compilador Java. O formato do gerador de código Java para autômatos RS está demonstrado na figura 5.

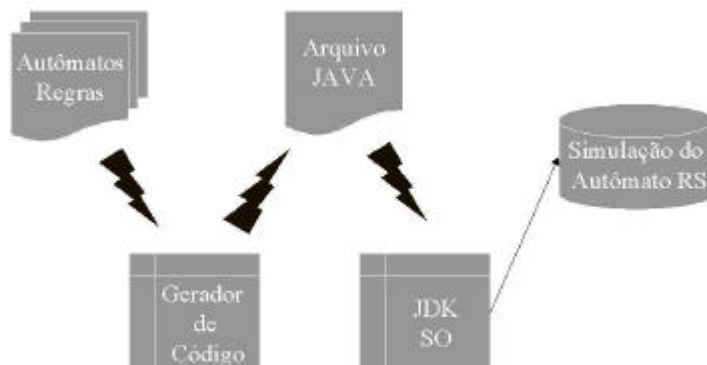


FIGURA 5 – O formato do gerador de código Java para autômatos RS

5 O GERADOR DE CÓDIGO JAVA PARA AUTÔMATOS RS

O gerador de código Java para autômatos RS 5.0 é auto-suficiente, não necessitando o uso de qualquer linguagem hospedeira. Além disso, sua versão final é multiplataforma, pois foi construído sobre a linguagem C, padrão ANSI. Este ambiente gerador de código receberá como entrada um arquivo com um programa fonte RS e tem um funcionamento semelhante a um compilador. Os programas fonte, no caso os arquivos *autômato* e *regras*, passarão pelas análises léxica, sintática e semântica. Na análise léxica, serão encontrados todos os *tokens* dos arquivos *autômato-regras* e passados para a análise sintática, que por sua vez, montará uma árvore sintática simples com as operações a serem executadas pelo autômato RS. A análise semântica entra em ação no mesmo momento, para verificar, entre outras coisas, a tipagem dos sinais e as condições booleanas encontradas no código. Não havendo erros em nenhuma das análises, a próxima etapa é a geração de código.

5.1 O Código Java para a simulação de Autômatos RS

Nesta seção, explicaremos as principais características do código Java gerado pelo Ambiente Gerador RS. Detalhes de implementação, como nomes de funções e chamadas a sub-rotinas, serão deixados de lado para nos concentrarmos, especificamente, nos seguintes detalhes:

1. Declaração e tipificação de variáveis e sinais;
2. Tratamento de asteriscos;
3. Representação do autômato em Java;
4. Tratamento de exceções internas;
5. A interface com o usuário;
6. A execução do sistema.

A geração do código Java passa por três fases distintas: a leitura dos sinais de entrada, a geração de regras e a geração do autômato. Na primeira fase, são identificados todos os sinais de entrada e também todas as variáveis. Na etapa posterior, é realizada a tradução de comandos e atribuições RS para a linguagem Java. Em uma terceira fase, a ordem de execução das regras é definida. Partindo-se da definição do autômato RS, constrói-se a rotina *autômato* do código Java, que basicamente conterá chamadas a procedimentos e funções anteriormente traduzidos.

5.1.1 Declaração e tipificação de variáveis e sinais

Como não existe declaração de tipo em RS, o compilador define todas as variáveis do sistema como do tipo *int*. Assim, portanto, toda e qualquer variável declarada em um programa fonte RS, vai ser interpretada no programa Java gerado como uma variável inteira.

5.1.2 Tratamento de asteriscos

Os asteriscos, na representação de um autômato RS, têm o objetivo de separar ações que podem ser executadas em paralelo. Para garantir o correto funcionamento dessas ações, deve-se garantir que elas terão acesso aos sinais com os valores originais dos mesmos. Isso significa que as mudanças que uma ação realizará em um sinal X não serão visíveis em outro. Entretanto, no fim do passo de reação, marcado pelo próximo asterisco, o valor deve ser corretamente atualizado. Torna-se claro que o Ambiente Gerador RS nunca deixará que duas ações, dentro de um mesmo passo de reação, modifiquem a mesma variável, pois estaríamos dentro de uma situação de inconsistência, na qual a solução seria a criação de uma seção crítica.

A idéia, porém, não é o uso de uma seção crítica, visto que apenas uma das ações poderá modificar a variável. Para realizar tal controle, usamos o mesmo esquema adotado pelo Compilador RS 4.0. Nele, uma cópia extra do sinal é alocado. Valores sempre são lidos do sinal original e escritos na cópia. Quando o passo de reação acaba, a cópia é usada para atualizar o sinal original. Convém ressaltar que o Ambiente Gerador RS verifica se uma segunda escrita está sendo feita sobre a cópia da variável. Caso seja uma nova escrita e os valores, da primeira e da nova, sejam diferentes, uma situação de inconsistência ocorreu. Então, o ambiente acusa um erro de execução.

5.1.3 Representação do autômato em Java

Para exemplificar a codificação de um autômato RS, vamos nos basear no exemplo da roleta de cinema (seção 4). Os trechos abaixo, foram retirados dos arquivos *roleta.aut* e *roleta.rul*. Eles representam o autômato *roleta* quando está esperando o estímulo *open*,

quando o estado atual é 1. À variável *peopleCount*, indicada abaixo, é atribuído o valor 0 para começar a nova contagem de pessoas. O sinal *openRoleta* é emitido ao ambiente externo para avisar que a roleta está aberta. E por fim, o autômato é passado para o estado 2 da simulação.

No exemplo abaixo, o trecho de código representado na tabela 1, é apresentado em Java, na figura 6. Note que a variável *user* recebe o sinal proveniente do ambiente externo.

<i>roleta.aut</i>	1 open [2, 3, *, go_to(2)]
<i>roleta.rul</i>	2. [] ==> [peopleCount:=0] 3. [] ==> [emit(openRoleta)]

TABELA 1 – Trechos de código dos arquivos *autômato-regras* da roleta

```

if(est_atual == 1)
    if(user.equals("open")) {
        peopleCount = 0;
        System.out.println("openRoleta");
        est_atual = 2;
    }

```

FIGURA 6 – Trecho de código em Java referente à tabela 1

Embora a representação dos autômatos RS seja bastante simples, foi necessário algum esforço para a construção de um compilador geral que permitisse o tratamento dos diversos níveis de aninhamento que este pode conter. Por exemplo, pode existir um comando de teste em que uma das opções se desdobra em outro teste, que se desdobra em outro teste, e assim por diante, sem previsão do número máximo de níveis.

5.1.4 A interface com o Compilador RS 5.0 e com o usuário

A interface do Ambiente Gerador com o Compilador RS 5.0 é feita através dos arquivos *autômatos-regras* gerados pelo compilador. Portanto, se não tivermos posse de tais arquivos, e apenas do próprio código fonte RS, teremos que compilar esse autômato no Compilador RS 5.0 para obtermos os arquivos necessários para a geração do código Java.

Ao serem obtidos os arquivos *autômato-regras*, basta ao usuário passar como parâmetro o nome do autômato a ser compilado, diretamente na linha de comando da execução do Ambiente Gerador. Abaixo, mostramos a utilização do Ambiente Gerador RS, para a *roleta*:

```
%> rs2java roleta
```

Após a compilação, será gerado o arquivo *roleta.java*. Esse código, ao ser compilado e executado sobre uma JVM (Java Virtual Machine), executará a simulação completa do autômato RS.

5.2 O funcionamento do Ambiente Gerador

O Ambiente Gerador de Código Java para Autômatos RS pode ser dividido em três partes distintas. Cada uma delas, prepara a execução do sistema para a etapa seguinte.

O primeiro procedimento a que o autômato RS é submetido, tem como função ler todos os sinais de entrada desse autômato. Essa etapa somente entra em contato com o arquivo *autômato*, pois é nele que se encontram todos os sinais de entrada. Isso acontece para armazenar todos os sinais de entrada na tabela de símbolos, para a formação do programa Java na terceira etapa do Ambiente.

Após a leitura completa do arquivo *autômato*, onde foram procurados os sinais de entrada do autômato em questão, é acionado o segundo procedimento de nosso Ambiente. Essa segunda etapa tem como função ler todas as regras do arquivo de *regras* e armazená-las em uma estrutura que é dividida em 2 partes. A primeira parte tem o índice da regra, que é usado para fazer a ligação entre os arquivos *autômato-regras*. A segunda parte, é o local onde será armazenada a regra RS propriamente dita. Nessa etapa, são feitas algumas alterações em algumas regras, para adaptá-las à linguagem de programação Java. Como por exemplo, a regra “*emit*”, é substituída nessa etapa pela instrução *System.out.println* a qual faz a emissão de sinais ao usuário.

A terceira etapa se divide em duas partes: a leitura do arquivo *autômato* e a criação do código Java. Mas ambas as tarefas são executadas ao mesmo tempo, pois conforme se recebe as operações do autômato RS, converte-se para a linguagem Java e salva-se no arquivo destino. Na leitura do arquivo *autômato* se identificarão, primeiramente, as variáveis do sistema e, posteriormente, todas as regras de transição de estados que o autômato poderá sofrer. E ao passo que cada regra de transição vai sendo lida, vai sendo instantaneamente traduzida para Java e sendo colocada no arquivo final. Como as regras do arquivo autômato não têm influência umas no funcionamento das outras, não há por que armazená-las em estruturas para uma posterior gravação. Então, conforme vão sendo lidas, vão sendo transformadas e armazenadas.

CONCLUSÕES

O presente artigo mostrou o projeto de um ambiente para a geração de programas Java para a simulação de autômatos RS. Embora no seu estado atual a linguagem RS possa ser considerada como tendo boas qualidades para os fins a que se destina, é natural que hajam aperfeiçoamentos para serem introduzidos. No futuro próximo, alguns melhoramentos já definidos deverão ser introduzidos. Também pretende-se criar um ambiente de execução mais apropriado para a simulação e teste de programas RS, baseado neste estudo aqui apresentado e, ainda, utilizar a linguagem no desenvolvimento de aplicações de maior porte. As aplicações práticas poderão indicar novos aperfeiçoamentos, os quais serão introduzidos em versões subseqüentes. As experiências realizadas até aqui permitem esperar que RS venha a se constituir, de fato, numa ferramenta útil para o projeto e a construção de sistemas reativos.

REFERÊNCIAS BIBLIOGRÁFICAS

BERRY, G. *Real Time Programming: Special Purpose or General Purpose Languages*. Research Report 1065, INRIA, 1989.

HALWACKS, Nicolas. *Synchronous Programming of Reactive Systems*. Dordrecht: Klumer Academic Publisher, 1993.

LIBRELOTTO, Giovani; TOSCANI, Simão; MONTEIRO, Luís F. The Distribution of the RS Language of the MDX Environment, Simpósio Brasileiro de Linguagens de Programação, *Anais...*, 2000.

MATTOS, Júlio. Proposta de geração de código VHDL a partir da Linguagem RS. *Trabalho Individual 812*, CPGCC-UFRGS, Porto Alegre, 1999.

PRICE, Ana; TOSCANI, Simão S. *Implementação de Linguagens de Programação - Compiladores*. Porto Alegre : Sagra Luzzato, 2000.

TOSCANI, Simão S. RS: Uma Linguagem para a Programação de Núcleos Reactivos. *Tese de Doutorado*. Lisboa: Universidade Nova de Lisboa, 1993.

ASPECTOS FORMAIS SOBRE LINGUAGENS DE PROGRAMAÇÃO

Fabrízio de Royes Mello¹

Guilherme Silva de Lacerda²

RESUMO

O propósito deste artigo é demonstrar a estruturação e organização de uma Linguagem de Programação. São abordados os principais conceitos de compiladores e interpretadores usados no desenvolvimento de um protótipo de linguagem denominada ZEUS.

Palavras chaves: Linguagem, Programação, Compiladores, ZEUS.

ABSTRACT

The purpose of this paper is showing that the structure and organization of a Programming Language. It is also approached many concepts of the compilers and interpreters, used in the development of a language prototype, called ZEUS.

Keywords: Language, Programming, Compilers, ZEUS.

1 INTRODUÇÃO

Com o nascimento da computação aliado à necessidade de se poderem realizar determinadas tarefas, os cientistas começaram a criar as primeiras linguagens de programação. Basicamente, as linguagens eram capazes de processar e realizar operações matemáticas gigantescas. Sua utilização era muito complexa e trabalhosa, pois a programação se dava de forma manual através de válvulas e cartões.

A evolução tecnológica e a caracterização de se formalizar modelos matemáticos que se faziam necessários para inúmeras atividades foram fatores determinantes para a evolução desta área.

A grande importância da linguagem de programação, não somente pela sua representação perante a ciência da computação, mas como ferramenta de apoio fundamental, propiciou a grande evolução das tecnologias atuais.

O trabalho proposto visa ao estudo da estruturação das linguagens de programação, baseadas em teorias formais que validam o modelo.

2 LINGUAGEM DE PROGRAMAÇÃO

“Linguagem é todo sistema de signos que serve de meio de comunicação entre indivíduos e pode ser percebido pelos diversos órgãos dos sentidos, o que leva a distinguir-se uma Linguagem Visual, uma Linguagem Auditiva, uma Linguagem Tátil, etc., ou, ainda, outras mais complexas, constituídas, ao mesmo tempo, de elementos diversos”. [FER86]

2.1 Gramáticas e Linguagens

A linguagem de programação é um mecanismo fundamental de comunicação entre programadores e a máquina, e acima de tudo, a representação de seu intelecto aplicado para solução de problemas.

¹ Acadêmico do 4º semestre do Curso de Informática da Universidade da Região da Campanha – fmello@urcamp.tche.br

² Bacharel em Informática – guilherm@urcamp.tche.br

Em [LEV95], define-se linguagem como um conjunto de strings pertencentes a um alfabeto ou símbolos. Porém, essa linguagem é considerada formal quando puder ser representada através de um sistema com sustentação matemática [CAM96].

A linguagem pode ser definida formalmente na figura 1, onde L é descrito como uma linguagem, contendo um conjunto \mathbf{a} de símbolos e \mathbf{b} de regras de construção.

$$L \hat{=} \{ \mathbf{a}, \mathbf{b} \}$$

Figura 1: Definição formal de linguagem

Uma linguagem é considerada infinita, passível de ambigüidades. Deve-se, portanto, restringir a sua definição para uma *gramática*. Em [CAM96], gramática é definida como um sistema de geração de linguagem, descrevendo um subconjunto de sentenças que fazem parte da linguagem. Também em [CAM96], é descrito um dispositivo formal de definição de linguagem que representa uma linguagem infinita através de uma gramática finita.

$$\text{Linguagem (Infinita)} \text{ \textcircled{R} Gramática (Finita)}$$

Figura 2: Dispositivo de definição de gramática

Conforme [RAN99], a gramática é definida como uma quádrupla $G=(\Sigma, N, P, S)$, onde se tem a gramática \mathbf{G} definida para uma linguagem em um alfabeto \mathbf{S} de símbolos terminais, composta por elementos \mathbf{N} de símbolos não-terminais; um conjunto \mathbf{P} de regras finitas de produção de sentenças; e, finalmente, \mathbf{S} símbolo inicial da gramática. Este trabalho não visa à discussão sobre assunto de gramática, porém, para um maior entendimento, pode-se consultar [HOP79] e [MEN97].

Deve-se utilizar, para a codificação de gramática, alguma notação formal que possa validá-la. Geralmente, utiliza-se a *BNF (Backus Naur Form)* para especificar, formalmente, a gramática de uma linguagem de programação [LEV95].

Após a definição da gramática da linguagem, devem-se enfatizar dois aspectos muito importantes definidos em [RAN99], quais sejam: análise e síntese.

2.2 Análise

A fase de análise, conforme [CAM97], é responsável pela quebra do programa fonte em partes componentes do processo, criando uma representação intermediária do programa. Nesta fase, o programa tradutor percorre o programa fonte, capturando e armazenando informações relevantes para o seu correto funcionamento.

Em [AHO86], o processo de análise é dividido em três fases:

- *Análise Linear ou Léxica (Scanning)*: para se desenvolver a análise léxica, o conhecimento é baseado na teoria de linguagens regulares e autômatos finitos [RAN99]. Basicamente, a análise léxica identifica os *lexemas*³ agrupando-os em *tokens*⁴. Outra função do analisador léxico é retirar a formatação do programa fonte, eliminando os espaços em brancos e comentários.
- *Análise Hierárquica ou Sintática (Parsing)*: responsável pelo reconhecimento da estrutura global do programa, identificando comandos, atribuições, expressões, entre outros [RAN99]. Realiza a verificação baseada nas regras de construção da linguagem, resultando em um grafo orientado, denominado *árvore sintática*. Em [AHO86], possui

³ *Lexemas* são símbolos terminais de uma linguagem

⁴ *Tokens* são classes que definem os componentes de uma linguagem

duas funções: (1) checar os *tokens* gerados pela análise léxica e; (2) dispor os *tokens* em uma estrutura de árvore, usadas pelas outras fases do tradutor.

- *Análise Semântica*: A aplicação da análise semântica é determinar o tipo de resultados intermediários, verificando a integridade e significado dos argumentos que pertencem à linguagem. Trata dos aspectos sensíveis ao contexto da sintaxe da linguagem. Em [AHO86], considera-se que a análise semântica pode ser feita durante a fase de análise sintática, na fase de geração de código intermediário ou na fase final de geração de código.

2.3 Síntese

Assim como a fase de análise, a fase de síntese é muito importante. Tem por objetivo unir as partes decompostas nas fases anteriores, dando origem a uma nova estrutura. A fase de síntese, segundo [AHO86, CAM97], possui três estágios:

- *Geração de Código Intermediário*: este processo é iniciado após a análise sintática e semântica. Consiste em percorrer a árvore em profundidade, gerando quádruplas ou triplas para cada uma das construções que estão na árvore [CAM97]. Durante o percurso, são geradas variáveis temporárias para cálculo de expressões que aparecem na árvore, bem como a utilização de comandos incondicionais e de desvio condicional nos quais são testados para determinar a ocorrência de desvios ou não [AHO86].
- *Otimização de Código*: relaciona-se com a eficiência de execução do programa, relacionando aspectos de performance e memória necessários [CAM97]. Existem alguns problemas conhecidos como os citados em [AHO86], relacionados com a otimização de código: (1) não existe algoritmo com a melhor otimização, somente garantindo que o código otimizado é melhor que o anterior; e (2) são algoritmos muito complexos, baseados em heurísticas. Com a otimização, os tradutores meramente tentam produzir um programa objeto melhor do que o produzido sem "otimização". As principais otimizações são realizadas em *loops* e decisões.
- *Geração de Código*: é a fase final do processo de tradução. Após a passagem por todos os estágios anteriores, isso garante que a fase de geração de código possa prosseguir, livre de erros [AHO86]. Os requisitos impostos de geração de código são muito severos, garantindo que o que foi escrito no programa fonte tenha realmente a saída esperada.

3 TRADUTORES

Em [RES99], define-se **Tradutor** como um programa responsável por traduzir programas escritos em uma linguagem de programação para uma outra linguagem de programação geralmente de menor nível. A principal função de um tradutor é mapear um comportamento esperado de programas em uma linguagem implementada, expresso pelo programa fonte, para uma linguagem alvo, expresso pelo programa destino.

O tradutor mais conhecido é o **parse generator** [LEE89]. Até 1975, escrever um compilador era um processo complexo, consumindo muito tempo de desenvolvimento. Partindo desse problema, *Lesk* e *Johnson* publicaram artigos sobre o Lex e o Yacc [LES75, JOH75]. Esses utilitários simplificaram muito a tarefa de construir um compilador [NIE98]. A figura 3 mostra o processo de tradução de um programa fonte.

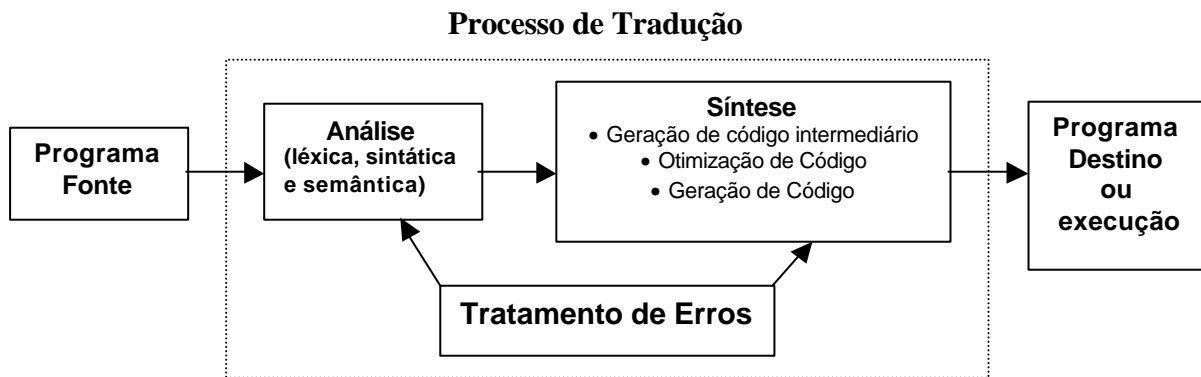


Figura 3: Processo de tradução

A seguir, são definidos dois tipos de programas tradutores.

3.1 Compiladores

Os compiladores são definidos em [AHO86,CAM97,RAN99 e REZ99] como sendo programas responsáveis pela conversão de um programa escrito em uma determinada linguagem de programação em um programa destino. O processo de compilação é baseado em uma total análise do programa fonte, gerando o resultado somente se não houver erros. Portanto, esse processo de tradução é mais lento, porém se ganhará em performance de execução.

Em [REZ99], pode-se definir uma classificação de compiladores, descritas a seguir:

- *Código de máquina*: binário absoluto, carregável, gerado em um passo único.
- *Código de máquina e Rotinas de suporte*: binário realocável, endereços relativos, gerado em dois passos (análise sintática; análise semântica e geração de código).
- *Código de máquina virtual*: simbólico, baseado em múltiplos passos.

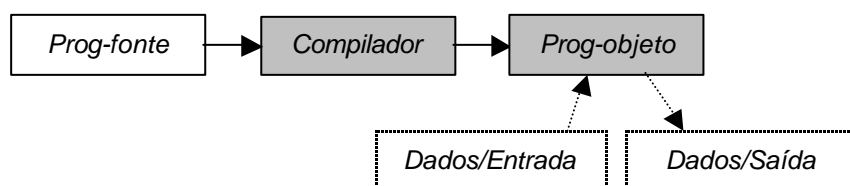


Figura 4: Processo de compilação

3.2 Interpretadores

Segundo [REZ99], os interpretadores são programas responsáveis pela tradução e execução simultânea do programa fonte. É um processo mais rápido para tradução, pois é traduzida instrução por instrução sendo executada paralelamente, porém ficando comprometido todo o tratamento de erros. Outra característica importante dos interpretadores é que o resultado é apenas a execução do programa, sem gerar um programa destino.

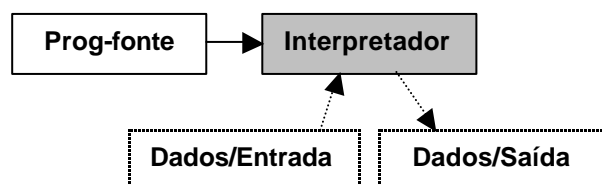


Figura 5: Processo de interpretação

4 ZEUS – UM PROTÓTIPO DE LINGUAGEM DE PROGRAMAÇÃO

Para exemplificar os conceitos e técnicas abordados ao longo deste trabalho, procurou-se desenvolver um protótipo de linguagem de programação com propósitos acadêmicos, objetivando o desenvolvimento da lógica de programação.

Seu fundamento está baseado em uma fusão da Linguagem *Pascal* com a Linguagem *Basic*. A linguagem ZEUS⁵ é muito simples e de fácil aprendizado, proporcionando a identificação de todo o fluxo lógico estruturado, necessário para resolução de problemas e execução de tarefas. Possui mecanismos de “*ignore-case*”, não importando a escrita de comandos. Outra peculiaridade da linguagem é a referência de variáveis, sendo fracamente tipificada.

Sua gramática foi definida utilizando a notação BNF [LEV95], como pode ser visto na Figura 6.

```
...  
Comando ::=  
    Atribuicao | Repetitivo | Vazio | Condicional | Entrada | Saida  
...  
Condicional ::=  
    Se Expressao Entao Sequencia_de_Comandos  
    [ Senao Sequencia_de_Comandos ] Fim_Se  
...
```

Figura 6: Trecho da notação BNF da Linguagem Zeus

A Figura 7 apresenta o Autômato Finito, parcial, que valida a Linguagem.

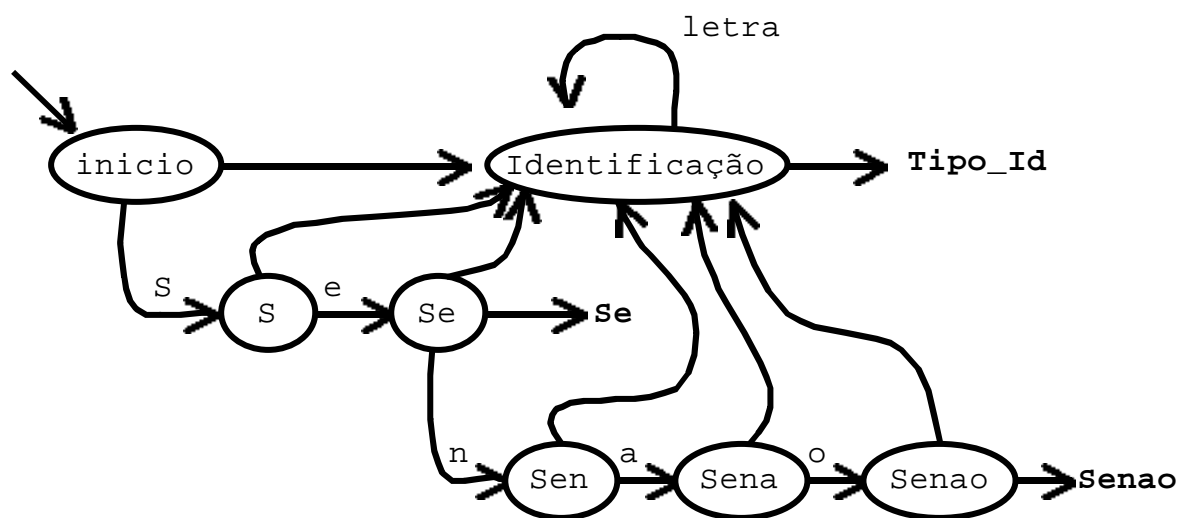


Figura 7: Autômato finito que gera os tokens baseados na definição da figura 6

O Zeus foi totalmente escrito em Linguagem C, garantindo sua portabilidade para diferentes plataformas como AIX, WIN9x, GNU-Linux, SCO Unix, entre outros.

⁵ Mais informações sobre a Linguagem ZEUS podem ser obtidas em:
<http://www.urcamp.tche.br/~guilherm/zeus>

5 CONSIDERAÇÕES FINAIS

A compreensão de como se estrutura uma Linguagem de Programação proporciona um melhor entendimento de diversas teorias computacionais. Foi possível desenvolver um trabalho que abordasse os principais conceitos de Linguagens Formais e Autômatos bem como Tradutores de Linguagens.

A contribuição deste estudo é no sentido de melhor compreender as diversas tecnologias atuais, e até mesmo futuras. Para o desenvolvimento de ferramentas CASE e geradores de aplicativos, por exemplo, é fundamental ter-se conhecimento pleno destas teorias. Pode-se, ainda, ir ao encontro de estudos referentes a Processamento de Linguagem Natural, a forma de representação matemática de algoritmos, heurísticas, entre outros.

REFERÊNCIAS BIBLIOGRÁFICAS

- [AHO86] AHO, Alfred V. SETHI, Ravi. ULLMAN, Jeffrey. Compilers: Principles, Techniques, and Tools. Stanford: Addison-Wesley, 1986.
- [CAM96] CAMPANI, Carlos. Linguagens Formais. Notas de Aula. Pelotas: UFPel, 1996.
- [CAM97] CAMPANI, Carlos. Compiladores. Notas de Aula. Pelotas: UFPel, 1997.
- [FER96] FERREIRA, Aurélio Buarque de Holanda. Novo Dicionário da Língua Portuguesa. 2. ed. Rio de Janeiro: Nova Fronteira S.A., 1986.
- [HOP79] HOPCROFT, John. ULLMAN, Jeffrey. Introduction to Automata Theory, Languages and Computation. Stanford: Addison-Wesley, 1979.
- [JOH75] JOHNSON, Stephen C. Yacc: Yet Another Compiler Compiler. Technical Report 32. New Jersey: Bell Laboratories, 1975. Acessado em Setembro de 2000. Disponível por WWW em <<http://epaperpress.com>>
- [LEE89] LEE, Peter. Tools for Practical Compiler Generation. Cambridge: The MIT Press, 1989.
- [LES75] LESK, M. E. SCHIMIDT, E. Lex – A Lexical Analyzer Generator. Technical Report 39. New Jersey: Bell Laboratories, 1975. Acessado em Setembro de 2000. Disponível por WWW em <<http://epaperpress.com>>
- [LEV95] LEVINE, John. MASON, Tony. BROWN, Doug. Lex & Yacc. 2. ed. USA: O'Reilly & Associates, 1995.
- [MEN97] MENEZES, Paulo F. B. Linguagens Formais e Autômatos. Porto Alegre: Sagra Luzzato, 1997.
- [NIE98] NIEMANN, Thomas. A Guide to Lex & Yacc. Portland, Oregon. Acessado em Setembro de 2000. Disponível por WWW em <http://epaperpress.com/y_man.html>
- [RAN99] RANGEL, José Lucas. Compiladores. Notas de Aula. Rio de Janeiro: PUC-Rio, 1999.
- [REZ99] REZENDE, Pedro. Tradutores, compiladores e interpretadores. Notas de Aula. Acessado em novembro de 1999. Disponível por WWW em <<http://www.cic.unb.br/docentes/pedro/>>

UM ESTUDO SOBRE A SOLUÇÃO PARALELA DE SISTEMAS DE EQUAÇÕES LINEARES UTILIZANDO A TÉCNICA “*BULK SYNCHRONOUS PARALLEL*”

Elza Marisa P. de Figueiredo Chagas¹

RESUMO

É objeto deste trabalho avaliar as potencialidades da técnica de programação paralela chamada “*Bulk Synchronous Parallel*” (BSP) na solução paralela de sistemas de equações lineares usando o método do Gradiente conjugado, através de uma análise comparativa com relação à outra técnica usual de programação paralela, por troca de mensagens, utilizando a biblioteca “*Message Passing Interface*” (MPI).

Palavras Chaves: Programação Paralela, BSP, Sistemas de Equações Lineares, Gradiente Conjugado, MPI.

ABSTRACT

This work is an investigation on the capabilities presented by the “Bulk Synchronous Parallel” (BSP) parallel programming on its application to the solution of systems of linear equations using the Conjugate Gradients iterative method. The BSP technique is compared in terms of ease of use and efficiency with the message-passing parallel programming technique, as expressed by the “Message Passing Interface” (MPI).

Keywords: Programming Parallel, BSP, Systems of Linear Equations, Conjugate Gradients, MPI.

1. INTRODUÇÃO

Atualmente, a ciência da computação tem visto uma explosão de interesses no campo da computação paralela. Do ponto de vista teórico, isso tem fornecido uma série de mudanças de problemas com novos grupos de regras para o desenvolvimento e análise de algoritmos.

Além do mais, o campo da matemática numérica, tais como soluções iterativas de sistemas não singulares de N equações lineares da forma $Ax = b$, tem recebido importante atenção por parte de cientistas e engenheiros, pois a solução de tais sistemas tem sido utilizada na resolução de muitos problemas complexos, incluindo programação linear, análise de dinâmica de fluídos, modelagem climática entre outros.

Como tais sistemas são usualmente grandes e esparsos e precisam de algoritmos sofisticados para resolvê-los, há uma grande necessidade de utilizarmos computadores paralelos. Entretanto, sabemos que as técnicas antigas de desenvolvimento e validação de algoritmos com modelos e dados simples – computação seqüencial – não são válidas para estes tipos de problemas e estão longe de representar situações do mundo real.

Assim, precisamos de modelos e quantias substanciais de dados para validar os algoritmos. Com isso, este processo necessita de grandes recursos computacionais – os supercomputadores - ou seja, máquinas com mais de um processador onde as tarefas que não colidem são executadas em paralelo, reduzindo, com isso, o tempo de execução dos algoritmos.

¹ MS. em Matemática Aplicada com ênfase em matemática computacional pela UFRGS, professora dos departamentos de Informática e Matemática das Faculdades Integradas de Palmas. E-mail: quissala@bol.com.br.

Entretanto, essas execuções paralelas utilizam algoritmos e técnicas especiais que trabalham com a programação paralela. Infelizmente, há um grande problema neste tipo de programação, ou seja, não há, por parte dos projetistas, um padrão de aceitação de como os computadores ditos paralelos poderiam controlar esta portabilidade.

Há pesquisadores que têm usado coleções de *wokstations* conectadas por LAN, simulando com isto um computador paralelo simples. Conforme escrito em [GEI 94], o processo de clusterização de *wokstations* dentro de computadores virtuais é extremamente fácil desde o aparecimento padrão de ambientes de passagem de mensagens tais como “*Message Passing Interface*” (MPI) e “*Bulk Synchronous Parallel*” (BSP). Com esses ambientes, a portabilidade foi alcançada. No entanto, a programação paralela por troca de mensagens é tediosa e suscetível a erros de execução do programa.

Sendo assim, podemos observar que é necessário um modelo abstrato de computador paralelo, escondendo os detalhes físicos de arquiteturas particulares de programas de forma a torná-los portáteis. Outra característica desejável deste modelo seria a predicabilidade, ou seja, a habilidade de se efetuar uma análise precisa de como programas portáteis irão rodar em diferentes arquiteturas reais. O modelo conhecido como BSP fornece justamente tal modelo.

Este artigo discorre sobre o trabalho de dissertação defendido em abril do ano 2000 e seu objetivo principal é apresentar algoritmos, utilizando o modelo BSP, que resolva (ou pelo menos alcance uma dada tolerância) sistemas de equações lineares esparsos utilizando o método iterativo de Gradiente Conjugado.

2. COMPUTADORES PARALELOS

Atualmente, os problemas que nos cercam são considerados cada vez mais complexos e a necessidade de resolvê-los de forma rápida torna-se cada vez mais imprescindível. Portanto, em termos de processamento, há três maneiras principais de diminuir o tempo de execução de programas, aumentando com isso a produtividade:

- usar processadores² que executam determinada tarefa de forma mais rápida;
- projetar não só um algoritmo correto, mas um algoritmo que seja considerado eficiente, ou seja, que utilize de forma racional os recursos de máquina, principalmente acessos a disco;
- finalmente, usar processamento paralelo.

O termo “computação paralela” refere-se, de forma geral, ao uso de vários recursos computacionais cooperando para a execução de determinada tarefa. Assim, a operação de caixas bancárias automáticas se enquadra neste tipo de computação.

Segundo [MEN 91], um computador paralelo pode ser definido como sendo uma coleção de elementos de processo que cooperam para resolver um determinado problema. Ele é formado por um conjunto de processadores que podem trabalhar conjuntamente para resolver problemas de cooperação computacional. Esta definição é comumente usada quando estamos tratando supercomputadores paralelos que têm centenas ou milhares de processadores, cadeias de *workstations*, *workstation* com vários processadores e sistemas embutidos.

Há muitas razões para que o usuário utilize programação paralela. Algumas destas razões são destacadas abaixo:

- aumento de performance, ou seja, as máquinas paralelas podem realizar mais trabalho do que as máquinas seriais;

² Processadores são máquinas físicas na qual o processo executa.

- aumento na utilidade, ou seja, as máquinas paralelas utilizam de forma mais útil os seus recursos do que máquinas seriais.

3. MODELO DE PROGRAMAÇÃO PARALELA

Para criarmos um programa paralelo, precisamos, num primeiro momento, identificar as tarefas que podem ser executadas em paralelo. Logo após, devemos dividir as tarefas e dados para todos os processadores. Num terceiro momento, devem ser realizados gerenciamento de dados, de comunicação e de sincronização. Por último, não devemos nos esquecer dos aspectos relacionados com a entrada e saída de dados, mensagens, etc.

Quanto aos modelos de programação paralela, em [SON 94], encontramos alguns deles. O primeiro trabalha sob um prisma de passagem (troca) de mensagens, ou seja, programas que enviam mensagens, como *softwares* baseados em tarefas e canais, criam várias tarefas, onde cada tarefa encapsula dados locais e é identificada por um nome único. Assim, tarefas interagem com as demais enviando mensagens através do nome.

Este modelo não impede a criação dinâmica de tarefas, a execução de várias tarefas num processador ou a execução de diferentes programas utilizando diferentes tarefas. Porém, na prática, a maioria destes modelos criam um número fixo de tarefas idênticas que iniciam um programa e não permitem a criação ou destruição de tarefas durante a execução deste programa. Na verdade, estes sistemas implementam um único programa com vários dados, pois cada tarefa executa o mesmo programa, mas opera em dados diferentes.

Outro modelo de programação paralela comumente usado, paralelismo de dados, utiliza conceitos baseados na concorrência, onde uma mesma operação usa um conjunto diferente de estrutura de dados. Neste tipo de modelo, o programador deverá informar como os dados serão divididos entre as tarefas.

O último modelo abstrato abordado por [SON 94] tem como característica principal o compartilhamento da memória. Neste modelo, parte das tarefas terão um espaço comum de memória onde realizaram leituras e escritas que serão executadas de forma assíncrona. Podem ser usados vários mecanismos, tais como “fechaduras” e “semáforos”, para controlar o acesso à memória compartilhada.

Uma vantagem deste modelo, do ponto de vista do programador, é que não há necessidade de especificar a comunicação de dados. Com isso, esse modelo pode simplificar o desenvolvimento de programas paralelos, mas pode dificultar a administração da localização e a criação de programas determinísticos.

4. GRADIENTE CONJUGADO

Em qualquer problema concreto, é necessário não apenas saber que uma solução existe, mas também chegar a essa solução em termos numéricos. Os problemas que aparecem na prática, no entanto, envolvem tantas variáveis ou são tão difíceis que se tornam praticamente impossível de serem resolvidos sem o uso de recursos como calculadora, computadores, etc.

Há a necessidade, então, de desenvolver processos numéricos que possam ser usados em computação eletrônica. Esses processos podem ser divididos em dois grupos: processos exatos e processos iterativos.

Por processo exato entendemos um processo através do qual se obtém a solução numérica de um problema por meio de um número finito de operações elementares. O método de resolução de sistemas lineares por redução da matriz é um exemplo de processo numérico exato que pode ser programado e usado em computadores.

Um processo iterativo de resolução de um problema é um processo em que se obtém uma seqüência de soluções aproximadas do problema, tal que cada termo da seqüência é obtida

a partir das seqüências anteriores por um processo bem determinado e que, num certo sentido, está “cada vez mais perto” da solução correta do problema.

A escolha de um ou outro processo vai depender de uma série de fatores tais como: precisão desejada dos resultados³, capacidade de memória do computador empregado e tempo despendido em computação.

O Gradiente Conjugado (CG) é um dos métodos iterativos mais populares para resolver grandes sistemas de \mathbf{N} equações lineares,

$$\mathbf{Ax} = \mathbf{b} \quad (1.1)$$

onde A é uma matriz conhecida, \mathbf{b} é um vetor conhecido e \mathbf{x} é um vetor desconhecido. A matriz A é do tipo quadrada, positiva definida ou positiva indefinida. Estes sistemas surgem em muitas conexões com aplicações importantes, tais como diferenças finitas e elementos finitos, ou nos métodos para solucionar equações diferenciais, para análise estrutural e análise de circuitos.

O CG é simplesmente o método das direções conjugadas onde as direções de busca são construídas pela conjugação dos resíduos, isto é, setando $\mathbf{u}_i = \mathbf{r}_i$. Em [CHA 00], é apresentado o desenvolvimento detalhado das equações que estão sendo utilizadas para o desenvolvimento do método CG.

As equações que governam o método CG são expressas em termos de três operações básicas: produtos internos entre dois vetores, produto matriz-vetor e acumulações de vetores. As duas primeiras envolvem comunicação entre os processadores, ao passo que a última pode ser feita de forma independente.

Assim sendo, utilizamos o algoritmo (1.1), mostrado abaixo, para a implementação paralela do método do CG.

```

procedure CG_par(in:A,b,x,e i_max; out:x)
  x(0) = x, d(-1) = r(0) = b - Ax(0)
  r(0) = r(0)Tr(0), w(0) = Ad(0), x(0) = d(0)Tw(0)
  for i = 0,1,...,i_max do
    a(i-1) = r(i-1)/x(i-1); x(i) = x(i-1) + a(i-1)d(i-1); r(i) = r(i-1) - a(i-1)w(i-1)
    if ||r(i+1)|| < e||b|| then stop endif
    s(i) = Ar(i); r(i) = r(i)Tr(i); d(i) = r(i)Ts(i); b(i) = r(i)/r(i-1)
    d(i) = r(i) + b(i)d(i-1); w(i) = s(i) + b(i)w(i-1); x(i) = d(i) - b(i)2x(i-1)
  endfor
  x = x(i)

```

Algoritmo 1.1 – Método do Gradiente Conjugado adequado para computação paralela.

O algoritmo (1.1) necessita, no entanto, de um vetor a mais (\mathbf{s}_i) e realiza uma acumulação vetorial a mais; mas esse é um pequeno preço a pagar, se considerarmos que

cada processador necessitará armazenar, no máximo, $\frac{en}{ep}$ elementos a mais (devido ao

vetor \mathbf{s}_i) e realizará uma operação de acumulação que é independente entre os processadores. O método CG está implementado no pacote PIM seguindo o algoritmo (1.1).

³ Aqui devemos ressaltar que a expressão exato não quer dizer que se encontre a resposta correta, sem erro algum. Este erro quase sempre aparece na prática, seja através de erros de arredondamento, seja pelas aproximações inerentes ao processo utilizado.

5. LINGUAGEM DE PROGRAMAÇÃO BSP

O modelo de programação paralela, chamado de “*Bulk Synchronous Parallel*” (BSP), promete facilitar a programação paralela. Desenvolvido pela equipe de pesquisadores das universidades de *Oxford* e de *Harvard*, ele oferece um mecanismo simples de sincronização. Além disso, possui um nível de abstração que faz com que os programas portáteis alcancem arquiteturas paralelas.

Outra facilidade oferecida pelo modelo BSP é o fornecimento de uma máquina abstrata cujas operações são de alto nível. Com isso, nestes modelos, os projetistas não necessitam organizar certos detalhes que outros sistemas não conseguem resolver sozinhos, tais como mapeamento de processos para processadores e controle das ações de comunicação individual de cada processador.

Com relação às vantagens oferecidas por este modelo, podemos citar as seguintes:

- oferta de *softwares* portáteis e eficientes em uma grande variedade de arquiteturas;
- implementação de bibliotecas de *software* livremente disponível;
- desempenho previsível baseado nos quatro parâmetros de custo de hardware abordados em [CHA 00];
- desenvolvimento de algoritmos, com análise de desempenho real para um grande alcance de multiprocessadores e redes *workstations*;
- forte fundamentação teórica.

O modelo BSP assume um conjunto de pares de processador e memória. Esses pares são conectados por uma rede de comunicação que, eficientemente, suporta um espaço de memória global e um mecanismo de barramento sincronizado dos processos. A execução de um programa BSP é realizada em fases e toda a comunicação global acontece entre essas fases.

Cada fase é chamada de superpasso, o qual consiste num número de linhas que rodam em paralelo. Essas linhas realizam somente comunicações locais até que elas alcancem um ponto ou uma barra de sincronização. Numa barra, todas as linhas devem esperar até que o último comando ou instrução seja lido, dentro de um tempo total de comunicação. Com isso, esse modelo separa completamente a sincronização, e assim a sincronização de mensagens individuais passa a não ser mais preocupação do projetista, mas do sistema em questão.

É importante ressaltar que o modelo **BSP** não se preocupa se um determinado computador paralelo implementa barramento sincronizado via *hardware* ou se ele implementa este barramento via *software*, pois isto afetaria somente a performance absoluta do sistema. Como exemplo disto, podemos citar o supercomputador paralelo Cray modelo T3D. Ele suporta barramento sincronizado via *hardware*, pois fornece a cada nodo um processador e uma memória com um registrador especial de barramento.

Além disso, o programador não se preocupa em conhecer o mecanismo usado para a comunicação. Como o barramento de sincronização sempre ocorre, logicamente, após operações coletivas de comunicação e antes de um período de execução local, então podemos utilizar este conhecimento para dobrar o barramento de sincronização com uma comunicação coletiva, reduzindo o custo efetivo desses mecanismos. O método exato de realizar esta implementação depende do objetivo envolvido na arquitetura.

Outro aspecto importante suportado pelo **BSP**, cujo objetivo é garantir a portabilidade de programas, é a maneira como ele fornece um modelo de custo analítico para avaliação do desempenho dos algoritmos paralelos.

Este ambiente de programação paralela é aplicado a todo tipo de máquina paralela: arquiteturas de memória distribuída, multiprocessadores de memória compartilhada e redes de *workstations*. Isso fornece um *framework* consistente dentro do qual desenvolvem-se *softwares* paralelos portáteis à arquiteturas paralelas escaláveis.

6. IMPLEMENTAÇÃO PARALELA DO CG USANDO BSP

O método CG é um dos métodos iterativos presentes no pacote *Parallel Iterative Methods* (PIM) [CUN95], o qual é utilizado em uma variedade de máquina paralelas, incluindo supercomputadores vetoriais (Cray Y-MP e NEC SX-3), computadores de memória distribuída (Intel Paragon, Thinking Machines CM-5), de memória distribuída com acesso global (Cray T3D, SGI Power Challenge, SGI Origin 200) e redes de *workstations*.

Escrito em Fortran 77, o PIM permite tal utilização por ter sido projetado de forma modular, encapsulando em três rotinas as operações produto matriz-vetor, soma de escalares e cálculo de normas de vetores, as quais exigem comunicação entre os processadores. Devido ao fato do PIM ser altamente modular, podemos concluir que ele também é altamente portátil.

Como demonstrado em [CHA 00], é fácil notar que podemos escrever uma rotina de produto matriz-vetor valendo-nos da operação de redução oferecida pela biblioteca do BSP – o BSPLib. Infelizmente, a implementação que temos disponível (BSPLib v. 1.4), instalada na máquina SGI Origin 200 (com dois processadores MIPS R10000 de 180 MHz e 128MB de memória RAM agregada), não funcionou de forma alguma, levando à falha da execução do programa. Dessa forma, fomos forçados a escrever uma rotina específica que realizasse operações de redução.

7. CONCLUSÕES

Nosso principal objetivo, neste trabalho, é avaliar as potencialidades do modelo de programação paralela BSP, através de uma análise comparativa com relação à utilização da biblioteca MPI. Para tanto, foi escolhido o método iterativo CG, já implementado no pacote PIM, como ferramenta para a solução de sistemas de equações lineares e como recurso para avaliação da biblioteca BSPLib versão 1.4.

Foram realizados testes com os programas SDIAG e SPOLY, presentes na distribuição do pacote PIM (versão 2.2), com as modificações necessárias ao uso da biblioteca BSP, incluindo as rotinas discutidas mais detalhadamente em [CHA 00]. Cabe ressaltar que esses programas utilizam pré-condicionadores – diagonal e polinomial, respectivamente – os quais são implementados em paralelo, no primeiro caso sem comunicação entre os processadores e, no segundo, utilizando a rotina de produto matriz-vetor apropriado.

Os testes foram realizados no computador paralelo SGI ORIGIN200, instalado no Laboratório Integrado de Computação Científica do Instituto de Matemática da UFRGS. Este computador é dotado de dois processadores MIPS R10000 de 180 MHz, os quais compartilham o acesso a uma memória de 128 *Mbytes*.

Para o caso denso, foram feitos testes utilizando sistemas de ordem $n=1000$, $n = 2000$ e $n = 4000$. O termo independente \mathbf{b} foi escolhido de forma que a solução seja o vetor

$(1,1,\dots,1)^T$ e as iterações foram realizadas até que o critério de parada $\frac{\|r_k\|_2}{\|b\|_2} < 10^{-10}$ fosse

satisfeito. Já para a solução da EDP, utilizou-se uma discretização em diferenças finitas com malhas de 100×100 e 200×200 células, resultando em sistemas de ordem $n = 10000$ e $n = 40000$; foi utilizado o mesmo critério de parada.

Analisando os resultados obtidos, podemos observar que o desempenho de ambas as implementações é sofrível, para os valores de N considerados, apesar de que o *speed-up*

crece com N . Por outro lado, percebe-se que a implementação em BSP é semelhante àquela em MPI.

Por outro lado, podemos observar ainda que o desempenho da versão em MPI é melhor do que em BSP. Percebe-se, aqui, no caso $n = 40000$, uma aceleração aceitável sendo oferecida pela versão em BSP, correspondendo a aproximadamente 56% de eficiência. Pudemos comprovar esses resultados utilizando a ferramenta *bspprof*, integrante da BSPLib e que fornece uma avaliação do desempenho de um programa.

Há, no entanto, uma certa variação no tempo de duração de cada troca de dados, a qual atribuímos à mudança na carga dos processadores ao longo da execução.

REFERÊNCIAS BIBLIOGRÁFICAS

- [CHA00] CHAGAS, E.M.P.F.. **Um estudo sobre a solução paralela de sistemas de equações lineares utilizando a técnica “Bulk synchronous parallel”**. Tese de Mestrado, Programa de Pós-Graduação em Matemática Aplicada. Universidade Federal do Rio Grande do Sul. Porto Alegre, 2000.
- [CUN95] DA CUNHA, R.D.; HOPKINS, T.R.. **The Parallel Iterative Methods (PIM) package for the solution of systems of linear equation on parallel computers**. *Applied Numerical Mathematics*, 19, pp. 33-50, 1995.
- [GEI94] GEIST, A.; BEGUELIN, A.; DONGARRA, J.; JIANG, W.; MANCHEK, R.; SUNDERAM, V.. **PVM: Parallel Virtual Machine – A User’s Guide and Tutorial for Networked Parallel Computing**. Cambridge, MA, 1994.
- [MEN91] MENDES, KIRNER. **Sistemas operacionais distribuídos**. Campus, São Paulo, 1991.
- [SON94] SONG, S.W.. **Computação maciçamente paralela: potencial e limitações**. *Anais do Colóquio – Questões Metodológicas em Ciências Cognitivas*. Setembro 1996. pp. 106-108.

ABORDAGENS SOBRE ESPECIFICAÇÃO DE HIPERDOCUMENTOS

Carlos Emilio Padilla Severo¹

RESUMO

As aplicações baseadas em hiperdocumentos utilizam tecnologias distintas e não possuem documentação específica sobre os métodos utilizados durante os seus desenvolvimentos. A elaboração de um hiperdocumento pode ser complexa, principalmente relacionada com a combinação de navegação controlada pelo usuário e a natureza dos dados multimídia. Outro aspecto é a manutenção da aplicação que se torna complicada. Face a estes problemas, surgiram métodos de projeto que oferecem ferramentas conceituais para a construção de abstrações sobre o modelo de dados hiperemídia. Sendo assim, este trabalho tem por finalidade abordar os problemas de elaboração de hiperdocumentos, apresentando métodos para especificação de aplicações.

PALAVRAS-CHAVES: especificação de hiperdocumentos, abordagens de especificação, OOHDM, RMM, HMT, HMBS/M.

ABSTRACT

The applications based on hyperdocuments employ different technologies and they don't have specific documentation about methods utilized during their developments. The elaboration of a hyperdocument can be complex, mainly related with the combination of user controlled navigation and the multimedia data features. Another aspect is the maintenance of the application that becomes complicated. In face to these problems, project methods that offer conceptual tools for the abstractions construction on the hypermedia data model. Being like this, this work has for purpose to approach the problems of hyperdocuments elaboration, showing methods for specification of applications.

KEYWORDS: hyperdocuments specification, approaches of specification, OOHDM, RMM, HMT and HMBS/M.

1. INTRODUÇÃO

Atualmente, observa-se que o uso da tecnologia de hiperdocumentos vem obtendo um expressivo crescimento. Sendo utilizados em uma grande variedade de propósitos, hiperdocumentos possibilitam uma estratégia poderosa para a interação com clientes, tornando-se assim uma poderosa ferramenta de apoio ao comércio eletrônico. O projeto de um hiperdocumento envolve uma equipe formada por autores, projetistas de conteúdo, programadores, músicos e outros profissionais [TUR 98]. Essa equipe deve estar capacitada para estruturar e organizar o domínio complexo que é uma aplicação baseada em hiperdocumentos, tornando a aplicação clara e acessível aos diferentes perfis de usuários.

O projeto deve seguir um modelo que possibilite o desenvolvimento de uma aplicação clara e de fácil operação por parte do usuário [GAR 93]. Além disso, deve ser fácil de ser mantida do ponto de vista dos autores. Tal como em aplicações convencionais, torna-se clara a necessidade de uma abordagem sistemática e formal para a definição da estrutura organizacional de uma aplicação baseada em hiperdocumentos. O uso de um modelo formal disciplina o autor a projetar a aplicação antes de desenvolver o conteúdo das páginas da aplicação [SCH 95].

¹ Professor do CCEI-URCAMP – Mestrando em Ciências da Computação – E-mail:emilio@urcamp.tche.br

Um bom método de projeto deve ser eficiente de se usar e deve produzir um hiperdocumento coerente que ajude o usuário a obter a informação desejada [IZA 95]. Para isso, o método de projeto deve tratar o domínio da aplicação e as conexões entre elas, como representá-las, manipulá-las e armazená-las. Além disso, um ponto importante a ser considerado é a dinâmica de interação com o usuário. Dessa forma, projetos de hiperdocumentos devem incorporar características específicas, diferentes dos métodos tradicionais da Engenharia de Software.

2. PROBLEMAS DE ESPECIFICAÇÃO DE HIPERDOCUMENTOS

Existem problemas associados ao processo de especificação de hiperdocumentos. Alguns estão ligados aos problemas típicos da Engenharia de Software. Todavia, existem outros problemas que são específicos da modelagem de aplicações hipermídia. Tais problemas são mencionados nesta seção. Posteriormente, serão apresentados métodos atuais que tratam os problemas de especificação de hiperdocumentos, onde serão verificadas suas principais características. Em seguida, o trabalho será finalizado com uma breve conclusão.

2.1. Especificação do domínio da aplicação

Os leitores de um hiperdocumento encontram dificuldades na compreensão do que está sendo informado. Isso deve-se ao fato de que o projetista, durante a especificação da aplicação, encontra dificuldades para a estruturação do hiperdocumento, pois muitas vezes não possui diretrizes para guiá-lo no processo de projeto e desenvolvimento de aplicações [THU 91]. Muitas aplicações hipermídia possuem um domínio muito complexo. Essa complexidade é verificada devido a exigências adicionais de consistência durante o projeto da aplicação [GAR 93]. Esses problemas são devidos a uma questão importante da Engenharia de Software que é a obtenção de modelos sólidos do domínio da aplicação para que possam ser mapeados, de forma simples e direta, para um modelo de projeto da aplicação.

2.2. Definição das estruturas de navegação

Os problemas de navegação têm sido amplamente discutidos na comunidade de desenvolvimento de aplicações hipermídia. Esses problemas se devem à fraqueza de mecanismos de especificação de estruturas navegacionais [NAN 93]. Dessa forma, grandes redes de hiperdocumentos são difíceis de serem gerenciadas por seus criadores, bem como são difíceis de serem compreendidas por seus usuários. Tendo em vista esses problemas, a definição de um modelo formal permite a abstração e a separação entre a estrutura e o conteúdo do hipertexto, possibilitando uma melhor compreensão da semântica de navegação e fornecendo uma interpretação consistente para propósitos de implementação. Durante certo tempo, a comunidade de hipertexto enfrentou o problema da desorientação propondo ferramentas ("browsers", mapas, etc.). Entretanto, um bom projeto de navegação pode auxiliar na superação de todos esses problemas [ROS 96].

2.3. Sincronização de elementos multimídia

A combinação da tecnologia de hipertexto com a multimídia cria novos problemas de especificação de aplicações. Um grande problema é a sincronização da mídia, que muitas vezes se perde entre as fases de projeto, sendo tratado como problema de navegação ou problema de interface. Em uma boa especificação esses problemas podem ser solucionados com apoio de boas primitivas de especificação, sendo necessário integrá-las ao método de projeto [ROS 96].

2.4. Manutenção da aplicação

A maioria dos problemas relacionados às aplicações hipermídia envolvem aspectos de especificação e projeto. Poucas referências são encontradas em relação à manutenção de aplicações. Esse aspecto é verificado devido a pouca idade da hipermídia. Muitas aplicações são desenvolvidas e comercializadas como pacotes fechados, no formato de um

CD-ROM, onde não existe a possibilidade de serem modificados de acordo com as necessidades do usuário. Contudo, a hipermídia está ganhando projeção, surgindo novos problemas associados a expansão da WWW (*World Wide Web*), onde aplicações baseadas na tecnologia hipermídia necessitam de manutenção. Somente agora os projetistas de aplicação estão começando a enfrentar estes problemas [ROS 96].

3. ABORDAGENS DE PROJETO DE HIPERDOCUMENTOS

O desenvolvedor, durante o projeto de uma aplicação hipermídia, enfrenta pelos menos duas categorias correlacionadas de problemas. A primeira são as tarefas globais de projeto, onde são identificadas todas as classes de informação, elementos e estruturas navegacionais da aplicação. Uma outra categoria são as tarefas locais, onde são colocados conteúdos nos nós de informação.

A autoria de um hiperdocumento pode ser classificada em duas dimensões que estão associadas às categorias mencionadas no parágrafo anterior. A autoria em ponto grande e autoria em ponto pequeno. A autoria em ponto grande está relacionada ao projeto global da aplicação envolvendo aspectos estruturais; preocupa-se mais com a modelagem do domínio da aplicação. Já a autoria em ponto pequeno trata do desenvolvimento do conteúdo dos nós de informação, mais dependente de ferramentas utilizadas para implementação. A autoria em ponto grande é a etapa onde se concentram o processo de especificação de requisitos e projeto do hiperdocumento. Essa é a principal preocupação dos métodos de projeto encontrados na literatura.

Os seguintes métodos serão descritos neste trabalho: OOHDM, RMM, HMT e HMBS/M, pois cercam muitas abordagens atuais para a especificação dos requisitos e projeto de hiperdocumentos, tais como entidade-relacionamento, orientação a objetos, diagramas de estados e redes de petri.

3.1. OOHDM (*Object-Oriented Hypermedia Design Model*)

O método OOHDM é uma abordagem de desenvolvimento de aplicações hipermídia baseada em modelos. O método compreende quatro etapas, onde o modelo da aplicação é construído e enriquecido a cada passo. Cada etapa é baseada na anterior, cujo estilo de desenvolvimento é interativo e incremental [SCH 95].

A modelagem conceitual é a primeira etapa do processo. Nesta etapa, o projetista define o modelo conceitual de domínio da aplicação através de mecanismos como classificação, composição, generalização e especialização. O produto final desta etapa é um modelo do domínio da aplicação envolvendo classes, sub-sistemas, relacionamentos e perspectivas de atributos.

O projeto navegacional é a próxima etapa. Nesta, o projetista estuda o perfil do usuário e as tarefas que o mesmo poderá executar na aplicação. A ênfase está nos aspectos cognitivos da aplicação hipermídia em relação às preferências do usuário. O mapeamento do modelo conceitual em objetos navegacionais gera o produto final desta etapa que é o conjunto de nós, elos, estruturas de acesso, contextos navegacionais e transformações navegacionais da aplicação.

Após o modelo conceitual e navegacional da aplicação, o projetista deverá modelar a interface. Esta etapa visa a obtenção dos objetos de interface perceptíveis durante a navegação para a descrição da interface navegacional. Com os mecanismos de mapeamento entre o modelo de navegação e os objetos perceptíveis será elaborada a interface, procedimentos de resposta a eventos externos e as transformações de interface.

A implementação da aplicação é obtida pelo mapeamento dos modelos de navegação e interface abstrata em objetos concretos avaliados em ambientes de implementação.

Exemplos de ferramentas para implementação de aplicações hipermídia são: *Hypercard*, *Toolbook*, *MacWeb*, *KMS*, *Guide*, *Microcosm*, entre outras.

3.2. RMM (Relationship Management Design Methodology)

O RMM é um método para projeto e desenvolvimento de aplicações hipermídia baseadas no gerenciamento de relacionamentos entre objetos de informação [IZA 95]. RMM é adequado para a especificação de aplicações que possuem um domínio regular e estruturado, através de classes, objetos e relacionamentos que podem ser estabelecidos entre estas classes. Exemplos de tais aplicações são catálogos de produtos e *front-ends* baseados em hipermídia para bancos de dados tradicionais. Como esses dados precisam de freqüentes atualizações, faz-se necessário adotar uma abordagem que automatize os processos de projeto e atualização dessas aplicações. Entretanto, em aplicações que não requerem constantes atualizações ou aquelas que possuem estruturas dinâmicas, RMM não é um método adequado para a especificação [IZA 95].

RMM utiliza como modelo de dados o RMDM (*Relationship Management Data Model*). O modelo de dados é um conjunto de objetos lógicos que fornecem uma abstração de uma porção do mundo real. O RMDM fornece uma linguagem para descrição dos objetos de informação e dos mecanismos de navegação em aplicações hipermídia. O RMDM possui uma linguagem que descreve os objetos de informação e os mecanismos navegacionais em uma aplicação hipermídia. Essa linguagem define as primitivas de domínio e primitivas de acesso.

O método RMM, como todo método de projeto de software, possui um ciclo completo de desenvolvimento. Esse ciclo envolve as etapas de estudo de viabilidade, análise de requisitos, projeto E-R, projeto de fatias, projeto navegacional, projeto da interface com o usuário, construção, teste e avaliação [IZA 95]. Em cada fase do método são gerados alguns produtos como resultado da especificação. Durante as fases iniciais são gerados os documentos de possibilidades, requisitos e definição de hardware. Através do documento de possibilidades será avaliada a necessidade, objetivos e análise de mercado para a futura aplicação, bem como um levantamento sobre mídias que serão utilizadas, canais de distribuição e custo/benefício.

Como o desenvolvimento de hiperdocumentos é diferenciado através da descrição das etapas envolvidas no projeto dos mecanismos de acesso à informação, Izakowitz [IZA 95] enfoca, principalmente, esses aspectos. Sendo assim, são enfatizados os seguintes passos: projeto E-R, projeto de fatias e projeto navegacional. Cada uma dessas fases gera, também, um produto. A fase de projeto E-R gera um diagrama entidade relacionamento que modela o domínio da aplicação, através das entidades e seus respectivos relacionamentos. Na fase de projeto de fatias o diagrama E-R é acrescentado por um modelo que especifica as partes de uma informação – denominado diagrama E-R+. Já a fase de projeto navegacional especifica o modelo de navegação da aplicação, definindo as estruturas de acesso às informações. Ao final dessa fase será gerado o diagrama RMDM. O processo de projeto não é linear, existe um *feedback* durante as fases de projeto

3.3. HMT (Hypermedia Modeling Technique)

O método HMT leva em consideração questões como: divisão do domínio da aplicação em nós de informação; tratamento das conexões entre os nós de informação; e a interação do usuário com a aplicação. Para tratar esses aspectos, HMT é subdividido em quatro etapas que utilizam modelos apropriados.

O modelo de objetos HMT descreve os objetos de domínio através de diagramas de objetos (ferramenta de descrição dos modelos), de acordo com as definições do OMT (*Object Modeling Technique*). Os conceitos de classes, atributos, operações, associações, cardinalidade, generalização e agregação são utilizados. A utilização do modelo de objetos

na especificação de hiperdocumentos tem como principal vantagem o tratamento explícito de relacionamentos. Os relacionamentos formam construções e não são implícitos em atributos de classes como em outros modelos.

O modelo de *HyperObject* é um refinamento do modelo de objetos. Esse modelo inclui características que não são modeladas no modelo de objetos, como: definição de novas classes e associações que definem caminhos de navegação desejados, identificação de mídias que serão utilizadas na aplicação e a identificação de classes abstratas. O modelo de *HyperObject* pode ser utilizado como uma ferramenta para complementar decisões de projeto em aplicações existentes que possuem um modelo de domínio já definido.

O modelo de navegação descreve os elos e as estruturas de acesso da aplicação. As associações determinam os caminhos de acesso entre os objetos do modelo e são definidas através de entidades conceituais utilizadas na modelagem conceitual.

A última etapa é o modelo de interface, onde é realizada a descrição de como o usuário perceberá os objetos de interface durante a navegação [NEM 98].

3.4. HMBS/M – An Object Oriented Method for Hypermedia Design

Assim como em outros métodos de projeto de aplicações hipermídia, HMBS/M possui uma seqüência de estágios de projeto. Começando pela modelagem conceitual, onde é elaborado um modelo do domínio da aplicação. Seguida por uma modelagem navegacional, que define os possíveis caminhos de navegação da aplicação baseados no perfil e tarefas que o usuário irá realizar durante a utilização do hiperdocumento. Em seguida, vem a fase de projeto da interface com o usuário com a criação de interfaces baseadas no perfil do usuário. As fases de implementação e teste concluem o método. A estrutura organizacional e a dinâmica do comportamento navegacional do hiperdocumento são especificados através do modelo HMBS (Hyperdocument Model Based on Statecharts [TUR 98].

A análise do domínio da aplicação é a primeira etapa, onde será criada a representação do hiperdocumento. O modelo conceitual de HMBS/M é derivado do método *Fusion*, que utiliza o paradigma de projeto de software orientado a objetos. Este método descreve as classes pela utilização de mecanismos como agregação e especialização/generalização e, ainda, os relacionamentos entre as classes do domínio. Nesta etapa são criados dois modelos para a especificação do domínio da aplicação: o modelo de objetos e o modelo de fatias. O modelo de objetos define as informações de domínio mais relevantes (classes), através da identificação de seus atributos, bem como seus relacionamentos. No modelo de fatias (*slices*) são representadas partes de informação e seus relacionamentos. Esse modelo é semelhante ao adotado em RMM [IZA 95].

Durante a modelagem navegacional, um conjunto de contextos navegacionais e estruturas de acesso são definidas a partir dos modelos gerados na fase de modelagem conceitual. Para isso, leva-se em consideração o perfil do usuário e as tarefas que serão realizadas durante a navegação. Ao final dessa etapa, dois modelos serão produzidos, um modelo navegacional de tipos e um modelo navegacional de instâncias. O modelo navegacional de tipos descreve uma visão sobre o modelo conceitual de domínio da aplicação, levando-se em consideração os contextos navegacionais e estruturas de acesso identificadas a partir do modelo. No modelo navegacional de instâncias, as classes conceituais modeladas são instanciadas em objetos estruturais, que representam os estados de acordo com as semânticas do modelo HMBS (*Hyperdocument Model Based on Statecharts*).

A interface é definida através da especificação de canais de apresentação. Os canais de apresentação são dispositivos abstratos que determinam a aparência do hiperdocumento através da definição do aspecto visual dos elementos de interface, de acordo com as definições em [HAR 94].

O ambiente de autoria denominado HySCharts [TUR 99] interpreta e implementa aplicações especificadas através do método HMBS/M. HySCharts realiza o mapeamento dos modelos navegacionais e de interface em uma aplicação final. Outra utilidade para o HySCharts é a prototipação de hiperdocumentos que pode ser útil aos projetistas. Uma outra alternativa é a tradução da especificação HMBS/M em elementos de um determinado ambiente de implementação.

4. CONCLUSÃO

Este trabalho realizou um estudo sobre as principais abordagens de especificação de hiperdocumentos amplamente discutidas na literatura. Foram apresentadas as seguintes abordagens: OOHDM, RMM, HMT e HMBS/M. As principais características dessas abordagens foram analisadas, tais como: fases de projeto, primitivas de modelagem e separação entre estrutura e conteúdo. Para isso, foram expostas as características comuns desejadas para cada uma das abordagens de especificação de hiperdocumentos. Tais características visam tratar os problemas associados a especificação de aplicações baseadas em hipermídia.

As principais características das abordagens de especificação de hiperdocumentos relatadas neste trabalho são: especificação de domínio de aplicações através de mecanismos de abstração de alto nível, a fim de delimitar o problema; especificação da navegação, na forma de estruturas de acesso e contextos de navegação para auxiliar o usuário durante a operação da aplicação; especificação de um modelo de interface independente do ambiente de implementação, permitindo que a interface abstrata possa ser implementada em qualquer ferramenta; e, por fim, a implementação do hiperdocumento através do mapeamento das estruturas de navegação em objetos de interface concretos.

Cada característica deve ser implementada em fases distintas de projeto, onde cada problema possa ser tratado individualmente. A divisão do projeto em fases permite que o processo de desenvolvimento seja incremental, onde cada fase de projeto constrói um modelo baseado na fase anterior.

Como pôde ser observado, existem diversas abordagens de especificação de hiperdocumentos. As citadas neste trabalho permitem uma visão geral sobre o que se tem realizado em termos de pesquisa e desenvolvimento de modelos de apoio ao processo de desenvolvimento de aplicações baseadas em hiperdocumentos.

A utilização de uma abordagem de especificação de hiperdocumentos deverá se tornar uma prática usual pelos desenvolvedores, pois possibilita a organização e estruturação do processo de projeto de aplicações de forma evolutiva e natural, que vai desde a definição até a implementação da aplicação final. A especificação de grandes aplicações sem a utilização de um método formal pode ser um empreendimento pouco produtivo, confuso e gerar um produto ineficiente.

REFERÊNCIAS BIBLIOGRÁFICAS

- [CAR 98] CARVALHO, M. R. HMBS/M – An object oriented method for the design and development of hypermedia applications. M.Sc. Dissertation, ICMC-USP, São Carlos, 133p., 1998 (in portuguese).
- [GAR 93] GARZOTTO, F.; SCHWABE, D. e PAOLINI, P.: "HDM- A Model Based Approach to Hypermedia Application Design", ACM Transaction on Information Systems, Vol. 11, #1, Jan. 1993, pág. 1-26.
- [HAR 94] HARDMAN, L.; BULTERMAN, D. e VAN ROSSUM, G.: "The Amsterdam Hypermedia Model: Adding Time and Context to the Dexter Model", Comm ACM. Feb. 94, vol. 37, N.2, pág 50-63.

- [IZA 95] IZAKOWITZ, T.; STOHR, E. e BALASUBRAMANIAM P.: "RMM: A methodology for structured hypermedia design". Comm of the ACM, October 1995, pág 34-44.
- [NAN 93] NANARD, J. e NANARD, M.: "*Should Anchors be typed too? An experiment with MacWeb*". Proceedings of the Fifth ACM Conference on Hypertext (Hyprtext'93), Seattle, 1993, pág 51-62.
- [NEM 98] NEMETZ, F. *Hypermedia Modeling Technique: An object-oriented design model for hypermedia applications*. Disponível em <http://www.cutsys.com/CHI97/Nemetz.html> (última consulta 25/03/2000).
- [ROS 96] ROSSI, G. *An Object-Oriented method for the design of hypermedia applications*. D. Sc. Thesis, Informatics Department, PUC, Rio de Janeiro, 1996 (In Portuguese).
- [SCH 95] SCHWABE, D. e ROSSI, G.: *The Object-Oriented Hypermedia Design Model*. Communications of the ACM, August 1995, pág 45-46.
- [THU 91] THURING, M., HAAKE, J.M. e HANNEMANN, J.: "*What's ELIZA doing in the Chinese room? Incoherent hyperdocuments and how to avoid them*". In Proceedings of Hypertext'91. ACM Press, New York 1991, pag 161-177.
- [TUR 98] TURINE, M. A. S. *HMBS: A statechart-based method for the formal specification of hyperdocuments*. D.Sc. Thesis. IFSC-USP, São Carlos, 194p. 1998 (in Portuguese).
- [TUR 99] TURINE, M. A. S., OLIVEIRA, M. C. F. e MASIERO, P. C. *Hyscharts: a hyperdocument authoring and browsing environment based on statecharts*. Multimedia Tools and Applications, Kluwer Academic Publishers, 1999. (to apper).

TAMANHO DA ORGANIZAÇÃO E O INDIVÍDUO

João Paulo Lunelli¹

RESUMO

Este artigo baseia-se em uma pesquisa bibliográfica, sobre os impactos causados nos indivíduos, em função do tamanho das organizações, as formas que estes encontram para se adaptar ao ambiente organizacional, que tipos de relações são buscadas, para que dentro das diversas culturas existentes e características em dado momento, os indivíduos consigam um bom desempenho organizacional, como, também, de satisfação profissional e pessoal.

Palavras-chave: análise organizacional, tamanho, indivíduo e cultura.

ABSTRACT

This paper is based on a bibliographic research about the impact on individuals of the organization size and the ways such individuals find to adapt themselves to the organizational environment. It also deals with those kinds of relationships which are strived for so that within the several existent cultures and their characteristics at a given moment, individuals arrive to a good organizational performance along to professional and personal satisfaction.

Keywords: Organizational Analysis, Size, Individuals, and Culture.

1. INTRODUÇÃO

A presente abordagem consiste numa análise do impacto do tamanho organizacional sobre o indivíduo, inspirada na leitura de HALL (1984) que afirma: “o fator tamanho tem um impacto imediato, que diminui à medida que aumenta o tempo em que uma pessoa está na organização ou em contato com ela” (...), e “o tamanho da organização é, provavelmente, a primeira coisa que uma pessoa observa”.

Nota-se que os indivíduos, de uma maneira ou outra, mantêm contato com uma organização, seja ela formal ou informal, pública ou privada, comercial, industrial ou de serviços e, assim, questiona-se quais são os principais elementos determinantes na relação entre o indivíduo e a organização e como esta se processa.

De acordo com o pensamento contingencial, o meio organizacional, interno e externo, age diretamente no psique do indivíduo e, assim, pode determinar fortes influências na sua motivação e nas suas expectativas no processo de trabalho.

A impressão subjetiva sobre o tema e que HALL (1984) menciona, originou indiretamente dois tipos de estudos. O primeiro refere-se ao grupo informal que se cria dentro da própria organização. Muitas variáveis relacionadas às expectativas do indivíduo estão fortemente atreladas a esses grupos. Os grupos informais determinam a rápida integração do indivíduo na organização, como também, determinam o nível de rejeição, isto porque o grupo informal é que modera a interação entre os indivíduos. O segundo, são as questões relacionadas às comunidades. Quanto a este tema, estudos revelam que o indivíduo vivendo numa grande comunidade não consegue interagir com todas as pessoas, mas mantém relacionamento com um grande número delas, estranhas até mesmo a sua família.

Reportando-se ao meio organizacional, em uma organização de grande dimensão, o indivíduo não consegue se relacionar e interagir com todo o seu universo, relacionando-se com um grupo representativo de pessoas, o que lhe dá uma espécie de âncora para a sua sobrevivência na organização e na comunidade como um ser social.

¹ Professor do CCEI – URCAMP. Mestrando em Administração (UFRGS). E-mail: jlunelli@alternet.com.br

Nesta relação do impacto do tamanho da organização, este estudo de pesquisa bibliográfica centralizará elementos como tipos de grupos, relações comunitárias e cooperativas, desempenho e as reações do indivíduo, a satisfação do indivíduo, cultura da organização e do indivíduo, estrutura, formalização e tecnologia e fatores ambientais.

2. GRUPOS FORMAIS E INFORMAIS

Ao se analisar a teoria das organizações, considerando-se as contribuições da sociologia, da antropologia, e mesmo da psicossociologia, verifica-se, em diversos momentos, inferências sobre a influência e a possibilidade de se lidar, no campo organizacional ou da administração, com grupos formais e informais. O indivíduo apresenta-se com uma natureza impelido a conviver com seus semelhantes, formando grupos de trabalho e social, que interagem e, por isso, são capazes de ação conjugada, a fim de alcançar objetivos comuns.

CARVALHO E SERAFIM (1998) estabelecem dois tipos de grupos de trabalho: os grupos formais e informais. Nos estudos de DEJOURS (1994), observa-se que o grupo de trabalho é estruturado por indivíduos que ele denomina como atores, capazes de atuarem numa organização de modo coletivo, desenvolvendo a inventividade, a cooperação e confiança, ressaltando, além do ator principal, que vem a ser o formal, o ator “quebra-galho” (ator informal). Se existem vários obstáculos que no ambiente de trabalho tornam difícil o estabelecimento do diálogo, existem também condições que o favorecem. Esses fatores encontram-se sobretudo na evolução dos valores no Ocidente, na dinâmica interna dos grupos informais e nas práticas adotadas em seus setores pelos líderes de turma (Chanlat e Bardad in Chanlat, 1996).

Uma das características mais marcantes da sociedade ocidental tem sido a emancipação da pessoa, cujos direitos de expressão, ao longo do tempo, são reforçados continuamente. À parte das manipulações que a mídia possa fazer, nenhum ocidental que hoje viva numa sociedade democrática pode negar o fato de que é um cidadão, que exerce por inteiro sua cidadania. Isso torna cada vez mais insustentável esse cidadão não desfrutar dos mesmos direitos no ambiente de trabalho.

Sabe-se há muito tempo que se formam grupos espontâneos no interior de qualquer estrutura oficial. Ao se examinar mais de perto o funcionamento de tais grupos, constata-se rapidamente que eles deveriam ser, no plano das trocas, dos intercâmbios, um maravilhoso exemplo a ser seguido. Curiosamente, todos os obstáculos ao diálogo que se encontra nas estruturas formais são inconcebíveis nos grupos informais e é, talvez isso, que explica sua originalidade, sua força e sua coesão. Nos grupos informais unidos, em geral, não se é avaro com as palavras; cada um sente-se um indivíduo e vê confirmar no olhar dos outros o sentimento de sua importância; o prazer provém de poder falar na primeira pessoa, de dizer “eu”, e de ser ouvido. O tempo de fala (*parole*) não é medido nem contado, e cada um está sempre moralmente disponível para o outro. As narrativas são marcadas por um clima caloroso, de afeição, de respeito mútuo e de amizade. Se surgem dificuldades nas relações entre alguns membros, os problemas são abordados de frente e sua resolução pressupõe que as pessoas envolvidas são capazes. A cultura do grupo informal aceita a metacomunicação, o que normalmente é muito difícil, ou mesmo, impossível no contexto da autoridade hierárquica ou funcional; um subordinado agir dessa maneira, seria considerado uma insolência inaceitável. O fato de pertencer ao grupo exige que todos os membros tenham comportamentos de solidariedade, colaboração, predisposição ao auxílio mútuo, reciprocidade e fidelidade à palavra.

A observação do que se passa no ambiente de trabalho revela a onipresença das atividades do “falar dos outros”, que tem uma dimensão muito mais profunda do que geralmente cremos. Ocorre o “falar dos outros” quando uma ou várias pessoas falam, bem ou mal, de

um ou de vários indivíduos ausentes. É dessa forma que os membros de um grupo pouco a pouco criam as normas e se distinguem dos demais grupos. Por ocasião dessas conversas, eles mutuamente fornecem exemplos dos comportamentos que devem ser seguidos ou proscritos; fazem o outro saber a sorte que o espera se ele se comportar como aquele do qual se fala mal ou bem. Durante as atividades de “falar dos outros”, as pessoas se constroem e se reforçam mutuamente em suas crenças individuais e coletivas. Nessa qualidade, tais atividades desempenham um papel fundamental para manter e desenvolver a identidade de cada um, e para atenuar os efeitos da violência vivida na estrutura formal. Quanto mais a estrutura formal obriga os empregados a se calarem, tanto mais eles se recuperam - e se embriagam de palavras nos grupos informais. Se os executivos buscassem refletir sobre as fontes de prazer no trabalho (Dejours, 1990 in Chanlat, 1996), poderiam descobrir entre elas a posição de destaque do ato de falar, de tomar a palavra, nos grupos informais e nas crises como as provocadas pelas greves, que freqüentemente deságuam em verdadeiras explosões de discursos, de falas, que vêm suprir necessidades por muito tempo reprimidas.

2.1 Relações comunitárias ou cooperativas

HALL (1984) diz que “uma fonte de comprovação indireta sobre o impacto do tamanho sobre o indivíduo provém dos estudos de comunidades”. Sem que haja uma pesquisa mais aprofundada sobre estas idéias, parece claro que os efeitos imediatos do tamanho da organização sobre o indivíduo são moderados, e desta forma os “estudos da participação como membro de organizações voluntárias de tamanhos variáveis, geralmente, concluem que há menor participação nas organizações maiores”. Quando as condições éticas e intersubjetivas da visibilidade estão reunidas, é possível abordar uma etapa decisiva para toda organização do trabalho - a que permite regular a tensão inevitável entre individualismo e cooperação”.

Na análise do posicionamento de Hall e Dejours, a relação do homem em comunidades ou cooperação e a forma e circunstâncias de como ele interage com outros, pode diminuir as tensões normais que surgem no momento em que o indivíduo ingressa numa organização. Isso motiva a geração de um processo cooperativo que pode se fortalecer, ao longo do tempo, criando as comunidades.

KAHN et all (1964) in HALL (1984) entendem que a relação de cooperação no grupo de trabalho pode ser uma das causas geradoras do estresse, especialmente, nas grandes organizações e, “que este estresse não pode ser diminuído ou erradicado simplesmente diminuindo a organização”, até atingirem um tamanho menor, já que “as conseqüências econômicas de tal medida seriam *trágicas*”.

2.2 Desempenho da organização e as reações do indivíduo

O desempenho e as reações do indivíduo no trabalho são analisados sob diferentes pontos, com maior ênfase, na área dos recursos humanos, essencialmente relacionando as características de personalidade, capacidade e habilidades individuais.

KAHN et all (1964) in HALL (1984) “as pesquisas existentes são incontundentes, com respeito às relações entre o tamanho e o desempenho e as reações do indivíduo diante da situação de trabalho”.

GOLDNER (1959) in CLEGG (1999) afirma que o modelo do racionalismo prescreve o “mapa” de uma estrutura autoritária em que os indivíduos e grupos são obrigados a seguir certas leis, e que os princípios de funcionamento eficiente e eficaz foram promulgados como um axioma para dirigir todas as formas de prática e análise organizacional. CLEGG (1999) ainda cita que as variáveis política, cultural, moral e histórica deste reduz o trabalho interpretativo a um mero processo de cognição dominado por regras e programas

operacionais padronizados, vital para um bom desempenho de atores individuais e organizacionais.

“O indivíduo racional é, e deve ser, organizado e institucionalizado” SIMON (1957). CLEGG (1999) analisa essa afirmação dizendo que “os seres humanos tornam-se ‘matéria prima’ transformada pelas tecnologias da sociedade moderna e membros bem comportados e produtivos da sociedade, pouco propensos a interferir nos planos de longo prazo das classes dominantes e elites”.

Em outra perspectiva que procura tornar compreensível a experiência humana e captar sua complexidade e riqueza, torna-se cada vez mais pertinente observar situações no mundo do trabalho, onde as reações do indivíduo podem ser evidenciadas através do desempenho da organização, refletido na citação abaixo:

“Em um mundo essencialmente dominado pela racionalidade instrumental e por categorias econômicas rigidamente estabelecidas, os homens e as mulheres que povoam as organizações são considerados, na maioria das vezes, apenas recursos, isto é, como quantidades materiais cujo rendimento deve ser satisfatório do mesmo modo que as ferramentas, os equipamentos e a matéria-prima. Associados ao universo das coisas, as pessoas empregadas nas organizações transformam-se em objetos (Chanlat, 1996, v. 1, p.25). (...). Logo, pode-se compreender melhor porque no momento atual, numerosos trabalhadores, sobretudo os mais jovens, procuram cada vez mais realizar-se enquanto pessoas fora do trabalho tradicional” (Adret, 1977; Gorg, 1988; Offe, 1985 in Chanlat, v.1, p.25).

2.3 A satisfação do indivíduo na organização

O ser humano é um animal motivado por necessidades e passa sua vida procurando satisfazê-las. A organização é o meio para o indivíduo suprir algumas de suas necessidades, um lugar para aprender e crescer, para buscar uma melhor qualidade de vida. Uma afirmação feita por MCGREGOR (1960) salienta quão importante é a motivação humana para o correto uso dos potenciais humanos em uma organização.

A satisfação com o trabalho é definida por ROBBINS (1999) como “a diferença entre a quantidade de premiações que os trabalhadores recebem e a quantidade que eles acreditam que deveriam receber.” Salientando que a crença dos gerentes ao longo dos anos é de que os empregados satisfeitos são mais produtivos que os insatisfeitos, o que os leva ter atitudes mais positivas em relação à organização. Não há relação consistente indicando que a satisfação leva à produtividade. E, afinal, não é a produtividade que interessa??. “Seria ingênuo supor que apenas satisfação teria um grande impacto no comportamento do empregado”(ROBBINS,1999).

O indivíduo está orientado, em seu comportamento, por valores, isto é, por estimações e avaliações, das quais decorre a sua concepção de mundo, e seu ideal de realização própria e social, e que consubstanciam sua ética de convicção. A observância mínima dessa ética, indispensável para a segurança e integridade interna do indivíduo, ocasionalmente o torna polêmico, envolvido em situações conflitantes. Seria utópico admitir-se – como salienta Whyte – que houvesse a possibilidade de instalar-se harmonia perfeita entre os valores do indivíduo e os da organização (RAMOS,1983).

2.4 Cultura da organização e do indivíduo

Quando se examina a maneira de agir do indivíduo sobre as organizações, pode se perceber que se está diante de processos de construção de pré-julgados intelectuais que acabam guiando as ações econômicas e organizacionais. E, ao se procurar as razões do

predomínio dessas construções, constata-se a ação de mecanismos culturais cujo escopo é bem mais amplo do que as arenas econômicas organizacionais. Assim, é central e adequado procurar-se analisar a lógica cultural que circunscreve o espaço das possibilidades de mudanças organizacionais.

BOURDIEU(1997) analisa o núcleo cultural das disputas políticas existentes na organização e diz:

“ A luta política é uma luta cognitiva (prática e teórica) pelo poder de impor a visão legítima do mundo social ou, mais precisamente, pelo reconhecimento, acumulado sob a forma de capital simbólico de notoriedade e de respeitabilidade, que fornece a autoridade necessária para impor o conhecimento legítimo do sentido do mundo social, de seu significado atual e da direção para onde ele vai e deve caminhar.”

MAYO (1959), numa fase avançada de suas experiências conclui que os indivíduos no trabalho são, antes de tudo, grupos que desenvolvem hábitos de relacionamento entre si, com o trabalho, e com as normas da própria organização.

BARNARD (1938), ao tratar das funções do executivo, já colocava como uma das funções dos administradores a administração dos significados, o que também remete o aspecto simbólico e cultural ao plano das preocupações básicas dos administradores.

MERTON (1964), ao propor análises de práticas sociais, a partir dos conceitos de funções manifestas e funções latentes, deixou contribuições importantes na medida em que permitiu aprofundar o conhecimento da interação dos membros da organização entre si e com a estrutura, permitindo-se esclarecer uma diversidade de padrões sociais de comportamento aparentemente irracionais, mas com uma funcionalidade intrínseca para os grupos e organizações.

Uma vez que as organizações estabelecem uma relação de recíproca interação com o seu ambiente externo, serão afetadas pela dinâmica das forças sócio-políticas e econômicas, KATZ e KAHN (1973) e PERROW (1972), autor que enfatiza a influência da cultura social nas organizações. Tais forças, por sua vez, assumirão papel relevante no processo em que se forma a identidade, o padrão das relações sociais e, também, da cultura nessas organizações.

FREITAS(1991) assinala a década de 80 como o momento em que a Teoria Organizacional suscitou “um discurso forte sobre Cultura Organizacional”. Corroborando esta perspectiva TOMEI et al.(1993) colocam, ainda, que o marco inicial dos estudos de cultura é decorrente da introdução da abordagem contingencial das organizações.

Quanto à cultura organizacional FLEURY (1986; 1990) procura inserir na discussão sobre o conceito a questão ideológica e do poder, intrínseca aos elementos da cultura - dos quais fazem parte os mitos, os ritos e rituais, entre outros elementos - o seu papel de legitimação da ordem vigente, e ocultamento das contradições presentes nas relações sociais de dominação. Esta inserção, se não resolve alguns dos problemas apontados por AKTOUF (1991), pelo menos incorpora algumas questões levantadas como base de suas críticas aos conceitos tradicionais de cultura.

Acrescentando uma dimensão política ao conceito de SCHEIN, FLEURY (1990) define cultura organizacional como:

"um conjunto de valores e pressupostos básicos expressos em elementos simbólicos, que em sua capacidade de ordenar, atribuir significações, construir a identidade organizacional, tanto agem como elemento de comunicação e consenso, como ocultam e instrumentalizam as relações de dominação".

Assim, a cultura organizacional começa a contribuir para o entendimento da complexidade da organização e a colaborar na sua elaboração estratégica e operacional, determinando o aprofundamento das discussões acadêmicas em torno da necessidade de compreensão ampla, total da empresa, uma realidade que, por si mesma, não comporta abordagens reducionistas.

3. FATORES QUE INFLUENCIAM NA RELAÇÃO DO INDIVÍDUO COM A ORGANIZAÇÃO

A análise do tema pretende integrar estudos relativos a alguns fatores influenciadores na relação do indivíduo com a organização. A estrutura é considerada como determinante das atitudes e reações em relação à satisfação e ao compromisso do indivíduo com o processo de trabalho. Como exemplo, a quantidade de controle por parte de gerente/supervisor, a existência ou não de uma estrutura hierárquica, poderá influenciar na maneira como o empregado percebe seu papel, o que gerará ou não uma satisfação com relação à situação.

James & Jones (1976, in Loscocco & Roschelle, (1991) concluem que características organizacionais, por si sós, não informam muito a respeito das reações dos empregados. É necessário saber quantas variáveis traduzem-se em características particulares do trabalho, de maneira a entender como elas afetam os empregados.

No entanto, SMITH & CARAYON (1994) relacionam fatores organizacionais a fatores psicossociais, por definirem que quando um trabalhador processa cognitivamente um fator organizacional, este tornar-se-á um fator psicossocial. E, na literatura de ergonomia, fatores psicossociais estão sendo relacionados a doenças ocupacionais dos trabalhadores (ex. Bammer & Blignaut, 1988; Bigos et al, 1991; Bongers et al, 1993; Carayon & Smith, 1994; Fitzgerald, 1992; Fordice, 1993).

Oportunidade de promoção é também considerada uma importante característica do trabalho. O estudo realizado por KANTER (1977) in Loscocco & Roschelle (1991), em uma corporação de suprimentos industriais, argumenta que oportunidades para mobilidade para cargos mais altos é muito importante para o desenvolvimento de atitudes positivas a respeito do trabalho. Enquanto, HARLAM (1989) in Loscocco & Roschelle (1991) demonstra que não existe apenas uma influência avaliativa estrutural, mas também, uma independente - as percepções dos trabalhadores com relação à justiça do sistema de promoções - nas atitudes dos empregados. O contexto social relativo à empresa e ao trabalhador também é entendido como sendo um fator importante.

"A extensão vertical e horizontal das estruturas tem por efeito multiplicar os níveis hierárquicos, do que resulta ser difícil visualizar claramente a divisão de responsabilidades; as possibilidades de cruzamento multiplicam-se e, por conseqüência, as fontes de conflito são ampliadas e tende a se instaurar um clima de desconfiança generalizada, quer seja entre os diversos níveis quer seja entre as diversas unidades" (Jaques in Chanlat, 1985, p. 155-165).

Quanto ao fator tecnologia, este possui um papel importante na estrutura, projeto, tamanho e no subsistema pessoal de uma organização. A tecnologia é um fator primordial para que a organização transforme entradas em saídas e, esta restringe o que os seres humanos em uma organização podem fazer influenciando a dinâmica das forças em uma organização. É considerada uma característica poderosa, podendo trazer vantagens e desvantagens, que dependendo de sua quantidade trará o fortalecimento dos relacionamentos existentes entre os subsistemas social e tecnológico e das dimensões da organização - complexidade, formalização, centralização e integração. O crescente uso das tecnologias de informação está subjacente ao crescimento de demandas cognitivas que requerem maior qualificação do trabalhador e, em decorrência, tornam o trabalho mais complexo.

A evolução tecnológica tem inúmeras implicações práticas, uma das mais evidentes é o caráter "labour-saving" das novas tecnologias em geral, que induzem os fenômenos de uma grande importância para o quadro temporal de trabalho, como o tempo necessário à produção.

Outro aspecto relevante para este estudo é a centralização, que se refere à distribuição de poder nas organizações HALL(1984). Segundo HAGE (1980) centralização é definida como "o nível e a variedade de participação dos grupos nas decisões estratégicas em relação ao número de grupos na organização". Tal abordagem enfatiza o fato de que o poder é exercido de diversas maneiras e numa variedade de locais dentro de uma organização.

Para Van de Ven e Ferry (1980) in Hall (1984, p. 81):

"centralização é definida como o locus de decisão que determina a autoridade dentro de uma organização. Quando a maioria das decisões é tomada hierarquicamente, uma unidade organizacional é considerada centralizada; uma unidade descentralizada, geralmente, implica que a fonte principal do processo decisório foi delegada pelos gerentes de linha para o pessoal subalterno".

Analisa-se, ainda, que a formalização, possivelmente, é muito maior nas grandes organizações, onde o controle sobre as atividades dos indivíduos é exercido com maior rigor, enquanto que tal controle é menos observável na relação do indivíduo com as pequenas organizações.

RIGGS (1964) menciona o formalismo como "a discrepância entre a conduta concreta e a norma prescrita que se supõe regulá-la, correspondendo ao grau de discrepância entre o prescritivo e o descritivo, entre o poder formal e o poder efetivo, entre a impressão que nos é dada pela constituição, pelas leis e regulamentos, organogramas e estatísticas e os fatos e práticas reais do governo e da sociedade.

As características ambientais são restrições às organizações RANSON, HININGS E GREENWOOD(1980). Quando se visualizam organizações como organismos, admite-se que a sobrevivência continuada de uma organização é dependente de um relacionamento apropriado, interativo e interdependente, entre a organização e seu ambiente. Dessa maneira, meios ambientes são cuidados até enquanto restringem ou colocam em perigo a sobrevivência organizacional. Os ambientes organizacionais são problemáticos em função do grau de incertezas que eles apresentam para os decisores organizacionais. Tal incerteza é indicada pela força das ligações interorganizacionais e as taxas de mudança dos elementos organizacionais em um ambiente.

O fato é que as organizações tentam manipular seu ambiente, e as percepções do ambiente geram ações nas organizações, que podem vir a influenciar nas atitudes e na relação do indivíduo com a organização.

4. CONCLUSÃO

Esta abordagem subjetiva, que contempla inúmeras variáveis e fatores quanto ao impacto do tamanho da organização sobre o indivíduo, é importante e merece um estudo mais aprofundado para tornar mais clara a relação indivíduo/tamanho da organização. Entretanto, os autores aqui abordados deixam transparecer que é praticamente impossível diminuir ou eliminar este impacto, apenas diminuindo as organizações; que o atual estágio da tecnologia jamais vai permitir um retorno da história, isto é impossível.

Por outro lado, é nas grandes organizações que existem naturalmente os grupos informais que se organizam em comunidades e amparam os recém chegados, e que ao mesmo tempo servem de elemento moderador da organização. Assim, os grupos informais organizam-se

em forma comunitárias para melhor desempenharem suas atividades e assim tornar reais os objetivos oficiais, embora em algumas vezes aconteçam pequenos afastamentos destes.

Dessa forma, para se pensar em diminuir ou eliminar o impacto do tamanho da organização sobre o indivíduo, faz-se necessário estudar formas de trabalho que fortaleçam os grupos, que incentivem a criação de comunidades e desenvolvimento de processos cooperativos que possam diminuir tensões, eliminar riscos de conflitos e aumentar o grau de satisfação dos indivíduos. Torna-se importante que as organizações ou os investidores e administradores implementem processos de trabalho mais descentralizados e que concedam maior autonomia aos grupos.

5. REFERÊNCIAS BIBLIOGRÁFICAS

AKTOUF, Omar. O simbolismo e a cultura de empresa: dos abusos conceituais à lições empíricas. In: CHANLAT, Jean-François (coord). TÔRRES, Ofélia de L. Sette. O indivíduo na organização: dimensões esquecidas. São Paulo: Ed. Atlas, v. II, 1996.

BARNARD, C. The function of the executive. Cambridge. Harvard University Press, 1938.

BOURDIEU, P. "Méditations pascaliennes". Ed. Seuil, p. 221-2, 1997.

BOURDIEU, P. Distinction: a social critique of the judgement of taste. London: Routledge and Kegan Paul, 1984.

CARVALHO, Antônio V. e SERAFIN, Ozélia C. Gomes. Administração de Recursos humanos. São Paulo: Pioneira, v. 2, 1998.

CHANLAT, Jean-François (coord). TÔRRES, Ofélia de L. Sette. O indivíduo na organização: dimensões esquecidas. São Paulo: Ed. Atlas, v. I, II, III, 1996.

CLEGG, Stewart, HARDY, Cynthia, NORD, Walter. Handbook de estudos organizacionais. São Paulo: Atlas, v. 1, 1999.

DEJOURS, Christophe. Psicodinâmica do trabalho: contribuições da Escola Dejouriana à análise da relação prazer, sofrimento e trabalho. São Paulo: Atlas, 1994.

_____. O fator humano. Rio de Janeiro: Ed. Fundação Getúlio Vargas, 1997.

MAYO, E. The Social Problems of an Industrial Civilization. London: Routledge and Kegan Paul, 1959.

MERTON, R. K. Ciencia, Tecnologia y Sociedad en la Inglaterra do século XVII. Madrid: Alianza, 1964.

FLEURY, Maria Tereza L. O simbólico nas relações de trabalho - um estudo sobre as relações de trabalho na empresa estatal. São Paulo. Tese de livre-docência apresentada USP/FEA, 1986. 90

FREITAS, Maria Ester de. Cultura Organizacional - formação, tipologias e impacto. São Paulo: Makron, 1991.

HALL, Richard H. Organizações: estrutura e processos. 3 ed. Rio de Janeiro: Printice/Hall do Brasil, 1984.

JAQUES, E. Structure d'organisation et créativité individuelle. In: CHANLAT, A, DUFOUR, M. (Dir.). La rupture entre l'entreprise et les hommes: le point de vue des sciences de la vie. Montréal: Québec/Amérique; Paris: Éditions d'Organisation, 1985, p. 155-165.

KATZ, D., KAHN R. L. Psicologia Social das Organizações. São Paulo: Atlas, 1972.

PERROW, C. Análise Organizacional; Um enfoque Sociológico. São Paulo: Atlas, 1972.

- RANSON, S., HINNINGS, P. GREENWOOD, R. The structuring of organizational structures. *Administrative science quarterly*.v. 25, n. 1, p. 1-17, 1980.
- RIGGS, F.W. *A ecologia da administração*. Rio de Janeiro, Fundação Getúlio Vargas, 1964.
- ROBBINS, Stephen P. *Comportamento organizacional*. Rio de Janeiro: LTC, 1999.
- SILVA, Clóvis L. M. da, VIEIRA, Marcelo M. Falcão e DELLAGNELO, Eloise H. L. *Ciclo de vida, controle e tecnologia: um modelo para análise das organizações*. Canela-RS: XVI Encontro anual da ANPAD, 1992.

A IMPORTÂNCIA DO MARKETING DE RELACIONAMENTO E A IMPLEMENTAÇÃO DO PLANEJAMENTO DE MARKETING

Prof^o. Ricardo Leal Cougo

ABSTRACT

The present work searches to approach the main activities of the marketing, as well as the process of implementation of the planning of marketing. The current markets are in constant changes, run over for the technological globalization and innovations that quickly present new forms of production and commerce. These changes are taking the companies to a confusion in relation to the strategy to be followed. To try to remain itself, they has mainly, reacted cutting to its costs and many times reducing its force of work. However, even enterprises well accomplished in cost cutting may have some difficulties to improve their cash flow without a accurate eyesight of market.

INTRODUÇÃO

Vive-se a era de grandes transformações. Com a popularização da grande teia, não estamos mais limitados às fronteiras físicas da cidade ou do país em que vivemos. Estamos, sim, inseridos numa sociedade globalizada, onde a informação flui de forma instantânea. Uma grande massa de profissionais experimenta expansão do foco de atuação em suas áreas. Essas mudanças exigem atualizações constantes. Aqueles que não estão se adaptando aos novos requisitos enfrentarão sérias dificuldades em breve. Nas empresas, a situação não é diferente. O que se experimenta são grandes mudanças de paradigmas administrativos e culturais. Estamos vivendo uma turbulência, no meio da qual muitos não sabem o que fazer. É a pressão exercida pelo rolo compressor da globalização, que exige melhora constante de qualidade e aumento da competitividade. Nesse cenário de revolução tecnocultural entra a figura do *Marketing*, como ferramenta de potencialização da competitividade empresarial. Essa potencialização deve ser provida de forma igualmente globalizada às necessidades de autonomia e, ao mesmo tempo, de interconectividade das várias áreas empresariais.

O *Marketing* implica uma atitude generalizada global da empresa. Nesse sentido pode falar-se dele como uma ideologia, **como um valor cultural que envolve toda a empresa**. Um cliente não é alguém que compra uma vez para provar, é alguém que nos elege sempre. Para isso, deve-se conseguir uma vantagem comparativa com respeito à concorrência, que nos permita satisfazer a nossos clientes melhor do que faria esta. O *Marketing* é dinâmico, quer dizer, as necessidades e desejos dos clientes evolui constantemente e uma empresa exitosa deve evoluir com eles (antecipar-se) para que os concorrentes não levem vantagem. Em alguns lugares do mundo, as pessoas despertam diariamente para enfrentar diferentes tipos de experiências. Uma família chinesa pode ter pouca escolha sobre que alimento comerá ou de onde virão suas roupas. Um camponês nas montanhas da Jamaica pode despertar em uma choupana com a esperança de colher apenas o suficiente para sua sobrevivência. Alguém em uma grande cidade como Tóquio pode ter muitas escolhas, mas não estará familiarizado com produtos de marcas Maxwell House, General motors e Oscar Mayer. O que explica essas diferenças e o que isso tem a ver com *Marketing*? Neste Trabalho, buscar-se responder a questões como essa. Você verá o que é *Marketing* e por que ele é importante. Explora-se, também, como o *Marketing* e seu planejamento afetam a qualidade de vida de diferentes sociedades e por que ele é tão crucial para o desenvolvimento econômico e para nossa economia global.

O MARKETING E SEUS RELACIONAMENTOS

Se a maioria das pessoas for forçada a definir *Marketing*, inclusive alguns gerentes de empresas, eles afirmarão que *Marketing* significa “venda” ou “propaganda”. É verdade que são partes de *Marketing*, mas *Marketing* é muito mais do que venda e propaganda. Assim, pode-se perceber que *Marketing* desempenha um papel essencial ao fornecer aos consumidores bens e serviços que satisfaçam suas necessidades. A produção é uma atividade econômica muito importante. A maioria das pessoas não fabrica os produtos que usa.

Claramente, o alto padrão de vida que a maioria das pessoas de economias avançadas desfruta é possível através da produção especializada. Como consumidores, as pessoas pagam pelo custo das atividades de *Marketing*. Em Economias avançadas, os custos de *marketing* representam 50 centavos de cada dólar consumido. Para alguns bens e serviços a porcentagem é muito maior. *Marketing* afeta quase todos os aspectos de sua vida diária. Todos os bens e serviços que você consome, as lojas onde você compra e os programas de rádio e televisão pagos pelos anunciantes existem em função de *Marketing*. Mesmo seu currículo faz parte de uma campanha de *marketing* para vendê-lo a algum empregador. Algumas disciplinas curriculares são interessantes no início, mas não serão mais relevantes após você concluí-las. No entanto, tratando-se de *Marketing*, você será um consumidor que vai lidar com essa disciplina pelo resto de sua vida. Uma razão fundamental para se estudar *Marketing* é que ele é responsável por grande parte do crescimento e desenvolvimento econômico. *Marketing* estimula a pesquisa e novas idéias, resultando em novos bens e serviços. *Marketing* proporciona aos consumidores a escolha entre muitos produtos. Se esses produtos os satisfizerem, pode resultar em mais emprego, melhores salários e um padrão de vida mais elevado. Um sistema de *Marketing* eficaz é importante para o futuro das nações. O *Marketing* é um conjunto de atividades desempenhadas por organizações e também um processo social. Em outras palavras, *marketing* existe nos níveis micro e macro. Assim, usa-se duas definições de *marketing*. A primeira examina os consumidores e as organizações que os servem. A segunda adota uma visão ampla de nosso sistema completo de produção-distribuição.

UNIÃO DE PRODUTORES E CONSUMIDORES ATRAVÉS DO MARKETING EFICAZ

Marketing eficaz quer dizer entregar bens e serviços que os consumidores desejam e necessitam. Consiste em obter produtos para eles no tempo certo, no local certo e no preço que estão dispostos a pagar. Não é uma tarefa fácil, principalmente se você refletir sobre a variedade de bens e serviços que uma economia altamente desenvolvida pode produzir e os muitos tipos de bens e serviços que os consumidores desejam. *Marketing* eficaz em uma economia avançada é mais difícil porque produtores e consumidores estão separados de várias maneiras. As discrepâncias de quantidade e de sortimento complicam ainda mais a troca entre produtores e consumidores. Isto é, cada produtor especializa-se em fabricar e vender grandes quantidades de um sortimento limitado de bens e serviços, mas cada consumidor deseja apenas pequenas quantidades de um sortimento amplo de bens e serviços.

O propósito de um sistema de *macromarketing* é superar essas separações e discrepâncias. As funções universais de *marketing* ajudam a fazer isso. As *funções universais de marketing* são compra, venda, transporte, estocagem, padronização e classificação, financiamento, risco e informação de mercado. Elas devem ser desempenhadas em todos os sistemas de *macromarketing*. As formas como essas funções são desempenhadas e por quem o são podem diferir entre países e sistemas econômicos. Geralmente, a troca envolve compra e venda. A função compra significa a procura e a avaliação de bens e serviços. A função venda envolve a promoção do produto. Inclui o uso de venda pessoal, propaganda e

outros métodos de venda em massa. Provavelmente, é a função de *Marketing* mais visível. A função transporte significa o movimento de bens de um local a outro. A função estocagem envolve manter os bens até que os consumidores necessitem deles. Padronização e classificação envolvem sortimento de produtos de acordo com tamanho e qualidade. Isso torna a compra e a venda mais fáceis porque reduz a necessidade de inspeção e amostragem. O financiamento fornece dinheiro e o crédito necessários para produzir, transportar, estocar, promover, vender e comprar produtos. O risco envolve assumir a incerteza que faz parte do processo de *marketing*. Uma empresa nunca pode estar segura de que os consumidores desejarem comprar seus produtos. Estes podem ser destruídos, roubados ou desatualizados. A função informação de mercado envolve a coleta, análise e distribuição de todas as informações necessárias para se planejar, conduzir e controlar as atividades de *marketing*, sejam elas no mercado local ou no exterior.

De um ponto de vista macro, essas funções de *marketing* fazem parte do processo de *marketing* e devem ser executadas por alguém. Nenhuma delas pode ser eliminada. Em uma economia planejada, algumas funções podem ser desempenhadas por órgãos governamentais. Outras podem ser deixadas para os produtores e consumidores.

Em um sistema dirigido pelo mercado, as funções de *marketing* são desempenhadas por produtores, consumidores e por uma variedade de especialistas de *marketing*. Algumas funções de *marketing* podem ser desempenhadas por uma variedade de *facilitadores*, empresas que desempenham uma ou mais funções de *marketing*, além de compra e venda.

Incluem as agências de propaganda, empresas de pesquisas de *marketing*, laboratórios independentes de testes, armazéns públicos, empresas transportadoras, empresas de comunicação e instituições financeiras (também bancos). Através da especialização ou de economia de escala, os intermediários e facilitadores de *marketing* estão, freqüentemente, em condições de desempenhar melhor as funções de *marketing*, e a custo mais baixo, do que os produtores e consumidores. Estes ficam com mais tempo para produção e o consumo.

A partir de um ponto de vista macro, todas as funções de *marketing* devem ser desempenhadas por alguém. Entretanto, de um ponto de vista micro, nem todas as empresas devem desempenhar todas as funções. Além disso, nem todos os bens e serviços exigem todas as funções em todos os níveis de sua produção. Por exemplo, serviços puros, como uma viagem de avião, não precisam de estocagem. Porém, exige-se estocagem na fabricação do avião e enquanto ele não está em serviço. Alguns especialistas de *marketing* desempenham todas as funções, outros especializam-se apenas em uma ou duas funções. Por exemplo, empresas de *pesquisa de marketing* são especializadas apenas na função informação de mercado.

DEFINIÇÃO DE MERCADOS-ALVOS PELA SEGMENTAÇÃO

A *Segmentação de Mercado* é um processo de duas etapas: 1ª) nomear mercados-produtos amplos e 2ª) segmentar esses mercados-produtos amplos para selecionar mercados-alvos e desenvolver compostos de *marketing* adequados. Esse processo de duas etapas não é bem entendido. Freqüentemente, os esforços para segmentar um mercado pela primeira vez fracassam porque as empresas começam com mercado de massa total e tentam definir uma ou duas características demográficas para segmentar esse mercado. Geralmente, o comportamento do consumidor é bastante complexo para ser explicado apenas por uma ou duas características demográficas. Por exemplo, nem todos os homens idosos compram os mesmos produtos ou marcas. Normalmente, outras dimensões devem ser consideradas, partindo-se das necessidades dos consumidores. Os segmentadores buscam aumentar o potencial de vendas. Um segmentador que satisfaz o mercado-alvo pode não ter nenhuma concorrência real. Para selecionar as dimensões de segmentação, partimos de dois tipos

diferentes de dimensões. As dimensões qualificadoras são aquelas relevantes para incluir um tipo de consumidor em um mercado-produto. As dimensões determinantes são aquelas que realmente afetam o consumidor que compra um produto ou marca específica em um mercado-produto. Por exemplo, um comprador de carro precisa ter dinheiro suficiente ou crédito para adquirir um veículo e segurá-lo. Nosso comprador também precisa de carteira de motorista. Isso ainda não garante a compra. Ele deve ter uma necessidade real, como um trabalho que exija “rodas” ou escolares para serem transportados. Essa necessidade pode motivar a compra de algum carro. Mas essas dimensões qualificadoras não determinam que marca ou modelo específico a pessoa pode comprar. Isso depende de interesses mais específicos como condições de segurança, desempenho ou aparência, desejados pelo consumidor. Determinar as dimensões relacionadas a essas necessidades afeta o tipo de carro que será escolhido pelo comprador.

Em se tratando de *Marketing* Internacional aumenta a exigência da atenção quanto à segmentação. Há cerca de 140 países com culturas únicas. Eles diferem grandemente em língua, costumes, ética nos negócios, crenças, religiões, raça e padrões de distribuição de renda. Essas diferenças adicionais podem complicar o processo de segmentação. O que é ainda muito pior, as informações críticas estão, com frequência, menos disponíveis e menos confiáveis à medida que as empresas entram em mercados internacionais. O número de variáveis aumenta, mas a quantidade e a qualidade das informações diminui. Essa é uma das razões por que algumas empresas concluem que as operações e decisões locais devem ficar a cargo de pessoas locais.

Pelo menos, eles possuem sensibilidade para seus mercados. A segmentação de mercados internacionais pode exigir mais dimensões. Mas um método prático acrescenta apenas uma etapa às abordagens discutidas acima. Primeiro, as empresas segmentam por país ou região, examinando características demográficas, culturais e outras, incluindo o estágio de desenvolvimento econômico. Isso pode ajudá-las a encontrar submercados razoavelmente similares. Depois, dependendo de a empresa estar visando consumidores finais ou mercados industriais, aplicam-se as mesmas abordagens básicas anteriormente discutidas.

O *Posicionamento* é outra abordagem importante, mostra como os consumidores localizam as marcas atuais e/ou propostas em um mercado.

Exige alguma pesquisa de *marketing* formal, mas pode ser útil quando as ofertas concorrentes são muito similares. Os resultados são, geralmente, anotados em gráficos para ajudar a mostrar onde os produtos estão posicionados em relação aos concorrentes.

Normalmente, as posições dos produtos estão associadas a duas ou três características do produto que são importantes para os consumidores-alvo.

Os gerentes constroem gráficos para as decisões de posicionamento pedindo aos usuários do produto que façam julgamentos de diferentes marcas, incluindo sua marca “ideal” e depois, usem programas de computador para resumir as avaliações e anotar os resultados. Os detalhes das técnicas de reposicionamento, às vezes chamadas de “mapeamento de percepção”, estão além do escopo deste texto. O posicionamento ajuda os gerentes a entenderem como os consumidores vêem seu mercado. Ele pode ser útil como parte de uma análise mais ampla de mercados-alvo. Posicionar uma marca em relação a outras pode levar uma empresa ao fracasso quando ocorrem mudanças básicas nos mercados. Por exemplo, os sabonetes em barra podem estar perdendo popularidade para os sabonetes líquidos, ou outros produtos como óleos de banho ou loções para limpeza de pele podem fazer parte da concorrência relevante. Os gerentes não veriam essas mudanças se ficassem fixados apenas às marcas alternativas de sabonetes, isto é, a um foco muito restrito.

PLANEJAMENTO DE MARKETING

Um estudo abrangente da Conference Board busca resumir a prática do *Marketing* nos anos 90. Neste trabalho constata-se que mais empresas estão descobrindo o conceito de *Marketing* e ajustando-se ao propósito de conquistar e satisfazer os consumidores, ao invés de limitar-se à produção de bens e serviços. Os **planos de negócios** estão sendo mais orientados para o consumidor e para os concorrentes e mais discutidos e realistas do que foram no passado.

Os planos retratam mais o conteúdo das funções de *Marketing* e estão sendo desenvolvidos em equipe. Crescentemente, os executivos de *Marketing* vêm-se, primeiramente, como administradores profissionais e, depois, como especialistas. A alta administração está tornando-se mais envolvida na preparação ou aprovação das decisões de *Marketing*. O Planejamento vem tornando-se um processo contínuo durante o ano para responder às condições de mercado rapidamente mutantes. Entretanto, o levantamento constatou que os procedimentos e o conteúdo do planejamento de *marketing* variam consideravelmente entre as empresas. O plano é chamado de **plano de negócio, plano de marketing** e, às vezes, **plano operacional**. A maioria dos **planos de Marketing** cobre o período de um ano, mas alguns envolvem alguns anos. Os planos variam em extensão, de 10 a 50 páginas. De acordo com executivos de *marketing*, as deficiências mais frequentes dos *planos de marketing* atuais são a falta de realismo, análise competitiva insuficiente e foco restrito no curto prazo.

O Planejamento corporativo, divisional e das unidades de negócios é parte integral do processo de *marketing*, deve-se primeiramente, examinar como uma empresa define seu negócio. A tarefa de qualquer negócio é entregar valor ao mercado e obter algum lucro. Há pelo menos duas visões do processo de entrega de valor.

A visão tradicional é que a empresa fabrica algo para, depois, vender. Por exemplo, a empresa de Thomas Edison inventa o fonógrafo e, depois, contrata pessoas para fabricá-lo e vendê-lo. Nesta visão, *Marketing* entra em ação na segunda metade do processo de entrega de valor. A visão tradicional afirma que a empresa sabe o que fabrica e que o mercado comprará unidades suficientes para ela obter lucro.

Esta visão tradicional tem chance de ser bem-sucedida em economias de escassez. Por exemplo, muitos consumidores do leste Europeu estão desesperados à procura de bens, não importa onde sejam fabricados.

Normalmente, não estão preocupados com qualidade, características ou estilo. Mas a visão tradicional do processo empresarial não funciona em economias mais competitivas, onde as pessoas possuem muita escolha. Realmente, o *mercado de massa* está pulverizado em muitos micromercados, cada um com seus próprios desejos, percepções, preferências e critérios de compra. Contudo, o concorrente inteligente deve desenhar a oferta para mercados-alvo bem definidos. Essa crença é a nova visão do processo de negócio, que coloca *marketing* no início do processo de planejamento. Em vez de enfatizar a fabricação e a venda, as empresas que adotam essa visão vêm-se como parte de uma seqüência de criação e entrega de valor. Para assumir as responsabilidades, os *administradores de marketing*, seja a nível corporativo, divisional ou de unidade de negócio, seguem um processo de *marketing*. Trabalhando conforme os planos estabelecidos em níveis hierárquicos superiores, os gerentes de produtos preparam um plano de *marketing* para produtos, linhas de produtos ou marcas.

OS RELACIONAMENTOS DO PLANEJAMENTO DE MARKETING

As atividades incluem realizar a avaliação da situação, determinar os objetivos, desenvolver as estratégias de objetivos e posicionamento, decidir programas de ação para os

componentes da mistura de *marketing* e preparar declarações financeiras de apoio (orçamentos e projeções de lucros e perdas).

O típico processo de planejamento em uma companhia que tem uma abordagem de planejamento efetiva envolve bastante coordenação e interação entre áreas funcionais. Produtos múltiplos exigem negociações para atingir uma concordância quanto à quantidade do tempo da força de vendas devotada a vários produtos. A característica de Planejamento fornece a visão de um CEO de como os planos são coordenados por um fabricante de produtos ao consumidor. A seqüência do Planejamento de *Marketing* se estabelece como segue: atualização de dados históricos, coleta de dados da situação presente, análise de dados, desenvolvimento de objetivos e programas estratégicos, desenvolvimento de documentos financeiros, negociação do plano final, medida do progresso dos objetivos e auditoria.

Nosso processo de planejamento começa com uma reunião de três ou quatro horas que discute as questões estratégicas. A reunião só inclui três pessoas, o Presidente da Divisão, o Presidente da Diretoria e o Presidente da Corporação. Algumas questões que versam sobre os temas anteriores são: Quais são as questões? Qual é a dinâmica do mercado? Como eu o vejo hoje e como será nos próximos anos? O que deveríamos estar gastando em termos de proteção de nossa fatia de mercado? Para onde está indo o mercado a partir de um ponto de vista demográfico?

Essa é uma discussão estratégica de alto nível. Nós simplesmente nos sentamos e discutimos o negócio, sem grande parte dos dispositivos, sem várias figuras, mais informal do que formal, a fim de obter a concordância deles, dos Presidentes (superiores), de que realmente estas são as questões que também os preocupam e que precisamos de desenvolver planos em torno dessas questões.

Concordamos que existem cinco ou seis imperativos estratégicos. Em cada caso, concordamos que um objetivo em particular tem de ser atingido como sucesso ou a divisão não vai atingir seus esforços comerciais e que dispomos de um horizonte de três anos. Os objetivos podem se referir à parcela de mercado, tecnologias de novos produtos, desenvolvimento administrativo ou questões organizacionais que devem ser encaradas. A preparação, tipicamente, é feita pelo departamento de *Marketing*, supervisionado pelo Gerente ou Diretor de serviços de *marketing* e planejamento. É mais ou menos ele que dirige a comissão que prepara o plano. Então, assim que o plano estiver pronto, tem-se uma reunião na administração central. Estudamos áreas muito específicas dentro de um relatório de 35 a 50 páginas. Este é basicamente o jeito como as coisas funcionam. Você tem uma idéia do que é sua função, mas outras pessoas na organização podem ter uma idéia bem diferente. Pensamos que a chave para ser obtida uma concordância quanto à missão é “ o que realmente estamos querendo concretizar?”

O PROCESSO DE IMPLEMENTAÇÃO

Um plano bem elaborado para implementação identifica as tarefas a serem implementadas, quem é responsável pela implementação, a época e localização da implementação e como ela deverá ser feita. Os encarregados das vendas devem focalizar todas as contas agora usando um produto competitivo.

Um plano deve ser desenvolvido para converter cinco por cento dessas contas para o produto da companhia durante todo o ano. As relações de contas são preparadas e distribuídas pela gerência do produto.

O plano de *Marketing* é usado para identificar as unidades operacionais e os gerentes responsáveis pela implementação das várias atividades do plano. Os prazos finais indicam o tempo disponível para a implementação. No caso do plano supra, o gerente de vendas é responsável pela implementação por meio da força de vendas.

Os Gerentes são importantes facilitadores do processo de implementação. Alguns executivos são melhores que outros. Bons planejadores e implementadores freqüentemente possuem diferentes pontos fracos e fortes. Nem sempre um bom planejador é bom na implementação. Principais habilidades desejáveis para implementação:

- habilidade de entender os outros e saber pechinchar;
- força para ser justo a fim de colocar as pessoas e os recursos onde serão mais eficientes;
- eficiência sobre os aspectos críticos do desempenho ao administrar as tarefas de *Marketing*;
- habilidade de criar uma organização informal necessária ou uma rede para enfrentar cada problema com o qual se confronte.

PLANEJAMENTO DOS PROGRAMAS DE MARKETING

Para transformar a estratégia de *marketing* em programas de *marketing*, os administradores devem aplicar decisões básicas sobre as despesas, composto e alocação de *marketing*. Tipicamente, as empresas estabelecem seus orçamentos de *marketing* baseando-se em uma porcentagem da meta de vendas. Segundo a organização também precisa decidir como dividir o orçamento de *marketing* entre as várias ferramentas do composto de *marketing*. O composto de *marketing* é um dos conceitos-chaves da moderna teoria de *marketing*. Há muitas ferramentas no composto de *marketing*. McCarthy popularizou uma classificação dessas ferramentas chamada os **4 ps**: produto, preço, praça (distribuição) e promoção. As decisões do composto de *marketing* devem ser tomadas considerando os canais de distribuição e os consumidores finais. A empresa prepara um composto da oferta de produtos, serviços e preços, utilizando um composto promocional de promoção de vendas, propaganda, força de vendas, relações públicas, mala direta e *telemarketing* para atingir os canais de distribuição e os consumidores-alvo.

Nem todas as variáveis do *composto de marketing* podem ser ajustadas a curto prazo. Tipicamente, a empresa pode alterar seu preço, tamanho da força de vendas e despesas de propaganda a curto prazo.

Os profissionais de *marketing* devem decidir sobre a alocação do orçamento de *marketing* entre os vários produtos, canais, mídias de promoções e áreas de vendas. Quanto deve gastar com as máquinas elétricas ou eletrônicas? Com venda direta x distribuidores? Com mala direta x anúncios em revistas técnicas. Para fazer essas alocações, os administradores de *marketing* utilizam as funções de respostas de vendas, que mostram como as vendas seriam afetadas pela quantidade de dinheiro alocado a cada aplicação possível. A ferramenta mais básica do composto de *marketing* é o produto, a oferta tangível da empresa para o mercado, que inclui qualidade, design, características, marca e embalagem.

Uma ferramenta crítica do composto de *marketing* é o preço, a quantidade de dinheiro que os consumidores pagam pelo produto. A praça (distribuição) é outra ferramenta-chave do composto de *marketing*. Inclui várias atividades assumidas pela empresa para tornar o produto acessível e disponível aos consumidores-alvo. A promoção é a quarta ferramenta do composto de *marketing*. Inclui todas as atividades desempenhadas pela empresa para comunicar ao mercado-alvo seus produtos e promovê-los. Os **4 ps** representam a visão dos vendedores em relação às ferramentas de *marketing* disponíveis para influenciar compradores. Robert Lautgerborn sugeriu que os **4 ps** dos vendedores correspondam aos **4 cs** dos consumidores.

Provavelmente, há muitas surpresas e desapontamentos na fase de implementação dos planos de *marketing*. Por essa razão, a organização precisa de feedback e controle. Há três tipos de controles de *marketing*:

Controle do plano anual – é a tarefa de assegurar que a empresa está atingindo suas metas de vendas, lucros, etc. Definição de metas por mês ou trimestre, mensurar o desempenho contínuo no mercado, causas de hiatos no desempenho e desenvolver ações corretivas para preenchimento de hiatos entre metas e desempenho.

- *Controle de rentabilidade* – Não é tarefa simples, o sistema contábil, raramente, é preparado para relatar a rentabilidade real de diferentes entidades e atividades de *marketing*.
- *Controle estratégico* – é a tarefa de avaliar se a estratégia de *marketing* da empresa está apropriada às condições de mercado.

Um plano de *marketing* deve começar com um breve sumário, contendo suas principais metas e recomendações. Veja o conteúdo abaixo:

1. Sumário executivo e índice de conteúdo	Apresenta uma breve visão do plano proposto
2. Situação atual do <i>marketing</i>	Apresenta dados históricos relevantes sobre o mercado, produto, concorrência, distribuição e macroambiente
3. Análise de oportunidades e assuntos	Identifica as principais ameaças/oportunidades, forças/fraquezas e assuntos relativos ao produto
4. Objetivos	Define as metas financeiras e de <i>marketing</i> do plano em termos de volume de vendas, participação de mercado e lucro
5. Estratégia de <i>Marketing</i>	Apresenta a abordagem ampla de <i>marketing</i> que será usada para atingir os objetivos do plano
6. Programas de Ação	Apresenta programas de <i>marketing</i> especiais preparados para atingir os objetivos do plano
7. Demonstração de resultado projetado	Prevê o resultado financeiro esperado do plano
8. Controles	Indica como o plano será monitorado

CONCLUSÃO

Ainda que o *Marketing* pareça ser parte integral do êxito dos negócios no mundo moderno, não foi sempre assim. Só quando uma sociedade alcança uma etapa em que satisfaz suas necessidades de produção e vendas faz a transição à fase orientada ao *Marketing*. Fase orientada ao *Marketing* se caracteriza por uma visível transição das táticas de venda agressiva com a meta de satisfazer as necessidades e desejos do cliente e conseguir além disso um volume rentável de vendas. O *Marketing* substitui as vendas e se converte em prioridade. Nas empresas lucrativas criam-se departamentos de *Marketing* para fazer com que as ofertas de produtos e serviços correspondam às exigências do mercado. Estes departamentos começam a desempenhar um papel central tanto em planejamento a curto como a longo prazo.

O *Marketing* implica uma *atitude generalizada global da empresa*. Nesse sentido pode falar-se dele como uma ideologia, como um valor cultural que envolve toda a empresa. A área de *Marketing* é simplesmente o braço executor desta teoria. Há que criar uma convicção geral, em cada um dos funcionários, de que o que o cliente compra não são somente os produtos ou serviços que divulgamos, mas fundamentalmente compra a lealdade. Um cliente não é alguém que compra uma vez para provar, é alguém que nos elege sempre. Para isto deve-se conseguir uma vantagem comparativa com respeito aos

concorrentes, que nos permita satisfazer a nossos clientes melhor que os outros. Isto é o primeiro ponto claro para planejar uma estratégia competitiva. Os clientes não compram apenas um produto, mas sim a promessa da solução do problema, de uma necessidade, ou um desejo de satisfação. Tudo isso é prova incontestável de que o *Marketing* é dinâmico, quer dizer, as necessidades e desejos de nossos clientes evoluem constantemente e uma empresa exitosa deve evoluir com eles para que algum concorrente não leve vantagem. Fundamentalmente, *Marketing é conseguir um maior valor agregado para nosso cliente do que possa oferecer o concorrente*. Esta é a teoria única do *Marketing*. Não existe um *marketing* educativo ou outro industrial, existem diferentes facetas e proporções no uso de cada variável.

De outra forma, o estabelecimento do planejamento requer o atendimento de quatro níveis que são: corporativo, divisional, unidade de negócio e produto. O Planejamento estratégico dos negócios envolve atividades de definição da missão do negócio, análise das oportunidades e ameaças externas, análise de forças e fraquezas internas, formulação de programas de apoio, implementação de programas, feedback e controle. Cada nível de produto dentro de uma unidade de negócio deve desenvolver um plano de *marketing* para atingir suas metas. O Plano de *Marketing* é um dos produtos mais importantes do processo de *marketing*. Dentro desse processo, a qualidade deve ser vista como uma arma competitiva que abarca todo o necessário para satisfazer aos clientes melhor que a rivalidade. O cliente não é somente aquele ao que vai dirigido nosso produto ou serviço final. A qualidade não é um problema técnico que compreende os produtos, inspetores e peritos. É um problema estratégico que envolve toda a organização, começando pela alta cúpula. As modernas organizações são identificadas por forte cooperação e foco no consumidor. Um plano estratégico de *marketing* tem que ser apropriadamente implementado. A implementação dos planos de *marketing* exige habilidade em reconhecer e diagnosticar um problema e avaliar os resultados obtidos. Assim é importante o controle do plano anual para assegurar que a empresa alcance as vendas, lucros e metas estabelecidas. E nesse sentido, o controle estratégico cumpre função fundamental, fazendo a reavaliação periódica da empresa e de sua abordagem estratégica em relação ao mercado, usando as ferramentas da revisão da eficácia de *marketing* e da auditoria de *marketing*. As empresas devem também propor revisões da atividade de *marketing* e revisões de sua responsabilidade ética e social.

BIBLIOGRAFIA

KOTLER, P. Administração de *marketing*: análise, planejamento, implementação e controle/ SP: Atlas, 1998.

URBAN, Glen L. & STAR, Steven H. Advanced *marketing* strategy: phenomena, analysis and decision. Englewood Cliffs, Prentice Hall, 1991.

MCCARTHY, E, J. *Marketing* essencial: uma abordagem gerencial e global/ SP: Atlas, 1997.

CRAVENS, D. Estratégia de *marketing*, implementação e controle: Manual do PPGA, disciplina: *Marketing* Estratégico, Prof. Dr. Carlos Alberto Vargas Rossi.

MATTAR, F. N. Pesquisa de *marketing*: metodologia, planejamento. São Paulo: Atlas, 1997.

REFLEXÃO COMPUTACIONAL

Acauan Pereira Fernandes¹

RESUMO

Este artigo pretende introduzir e discutir o paradigma da reflexão computacional e seus principais conceitos, assim como mostrar os benefícios e dificuldades decorrentes de sua utilização. Mostra também o estado da arte de sua aplicação por meio de protocolos de meta-objetos em linguagens de programação. Finalmente, discute como utilizar a reflexão computacional para introduzir de forma transparente, não intrusiva e reutilizável, requisitos não-funcionais em um sistema, focalizando o aspecto do gerenciamento de atomicidade.

ABSTRACT

The goal of this paper is to introduce and discuss computational reflection as well as its most important concepts. It also intends to show what may be achieved through its use. This means not only benefits but also problems to be dealt with. The state-of-art in meta-object protocols is shown. Finally, it discusses how to use computational reflection to introduce non functional requirements in a transparent, non intrusive and reusable way, focusing the atomicity management aspect.

1. O PARADIGMA REFLEXIVO

1.1 Introdução

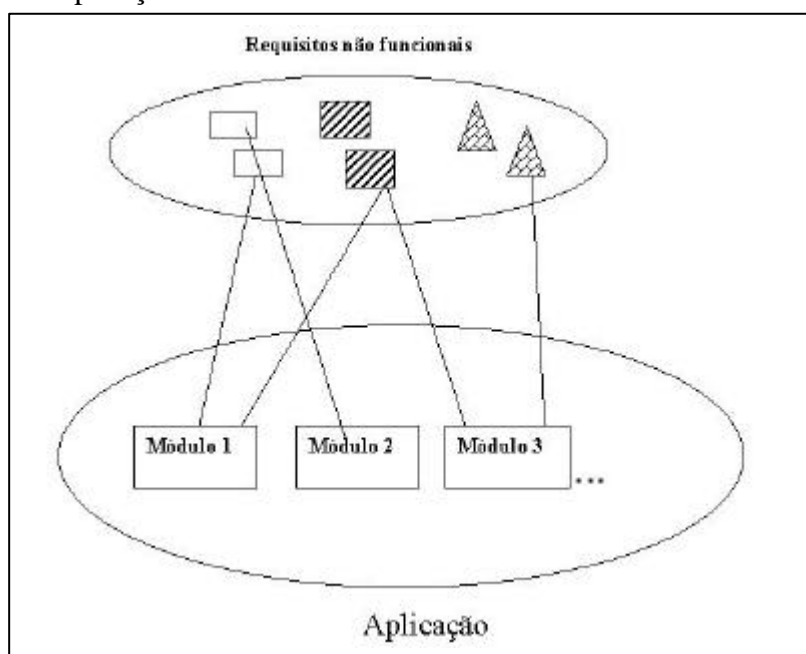
Um dos grandes problemas encontrados por desenvolvedores de sistemas computacionais é que, uma vez definidos e implantados, torna-se extremamente difícil e complexo alterar-se o seu comportamento sem alteração de seu código. As demandas ocorrentes na prática, entretanto, são muito mais abrangentes e invariavelmente exigem do sistema uma flexibilidade muitas vezes inatingível por meio dos mecanismos tradicionais existentes. Com este cenário em mente, percebe-se o enorme potencial de aplicação de um paradigma relativamente desconhecido: a reflexão computacional. Apresentada em 1987 por Patti Maes (MAES, 1987), pode ser definida como qualquer atividade executada por um sistema sobre si mesmo, no intuito de obter informações sobre suas próprias atividades tendo como objetivo melhorar seu desempenho, introduzir novas capacidades ou ainda resolver seus problemas escolhendo a melhor forma de procedimento (LISBÔA, 1998) (CAZZOLA, 1999). Neste sentido, sua preocupação deixa momentaneamente de ser o problema para o qual foi desenvolvido, ou seja, suas especificações funcionais, e passa a ser sua própria execução.

A utilização de reflexão computacional permite que um sistema monitore seu contexto presente e, a partir dessa análise, decida o que fazer a seguir. Essa capacidade confere a ele um alto grau de flexibilidade, visto que não há mais necessidade de seguir passos definidos *a priori*. Sua aplicação mais visível tem sido na área de processamento distribuído, persistência de dados, controle de concorrência e tolerância a falhas (STROUD, 1995). Conforme será visto, a introdução desses requisitos não funcionais de forma transparente e não intrusiva à aplicação pode ser levada a cabo de forma extremamente eficiente e sofisticada através da utilização deste paradigma.

¹ Bacharel em Ciência da Computação (UCPel) e Agronomia (URCAMP). Mestrando 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).

1.2 Arquitetura reflexiva

A arquitetura usada em um sistema com suporte à reflexão é chamada arquitetura reflexiva e divide esse sistema em duas partes logicamente distintas: um nível base, onde estão os dados e código responsáveis pela execução dos requisitos para os quais a aplicação foi desenvolvida, e um meta-nível, em que se encontram estruturas e ações que devem ser executadas, dinamicamente, em resposta às ações do nível base do programa (LISBÔA, 1997). O domínio do primeiro nível é o mundo real, enquanto que o do segundo é o primeiro, ou seja, a própria aplicação. As classes e suas instâncias (objetos) localizadas no meta-nível são chamadas respectivamente de meta-classes e meta-objetos. Essa divisão característica permite a reutilização separada e independente dos níveis, mas, principalmente, torna possível a separação do código que lida com requisitos funcionais do que lida com requisitos não-funcionais. Entre outras vantagens, simplifica o desenvolvimento de software, liberando o programador de manipular outros contextos que não o específico da aplicação.

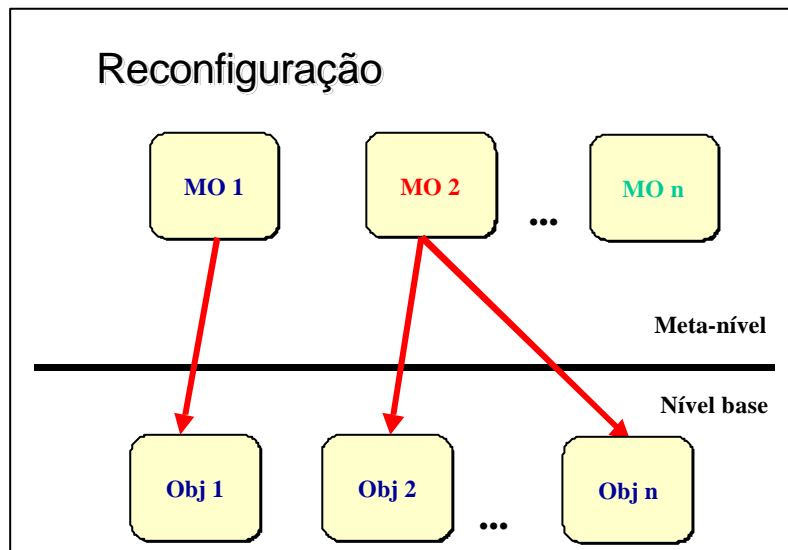


Requisitos não-funcionais no meta-nível (adaptado de LISBÔA, 1997)

Conforme visto na figura anterior, os requisitos não-funcionais de uma aplicação podem ser isolados nas meta-classes do meta-nível, incorporando assim capacidades inerentes à orientação a objetos, como encapsulamento e reutilização. Encapsulamento porque cada requisito pode estar restrito a um grupo de meta-classes, podendo ser tratado de forma isolada e independente. Reutilização porque essas meta-classes podem ser novamente utilizadas em muitos outros sistemas que as necessitem. Essa possibilidade torna-se extremamente importante no incremento da velocidade de desenvolvimento de aplicações, pois permite ao programador incorporar tais características a seus sistemas de forma rápida, simples e eficiente. Basta selecionar as classes que fornecem suporte às características desejadas no sistema e introduzi-las no mesmo, sem necessidade de recodificação ou mesmo entendimento de seu funcionamento, pois o encapsulamento demanda do programador conhecimento apenas sobre as chamadas a realizar sobre os métodos das classes em questão. Além disso, a manutenção de tais programas torna-se muito mais simples, pois pode ser feita de modo estanque apenas sobre as partes que necessitem alteração. Eventuais melhorias nas implementações dos requisitos não-funcionais podem ser feitas através de simples substituições de meta-classes, sem que haja necessidade de alteração no código da aplicação.

1.3 Mecanismo de reflexão

O modo pelo qual a reflexão computacional atua em um sistema é chamado de mecanismo de reflexão. Através desse mecanismo são implantadas as características anteriormente discutidas. As classes e meta-classes são criadas respectivamente no nível base e no meta-nível da aplicação. Deve-se, então, estabelecer as conexões entre ambos os níveis, de forma que os meta-objetos sejam associados aos objetos que devem monitorar. É importante notar que essa relação não é inflexível nem inalterável, ou seja, um meta-objeto pode estar associado a um ou mais objetos e vice-versa. Da mesma forma, as conexões podem ser estabelecidas e terminadas durante a execução da aplicação.



Associação de objetos do nível base ao meta-nível

Quando um objeto do nível base associado a um meta-objeto recebe uma chamada a um de seus métodos, essa chamada é redirecionada ao meta-nível, em um processo denominado reificação (LISBÔA, 1997). Este processo de ativação do meta-nível pode ser realizado de forma explícita pela aplicação ou implícita.

1.4 Protocolos de meta-objetos (PMO)

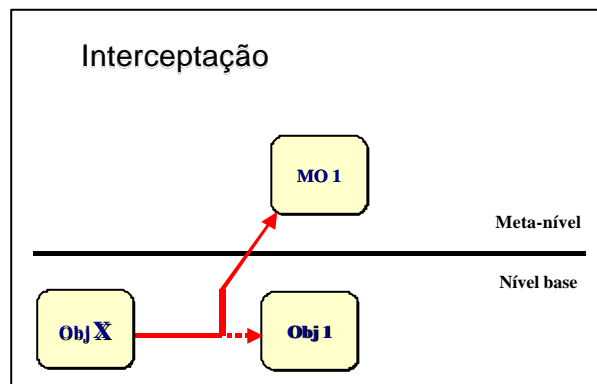
Para que a reflexão computacional seja aplicada, é necessária a existência de algum mecanismo (PMO – protocolo de meta-objetos) que permita a troca de mensagens e a passagem da execução do programa entre os dois níveis, ou seja, que possibilite a comunicação e monitoração do nível base pelo meta-nível e a tomada de controle do programa por este (LISBÔA, 1997). Essas informações precisam de uma interface para serem enviadas entre os níveis. Zimmermann (ZIMMERMANN, 1995) classifica os protocolos em implícitos ou explícitos, de acordo com o modo de ativação do meta-nível:

- *implícitos*: apresentam mecanismos para perceber e interceptar uma chamada a um método e redirecioná-la;
- *explícitos*: precisam que uma chamada seja feita de forma explícita no programa do nível base.

Os protocolos de meta-objetos também permitem um total controle sobre os objetos no nível base. Isso significa que é possível para um meta-objeto descobrir todos os atributos e métodos de um objeto ou classe, além de seus valores correntes. Mais que isso, a reflexão possibilita a criação em tempo de execução de novos objetos e a alteração de valores de atributos de objetos existentes.

1.5 Ativação do meta-nível

Há dois modos de transferência do controle do programa para o meta-nível. O primeiro é feito de forma explícita, através de um comando no programa do nível base. Desse modo, para que as computações a serem realizadas no meta-nível sejam disparadas, é necessário que seja inserida uma instrução explícita no nível base. O grande problema dessa técnica é a diminuição do grau de transparência obtido pela aplicação. Já na forma implícita, denominada interceptação de mensagens, não há necessidade de tal chamada. Cada vez que uma mensagem é enviada a um método de um objeto do nível base relacionado a um meta-objeto, tal mensagem é interceptada e redirecionada para o meta-objeto no meta-nível da aplicação. Esse mecanismo de interceptação deve ser implementado e suportado pelo protocolo de meta-objeto em utilização (PMO). Tais protocolos são implementações de reflexão computacional e apresentam diferentes características, conforme será visto.



Interceptação de mensagens

Quando a mensagem interceptada chega ao meta-objeto no meta-nível, este pode então decidir que procedimentos tomar e executar as computações necessárias. Um protocolo que suporte interceptação de mensagens pode permitir diversos cenários. O meta-objeto pode simplesmente retornar o controle do programa ao objeto do nível, sem executar computação alguma, pode interceptar a mensagem, realizar o que for necessário e retornar o controle ao objeto chamado, ou ainda interceptar a mensagem, realizar as computações que decidir necessárias e, a seguir, retornar o controle não ao objeto chamado do nível base, mas à origem da mensagem, sem que o destinatário sequer perceba que houve uma mensagem enviada a ele.

1.6 Implementações existentes

A despeito das possibilidades que a reflexão computacional oferece, ainda não é grande o número de linguagens de programação com suporte a tal paradigma. Entre elas, destaca-se a linguagem Java (SUN, 1999), que disponibiliza, através de uma API reflexiva, diversos serviços relacionados a esse paradigma. Tal API, embora limitada, permite a implementação de conceitos reflexivos básicos. Essa limitação, todavia, pode ser superada através do uso de outros protocolos de meta-objetos. Um dos mais completos é o meta-protocolo Guaraná, desenvolvido na Unicamp por Alexandre Oliva (OLIVA, 1998).

Há diversos protocolos de meta-objetos além do já citado Guaraná, como MetaJava (GOLM,1997) e OpenJava (TATSUBORI, 1999). Cada um desses protocolos apresenta características de implementação próprias, contudo praticamente todos ampliam os serviços da API nativa Java.

Alguns desses protocolos geram código intermediário que deve ser compilado pelo compilador Java, enquanto que outros alteram a máquina virtual Java para que esta suporte suas novas funcionalidades.

2. REFLEXÃO COMPUTACIONAL E REQUISITOS NÃO-FUNCIONAIS

2.1 Introdução

Uma das utilizações mais freqüentes da reflexão computacional tem sido a introdução transparente e reutilizável de requisitos não-funcionais. Estes são tipicamente processamento distribuído, persistência de dados, controle de concorrência e tolerância a falhas. Sua implementação ocorre através de meta-classes. Quando determinado objeto necessita suporte de algum desses requisitos, basta que seja associado à meta-classe correspondente. Isso torna tal implementação transparente à aplicação, pois a mesma não precisa ser alterada para suportar o requisito desejado, assim como possibilita alto grau de reutilização, pois as mesmas meta-classes podem ser utilizadas em quaisquer outros sistemas. A manutenção de tais sistemas é simplificada pelo encapsulamento dos requisitos em meta-classes.

2.2 Gerenciamento de atomicidade

Atomicidade é a propriedade de um programa que permite que suas operações sejam executadas de forma única, ou seja, ou todas são concluídas ou deve-se voltar ao estado inicial anterior ao início da execução da primeira operação (DATE, 1991). Seu gerenciamento prevê dois aspectos fundamentais: o controle de concorrência e a recuperação de estados (JALOTE,1995). O controle de concorrência impede que um objeto seja alterado por dois métodos ao mesmo tempo, levando a inconsistências. A recuperação consiste em meios de garantir a volta ao estado inicial do objeto em caso de falha de alguma das operações atômicas sobre o mesmo. Para atingir tal controle, é possível o desenvolvimento de diversas meta-classes, cada uma implementando um método diferente de controle de concorrência e recuperação de estados. Outras classes seriam responsáveis pela decisão do método a ser utilizado. Desta forma, é possível realizar reconfiguração dinâmica do programa, obtendo-se um alto grau de adaptabilidade às mais diversas condições de execução. Em um determinado ambiente de rede com alto tráfego de dados, por exemplo, pode-se optar por métodos que utilizem menos comunicação entre objetos. Assim, a reflexão computacional pode ser uma ferramenta vital no desenvolvimento de software autoconfigurável.

Ao associar-se um objeto base a um meta-objeto, todas as mensagens enviadas àquele são interceptadas por este, o que torna o objeto base inacessível a outros objetos da aplicação. Apenas com a concordância de seu meta-objeto poder-se-á acessar tal objeto. Com isso, impõe-se um controle que impede que o objeto seja alterado concorrentemente por outros dois ou mais objetos da aplicação, resultando assim em um controle de concorrência. Da mesma forma, um objeto do nível base pode ser monitorado por um meta-objeto de maneira que cada vez que o mesmo estiver em vias de ser alterado, o meta-objeto possa salvar seu estado corrente e, caso seja necessário, posteriormente restaurá-lo, retornando à situação anterior a alguma eventual falha, garantindo sua recuperação.

Não apenas o controle de atomicidade, mas também a introdução de outros requisitos não-funcionais pode ser efetivada através do uso de mecanismos reflexivos. A transparência que pode ser obtida é um dos fatores mais importantes a corroborar no seu emprego. Esta pode ser obtida facilmente através da prospecção de atributos e métodos permitida pelos mecanismos reflexivos. É possível, em tempo de execução, descobrir toda a estrutura de um determinado objeto. A partir deste princípio, pode-se chegar a um código totalmente genérico, que não demande conhecimento prévio da classe do objeto a ser tratado. Desta maneira, o mesmo código pode atuar com qualquer classe, incrementando sobremaneira a transparência de tal solução. Uma meta-classe que implemente um determinado tipo de recuperação de estados não ficaria vinculada a uma única classe, pois seu código

descobriria a estrutura da classe e, a partir daí, executaria a recuperação. Uma única solução seria utilizável para qualquer classe.

3. CONCLUSÕES

Embora ainda não largamente difundido e utilizado, o paradigma reflexivo tem se mostrado como importante opção para implementação de diversas características cruciais em sistemas computacionais. Alguns fatores contribuem, porém, para que seu grau de utilização ainda não tenha atingido todo seu potencial. Muitas linguagens ainda não oferecem suporte a este paradigma. Além disso, há vários protocolos, cada qual com características próprias e diferentes graus de implementação de reflexão, alguns deles não muito simples.

Espera-se que, com o aperfeiçoamento dos protocolos existentes e introdução de suporte à reflexão em outras linguagens, todos os benefícios oferecidos por este paradigma possam ser largamente utilizados e tornem mais rápida e menos propensa a erros a tarefa dos programadores.

4. REFERÊNCIAS BIBLIOGRÁFICAS

CAZZOLA, W.; SAVIGNI, A.; SOSIO, A.; TISATO, F. *Architectural reflection: concepts, design and evaluation*. DSI – University of Milano, 1999

DATE, C.J. *Introdução a sistemas de bancos de dados*. Ed. Campus, Rio de Janeiro. 1991

GOLM, M.; KLEINODER, J. – *A platform for adaptable operating-system mechanisms*. Universidade de Erlangen, Alemanha. Disponível via Internet em <http://www4.informatik.uni-erlangen.de/TR/TR-14-97-10.abs.html>. Consultado em 14/12/1999.

JALOTE, P. – *Fault tolerance in distributed systems*. PTR Prentice Hall. New Jersey, 1995. Cap. 6. P. 217-253.

LISBÔA, M.L.B. – *Arquiteturas de meta-nível*. Simpósio Brasileiro de Engenharia de Software. Fortaleza, 1997

LISBÔA, M.L.B. – *Reflexão computacional no modelo de objetos*. Porto Alegre, CPGCC, UFRGS, 1998

MAES, P. *Concepts and experiments in computational reflection*. SIGPLAN notices, NY. OOPSLA 1987

OLIVA, A. – *Guaraná – uma arquitetura reflexiva*. Disponível via Internet em <http://www.sunsite.unicamp.br/~oliva/guarana/index.html>. Consultado em 11/12/1999.

STROUD, R.J.; WU, Z. *Using metaobject protocols to implement atomic data types*. Technical report series nr. 512 University of Newcastle upon Tyne, 1995

SUN - *The Java tutorial – a practical guide for programmers*. Disponível via Internet em <http://java.sun.com/docs/books/tutorial/index.html>. Consultado em 10/11/1999

TATSUBORI, M. – *Open Java*. Disponível via Internet em <http://www.hlla.is.tsukuba.ac.jp/~mich/openjava.html>. Consultado em 5/1/2000.

ZIMMERMAN, C. – *Advances in object-oriented metalevel architectures and reflection*. CRC Press. Caps. 1-4. 1995

SISTEMAS DE VISUALIZAÇÃO DE MÚSICA

Mikhail M. Polonskii¹

RESUMO

Este artigo apresenta uma análise do estado da arte da área de visualização de música. Uma breve história da música visualizada e um esquema generalizado de modernos sistemas de visualização de música são apresentados. Com base na análise e testes de softwares para visualização de música para a plataforma PC, foi elaborada uma arquitetura de sistema de visualização de música para essa plataforma. Um método de programação e controle de processo de visualização foi desenvolvido.

Palavras-chaves: computação gráfica em tempo real; multimídia; música.

ABSTRACT

This article presents an analysis of the state-of-the-art of music visualization. A brief history of visual music and a generalized structure of modern system for visual music production are presented. Based on analysis and tests of the music visualization softwares for PC platform, one new architecture of the system is proposed. A programming and a control methods for the music visualization are developed.

Key words: real-time graphics; multimedia; music.

1. INTRODUÇÃO

Visualização de música (*music visualization*) consiste, essencialmente, em geração, em tempo real, de gráficos animados em resposta ao sinal áudio. Música visualizada (MV) ou música colorida (*visual music or colour music*) é o resultado deste processo que visa alcançar maior efeito emocional por parte de espectadores. MV pode ser definida, por outro lado, como união sinérgica de música e artes visuais, sendo o papel principal da música.

Analisando MV de modo mais generalizado, podemos concluir que esta pertence a mais ampla área de sincronização vídeo-áudio na qual se encaixam, também, filmes visuais abstratos (*abstract films*) e sonorificação de imagens. Um esquema de classificação generalizado de sistemas de sincronização áudio-vídeo é mostrado na Fig.1.

Visualização *on-line* (em tempo real) é feita “ao vivo” ao contrário do processo de visualização *off-line*, que pode ser feito através de alguns softwares de renderização, como, por exemplo, o Adobe After Effects. Visualização em tempo real exige a utilização de softwares especiais que pertencem à área conhecida como *real-time graphics*.

Visualização automática (Fig.1), para que o resultado seja satisfatório, poderia ser feita com base num programa com elementos de inteligência artificial, pois o processo de percepção de música é indispensável e deveria ser efetuado pelo programa. Não foi encontrada, durante a elaboração deste artigo, nenhuma informação sobre softwares desse tipo.

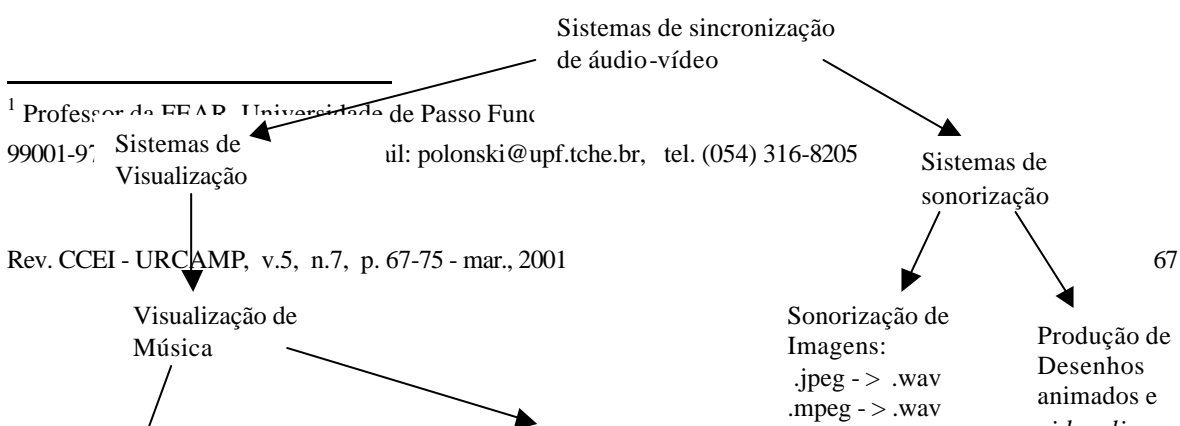


Fig. 1 Esquema de classificação de sistemas de sincronização áudio-vídeo.

Visualização supervisionada é feita através de um processo de programação, executado no modo *off-line*, e um processo de controle no modo *on-line*. Atualmente, existem softwares desse tipo, alguns deles foram testados pelo autor e os resultados desses testes são apresentados na seção 4 deste artigo.

Visualização manual é feita através de software que interpreta apenas comandos do artista e não utiliza informação musical. O artista (o usuário de sistema) pode enviar os comandos de controle através de dispositivos tradicionais, tais como teclado ou *joy-stick*, ou utilizando um instrumento especial, por exemplo, um teclado com interface MIDI (*Musical Instrument Digital Interface*). Nesse último caso o teclado serve apenas para controlar o processo de visualização.

Neste artigo serão discutidas questões de visualização de música no modo *on-line*, sendo a maior ênfase dada à parte técnica (software e hardware) de sistemas supervisionados, sem discutir questões filosóficas e apenas mencionando alguns problemas artísticos. Visualização *on-line*.

2. HISTÓRIA DA VISUALIZAÇÃO DE MÚSICA

O primeiro dispositivo capaz de sincronizar imagens e música foi projetado por volta de 1734 por Louis-Bertrand Castel que era padre, matemático e filósofo. O dispositivo, chamado de Ocular Harpsichord, uma espécie de órgão, gerava luzes coloridas em resposta a movimento de teclas. A construção desse primeiro instrumento áudio-visual continha um conjunto de películas coloridas transparentes que foram iluminadas por velas. Nos séculos XVIII e XIX foram construídos vários dispositivos desse tipo [1].

Uma nova etapa em visualização de música começou após a invenção da eletricidade, que possibilitou a construção de vários instrumentos à base de lâmpadas incandescentes e vidros coloridos. A primeira peça musical composta especialmente para execução áudio-visual foi, provavelmente, *Prometheus: The Poem of Fire* do compositor russo Scriabin,

em 1910 [3]. A estréia aconteceu em 1915 na cidade de Nova York, quando foi utilizado o *Colour Organ* de A. Wallace Remington.

Na década de 1920s, nos EUA, T. Wilfreed construiu seu órgão colorido Clavilux. As imagens geradas pelo Clavilux, o inventor chamou de “Lumia”. T. Wilfreed, além disso, projetou *lumia boxes*, que se constituíam em dispositivos parecidos com televisor moderno, que geravam imagens durante semanas e até meses sem nenhuma repetição. Na mesma época, nos EUA, M. H. Greenwalt, que era uma pianista talentosa, começou experimentações com modulação de luz. Foi ela que inventou o potenciômetro elétrico e a chave de mercúrio líquido para controlar a intensidade de luz. O instrumento áudio-visual que ela inventou foi chamado de Sarabat.

Paralelamente, na Alemanha, W. Ruttman e O. Fischinger começaram a produzir filmes musicais, usando projeção de animação colorida para acompanhar apresentações musicais. Em 1925, o compositor húngaro A. Laszlo publicou sua obra teórica “*Color-Light-Music*” e viajou pela Europa com o instrumento áudio-visual que inventou.

No final de 1930s, C. Dockum construiu o Mobicol Projector, que era capaz de produzir imagens com fronteiras nítidas ou imprecisas. Esse aparelho utilizava padrões preparados antecipadamente que durante a execução podiam ser movimentados e modulados em cores.

No final de 1940s, O. Fischinger inventou o Lumigraf, uma espécie de “órgão colorido”, e com esse instrumento áudio-visual interpretou várias peças musicais.

Com o advento de computadores, a área de visualização de música ganhou uma nova base para realização de antigas e novas idéias e abordagens. Um dos primeiros sistemas para compor MV à base dessa nova tecnologia foi, provavelmente, o GROOVE (*Generating Realtime Operations On Voltage Controlled Equipment*) projetado e programado pela Laurie Spiegel da Bell Telephone Laboratories, Murray Hill, New Jersey, em 1974-1976 [2]. Para essa tarefa, ela utilizou um *mainframe* DDP-224 da década de 60. Esse computador foi dotado de um sistema analógico/digital e foi programado em FORTRAN e *assembler*.

Os êxitos na área de informática, tais como a fabricação de microprocessadores de 32 bits de alta performance, DSP (*Digital Signal Processor*), tecnologia MMX, CD-ROM, DVD e projetores multimídia, possibilitam realizar algoritmos sofisticados em tempo real e fazer apresentações de MV ao vivo ou gravar fitas S-VHS e discos multimídia de alta qualidade.

3. ESTRUTURA DE SISTEMAS MODERNOS DE VISUALIZAÇÃO DE MÚSICA

O processo de visualização consiste em transformação de música, interpretada ao vivo ou pré-gravada, em imagens dinâmicas bidimensionais (gráficos animados). Um esquema generalizado para criar MV é apresentado na Fig. 2.

O sinal áudio (veja Fig. 2) é proveniente de uma fonte, por exemplo, de um áudio *player*, de um microfone, de um *mixer* ou de um instrumento musical. O computador é dotado de uma placa, por exemplo, Sound Blaster, que executa a digitalização do sinal áudio. No caso de utilização de um instrumento musical com a saída MIDI, o computador deve ser equipado com uma placa adequada e a estrutura do sistema é ligeiramente diferente (Fig. 3).

O computador áudio (Fig. 3) converte sinais MIDI em forma analógica e os envia ao amplificador áudio. Os comandos de controle podem ser digitados através do teclado do computador-vídeo que gera imagens dinâmicas com base nesses comandos e com base nos sinais MIDI. O computador áudio pode, também, ser substituído por uma placa de som, dotada da função MIDI e inserida no computador vídeo.

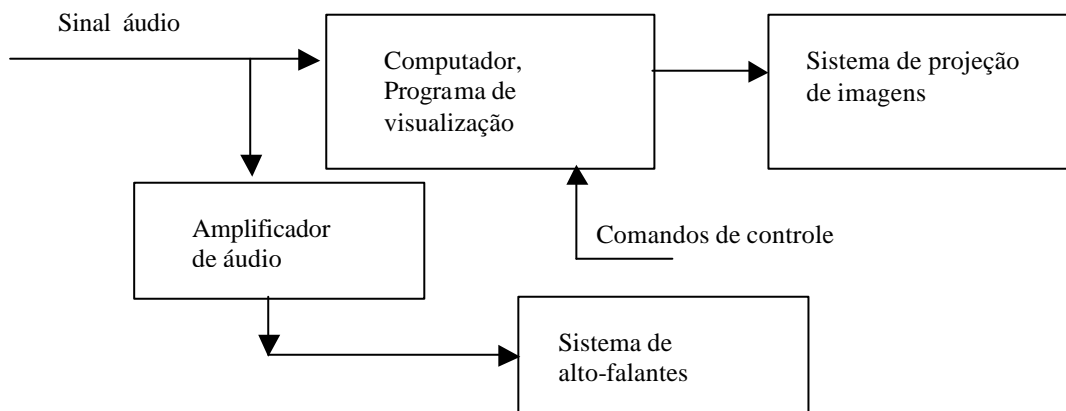


Fig. 2 Esquema generalizado de sistemas de visualização de música

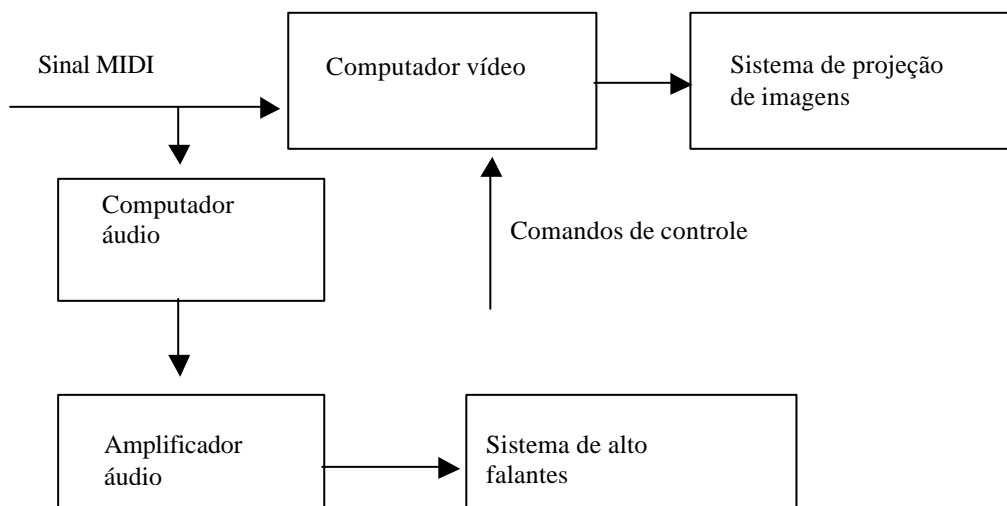


Fig. 3 Esquema do sistema com a entrada MIDI

O sistema de projeção de imagens (Fig. 2 e Fig. 3) para apresentações e shows pode ser realizada à base de um projetor multimídia do modelo VP800 da Lightware, por exemplo. Uma outra abordagem é construir o sistema de projeção à base de um conjunto de lâmpadas coloridas dotadas de refletores e controladas por *dimmers*.

4. SOFTWARES PARA VISUALIZAÇÃO DE MÚSICA

A parte mais importante de qualquer sistema moderno de visualização de música é o software que gera gráficos em tempo real (*real-time graphics*). Atualmente existem vários softwares (programas) de visualização, disponíveis em versões para computadores do tipo PC (DOS e Windows), Macintosh, Amiga, estações de trabalho (*workstations*) e até supercomputadores. Alguns desses programas foram testados durante a elaboração desse artigo e informações sobre outros foram encontradas na Internet. Além disso, uma ampla

lista de softwares para visualização de música com *sites* correspondentes pode ser encontrada na referência [3].

A maioria de softwares para MV com características avançadas foram desenvolvidas para a plataforma Macintosh. A essa categoria se referem os seguintes softwares: **Multicolor**, de Jeff Hoekman, [4]; **The Piano of Light**, de Jeffery Burns [9], **Max**, da Opcode [17]; **Imager**, de Fred Collopy [3]; **Sonnet**, de Fred Collopy *et al* [16]. Esses softwares, embora cada um deles possua suas particularidades, exploram ao máximo o excelente sistema gráfico da plataforma Macintosh.

O nicho do mercado que ocupa computadores Macintosh não é tão significativo como o de computadores compatíveis com PC IBM. Para a plataforma PC, também foram desenvolvidas várias softwares que podem servir para produção de MV. A seguir serão apresentadas informações sobre softwares de visualização de música para microcomputadores PC. Referente aos softwares, testados durante a elaboração desse artigo, serão apresentadas informações técnicas adicionais resultantes de uma análise crítica feita pelo autor.

Performer, da AuVisual [5], é um software *player* para PC Windows que gera imagens em resposta à entrada MIDI. As imagens geradas pelo Performer são parecidas com imagens geradas por laser sobre um fundo preto e são essencialmente figuras simétricas em relação aos eixos vertical e horizontal da tela do computador. Programação ou controle em tempo real não foram realizados nesse software.

ZipArt, de Paul Friedland [6], é um programa escrito em Delphi para Windows. Esse programa não possui nenhuma interface musical e permite apenas criar em tempo real imagens baseadas nas funções seno e cosseno, sendo o processo de criação baseado exclusivamente na percepção de música pelo artista.

Bomb, de Scott Draves [7], gera imagens suaves, automaticamente, ou em resposta ao teclado e/ou entrada áudio. Esse software, que está disponível nas versões para Windows, Linux, Macintosh e UNIX, é baseado numa fascinante mistura de modos e efeitos diferentes, porém as imagens geradas são bastante monótonas. O software trabalha com texturas, *cellular automata*, *harmonic geometry*, fractais e utiliza uma ampla biblioteca de imagens escaneadas. Infelizmente, o funcionamento desse programa não é previsível porque é em alto grau aleatório.

Visual Music Tone Painter, de Stephen Nachmanovitch da Free Play Production [8], converte sinais MIDI em imagens dinâmicas em tempo real, e está disponível apenas para licenciamento temporário para apresentações, concertos e shows. Para criar imagens, o programa utiliza vários objetos geométricos desde simples círculos, polígonos, espirais e ondas até as mais complexas formas tridimensionais (3D) como, por exemplo, cubos e figuras de Lissajous.

MIDIART, de Tony Fragiocomo [10], é um conjunto de dois softwares para PC Windows. O primeiro software, o MIDIART Live, utiliza arquivos MIDI ou entrada MIDI para gerar gráficos animados. O programa possui aproximadamente 20 geradores de formas que executam mapeamento direto de informação musical contida em MIDI, gerando, também, arquivos que podem ser rodados posteriormente.

O segundo software, o MIDIWorks, é usado para executar arquivos criados através do MIDIART Live e assim reproduzir MV. O MIDIART trabalha com uma paleta de apenas 256 cores e gera imagens de formas simplificadas. A abordagem de visualização adotada, ou seja, mapeamento direto de informação MIDI em gráficos através dos geradores, simplifica a sincronização de vídeo com áudio em detrimento de possibilidades de expressão artística.

3DMidi, de Marc Cardle [11], é mais um software para plataforma Windows, para criar MV com base em informação MIDI. Para rodá-lo, é preciso utilizar uma placa aceleradora compatível com OpenGL. O 3DMidi é parecido com o MIDIART, porém mais avançado, gera animação 3D e oferece uma sofisticada interface gráfica com o usuário. Analogamente ao MIDIART, ou qualquer outro software com mapeamento direto de informação MIDI para gráficos, o 3DMidi mostra uma sincronização quase perfeita entre vídeo e áudio. Além disso, o usuário pode modificar parâmetros e estilo de visualização no modo *on-line*.

Cthugha, de Kevin “Zaph” Burfitt e companhia [12], é um programa de visualização para plataformas de Windows, Linux, Macintosh e *workstations*. As primeiras versões foram desenvolvidas para ambiente DOS, sendo a última versão, a de número 5.3, lançada em 1995. A última versão para Windows que foi liberada em 1998, não é estável e é muito lenta em comparação com a versão para DOS. Existe, também, uma versão 3D que exige a utilização de uma placa aceleradora.

Cthugha gera imagens suaves que são moduladas por sinais áudio. O software utiliza vários padrões básicos, tais como “fogo” e “plasma”, cujas imagens são transformadas e distorcidas através de arquivos de tabelas. Além disso, o software usa arquivos de paletas e arquivos gráficos no formato PCX.

Esse software possibilita salvar parâmetros de imagens e posteriormente projetar essas imagens na tela novamente. A sua principal desvantagem é que o seu funcionamento é aleatório e não existe nenhuma possibilidade de começar a executar um certo padrão básico com um determinado tipo de transformação. O que é possível, é simplesmente começar a executar um dos padrões básicos, o que é feito através do teclado, ou o software inicializa um dos padrões básicos, automaticamente, depois de um intervalo de tempo, o qual, também, é aleatório.

A utilização de imagens prontas no formato PCX, cujo número pode chegar a vinte, aumenta a possibilidade de expressão artística, pois possibilita preparar imagens adequadas para cada peça musical. Infelizmente, a escolha da imagem, que é feita através de uma só tecla, também é aleatória. Além disso, a imagem escolhida pode sofrer distorções e transformações inesperadas porque o Cthugha não distingue padrões básicos, como “fogo” ou “plasma”, gerados em tempo real, de imagens pré-gravadas.

Panta Rhei, de Michael Hübner [13], é um programa de visualização de música para a plataforma Windows. Esse software, sem dúvida, foi inspirado pelo Cthugha e é mais avançado e mais estável se comparado com esse último. O Panta Rhei usa, além de arquivos de tabelas do Cthugha (.tab), mais um tipo de arquivos de tabelas com a extensão .COO, os quais possuem a mesma estrutura e tamanho que os primeiros e servem para executar distorções esféricas de imagens na tela.

Os experimentos com esse software mostraram que o Panta Rhei não tem sucesso com todos os estilos de música. A sua principal desvantagem é a impossibilidade de programar o processo de visualização. Além disso, o Panta Rhei não é estável e frequentemente apresenta a falta de sincronismo entre áudio e vídeo.

Geiss, de Ryan M. Geiss [14], é um aplicativo para Windows. Há duas modalidades do programa: *screensaver* (protetor de tela) e *plug-in* para WinAmp *player* [15]. Como *screensaver*, o programa pode ser utilizado para visualização de sinais áudio digitalizados através de uma placa de som. Como *plug-in*, esse aplicativo pode ser utilizado, em conjunto com WinAmp, para produzir MV a partir de arquivos de áudio de formatos diferentes, inclusive o MP3.

A última versão, a de número 4.24, possibilita salvar os parâmetros de animação gerada, aliás aleatoriamente, em tempo real; assim podem ser salvos até vinte e cinco conjuntos de

parâmetros. Cada um desses conjuntos poderá ser, posteriormente, chamado diretamente, isto é, não no modo aleatório, e assim será realizada uma visualização mais controlada do que no caso do Cthugha ou Panta Rhei.

Como um dos pontos fracos do Geiss, podemos considerar a impossibilidade de sintetizar imagens conforme a imaginação e criatividade artística do usuário. O usuário deve esperar até quando uma imagem exata aparecer na tela e, em seguida, salvar os parâmetros dessa imagem para uso posterior. Outra desvantagem é que o software não pode projetar na tela imagens pré-gravadas, por exemplo, imagens escaneadas ou criadas através de programas como Photoshop.

A experiência adquirida durante a utilização do Geiss mostra que esse programa não monitora o ritmo de música e que os movimentos de imagens na tela dependem da resolução escolhida e da velocidade do processador. Por exemplo, se rodarmos o Geiss num Pentium de 100Mhz e num AMD K6-2 de 450MHz, a diferença entre velocidades de deslocamento de imagens na tela nestes dois casos é de pelo menos duas vezes. Esse defeito prejudica a criação de MV, pois, às vezes, atrapalha significativamente a sincronização ente vídeo e áudio.

GEM e **PD** [18], sem dúvida alguma, é o mais avançado software desse tipo para a plataforma PC. O software permite a programação *off-line* através de uma linguagem visual, porém o controle *on-line* é bastante problemático. A maior desvantagem desse software é a necessidade de utilizar hardware especial para projetar na tela grande apenas uma parte da tela do monitor do PC.

5. PROPOSTA DE UMA ESTRUTURA DE SISTEMAS DE VISUALIZAÇÃO DE MÚSICA PARA A PLATAFORMA PC

Levando em consideração a análise dos softwares de MV para a plataforma PC feita na seção anterior, podemos concluir que todos eles não estão livres de defeitos. Antes de propor uma estrutura de sistema de MV para PC, que possua características avançadas e seja livre de defeitos típicos para sistemas existentes, vamos formular os pré-requisitos para um sistema de MV ideal. Os pré-requisitos são: uma interface gráfica amigável com o usuário, facilidade de programação e controle de sistema, diversidade de estilos de visualização, funcionamento determinado e não aleatório, meios para atender exigências de uma imaginação artística sofisticada.

Obviamente, preencher todos estes pré-requisitos é impossível, então será feita uma tentativa de se aproximar o mais perto possível de um tal sistema de visualização ideal. A idéia principal é utilizar no sistema primitivo, gráficos de alto nível, tais como “fogo”, “plasma”, “cascata”, “água”, “*star fields*” etc., com seus parâmetros ajustáveis durante o processo de programação (*off-line*) e durante o processo de visualização (*on-line*). Assim, o sistema vai funcionar nos dois modos: no modo de programação e no modo de execução.

No modo de programação, o usuário poderá selecionar primitivos gráficos para a execução e editar seus parâmetros através de uma interface gráfica com caixas de diálogo, botões e *sliders*. O conjunto desses primitivos editados vai compor o tal projeto de execução. O projeto de execução vai aparecer na tela de controle do computador como grupo de ícones que podem ser clicados e, em conseqüência, as imagens dos primitivos correspondentes serão exibidas na tela de visualização. Assim, a estrutura de hardware do sistema pode ser composta de dois computadores ou um só, mas com dois monitores vídeo (sob Windows 98).

Durante a execução, o usuário trabalha com o projeto de visualização, exibido na forma de um conjunto de ícones. Para projetar no *display* ou, através do projetor multimídia, na tela grande um certo primitivo gráfico, é preciso simplesmente clicar o ícone correspondente. Além disso, para obter uma indispensável liberdade de improvisação artística, o usuário

poderá editar os parâmetros dos primitivos gráficos durante a execução, ou seja, no modo *on-line*.

O sistema de visualização de música proposto pertence, conforme o esquema de classificação da Fig. 1, aos sistemas de visualização *on-line* supervisionados. Esse sistema é programável, usando uma linguagem visual de alto nível, e o comportamento dele é totalmente determinado. O usuário, para controlar o processo de visualização em tempo real, poderá utilizar o *mouse*, o teclado ou o *joy-stick*. O método de programação é simples e poderá ser feito através de uma interface gráfica amigável. O sistema será livre de um dos defeitos típicos de sistemas de visualização de música – perda de sincronismo entre áudio e vídeo. Isso acontecerá devido à possibilidade para o usuário de escolher os momentos certos de troca de primitivos e, também, em função de possibilidade de ajustar a velocidade de deslocamento e/ou deformação de imagem gráficas na tela de visualização. Os ajustes dessa velocidade poderão ser ou programados ou feitos em tempo real, isto é, no modo de execução.

Obviamente, o método de visualização proposto aqui subentende um certo trabalho preparatório, ou seja, subentende a etapa de programação. Embora essa etapa não seja absolutamente obrigatória, ela condicionará um resultado melhor.

CONCLUSÕES

Como se pode ver, a área de visualização de música, com o advento de microprocessadores de alta performance e meios de multimídia, ganhou, realmente, uma nova base para o seu desenvolvimento. Verificou-se que, atualmente, existem vários sistemas de produção de MV baseados em computadores compatíveis com PC IBM, em aparelhos de jogos (Play Station), em Macintosh, em Amiga, em estações de trabalho e até em supercomputadores. No entanto, constatou-se que, até o momento, o software, que é a parte mais importante do sistema de visualização, foi o menos desenvolvido para computadores PC.

Através de uma análise de informação disponível e testes feitos com vários softwares de visualização, foi proposta uma nova estrutura de hardware e software para PC IBM. O sistema proposto é programável e seu desempenho é determinado, isto é, não é aleatório. A programação será fácil, pois poderá ser feita através de uma linguagem visual com uma interface gráfica amigável. Improvisações na visualização em tempo real, ou seja, durante a execução, serão possibilitadas através do proposto método de controle *on-line*.

REFERÊNCIAS BIBLIOGRÁFICAS

1. MORITZ, W. The Dream of Colour Music and the Machines that Made it Possible. Animation World Magazine. Issue 2.1, April 1997.
2. SPIEGEL, L. Graphical GROOVE: Memorium for a Visual Music System. 1998. <http://www.dorsai.org/~spiegel/btl/vampire.html> .
3. Visual Music. Imagers & Lumia. <http://imagers.cwru.edu/index.html>
4. <http://www-ccrma.stanford.edu/~jhoekman>
5. <http://www.auvisual.com>
6. <http://proskovi.clara.net>
7. <http://www.draves.org/bomb>
8. <http://www.freeplay.com>
9. <http://www.snafu.de/~jeff>
10. <http://www.midiworks.com>
11. <http://members.xoom.com/midi3d>

12. <http://www.afn.org/~cthugha/>
13. <http://www.ndh.net/home/huebner>
14. <http://www.geisswerks.com>
15. <http://www.winamp.com>
16. COLLOPY, F; FUHRER, R.M.; JAMESON, D. Visual Music in Visual Programming Language. In IEEE Symposium on Visual Language, Tokyo, Japan, 1999. pp.111-118.
17. <http://www.opcode.com>
18. <http://www.crsa.ucsd.edu/~msp/software.html>

MARKETING: DA SEGMENTAÇÃO AO POSICIONAMENTO, UMA ESTRATÉGIA PARA ENFRENTAR MOMENTOS DE CRISE

Anderson Viera Martins¹

RESUMO

Pequenas e médias empresas em crise, normalmente, sugerem ou adotam como primeira ação para superação dos problemas, o corte de gastos com marketing, criando-se assim um círculo vicioso, no qual as entradas de capital passam a se destinar apenas para manter a empresa, às custas de sacrifícios nas receitas a médio prazo, denunciando a falta de novos investimentos, pondo assim, sua sobrevivência em risco. Tenta-se então, neste artigo, utilizando-se o paralelo entre uma empresa problemática e um doente, encontrar razões mais profundas para a crise e oferecer algumas soluções duradouras do ponto de vista do marketing, tais como a segmentação, a seleção de mercados-alvo e o posicionamento, bem como a utilização de estratégias de comunicação para enfrentar momentos conturbados. Nota-se, portanto, que nos momentos de crise, dá-se o seguinte paradoxo: conta-se com os menores recursos da história da empresa e se requer o máximo de estratégias de comunicação e marketing, sendo necessárias correções de reestruturação do organizacional sistêmico, além de novas atitudes no ambiente interno e externo.

Palavras-chave: Marketing; Segmentação; Mercados-alvo; Posicionamento; Estratégias de comunicação.

ABSTRACT

Small and medium size companies in crisis usually suggest or adopt, as first action to overcome the problems, the cut in marketing expenses, creating thus a vicious circle in which the income becomes strictly a way to maintain the company to the detriment of sacrifices at average date, denouncing the lack of new investments, and in this way putting their survival at risk. Making use of the parallel between a problematic company and a patient, we try in this article to find more profound reasons for the crisis and to offer some lasting solutions from the point of view of marketing, such as segmenting, targeting and positioning; as well as the use of communication strategies to face disturbed moments. There for we note that, in moments of crisis, we have the following paradox: we count on the smallest resources of the company's history and it requires the maximum communication and marketing strategies, being necessary corrections on the restructuring of the systemic organizational; besides we need to take new attitudes towards the internal and external environment.

Keywords: Marketing; Segmenting; Targeting; Positioning; Communication strategies.

1. INTRODUÇÃO

Quando uma pequena ou grande empresa entra em uma crise não circunstancial, uma das primeiras ações que é sugerida ou tomada é a de cortar os gastos com marketing.

Cria-se, assim, um círculo vicioso, no qual as entradas de capital se destinam a manter a empresa “viva”, pagando dívidas atrasadas, geralmente com fornecedores, a fim de não diminuir ainda mais as existências ou evitar o agravamento da situação financeira, sintoma mais clássico da crise às custas de sacrifícios nas receitas a médio prazo, além de pôr em risco sua sobrevivência. Tendo em vista que cortar gastos de publicidade, promoção ou treinamento de vendedores é menos doloroso que despedir empregados ou atrasar pagamentos, a curto prazo, o gerente, dono ou encarregado toma tal decisão com

¹ Bacharel em Administração de Empresas, Doutorando em Ciências Empresariais na Universidad del Museo Social Argentino – UMSA – Buenos Aires, Professor do CCEI – URCAMP Campus Universitário de São Gabriel, e-mail: anderson@sgnet-rs.com.br

consciência de que o “dia do pagamento” de tais decisões será em um médio prazo no qual poderá trabalhar com outras variáveis de ajuste. Nesse caso, compra tempo, sem se dar conta de que é tão só uma ilusão cosmética.

Poucas semanas depois suas receitas denunciam a falta de investimentos e suas finanças não lhe permitem injetar novos recursos porque a crise se agrava ou passa a uma fase terminal.

Seria fácil condenar a miopia do decisor. Porém, todo aquele que tenha passado por tal posição, ou compreenda profundamente o papel do dono de uma micro ou pequena empresa ou o gerente geral de uma empresa média, tem claro que, igualmente ao médico frente a uma emergência, devem-se tomar decisões impostas pela urgência, apesar de sua insensatez.

Tentar-se-á aqui, encontrar razões mais profundas para a crise e oferecer algumas soluções duradouras do ponto de vista “marketeiro”.

A fim de facilitar a exposição do tema, será traçado um paralelo entre a empresa em crise e um doente.

2. OS SINTOMAS

Um doente se apresenta a um consultório médico mostrando sintomas variados de sua enfermidade (baixa nas vendas, problemas de fluxo de caixa, necessidades de crédito, altos custos fixos, problemas de pessoal, etc.).

A relação entre as empresas e os sonhos de seus criadores tem sido amplamente estudada pelo desenvolvimento organizacional. Reconhecer que a crise se deve por razões estreitamente ligadas com disfunções de tal sonho implica, para os decisores, um primeiro passo para a cura ou saída da crise.

Como todo bom médico, tentar-se-á descobrir através dos sintomas, quais são as características mais profundas da enfermidade e, a partir disso, será solicitada uma série de análises para confirmar o diagnóstico.

Entre as razões mais comuns das crises, pode-se enumerar:

a. Mudanças na preferência do consumidor: depois de muitos anos de êxito, o público começa a preferir os produtos de outra empresa ou uma nova categoria de produto. Exemplo: enquanto a única opção que tiveram os brasileiros para a escolha de um automóvel eram os modelos tecnologicamente obsoletos oferecidos pelas revendedoras locais, estes não tinham concorrência; não incorporaram transformações substanciais nos modelos. Ao abrir a importação, e o público ter acesso a modelos japoneses, coreanos, espanhóis, os fabricantes passaram a oferecer níveis similares ou perderam mercados (na realidade ocorreram ambas, tanto aqui como no resto do mundo). Uma das primeiras variáveis para analisar é se estas mudanças são reversíveis (por meio da utilização de recursos como redesenho do produto, reposicionamento dos produtos existentes, extensões de linha, etc.) ou se são definitivas (o mercado para este produto específico desapareceu ou não apresenta a escala necessária para sua produção).

b. Mudanças na logística ou na distribuição: a velocidade de mudança nos esquemas de venda e distribuição se incrementou substancialmente nos últimos anos. Assim, os supermercados e hipermercados têm aumentado sua importância e poder de compra. O ingresso a novos canais suporta conhecimentos específicos que muitas vezes escapam ao pequeno e médio empresário, habituado a vender a um grande número de comerciantes varejistas. Entre as variáveis que devem ser levadas em conta, podem-se citar: a necessidade de utilizar *palets*, as entregas *just in time* nas distintas filiais, o conhecimento dos índices de rotação do produto nas gôndolas, a negociação conjunta e separada com os

diversos gerentes de filiais, os esquemas de comercialização de supermercados – ponta de gôndola, promoções, etc. – o treinamento de repositores, a utilização de código de barras e a manutenção de estoques informatizados.

c. Mudanças na estrutura das empresas: as políticas macroeconômicas implementadas no país têm levado, na maioria dos mercados, à realização de processos de concentração econômica com a presença de importantes empresas multinacionais que aparecem como novos jogadores (concorrentes) ou como concentradores de uma importante parcela do mesmo. Isso força as empresas existentes neste mercado (sejam pequenas, médias ou grandes) a reformular suas estratégias. Algumas das soluções encontradas para garantir sua sobrevivência têm sido: as alianças estratégicas, a reengenharia (para tornar as empresas mais flexíveis e próximas ao cliente), o marketing de guerrilha e a especialização.

d. Mudanças no produto e nos processos de fabricação: frente ao incremento na procura pela qualidade por parte do mercado e do cliente, as sociedades globais começaram a implantar certificados que garantam um mínimo de qualidade nos processos de fabricação. As normas ISO, são, quem sabe, as mais conhecidas e têm forçado muitas empresas a variar substancialmente sua maneira de produzir. Modificar a maneira de produzir inclui a melhoria do controle sobre o produto final. A introdução de tecnologia de fabricação e de gestão pode requerer, em muitos casos, contar com recursos financeiros que a empresa não possua, o que a leva a um caminho de múltiplos antagonismos (necessita de um empréstimo para mudar / nenhum banco empresta a empresas obsoletas / não pode mudar).

3. TERAPIAS ALTERNATIVAS

Tendo chegado a este ponto, o doente (o gerente ou o empresário) começa a procurar “terapias alternativas”, consultar “bruxas” e esperar magia, visto que a medicina tradicional não lhe deu resposta. Recorrerá assim ao que se denomina “marketing selvagem”, que são diversas prestações de serviços realizadas por profissionais de diferentes qualificações, ou por especialistas, que se supõe, poderão descobrir poderosas maneiras de atrair os clientes que até agora nosso paciente desconhecia. Também entram nessa categoria as novas embalagens desenhadas por um jovem estudante de desenho, amigo dos filhos, a campanha publicitária pensada em um dos momentos de folga, ou um plano para pequenos anunciantes de uma rádio local.

Tal como acontece na medicina tradicional, dependerá da doença do paciente para que:

- melhore ou se cure;
- não modifique o estado de sua enfermidade;
- piore;
- morra.

Nos casos “a” e “d”, tendo em vista a natureza da situação, pouco se pode acrescentar. Nos casos “b” e “c”, todavia, quem sabe o paciente se dê conta de que o tempo gasto em dúvidas e em corridas consultórios tenham passado e que deverá enfrentar sem remédio a cirurgia.

Não é pretensão aqui esgotar o assunto sobre as medidas necessárias, porque cada caso é específico. Mas, sem dúvida, algumas das soluções enunciadas no primeiro ponto se aplicarão agora. É importante recordar o caráter sistêmico das soluções a encarar. De nada serve gerar uma nova imagem publicitária se os pontos comerciais e os vendedores não correspondem à mesma. Pouco importam as transformações no produto se não se sabe comunicá-las aos clientes. Desde a estrutura gerencial até o *layout* da empresa, as reformas deverão ser integrais, arriscadas e conseqüentes.

Devido à proposta deste artigo, analisar-se-ão com mais detalhes, somente as medidas que são possíveis de implementar: o marketing segmentado e as estratégias de comunicação para enfrentar uma crise.

4. ESTRATÉGIAS DE COMUNICAÇÃO (OS OUTROS “P” DO MARKETING)

Segundo David Arnold, para entender os fatores críticos de sucesso em uma organização é de vital importância ter uma idéia clara da necessidade que o produto ou serviço satisfaz, enquanto que a diferenciação pela marca depende fundamentalmente das ações de posicionamento para satisfazer os desejos dos consumidores. Os desejos intangíveis ou emocionais dos consumidores são particularmente a fonte de lealdade de uma marca. Nos momentos de crise, a lealdade pela marca (e a posse de uma marca forte) são ativos incalculáveis na hora de batalhar pela sobrevivência. Após muitos anos de investimentos nela, chega a hora de capitalizar nossos investimentos na marca.

Nas lições de marketing básico eram enunciados os 4 “Ps” (produto, preço, promoção e praça) como ferramentas básicas para o trabalho. Esses quatro elementos compreendem o conjunto de atividades ou ferramentas com que conta uma empresa para influenciar o seu mercado alvo. A coordenação desses elementos de maneira sólida, simples e convincente constitui uma estratégia de marketing eficaz. Todavia, desde que foram enunciados por Philip Kotler, os 4 “Ps” têm-se multiplicado, e desde as promoções até o ponto de venda são muitas as novas variáveis que têm alcançado vôo próprio. O marketing se transformou em estratégico.

O mesmo Kotler afirmou que o primeiro passo no Marketing Estratégico é a investigação do mercado no qual se está interessado. Não existe outra maneira de abordá-lo. É necessário saber como é o mercado, quem faz parte dele, que necessidades, que desejos e expectativas existem; a ação destinada para tal é denominada de Investigação de Mercado. Uma vez sondado o mercado, descobre-se que sua composição não é homogênea visto que os consumidores são heterogêneos, ainda que agrupados em conjuntos ou segmentos. A maneira de classificá-los denomina-se Segmentação. O terceiro passo é concluir que não se dispõe de recursos para atender todas as necessidades dos clientes, nem a todos os grupos de consumidores, de maneira que se deve eleger um mercado-alvo. Finalmente o quarto passo: define-se a mensagem que se deseja transmitir. Posiciona-se.

Existem várias maneiras de se comunicar com o mercado-alvo. Uma das mais interessantes para ser analisada é a relação com a imprensa.

No mundo atual, as empresas têm à sua disposição um grande espaço dentro dos meios publicitários para difundir suas atividades. Hoje quase não existe um meio que não disponha de uma seção para Empresas e Negócios. Saber aproveitar essas oportunidades é fundamental. Enfrentar seguro e tranquilo uma entrevista é uma parte importantíssima deste processo. Desse momento depende a imagem da empresa.

Nos momentos de crise esses contatos se intensificam. Seja porque é a imprensa que começa a divulgar os problemas, seja porque, para evitar que ocorra, a empresa encara estratégias específicas de comunicação que incluem contato com os meios.

Esse espaço, todavia, pode chegar a se converter em uma faca de dois gumes se for mal utilizado. Uma informação dada incorretamente, confusa ou em um momento inadequado pode gerar resultados contrários àquilo que se pretendia. Ver na imprensa e nos jornalistas a encarnação do demônio não conduz a parte alguma. Saber se relacionar com os meios é um ponto chave na capacitação do gerente de hoje.

Nunca se deve esquecer que a informação é o ponto central da relação entre uma empresa e os meios de comunicação. Esse é o canal através do qual se deve buscar o caminho. A imprensa serve para dar conhecimento de informações e dados às atividades empresarias;

ao mesmo tempo, demanda que essas notícias sejam verdadeiras, claras e se revistam de um real interesse. Quando um jornal publica dados irrelevantes ou que não importam a ninguém, abre-se um espaço propício para a suspeita de que houver interesses comerciais na notícia.

5. COMO ENFRENTAR UMA CRISE?

Freqüentemente ocorrem graves incidentes – industriais, financeiros, sindicais, etc. – dos quais toma-se conhecimento através da imprensa. A cobertura jornalística desse tipo de fato tem aumentado o interesse público sobre as atividades próprias do mundo dos negócios, como por exemplo, as medidas de segurança e melhorias na qualidade de vida do funcionário que as empresas vêm adotando no dia-a-dia de suas atividades.

O efeito desses incidentes, especialmente quando se trata de situações mais ou menos críticas, pode gerar uma publicidade desfavorável ou afetar a imagem da empresa envolvida. Todavia, os efeitos negativos podem chegar a se reduzir ao mínimo caso sejam tomadas medidas rápidas e eficazes.

Nos momentos de crise, sempre é melhor considerar a imprensa como uma aliada do que como uma inimiga. É necessário tomar a iniciativa e comunicar as situações conflitantes antes que estas cheguem confusas ou distorcidas às redações. A melhor forma de contar os acontecimentos é tal qual ocorreram, ou seja, é ser transparente.

Os boatos: muitas vezes uma notícia que não está correta começa a circular e se transforma em uma verdade para todos. O que fazer para que essa informação não continue circulando e prejudique a imagem da empresa?

Em primeiro lugar, é necessário que se desminta o boato na própria imprensa. Através de memorandos ou reuniões, o pessoal da empresa deve saber que este dado que circula é falso.

Uma vez que o foco interno foi controlado, deve-se enfrentar a parte mais difícil: convencer a opinião pública de que o boato é falso. Para obter êxito, é necessário enfrentar o problema de frente e sair a desmenti-lo. Nada parece mais verdadeiro que um segredo não negado por ninguém.

O desmentido deve ser claro e não pode deixar lugar para dúvidas ou resistências. A melhor forma de explicar um fato é contar exatamente o que aconteceu. Porém, deve-se evitar o excesso de otimismo. As declarações podem ser feitas a vários meios de forma conjunta (organizando uma entrevista coletiva), ou convidando individualmente os jornalistas que possuam mais contato com a empresa.

Contar com uma imagem corporativa séria e forte, além de desenvolver “cintos de segurança” que tenham a capacidade de amortecer a publicação de boatos, assim como contar com a segurança de que a versão da empresa será escutada e incluída nos informes publicitários são outros passos necessários a serem tomados.

6. CIRURGIA

Voltando à empresa em crise, outra das possíveis opções que se deve analisar é a possibilidade de realizar ações de marketing segmentado. O marketing de massa se tornou demasiadamente caro para a maioria das empresas e produtos. Nem mesmo megaempresas como Nestlé ou McDonald's têm resistido à sedução do marketing segmentado.

Philip Kotler define a essência da estratégia do marketing moderno com as siglas STP (*Segmenting, Targeting and Positioning*): Segmentação, Seleção de mercados-alvo e Posicionamento. E acrescenta: “As companhias hoje em dia consideram cada vez menos eficaz a prática do marketing de massa ou o marketing baseado na variedade de produtos.

Os mercados massivos estão se “desmassificando”. Estão se decompondo em “micromercados”.

Em resumo, a proposta deste tipo de marketing inclui a identificação de mercados-alvo (*target*) e a elaboração de estratégias específicas para os mesmos.

Fortemente baseadas no marketing de relacionamento, as empresas tentarão conhecer melhor os seus clientes por meio de estudos de mercado, entrevistas e levantamento de dados sócio-demográficos. A partir desse momento passarão a estabelecer estratégias de comunicação que podem incluir, entre outros, o envio de *e-mails*, chamados telefônicos (*telemarketing*), o envio de *newsletters* com informação periódica de interesse ou de catálogos para venda direta, as reuniões para clientes ou a criação de clubes de afinidade.

Estabelecerão então as mudanças necessárias na comunicação publicitária (de acordo com o novo posicionamento). Como exemplo disso, poderia ser citada a substituição dos informes publicitários nos meios de massa (TV aberta) por informes específicos em programas da TV a cabo, ou a substituição de avisos em jornais de grande circulação, por avisos em revistas dirigidas ao segmento desejado.

Por último, a realização de eventos e estratégias de comunicação corporativa (desde folhetos até as campanhas na imprensa) completarão a imagem da empresa na mente do consumidor. Essas opções, por mais fáceis que pareçam, são de difícil arquitetura e requerem conhecimentos específicos em cada uma das áreas. Confundir chamadas telefônicas com *telemarketing* (que entre outras coisas inclui um guia, a manutenção eficaz de uma base de dados, o trato das resistências do cliente em potencial através do telefone, etc.) implica riscos graves para a realização com êxito das referidas estratégias.

Para o empresário em crise, esses tipos de propostas são tentadoras na errônea suposição de que implicam custos menores. Tal como as pesquisas têm demonstrado, o marketing segmentado não é mais barato. Implica em investir melhor os recursos para influenciar os consumidores que mais interessam, evitando assim o desperdício ou a redundância.

A médio prazo, e uma vez obtida uma base de dados sólida, é possível reduzir alguns investimentos, mas deve-se ter em mente que os índices de resposta de um *e-mail* qualquer variam substancialmente de acordo com as bases de dados utilizadas e o tipo de envio que se realiza, mas raramente superam o índice dos 20% de resposta (e geralmente giram em torno dos 3% e 5%). Nota-se, portanto, que se não for enviada uma quantidade suficiente de prospectos, não se obterá a quantidade necessária de vendas que permita pagar tal investimento (seja este um *e-mail*, uma campanha de *telemarketing* ou um convite a um evento).

7. FATORES EXTERNOS OU A GLOBALIZAÇÃO DA CONCORRÊNCIA

Já em 1983, Theodore Levitt anunciava na *Harvard Business Review* que “existe uma força poderosa que dirige o mundo para uma uniformidade convergente e se chama tecnologia. O resultado dessa corrente apresenta uma nova realidade comercial – o surgimento de mercados globais para produtos de consumo estandardizados em uma escala nunca antes imaginada. As companhias que venham a fazer parte dessa nova realidade podem obter benefícios de imensas economias de escala em produção, na distribuição, no marketing e no gerenciamento de sua estrutura organizacional. Traduzindo esses benefícios em preços reduzidos em nível mundial, ficam em condições de dizimar todos aqueles concorrentes que ainda se apegam a velhos conceitos que não condizem com a realidade mundial. Já desapareceram as diferenças entre as preferências nacionais ou regionais. A universalidade das preferências conduz, inexoravelmente, a estandardização de produtos, das formas de produção e das instituições comerciais e industriais”.

Em geral, as profecias de Levitt provaram estar corretas, em particular, as empresas encontraram algumas soluções para esse tipo de concorrência global. Muito além dos volumosos orçamentos publicitários investidos por marcas globais e o nascimento de meios de comunicação transnacionais, a pesquisa de mercado demonstrou que as diferenças locais ou regionais ainda subsistem e que existem segmentos insatisfeitos que oferecem oportunidades às empresas, o que gera a necessidade de uma reformulação radical de sua estratégia. Tal como nos ensinaram Al Ries e Jack Trout, em Marketing de Guerra, há mais de uma década, os líderes têm políticas que deixam amplos flancos abertos para o ataque guerrilheiro.

Grande parte da batalha será vencida se os líderes responsáveis pelas mudanças nas empresas em crise aprenderem, rapidamente, a livrar-se das estratégias de produtos que foram vitoriosas no passado e que já não são nos mercados atuais. Segundo afirmaram Hamel e Prahalad, mudar a concepção da missão empresarial – desafiando as definições convencionais do setor industrial e de limites de segmentos, postuladas por líderes – representa uma forma diferente de inovação competitiva. A inovação competitiva se baseia na premissa de que os líderes da concorrência têm a tendência a se “casar” com uma “fórmula” de êxito. Assim sendo, provavelmente a arma mais eficaz que possuem os inovadores é a capacidade de apresentar um plano totalmente novo, repleto de surpresas. A maior vulnerabilidade dos líderes provém do seu apego às políticas e aos procedimentos aceitos como infalíveis.

Tal como a velha dicotomia entre centralizar ou delegar, a discussão entre globalização e mercados locais não só é cíclica, como também permite que ambos os fenômenos subsistam e sobrevivam em qualquer momento da atualidade dos mercados mundiais. Isso vai além das atitudes governamentais que protegerão (ou não) as indústrias nacionais desse tipo de concorrência, tendendo para o desenvolvimento (ou não) de um mercado interno forte frente a este fenômeno de produção e comercialização global.

Todavia, os requisitos para jogar este jogo (suas barreiras de ingresso) têm elevado os custos à mercê do desenvolvimento de normas de qualidade e processos de produção cujo custo de implementação tem-se amortizado em níveis globais em um caso e deve realizar-se em nível local, em outro. Se bem que nos parece impensável, hoje, que o consumidor aceite produtos com padrão de qualidades mais baixos do que aqueles oferecidos pelo mercado nacional, até mesmo porque chegará o momento em que deverá pensar que a diferença de preço (custo) poderia ser o equivalente simbólico das altas nas taxas de desemprego nacional e que deve tomar uma decisão a respeito.

8. CONCLUSÃO

Nos momentos de crise, dá-se o paradoxo de que se conta com os menores recursos da história das empresas e se requer o máximo de estratégias de comunicação e marketing.

Internamente, torna-se necessário que a equipe de trabalho, tanto a nível gerencial como operacional, tenha plena confiança na continuidade da empresa e da viabilidade das medidas a serem adotadas. Nas empresas modernas “todos vendem”, desde a telefonista até o presidente. Se no momento da verdade (o contato com os clientes) o pessoal de linha transmite dúvidas, desilusões e fracassos, pouco é o que uma campanha publicitária, por mais maravilhosa que seja, pode fazer para consolidar o processo de saneamento.

Externamente, devem ser encontrados caminhos para melhor conhecer e se comunicar com os clientes atuais e potenciais de maneira efetiva, econômica e eficiente.

No organizacional sistêmico, trata-se, nada mais nada menos de uma reestruturação, em que tomando alguns elementos do passado e do presente, reformula-se o futuro através de uma estratégia. Um novo posicionamento, a necessária mística, a inteligência e os recursos materiais e humanos são parte substancial de tal reestruturação.

9. BIBLIOGRAFIA

KOTLER, Philip. Administração de Marketing: análise, planejamento, implementação e controle. 5^a ed. São Paulo: Atlas, 1998.

_____. Marketing para o século XXI: Como criar, conquistar e dominar mercados. 2^a ed. São Paulo: Futura, 1999.

KOTLER, Philip. e ARMSTRONG, Gary. Princípios de Marketing. Rio de Janeiro: Editora Prentice-Hall do Brasil Ltda, 1995.

LEVITT, Theodore. The Globalization of Markets. Harvard Business Review, ed. maio, 1983.

NICKELS, William G. e WOOD, Marian Burk. Marketing, relacionamentos, qualidade e valor. 1^a ed. Rio de Janeiro: Livros Técnicos e Científicos Editora S.A., 1997.

PORTER, Michael E. What is strategy?. Harvard Business Review, ed. novembro-dezembro, 1996.

RIES, Al. e TROUT, Jack. Marketing de Guerrilha. São Paulo: McGraw-Hill, 1986.

ROBERT, Michel. Estratégia: Como as empresas vencedoras dominam seus concorrentes. São Paulo: Negócio Editora, 1998.

COMUNICAÇÃO ORGANIZACIONAL: PERSPECTIVAS E MODELOS

Jeanice Jung de Andrade¹

RESUMO

O presente trabalho tem por objetivo discorrer sobre um dos assuntos mais relevantes no processo de administração, a comunicação organizacional. Na introdução, é feita uma retrospectiva do significado da comunicação em seus distintos contextos e sua importância. Na continuidade aborda-se o processo, barreiras, perspectivas e modelos de comunicação organizacional. Nas considerações finais procura-se apresentar a relevância da idéia da comunicação interativa entre o administrador e a organização para o alcance dos objetivos organizacionais. Palavras-Chaves: comunicação, perspectivas, modelos, administrador da comunicação.

ABSTRACT

The objective of this work is to discourse about one of the most important subjects of the management process, the organizational communication. The introduction presents a retrospective about the communication meaning in distinct context and its importance. It continuous, approaching the process, obstacles, perspectives and models of the communication organizational. The finals considerations tries to present the importance of the idea of iterative communication between the manager and the organizational to get the objectives of the organization. Keywords: communication, perspectives, models, communication manager

1. INTRODUÇÃO

A comunicação pode ser vista como um conjunto de fluxos de informações, de diálogo, de influência recíproca e forma de relação existente entre os componentes da organização e com os atores do meio ambiente em que está inserida. A comunicação pode-se definir também como o processo pelo qual a informação é trocada e compreendida por duas ou mais pessoas, objetivando motivar ou influenciar o comportamento.

O complexo processo de comunicação tem um componente educativo e a responsabilidade de desenhar um modelo capaz de fazer fluir e compreender as informações entre os interlocutores internos e externos para atingir os propósitos de desenvolvimento e competitividade das organizações. O fluxo de comunicação envolve:

- articulação das necessidades e situações críticas com o intuito de atingir soluções,
- aproximação dos elementos de apoio e especialistas com os outros elementos do ambiente organizacional externo e interno.

Tem-se, então, um fluxo multidimensional por sua abrangência e importância. Citando CALLETI (1983) *apud* BRAGA e KUNSCH (1993), "*a comunicação como ciência já nasce com uma vocação de poder*", fato que pode ser limitante para os ganhos sociais e interferir na competitividade econômica. O forte componente cultural da comunicação e considerando que o homem amarra-se a cadeias de significados que ele mesmo tece, a presença de um *administrador da comunicação* torna-se relevante e estratégico. Esse administrador tem a função básica de modelar um sistema de comunicação que capture no seu curso os fenômenos "densos" internos, adequando-os à identidade organizacional desejada e conforme o contexto de suas interfaces externas. Ao profissional, *administrador*

¹ Engenheira Agrônoma, Bacharel em Administração e Mestranda em Administração. Prof^ª da Universidade da Região da Campanha -Campus de S. do Livramento. (jeanice@terra.com.br/ jeanice@v-expressa.com.br)

da comunicação, cabe observar e interpretar as "teias de significados" e as relações sociais imergentes, para coordenar com coerência os elos de ligações e atingir os objetivos e metas da organização.

A correta percepção, sem a influência dos ruídos e interferências, é tarefa do *administrador da comunicação* e consiste em "saber ouvir" e pensar. A habilidade de "saber ouvir" está ligada à liderança e é um processo ativo. De acordo com HODGSON (1996) apud MARTINELLI e ALMEIDA (1997), a amplitude do ato de ouvir envolve a postura e expressão facial, hesitação e silêncio, ênfase e inflexões, tom de voz, coisas não-ditas, emoções ocultas do emissor e receptor quanto ao conteúdo verbal da mensagem.

A comunicação é também parte ativa do composto de marketing envolvendo as funções de informar e orientar os consumidores nos seus processos de compra. Adquire formas diferentes de acordo com as necessidades, publicidade, propaganda e promoção de vendas entre outras.

Dentro da estrutura organizacional contemporânea, a comunicação assume duas dimensões: a perspectiva organizacional e a perspectiva interpessoal. A primeira está relacionada com a estrutura organizacional e a segunda com a eficácia da comunicação.

2. IMPORTÂNCIA

O estudo da comunicação organizacional é relevante para conhecer o conteúdo e a estrutura das interações humanas diárias, formar e preparar pessoas adequadas com o conhecimento do processo, ferramentas e treinamento em organizações e também pela forte conexão entre a comunicação efetiva e a efetividade organizacional. Há problemas para o desenvolvimento da área devido à similaridade do conteúdo da comunicação organizacional com os diferentes e muitas vezes estudos competitivos na área que consideram a comunicação como uma "variável".

As pesquisas têm mostrado a importância da comunicação aos administradores para alcançar a efetividade organizacional e sua relevância na motivação dos empregados. Segundo MAYFIELD et alii (1998) há uma relação positiva entre uma liderança na comunicação oral, desempenho e satisfação no trabalho. Esta pesquisa também corrobora a necessidade de treinamento de pessoas em comunicação como suporte e habilidade para conduzir a organização aos objetivos previstos, através do desenvolvimento de programas adequados. Cada tipo de comunicação tem suas regras, mas a comunicação dos administradores eficazes deve ser apropriada para maximizar o desempenho dos trabalhadores. MARTILLA (1971), em suas pesquisas, conclui sobre importância da comunicação verbal nas organizações industriais, identificando em que estágios tem maior influência.

A comunicação compreende aspectos como o alvo, método, estrutura, *feedback*. O alvo definido e claro se faz necessário para a compreensão da comunicação e do que se deseja alcançar com a mesma. A determinação do método e da estrutura da comunicação deve-se encaixar sob a perspectiva da integração. A sinalização do *feedback* pelo receptor deve ser encorajada e considerada para o aperfeiçoamento do processo. O *feedback* como fonte de informações para estimular a comunicação recebida e influenciar o comportamento deve estar focalizado no comportamento, nas observações, em descrições e não em julgamentos, no compartilhamento de idéias, na exploração de alternativas, no volume de informações que se recebe e se pode usar, no tempo e lugar apropriado e no que realmente é dito e não em por que é dito.

3. COMUNICAÇÃO ORGANIZACIONAL

A comunicação organizacional flui através da estrutura hierárquica da organização de forma ascendente, descendente e horizontal. Comumente as informações ascendentes

compreendem problemas e particularidades, sugestões de aperfeiçoamento e disputas, informações de ordem financeira e contábeis. Nessa fase da comunicação, a presença de sistemas de informações gerenciais tem sido uma ferramenta de apoio.

As comunicações descendentes estão relacionadas com a implementação de metas, estratégias e objetivos organizacionais, instruções de trabalho, procedimentos e práticas, *feedback* sobre o desempenho e a transmissão de valores culturais, missão da empresa para a construção e solidificação da identidade organizacional.

A figura seguinte visualiza a comunicação descendente, ascendente e horizontal na organização.

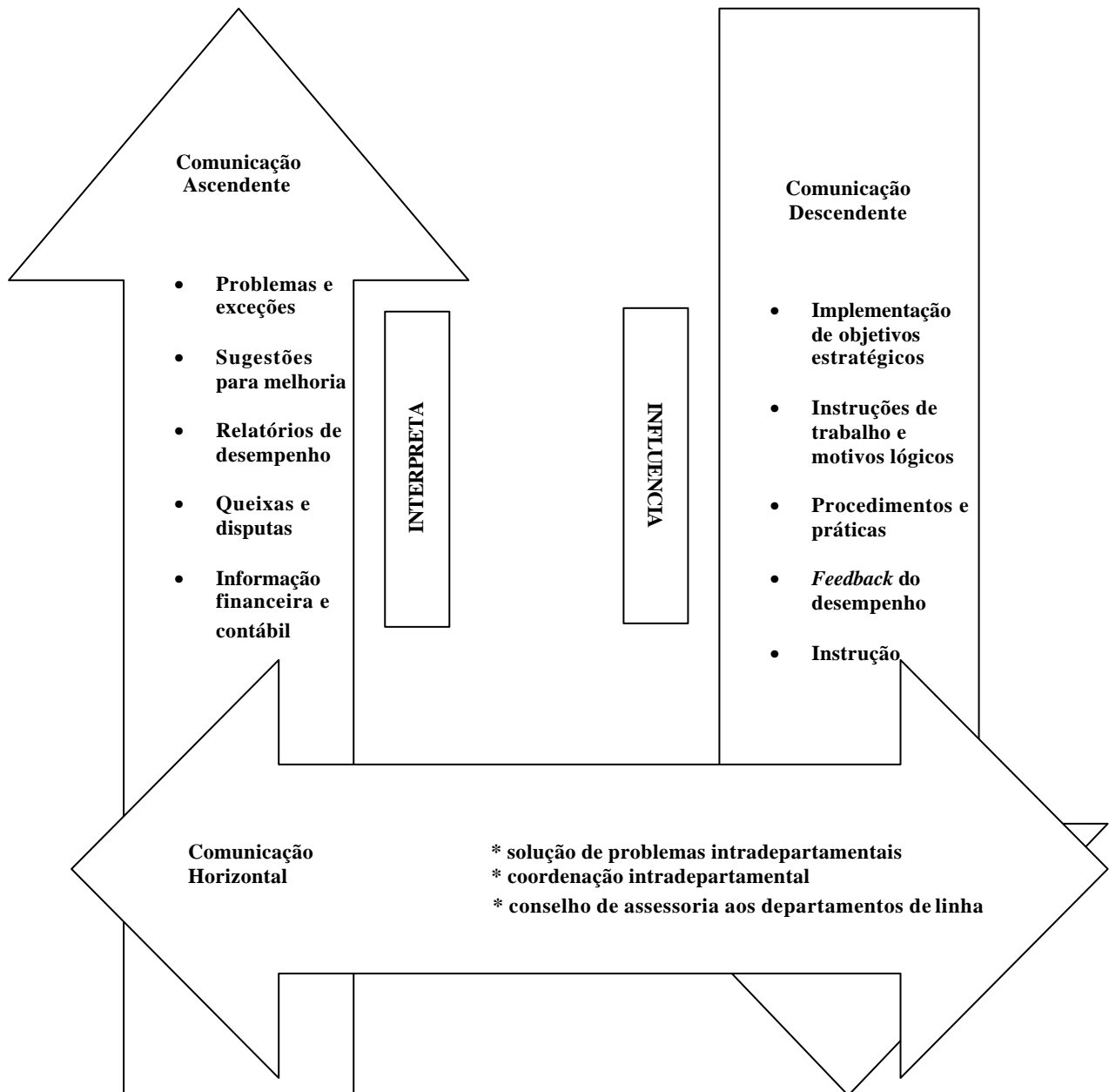


Figura 1. Adaptado por Richard Daft e Ricard M. Steers, Organizations: A Micro/Macro Approach, 1986.

Considerando o canal de comunicação e as mensagens, do canal mais rico para o canal mais pobre, tem-se a seqüência: presença física (conversa frente a frente), canais

interativos (telefone, mídia eletrônica), canais pessoais estáticos (memorandos, cartas) e canais impessoais estáticos (panfletos, boletins, relatórios pessoais).

3.1 - O processo de comunicação

A dinâmica da comunicação nas organizações é necessária, apresenta-se multifacetada e complexa, refletindo as normas, valores clima e objetivos organizacionais em cada função organizacional, de acordo com MINTZBERG (1973), PUTNAM (1982), HEATH & BRYANT (1992) *apud* BUCKLEY, MONKS & SINNOT (1998). Conforme SIMS (1994) *apud* BUCKLEY, MONKS & SINNOT (1998) conclui, é responsabilidade da função de relações humanas iniciar o canal de integração do processo de comunicação na organização, e os últimos autores citados, em suas pesquisas, demonstram o impacto das iniciativas qualitativas no processo de comunicação, levando o processo a estágios de maior maturidade.

Os canais informais de comunicação, ditos de administração interativa (*management by wandering around*) e o boato, não seguem a hierarquia da organização, circulam, conectando-se com qualquer um em qualquer direção na organização. A administração interativa consiste na conversação direta dos executivos de todos os níveis com os seus colaboradores¹, enfatizando-se a comunicação ascendente e descendente. O boato liga todos em todas as direções, proporcionando muitas vezes a ajuda para entender uma situação obscura e incerta. É utilizada pelos colaboradores para preencher lacunas de informações e esclarecer decisões gerenciais.

Na administração interativa, a comunicação é sob forma de redes centralizadas onde os problemas são solucionados mais rapidamente e com menos erros, enquanto a mesma situação em redes descentralizadas torna mais lenta a comunicação. Contudo, nas redes descentralizadas, as tarefas mais complexas são solucionadas com maior rapidez e complexidade, porque num cenário altamente competitivo, utilizam-se equipes para resolver problemas complexos com o compartilhamento das informações. As tendências recentes da comunicação em equipes compreendem a comunicação aberta e o diálogo, focalizando a perspectiva coletiva e não uma visão particular do assunto abordado.

3.2 - Barreiras ao processo de comunicação

As ações de codificar e decodificar são potenciais de erros na comunicação, isso porque o conhecimento, as atividades e o *background* atuam como filtros e passíveis de dar origem aos ruídos/barreiras na comunicação.

As barreiras ou ruídos envolvem aspectos individuais e organizacionais. Enquanto os aspectos individuais abrangem questões interpessoais e semânticas, tipos de canais e mensagens inconsistentes, as barreiras organizacionais relacionam-se com as diferenças de *status* e poder entre os indivíduos envolvidos, diferenças de objetivos e necessidades entre áreas da organização, rede de comunicação inadequada e até a inexistência ou resistência, para implantar os canais informais de comunicação da empresa.

3.3 - Melhoria no processo de comunicação

O *administrador da comunicação* organizacional deve manter uma atitude pró-ativa para o êxito do processo. Assim, alguns princípios são considerados: enfatizar as similaridades das metas e objetivos, reduzindo diferenças e maximizando as similaridades; adequar os estilos de comunicação; conhecer adequadamente os próprios objetivos; conhecer e entender a parte que recebe a comunicação, percebendo o mundo como o outro percebe e pensar/planejar antes de falar.

¹ colaboradores é o termo empregado para funcionários / empregados pelo autor.

Segundo LEWICK et al. (1996) *apud* MARTINELLI e ALMEIDA (1997), ainda devem ser considerados os seguintes aspectos na melhoria do processo: preparar e simplificar a mensagem; utilizar uma linguagem clara; ter atitudes positivas e brincadeiras adequadas; utilizar recursos visuais para reforçar a mensagem; falar de forma lenta e clara para o entendimento; adequar o uso de gestos acompanhando a expressão corporal do receptor (*feedback* não-verbal).

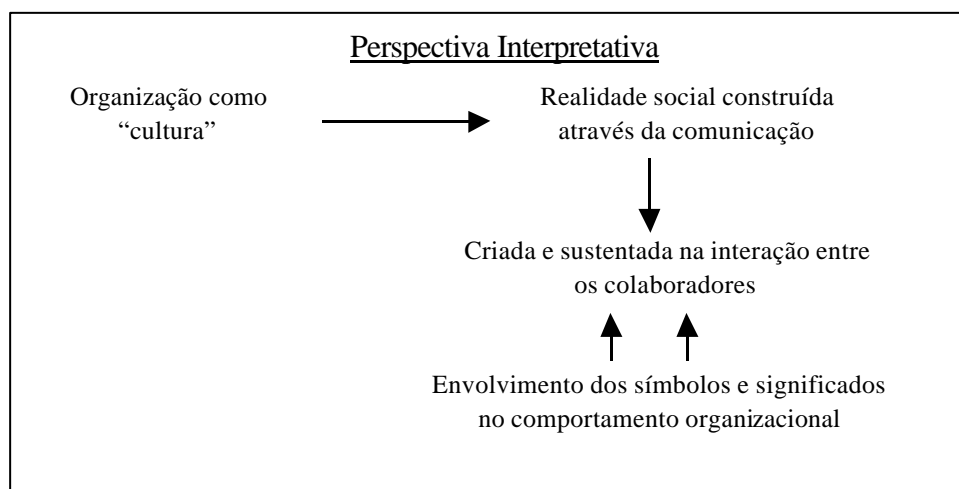
A criação de um sistema de informações, eficiente e amplo, aliado a uma política de "portas abertas" é uma das atitudes iniciais para o avanço da mentalidade sistêmica, onde a comunicação é elemento básico.

De acordo com LEITÃO & MARTINS (1998), desenvolver os sistemas de informações centrados na demanda, faz com que o usuário tenha a liberdade de elaborar suas mensagens através de recursos sistêmicos e melhor compreender os significados das distintas fontes de informação, ajudando a construir conexões associativas que incrementam a inteligência coletiva da empresa. Os autores continuam, concluindo que este conhecimento tácito, agregado às percepções que surgem do diálogo e da troca de informações, é capaz de orientar e provocar a reformulação dos pressupostos gerenciais da organização. A rede de comunicações é colocada pelos autores como componente da rede autopoietica que tem como função participar da produção ou transformação dos outros elementos da rede.

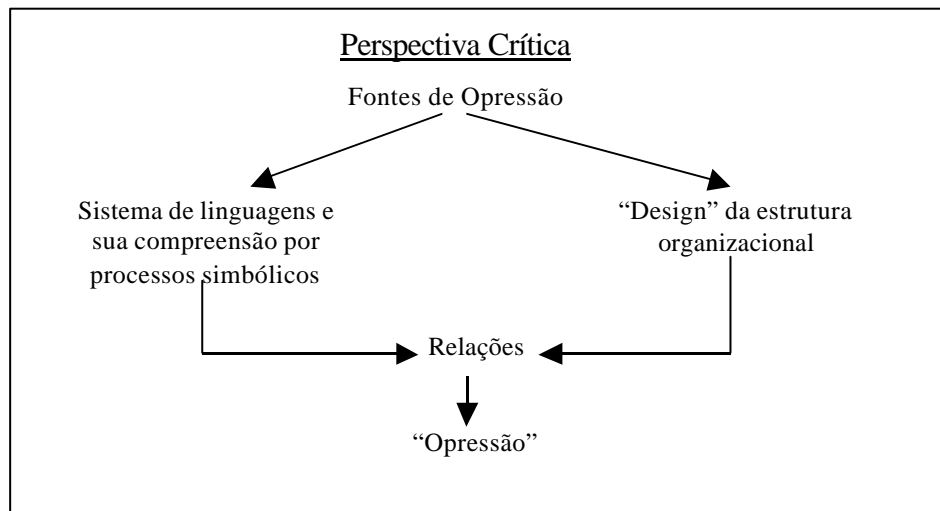
3.4 - Perspectivas da comunicação organizacional

A perspectiva tradicional focaliza os conceitos e métodos tradicionais das ciências sociais e a relação entre processo de comunicação e efetividade organizacional.

Na perspectiva interpretativa da comunicação, os objetivos compreendem descrever as experiências dos membros da organização e sua construção social da realidade organizacional. A organização constrói sua realidade social através da comunicação, numa interação entre indivíduos.



A perspectiva crítica posiciona a organização como instrumento de "opressão". Segundo CALETTI (1983) *apud* BRAGA e KUNSCH (1993), usa o poder no sistema de linguagem e compreensão por processos simbólicos. Há uma distorção sistemática da comunicação.



3.5 - Modelos interativos de comunicação

A comunicação conforme DUNCAN & MORIARTY (1998) pode ser construída num modelo de marketing de administração das relações, numa interatividade entre: as fontes das mensagens na empresa, as fontes das mensagens do marketing, as fontes de mensagens junto ao marketing de comunicação, as interfaces (funcionários, investidores, sistema financeiro, governo, reguladores, entre outros), clientes (distribuidores, fornecedores, competidores - consumidores, comunidade local, mídia e grupos de interesse).

De acordo com JEHIEL (1999), na estrutura ótima de comunicação pressupõe-se:

- que qualquer estrutura de comunicação com N unidades operacionais esteja associada à estrutura do grupo com um tamanho igual ou maior a dois;
- um grupo com ótima estrutura é irreduzível (igual à estrutura ótima do grupo);
- o aumento no tamanho da organização não necessariamente exige a reestruturação dos canais de comunicação e quando existe um grande número de unidades operacionais, a probabilidade de sucesso na comunicação decresce exponencialmente com o número de unidades operacionais.

Conforme SCHULER (1996), o modelo de administração da comunicação empresarial consiste em "*colocar cada um dos elementos do Sistema Concreto da Empresa na posição de cada um dos elementos do Sistema de Comunicação, aplicando-lhe em cada uma dessas posições as funções de administração.*"

De acordo com a autora, tem-se uma administração do todo, sendo cada elemento da empresa passível de ser emissor e receptor de informações, de tornar-se canal de circulação de informações, bem como mensagem por si próprio.

4. CONSIDERAÇÕES FINAIS

A comunicação constitui-se num processo e fluxo complexo, sistêmico e com comportamento interativo. A sua dinâmica reflete as normas, valores, clima e objetivos organizacionais em cada função organizacional. Por sua ação sistêmica e dinâmica, é uma alavanca para a aprendizagem organizacional.

O profissional administrador da comunicação é o elemento responsável em dar forma ao processo de comunicação na empresa, capturando as "teias de significado" e as correspondentes relações sociais emergentes e interativas que surgem na efetividade e motivação das pessoas. Tendo-se claramente determinada a relação da comunicação com o desempenho organizacional, importante é estruturar a comunicação de forma interativa.

O modelo de Administração da Comunicação Empresarial enfocando a estrutura organizacional, pessoal, ambiente e presença física como emissor, receptor, canal e mensagem é uma forma de interação entre a função e a comunicação.

Considerando a perspectiva organizacional e interpessoal, se a empresa estruturar-se para uma comunicação eficaz mas os funcionários não apresentarem habilidades comunicativas, a comunicação como um todo terá problemas. A estrutura organizacional também pode atrapalhar a comunicação mesmo existindo ótimas habilidades comunicativas dos indivíduos da organização.

A comunicação empresarial se constitui no cerne dos processos de negociações e considerando a contínua tomada de decisões e inter-relações organizacionais, avalia-se a relevância do aperfeiçoamento do processo de comunicação.

REFERÊNCIAS BIBLIOGRÁFICAS

BRAGA, Geraldo M. & KUNSCH, Margarida M. K. Organizadores. *Comunicação Rural: discurso e prática*. Viçosa:ed. UFV, 1993.

BUCLEY, Finian, MONKS, Kathy & SINNOT, Anne. *Communications Enhancement: A Process Dividend for the Organization and the HRM Department*. Human Resource Management. Vol. 32, nº 3 & 4: 221-234. Fall/Winter. 1998.

DAFT, Richard. *Administração*. Rio de Janeiro:LTC,1999.

DUNCAN, Tom & MORIARTY, Sandra E. *A Communication - Based Marketing Model for Managing Relationships*. Journal of Marketing. Vol. 62:1-13, abril. 1998.

JEHIEL, Philippe. *Information Aggregation and Communication in Organizations*. Management Science, vol. 45, nº 5: 659-669. Maio.1999.

LEITÃO, Sérgio Proença & MARTINS, Alexandre C. Lima. *Organização de aprendizagem e teoria da autopoiese*. Revista Administração Pública. Rio de Janeiro nº32 (4): 95-109,Jul./Ago. 1998.

MAYFIELD, Jacqueline R., MAYFIELD, Milton & KOPF, Jerry. *The Effects os Leader Motivating Language on Subordinate Performance and Satisfaction*. Human Resource Management. Vol. 37, nº3 & 4:235-248.Fall/Winter. 1998.

MOENAERT, Rudy K., SOUDER, William E., MAYER, Arnoud De & DESCHOOLMEESTER, Dirk. *R & De - Marketing Integration Mechanisms, Communication Flows and Innovation Success*. The Journal os Product Innovation Management. Vol. 11,nº 1:31-45. Janeiro. 1994.

MARTILLA, John. *Word-of-Mouth Communication in the Industrial Adoption Process*. Journal of Marketing Research. Vol.VIII:173-8. Maio.1971.

MARTINELLI, Dante P. & ALMEIDA, Ana Paula de. *Negociação: como transformar confronto em cooperação*.. São Paulo: Atlas, 1997.

SCHULLER, Maria. *A Administração da Comunicação Empresarial*. R. Bibliotecon & Comunicação. P. Alegre, 7:108-125. Jan./Dez.1996.

EXPORTAÇÃO DE APLICAÇÕES MULTIMÍDIA PARA O AMBIENTE WEB UTILIZANDO TOOLBOOK

Rodrigo Ferrugem Cardoso¹

RESUMO

Este artigo apresenta aspectos inerentes a multimídia, em especial no que se refere à exportação para o ambiente Web de aplicativos já desenvolvidos através da ferramenta *Asymetrix Toolbook*. Para melhor ilustrar os resultados, foi realizada a conversão para Web de um software já desenvolvido na área médica utilizando esta ferramenta, com a finalidade de analisar as características principais, descrever os passos necessários e apontar vantagens, desvantagens e problemas com relação à publicação de aplicativos já desenvolvidos.

Palavras-chave: Multimídia, WWW.

ABSTRACT

This paper present aspects about multimedia, specially in exportation for Web environment of previously developed titles with Asymetrix Toolbook tool. For better show the results, was converting for the Web an software developed in medical area whit this tool, with objective of analysis the main proper, describe steps necessities and to point advantages, disadvantages and problems with publication of titles previously developed.

Keywords: Multimedia, WWW.

1. INTRODUÇÃO

Ao contrário de como era há alguns anos atrás, na maioria dos casos, o software utilizado nos computadores está cada vez mais interativo e amigável ao usuário. A principal tecnologia empregada para produzir software que utilize vários recursos como imagens, textos e sons de forma interativa e desta maneira torne-se mais fácil e agradável ao usuário é a multimídia.

Existem várias formas tradicionais de distribuição desse tipo de aplicativo para clientes finais, no entanto, com a revolução da Internet, torna-se imprescindível que os aplicativos multimídia sejam disponibilizados na Web, o que traz uma série de dificuldades e características peculiares, principalmente porque para uma melhor eficácia, os aplicativos devem ser convertidos diretamente através da ferramenta de desenvolvimento, reduzindo assim, tempo, custos e automatizando as tarefas.

No decorrer deste artigo serão detalhados alguns aspectos da ferramenta de autoria multimídia denominada *Asymetrix Toolbook*, sendo que o enfoque principal será sobre as maneiras de disponibilização dos aplicativos desenvolvidos nesta plataforma para Web. Para obter melhores resultados, foram realizados diversos testes de conversão sobre um aplicativo multimídia já existente e utilizado na área médica denominado “ECG – Curso Básico”.

2. MULTIMÍDIA

Como afirma a nomenclatura, o termo multimídia é o conjunto de duas ou mais mídias que podem ser apresentadas de distintas formas, como por exemplo, sons, textos, imagens estáticas, gráficos e filmes.

¹ Professor do CCEI – URCAMP - Mestrando em Ciência da Computação – UFRGS, cardoso@inf.ufrgs.br

De acordo com [VAU94] Multimídia é qualquer combinação de texto, arte gráfica, som, animação e vídeo transmitido pelo computador. Se você permite que o usuário – o visualizador do projeto – controle quando e quais elementos serão transmitidos, isto se chama multimídia interativa. Se você fornece uma estrutura de elementos vinculados pela qual o usuário pode mover-se, a multimídia interativa torna-se hipermídia.

Através da utilização da multimídia existe uma maior aproximação entre usuário e o computador, sobretudo com usuários que não possuem muito contato com a área de informática, o que muitas vezes não ocorre em sistemas que utilizam a maneira de apresentação tradicional. Segundo [ROC92], a multimídia “... *permite uma comunicação mais natural com as máquinas, usando nossos sentidos, visão, audição e tato, e o diálogo homem-computador torna-se mais intuitivo, espontâneo e agradável*”.

3. FERRAMENTAS DE AUTORIA MULTIMÍDIA

Aplicações multimídia podem ser desenvolvidas usando linguagens de programação tradicionais como C++ e outras. Entretanto, além da dificuldade de aprender a forma de programação da linguagem, o número de instruções que devem ser escritas para criar uma simples aplicação multimídia é exaustivo, portanto a melhor alternativa é utilizar Ferramentas de Autoria Multimídia. Essas ferramentas são sistemas de alto nível que fornecem um ambiente unificado para criação de aplicações multimídia. Sua maior vantagem é que podem ser usadas por não-programadores como, por exemplo, especialistas em determinado domínio do conhecimento.

Uma ferramenta de autoria, normalmente, possui ferramentas para hipertexto, figuras, animação, vídeo, som, conectividade com bancos de dados e comunicação entre processos.

Segundo [WOR92] as linguagens de autoria possuem rotinas específicas para o desenvolvimento de um software multimídia. São mais fáceis de utilizar do que as linguagens de programação convencionais mas, mesmo assim, faz-se necessário o conhecimento de conceitos de programação. Essas linguagens possuem recursos tais como: editor de textos, editor gráfico, possibilidade de incluir sons e imagens, entre outros.

4. ASYMETRIX TOOLBOOK

Toolbook é uma ferramenta de autoria criada para implementar cursos “*e-Learning*” e aplicações multimídia. Através de sua utilização é possível implementar cursos sofisticados que podem ser disponibilizados via Web, intranet ou em CD-ROM.

Aplicativos simples podem ser criados facilmente através de objetos pré-programados existentes nos catálogos e convenientes templates. Para necessidades especializadas o *Toolbook* possui uma poderosa linguagem de programação denominada Openscript e está dividido em dois produtos comerciais, o *Toolbook II Instructor* e o *Toolbook Assistant*. A principal diferença entre essas ferramentas é que seu foco principal é distinto, sendo que a primeira é voltada para o desenvolvimento de aplicativos por profissionais da área, que trabalham com programação e podem utilizar os sofisticados recursos proporcionados pelo *OpenScript*, e a segunda, é voltada para usuários finais e possui uma interface bastante amigável que elimina a necessidade de programação permitindo que usuários criem aplicações simplesmente utilizando o mouse. Segundo [SUB96], o *toolbook* oferece um ambiente de programação orientado a objetos e dirigido a eventos; entretanto a linguagem de *scripting* não é orientada a objetos, pois não fornece por exemplo classes, herança ou encapsulamento. No ambiente de desenvolvimento, existem diversos tipos de objetos como botões, caixas de textos dentre muitos outros. Cada objeto pode possuir um script associado a ele, sendo que os scripts somente podem ser criados nas versões *Instructor*. Cada objeto também possui inúmeras propriedades que podem ser modificadas. Através de catálogos, é possível a criação de objetos com ações pré-definidas como, por exemplo, um botão.

4.1 Openscript

É a linguagem de programação do ambiente *toolbox*. Através dela, podem-se programar scripts que permitem a implementação de muitas funcionalidades adicionais ao ambiente. Essa linguagem não está disponível na versão *Assistant*, no entanto está presente em todas as versões *Instructor*. A linguagem também é de fácil utilização, pois sua sintaxe é similar ao inglês e possui uma grande quantidade de comandos. Cabe salientar que os aplicativos desenvolvidos em uma versão *Instructor* e que contenham scripts, quando são exportados ou abertos em uma versão *Assistant*, preservam os scripts originais que, no entanto, não podem ser acessados nesta versão, sendo somente executados.

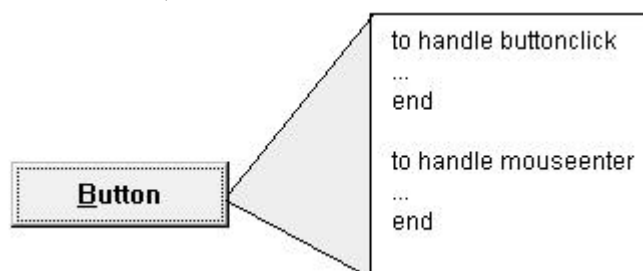


Figura 1 – Exemplo de script associado a um botão simples

5. EXPORTAÇÃO DE APLICATIVOS TOOLBOOK PARA WEB

A família de produtos *toolbox II* introduziu a possibilidade de criar páginas HTML com applets Java embutidos. A natureza do ambiente de desenvolvimento possibilita a criação de poderosas ferramentas interativas que podem ser utilizadas para ensinar, treinar, educar, simular, dentre inúmeras outras tarefas através de um rico ambiente de interação. Este é o resultado de sua capacidade gráfica e sua linguagem de programação.

Apesar da proposta de gerar aplicações independentes de plataforma, o *toolbox* possui completa dependência do sistema operacional Windows. Podem ser criadas páginas HTML baseadas no layout de um limitado número de objetos não-interativos, mas com limitações impostas pelo próprio HTML. Existem também um conjunto de applets Java que realizam uma variedade de tarefas adicionais interativas como, por exemplo, navegação e marcação de score.

A opção “*Export to HTML/Java*” é uma característica adicional que permite que sejam criados tags HTML “<APPLET></APPLET>” no interior da página, que são necessários para embutir um link em seus applets Java pré-construídos para prover a capacidade de interatividade.

Cada applet Java é seu próprio miniprograma, executando independentemente de todos os outros applets. É importante ressaltar que o *toolbox* não gera applets, ele pode somente criar um link em uma página HTML para um applet pré-construído que realiza uma tarefa específica pré-determinada.

As imagens originais do aplicativo são convertidas para o formato “gif” ou “jpeg”. Nos testes realizados, os arquivos de vídeo no formato “avi” não foram convertidos, impossibilitando seu uso devido ao tamanho excessivamente grande.

De acordo com [CSS99], pode-se distribuir uma aplicação feita no *Toolbook* pela Internet/Intranet e, nesse caso pode-se transformar a aplicação em páginas Web, no formato HTML puro ou HTML usando *applets* Java ou Active X. Pode-se, também, visualizar a aplicação por intermédio de um browser *Internet Explorer* ou *Netscape Navigator* e com o plug-in Neuron instalado. Neste caso, a aplicação é “empacotada” porém os arquivos gerados devem ser armazenados no servidor Web. Quando é escolhida a

opção “*Export to Web*”, cada página da aplicação é analisada de acordo com o tipo de objeto. Um arquivo HTML é então produzido duplicando a página em um browser Web.

5.1 Neuron

Neuron é um plug-in ou controle ActiveX que trabalha com Netscape Navigator ou Microsoft Internet Explorer para permitir o acesso e execução de aplicativos *Toolbook II* via Internet ou qualquer rede com protocolo TCP/IP. Neuron é na realidade, o conjunto de arquivos *runtime* do *toolbook* que juntos somam aproximadamente 6 Mb e que normalmente são distribuídos com aplicações regulares em CD-ROM, com adicionais auxiliares que são capazes de carregar os recursos do *toolbook* como books, system books, recursos de mídia dentre outros, através de uma conexão Internet, ao invés de diretamente do disco rígido/CDROM.

Justamente porque o plug-in Neuron passa seus books, arquivos de mídia, e outros, diretamente de *runtime Toolbook* para o computador cliente, este tem a capacidade de programação intacta. Isso inclui tudo que pode ser feito normalmente em uma aplicação.

Uma limitação da tecnologia Neuron é que toda aplicação tem que ser carregada para o diretório cache do cliente antes da primeira página estar visível. Após isso tudo, executa em velocidade normal. Neuron inclui uma simples maneira para escolher entre seus dois modos principais, o modo seguro e o não-seguro. No modo não-seguro, o aplicativo é executado de maneira que seja permitido acessar todas as características do *Toolbook II*. No modo seguro, é executado o aplicativo em um modo que permita acessar um pequeno subconjunto de suas características, protegendo, assim, o sistema de *books* que possa executar processos não-autorizados e que possam talvez causar problemas. [KAR97]

5.2 Impulse

Impulse é um utilitário de pré-processamento de aplicativos *toolbook* para fornecer melhor performance quando o aplicativo for acessado via Internet utilizando o plug-in Neuron. Sua finalidade é “quebrar” grandes aplicações em *books* com um grande número de arquivos pequenos. Cada um desses arquivos contém uma pequena porção da base de dados da aplicação.

Esse aplicativo reduz o tempo inicial de download para um arquivo “.tbk” (extensão padrão do *toolbook*) quando a aplicação é entregue na Web. Impulse particiona um arquivo “.tbk” em arquivos pequenos. Depois de usar *impulse*, somente a informação necessária para mostrar a página é carregada. Quando este utilitário é utilizado, é conveniente que sejam criados scripts na aplicação que examinem diretamente todas ou um grande número de páginas do aplicativo. Para examinar todas as páginas, muitos dados devem ser carregados do servidor, eliminando então a eficiência ganha através do uso do Impulse. Por exemplo, se existe um aplicativo que mostra aleatoriamente um subconjunto de páginas, todas as páginas são examinadas em ordem pelo script para serem exibidas randomicamente, o que faz com que todas as páginas sejam carregadas quando o aplicativo é aberto. Quando todo aplicativo deve ser carregado na abertura, não existe razão para utilização do impulse.

6. CONVERSÃO DO APLICATIVO: “ECG - CURSO BÁSICO”

O objetivo principal deste artigo é descrever os passos principais, problemas e maneiras alternativas para geração de páginas Web de aplicativos *Toolbook* já desenvolvidos. Para melhor ilustrar este trabalho, foram realizados testes sobre a exportação para Web de um software multimídia da área médica denominado “ECG – Curso básico”, desenvolvido por Waldomiro Carlos Manfroi.

O aplicativo médico apresentava as seguintes características antes da exportação para formato Web: 163 páginas, 44,8 MB sem incluir arquivos de vídeos, 342 objetos com scripts e 2932 linhas de script.



Figura 2 – Exemplo de tela do ECG

Os scripts no aplicativo original, desenvolvido e executado em PCs stand-alone através do módulo *runtime* do *Toolbook*, possuem, principalmente, as seguintes funções:

1. mostrar / ocultar a janela chamada de “menu”, através da qual se pode acessar informações sobre os autores e todos os capítulos da aplicação;
2. modificar a forma do cursor do mouse quando este está posicionado sobre um botão ou campo que possua um link com alguma página ou objeto ou execute uma determinada ação;
3. nos questionários presentes ao final de cada capítulo:
 - 3.1 - computar e calcular a pontuação obtida pelo usuário nos testes realizados;
 - 3.2 - modificar o texto das *hotwords* de alternativas “a), b), c), d), e)” para “x)” quando forem selecionadas para determinar qual opção foi escolhida pelo usuário;
 - 3.3 - mostrar se a alternativa escolhida pelo usuário estava correta ou não, através de sinalização por determinado período de tempo colocada nos campos “correto” ou “incorreto” localizados no canto inferior direito da tela;
4. navegação entre páginas utilizando efeitos de transição;
5. execução de vídeos;
6. mostrar / ocultar objetos existentes na página ou no background como um botão, por exemplo.

Foi constatado que aproximadamente 95% dos scripts contidos na aplicação de origem, são relativos aos testes para usuários executados ao final de cada capítulo e o restante está dividido entre os outros itens.

Diante das constatações realizadas sobre o aplicativo executando em uma plataforma PC stand-alone, foi determinada a metodologia de trabalho para obtenção dos resultados finais. Como o aplicativo em estudo foi previamente desenvolvido com a ferramenta *Toolbook*, foi utilizada a mesma ferramenta em duas versões para conversão para Web. Foi utilizada a versão 7.0 *Assistant* para geração dos aplicativos para Web e a versão 6.0 II *Instructor* para edição e análise dos scripts e demais funcionalidades do aplicativo.

Foram usadas essas duas ferramentas, pois na versão *Assistant*, mesmo existindo scripts nos objetos, estes não são mostrados e tampouco editáveis. Além disso, as versões *Assistant* limitam bastante o campo de ação do programador da aplicação já que poucas funcionalidades avançadas estão disponíveis. Como existia uma grande necessidade de manipulação e análise de scripts para o correto desenvolvimento do trabalho, foi necessária a utilização da versão 6.0 *Instructor* para realizar esta tarefa. Por outro lado, por ser mais recente, a versão 7.0 *Assistant* possui o módulo de publicação para Web mais eficiente, pois além de trabalhar com o formato DHTML, seus algoritmos são mais corretos e otimizados. Essa utilização de versões de duas ferramentas distintas, fez com que o trabalho se tornasse muito mais lento, pois a cada exportação para o formato Web ou necessidade de visualizar um Script, foi necessário salvar o aplicativo e abri-lo na outra versão.

6.1 RESULTADOS DA CONVERSÃO

Durante a conversão para Web, muitas dificuldades foram encontradas, fazendo com que o tempo necessário para a exportação do aplicativo fosse bem superior ao sugerido pelo fabricante do software em seus informativos. Após ser convertida para o formato Web, a aplicação apresentou várias características que foram:

- botões com hyperlinks são exportados e funcionam corretamente, no entanto, quando se utiliza a opção “*Java Where possible*” que trabalha com applets Java para manter a compatibilidade com navegadores com a versão 3.x, não basta simplesmente isso, pois como descrito em capítulos anteriores, o *Toolbook* não gera os Applets, que devem ser criados separadamente;
- as potencialidades de exportação para Web da versão 6.0 não são tão poderosas quanto as da versão 7.0 *Assistant*;
- scripts não funcionam na conversão para Web, tanto os criados pelo desenvolvedor quanto os sugeridos pela própria *Asymetrix*, que é o caso dos scripts gerados através da ferramenta *AutoScript*;
- list boxes não suportam formatação de caracteres ou transparência;
- clips visuais não suportam pré ou pós-efeitos contidos em Stages;
- clips inteiros podem ser executados, mas não uma porção de um clip;
- botões suportam título transparente quando o título é configurado para ser centralizado;
- questões com preenchimento em branco devem ter uma borda retangular e não suportam rolagem;
- caixas de texto não podem ser transparentes;
- figuras presentes na página são deslocadas de sua posição original no momento da conversão para Web.

Independentemente dos problemas encontrados, após vários ajustes, principalmente nos itens citados anteriormente, foi possível a conversão do aplicativo original para Web com grande parte de suas características preservadas e somando ao total 1,20 Mb e mantendo o mesmo número de páginas.

7. CONCLUSÃO

A ferramenta *Asymetrix Toolbook*, possui módulos de geração dos aplicativos desenvolvidos para o ambiente Web, no entanto, ainda deixa muito a desejar principalmente em termos de flexibilidade. Isto ocorre pelo fato de existirem muitas

possibilidades distintas de recursos a serem utilizados em aplicativos multimídia convencionais, o que normalmente é um artifício largamente explorado em aplicativos desta espécie, principalmente através de imagens, sons e vídeos. Como faz uso de muitos recursos desse gênero, que ocupam muito espaço de armazenamento, normalmente um aplicativo multimídia é contido em um CD-ROM que tem a capacidade aproximada de seiscentos megabytes. Analisando sob este ponto de vista, realmente pode-se concluir que é um desafio realizar a conversão destes aplicativos para páginas Web que não ocupem muito mais do que poucos megabytes de espaço, o que atualmente é um padrão aceitável para não degradar a velocidade na Web.

Através da análise dos testes realizados sobre o aplicativo “ECG – Curso Básico”, foi possível identificar as principais lacunas existentes neste tipo de conversão pelo *toolbook*. Apesar das dificuldades e de bastante tempo necessário para modificações no aplicativo original, foi exportada a aplicação corretamente para Web com a maioria das características iniciais preservadas, o que é um fato que serve como base para que os desenvolvedores multimídia possam converter suas aplicações já desenvolvidas para Internet, proporcionando seu uso por um número exponencialmente maior de usuários se comparado ao método de distribuição tradicional através de CD-ROM.

REFERÊNCIAS BIBLIOGRÁFICAS

- [KAR97] Karyn Pallay, Lori Schultz, and Gaylene Zweigle, PROGRAMMING IN OPENSRIPT, Asymetrix Corporation, 1997. Bellevue, WA.
- [MUL96] Multimedia Database Systems – Issues and research directions. Springer-Verlag Berlin Heidelberg 1996. New York. P. 284.
- [ROC92] ROCHA, Ana Regina C.; STAHL, Marimar M. América à Vista: O descobrimento numa viagem através de hipermídia. CONGRESSO IBEROAMERICANO DE INFORMÁTICA EDUCATIVA, 1992, Santo Domingo, República Dominicana. Memórias ... República Dominicana: [S. N.], 1992.
- [SAC97] SANTOS, Fabiano Calixto; VIEIRA, Marina Teresa Pires. Uma ferramenta de Autoria para Aplicações MHEG-5 In: WORKSHOP EM SISTEMAS MULTIMÍDIA E HIPERMÍDIA, 3. 1997, São Carlos. Anais . São Paulo: UFSCar, 1997.
- [SIL99] SILVEIRA, Sidnei Renato. Estudo de uma Ferramenta de Autoria Multimídia para a Elaboração de Jogos Educativos. Porto Alegre: PPGC – UFRGS, 1999. p. 31. Dissertação de Mestrado.
- [VAU94] VAUGHAN, Tay. Multimídia na Prática. São Paulo: Makron Books, 1994. p. 376.
- [WOR92] WOLFE, Christopher R. Using an Authoring System to Facilitate Student-Centered Discovery Orienteds Learning. Computers & Education. Oxford, v. 19, n.4, p. 335-340, 1992.

TÉCNICAS DE GERENCIAMENTO DE TRANSAÇÕES DISTRIBUÍDAS

Fábio Fagundes Silveira¹

RESUMO

Transações constituem operações de grande importância em Bancos de Dados, as quais consistem em seqüências de operações especificadas pelas aplicações. Entretanto, uma compreensão cuidadosa de transações é requerida para ambientes distribuídos, pois envolvem execução de código em múltiplos nodos do sistema. Este artigo aborda as técnicas de gerenciamento de transações para ambientes distribuídos, discutindo com maior ênfase um dos mecanismos de controle de concorrência, o *Timestamping*.

Palavras-chaves: Transações Distribuídas, Controle de Concorrência, Timestamping

ABSTRACT

Transactions are very important operations on databases, and consist of sequences of operations specified by applications. Nevertheless, an accurate understanding of them is required in distributed environments, because they involve code executions in multiple sites of the system. This paper discusses transactions management techniques for distributed environments, focusing on one of the concurrency control methods, the Timestamping.

Keywords: Distributed Transactions, Concurrency Control, Timestamping

1. INTRODUÇÃO

O conceito de Sistemas Distribuídos consolidou-se com dois avanços na tecnologia para uso de computadores: o primeiro, o desenvolvimento de microprocessadores de alta velocidade, e o segundo, o surgimento das redes locais, que permitiu que se conectassem várias máquinas e se pudesse, então, trocar informações em parcelas de tempo muito pequenas [TAN92]. A combinação dessas duas tecnologias não só tornou possível, mas também fácil, conectar-se um grande número de CPUs através de redes com excelente velocidade, ao que hoje denomina-se Sistemas Distribuídos. De acordo com [NOE98], Bancos de Dados Distribuídos são uma coleção de banco de dados que estão conectados por links de comunicação, sendo estes situados de forma local ou dispersos geograficamente, podendo envolver diferentes sistemas gerenciadores de banco de dados, sendo executados sobre diferentes arquiteturas que distribuem a execução de transações.

Um SGBDD (Sistema Gerenciador de Banco de Dados Distribuídos) é definido como um software que trata do gerenciamento de BDD (Banco de Dados Distribuído) e realiza as operações de forma que o sistema pareça para o usuário como se fosse um sistema centralizado. Segundo [NOE98], bancos de dados distribuídos devem encapsular sete funções de transparência: Localização, Performance, Cópia, Transação, Fragmentação, Mudança de Esquema e Transparência de SGBD Local.

Em Sistemas Computacionais Distribuídos, processos em nodos independentes tendem a acessar recursos compartilhados concorrentemente. Nessa situação é importante certificar-se de que os resultados dos acessos concorrentes sejam previsíveis [MIT95]. Uma vez que as ações (transações) sobre recursos compartilhados tornam-se potencialmente conflitantes, deve-se então, implementar algum mecanismo de sincronização.

¹ Bel. em Informática (URCAMP), professor do CCEI (Curso de Informática), mestrando em Ciência da Computação (UFRGS). E-mail: ffs@urcamp.tche.br / ffs@inf.ufrgs.br

Transações distribuídas podem ocorrer devido a diversos fatores, entre os quais incluem-se [DEM97]: (i) itens de informações pertencentes a uma operação podem estar distribuídos em diversos servidores (nodos); (ii) uma transação pode necessitar de acesso a mais de um recurso e, talvez, conseqüentemente, mais de um servidor; (iii) um servidor provendo um serviço pode ter que contactar outro servidor para poder prover esse mesmo serviço.

Vários são os problemas originados com transações distribuídas. Segundo [MAT98], pode-se citar o problema de como manter um estado consistente da base de dados com tanta replicação, a qual implica protocolos sofisticados de controle de réplicas. Outro refere-se que, para que sejam alcançados os objetivos da replicação (disponibilidade e desempenho), faz-se necessária, a avaliação da integração dos três tipos de replicação em SBDDs: replicação de dados, de processamento e de comunicação.

Existe, ainda, o fato de transações poderem geralmente ser executadas em paralelo, pelo fato de as informações residirem em localizações diferentes, enfatizando com isto, que noção de controle de concorrência e a capacidade de recuperação de dados constituem grande esforço de pesquisa na área de teoria de bancos de dados distribuídos.

2. TRANSAÇÕES E CONCORRÊNCIA

Uma transação é entendida como uma série de operações sobre itens no banco de dados que podem vir a alterar o estado do mesmo. Neste ponto, encontra-se o conceito de **atomicidade da transação** [REI97]. Transações atômicas são constantemente empregadas em sistemas distribuídos, caracterizando-se pelo fato de que, se algo impede o término desta transação, o banco de dados deve ser restaurado ao estado anterior ao início da execução desta transação (*RollBack*), evitando com isto, que o mesmo passe a um estado de inconsistência.

As transações devem possuir quatro propriedades principais (*ACID*) [REI97] [NEU95]:

- atomicidade: uma transação deve ser executada em sua totalidade, caso contrário, não deve executar absolutamente nenhuma de suas operações.
- consistência: transformações preservam a consistência do banco de dados, isto é, uma transação transforma um estado consistente do banco de dados em outro estado consistente, sem necessariamente preservar a consistência nos pontos intermediários da transação.
- isolamento: as transações devem estar isoladas umas das outras, sendo esta propriedade também conhecida como *serialização*. Se duas ou mais transações forem processadas em paralelo, o resultado será o mesmo se elas forem processadas seqüencialmente.
- durabilidade: também chamado, de *persistência*, corresponde ao fato de que, uma vez atualizado o banco pela transação, tal atualização permanecerá (sem possibilidade de *rollback*), mesmo se existir um problema subsequente no sistema.

Em bancos de dados distribuídos, ao contrário da abordagem centralizada, é possível que ocorra execução de transações em paralelo, decorrendo disto um aumento de complexidade de ações como, por exemplo, como determinar quando realizar um *commit* de uma transação. A filosofia por trás desta idéia é verificar que o término das transações num sistema distribuído seja uniforme, ou seja, ou as transações são abortadas em todos os nodos (*sites*), ou são atualizadas nestes nodos, conforme princípio da atomicidade [NOE98]. Para lidar com tal situação, a maioria dos sistemas distribuídos utilizam um protocolo denominado *two-phase commit*, cujo princípio de funcionamento consiste no fato de envio de mensagens por parte de cada nodo para um nodo chamado “coordenador” que poderá, então, saber se os demais nodos estão habilitados para realizar recuperações, se algum processo de atualização falhar por alguma razão. Logo após o envio das mensagens ao coordenador, este, por sua vez, poderá enviar as decisões de atualização (*commit*) diretamente aos nodos. Apesar de bastante confiável, de acordo com [MIT95], uma

possível falha deste protocolo pode acontecer: ao longo do tempo que a mensagem de *commit* enviada pelo coordenador para os demais nodos leva para chegar, pode acontecer de um dos nodos que espera por tal mensagem, falhar. Isto pode levar a um estado inconsistente do banco de dados, pois os demais nodos terão realizado um *commit*.

Para permitir que múltiplas transações sejam executadas simultaneamente em diversos nodos, algum mecanismo torna-se necessário para impedir que elas não interfiram umas nas outras. Este mecanismo é chamado de *Controle de Concorrência*, no qual existem diversos métodos utilizados para sua implementação.

3. GERENCIAMENTO DE TRANSAÇÕES DISTRIBUÍDAS

Dentre o conjunto de principais recursos requeridos a qualquer sistema de banco de dados destaca-se, em especial, sua capacidade de executar transações. Em ambientes distribuídos, tal operação requer definições cuidadosas pelo fato delas envolverem execução de códigos em múltiplos nodos do sistema.

Alguns sistemas distribuídos permitem processamento paralelo dentro de transações. Isto é, agentes múltiplos dentro de uma transação executam, simultaneamente, em nodos múltiplos. Devido a essas características, sistemas distribuídos apresentam vários problemas específicos [DAT88], entre os quais podem-se citar: Processamento de Consultas, Propagação de Atualização, Controle de Concorrência, Protocolos de *Commit* e Gerenciamento de Catálogos

O problema no gerenciamento de transações é o fato de elas serem executadas em paralelo, em diversos nodos. Tal circunstância leva à utilização de mecanismos que controlem a ordem em que tais transações são executadas. A seguir, são definidos os mecanismos de controle de concorrências distribuídos.

4. CONTROLE DE CONCORRÊNCIA EM TRANSAÇÕES DISTRIBUÍDAS

Controle de concorrência é a parte de manipulação de transações que negocia como múltiplos acessos a recursos compartilhados do sistema possam ser executados sem causar conflitos permitindo, com isso, o compartilhamento dos recursos de forma completamente transparente aos usuários [DEM97].

A seguir, discutem-se três mecanismos capazes de tornar tal circunstância possível, ressaltando-se que, cada um destes, apresenta certas vantagens e desvantagens.

4.1. Bloqueio em Transações Distribuídas

Segundo [TAN92], o precursor e mais amplamente difundido algoritmo de controle de concorrência é o bloqueio (*locking*). Esse mecanismo consiste no fato de que quando algum processo (o que dá início à transação) requer operações de leitura ou gravação sobre registros (ou outro objeto), como parte da transação, ele deve primeiro obter o bloqueio do recurso ou dos recursos que irá utilizar. Outros processos que requeiram tais recursos não irão obtê-los, pelo fato destes estarem bloqueados. Os bloqueios podem ser gerenciados de vários modos, como a partir de gerenciadores de bloqueio centralizados em um *site* ou a partir de gerenciadores de bloqueios replicados, que requerem o consentimento da maioria dos *sites* durante uma requisição de bloqueio. Em ambos os mecanismos, o gerenciador mantém uma lista de objetos bloqueados e rejeita todas as tentativas de bloqueio sobre esses objetos, que já se encontram nessa situação por outros processos. Como explanado, existem dois tipos de bloqueios: de leitura (*read lock*) e o de gravação (*write lock*). Uma transação que detém um *read lock*, pode atualizá-lo para um *write lock* garantindo que nenhuma outra transação detenha um *read lock* para o mesmo objeto. Se um *read lock* está associado a um objeto, outras requisições de *read lock* são concedidas, pois estas certificam-se de que os objetos não serão atualizados podendo, então, ser concedidos também a outras transações. Em contrapartida, quando um objeto encontra-se bloqueado

com *write lock*, nenhum outro tipo de *lock* é concedido. Conseqüentemente, *read locks* são compartilhados, enquanto que *write locks* devem ser exclusivos [TAN92].

Por simplicidade, foi aqui descrito como unidade de bloqueio, o termo objeto. Na prática, este pode assumir várias formas, tais como um registro, um atributo ou até mesmo um banco de dados inteiro. Esta particularidade, com relação ao tamanho do objeto, leva ao conceito de *granularidade de bloqueio*. Quanto menor a granularidade, mais preciso será o bloqueio e mais paralelismo poderá ser alcançado [TAN92]. Por outro lado, granularidades menores requerem mais bloqueios e podem levar a *deadlocks*.

As transações podem somente obter um bloqueio se elas não tiverem previamente liberado qualquer bloqueio. Isto implica em que transações experimentem fases de requisições e fases de liberações para obterem e liberarem bloqueios. Isso pode conduzir à situação de *deadlock*, na qual uma transação está esperando por um objeto bloqueado por outra transação, que em fila, está bloqueada esperando pelo objeto bloqueado pela primeira. Entretanto, a maioria das transações implementadas por *locking* usam o que é chamado de *two-phase locking*. Todas as informações de *lock* são mantidas centralizadas de onde são acessíveis para todos os processos das transações. Estes executam o código para bloqueio e sincronizam uns aos outros [REI97].

É possível uma transação requisitar múltiplos bloqueios, visto que suas operações podem ser passíveis de execução em diversos nodos do sistema (*sites*). É possível uma transação esperar devido a duas razões: primeiro, para obter um bloqueio e, segundo, para receber uma mensagem de um coortes (agente na mesma operação) da mesma transação. Isto pode levar à dedução de que um *deadlock* global ou centralizado pode ocorrer. Podem, entretanto, utilizar-se de mecanismos para detecção de *deadlocks*, não sendo estes, contudo, totalmente seguros.

4.2. Controle de Concorrência Otimista em Transações Distribuídas

Uma segunda forma de manipular múltiplas transações concorrentes é o controle de concorrência otimista [KUN81]. A filosofia por trás desta técnica é extremamente simples: tentar realizar as atualizações, baseadas no fato de que elas, geralmente, não intervêm umas nas outras. Entretanto, conflitos não estão totalmente descartados, sendo necessário manter registros de que objetos estão sendo lidos ou escritos. No momento da atualização, ele verifica as demais transações para ver se algum dos seus objetos foi atualizado desde que a transação foi iniciada. Em caso afirmativo, a transação é abortada, caso contrário, a atualização é realizada. A grande vantagem desta técnica é a eliminação de *deadlocks*, permitindo máximo paralelismo devido ao fato de os processos não aguardarem por bloqueios. A desvantagem é que, alguma vezes, falhas podem ocorrer, sendo que as transações devem ser reinicializadas.

Em ambientes distribuídos, cada servidor envolvido necessita validar o acesso aos iens de dados que ele manipula. Isto é realizado durante a primeira fase do protocolo *two-phase commit* [DEM97]. Na figura 1, é apresentado um exemplo da utilização desta técnica:

T	U
Read (A) at X	Read (B) at Y
Write (A)	Write (B)
Read (B) at Y	Read (A) at X
Write (B)	Write (A)

Fig. 1: Exemplo de controle de concorrência otimista [DEM97]

Analisando-se o exemplo da figura 1 e assumindo-se que os dados são acessíveis, T precede U em X e U precede T em Y, a validação é realizada progressivamente. Como

somente uma transação pode realizar a fase de validação/atualização por vez, ocorre *deadlock* de atualização. No intuito de melhorar a performance do processo de validação, ele é geralmente realizado em paralelo. Isso resolveria o problema do *dealock* de atualização. Entretanto, torna-se agora possível para os dois servidores do exemplo, serializar as transações **T** e **U** em diferentes ordens. Tal problema pode ser evitado passando-se a realizar validações globais logo após as validações locais [DEM97].

4.3. *Timestamping*

Um método completamente diferente para implementar controle de concorrência é o chamado *timestamping*, o qual consiste na designação de um valor (*timestamp* ou simplesmente, identificador) para cada transação, no exato momento em que aquela ocorre (BEGIN_TRANSACTION). Através da utilização de algoritmos, como o de *Lamport* [LAM90], pode-se certificar de que estes identificadores são únicos, os quais correspondem à sua finalidade e princípio básico de funcionamento. Em outras palavras, a idéia fundamental desta técnica é a seguinte [DAT88]:

1. cada transação recebe um identificador (*timestamp*) único global;
2. as operações de atualizações não são aplicadas fisicamente sobre o banco de dados até o término (bem-sucedido) da respectiva transação;
3. cada um dos objetos do banco de dados (conforme conceito já definido anteriormente), carrega o *timestamp* da transação que por último o leu (*read*), e o *timestamp* da transação que por último o atualizou (*write*). De acordo com [BER80], diversas implementações são possíveis;
4. se uma solicitação de operação sobre um banco de dados, feita por determinada transação *T1*, conflitar com alguma outra operação já executada, baseada na solicitação de outra transação mais jovem *T2*, a transação *T1* é cancelada e reiniciada;
5. quando uma transação é reiniciada por algum motivo, esta receberá um novo identificador (*timestamp*).

Ressalta-se, entretanto, que uma operação *T1* encontra-se em conflito com *T2* se, e somente se: (i) a operação de *T1* for de leitura, e *T2* já tiver atualizado o objeto em questão, ou (ii) a operação for de atualização, e *T2* já tiver lido ou atualizado o objeto em questão. Esta situação só é possível ocorrer durante o processo de atualização (*commit*) de *T1*, baseado no item “2” acima.

Supondo que as transações sejam pequenas e com tempo de execução longo, então elas serão executadas normalmente quando um processo tentar acessar um objeto, seja para operações de leitura ou escrita, pois o *timestamp* do objeto (identificador de *read* e *write*) será menor (mais antigo) do que o identificador da transação atual. Entretanto, quando essa característica de ordenação está incorreta, significa que uma transação que começou depois da atual obteve êxito para acessar um objeto e atualizá-lo, decorrendo disso que a transação corrente está “atrasada”, devendo então ser cancelada (reinicializada, obviamente com um novo *timestamp*) [DAT88]. Dessa forma, este mecanismo é também otimista, sendo os detalhes de implementação um pouco diferentes. No método descrito por [KUN81], espera-se que transações concorrentes não usem os mesmos objetos. No método *timestamping*, não importa se as transações acessam os mesmos objetos, contanto que transações com identificações mais baixas ocorram primeiro.

Um modo simples de exemplificar concorrência utilizando *timestamp* [TAN92] é considerar três transações : *alfa*, *beta* e *gama*. A primeira, *alfa*, foi iniciada a um certo tempo atrás e utilizou todos os objetos requisitados por *beta* e *gama*, logo, todos esses objetos possuem *timestamps* referentes à transação *alfa*. *Beta* e *gama* são posteriormente iniciadas, com *beta* possuindo um *timestamp* menor que *gama*.

Considera-se, no exemplo, primeiramente, uma operação de escrita pela transação *beta* sobre um objeto, cujo *timestamp* é denominado T, e o *timestamp* de leitura e escrita do objeto a ser acessado é T_R e T_W respectivamente. Desde que *gama* não tenha acessado este objeto e executado um *commit*, ambos, T_R e T_W , corresponderão ao *timestamp* da transação *alfa*, sendo evidentemente menores que T. Conforme o exemplo, na figura 2(a) e (b) vê-se que T é maior que T_R e T_W , logo a operação de escrita é permitida, tornando-se permanente quando *beta* realizar *commit*. O *timestamp* da transação *beta* é então atribuído ao identificador *write* do objeto. Entretanto, na figura 2(c) e (d) as requisições de *beta* não serão aceitas, pois *gama* já havia previamente lido ou escrito no objeto e realizado uma operação *commit*, sendo então, a transação *beta* cancelada, podendo esta ser reinicializada com um novo *timestamp*.

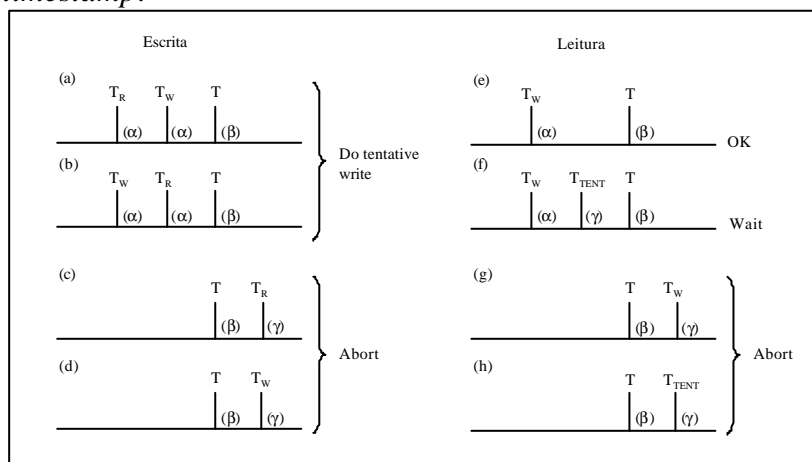


Fig. 2: Controle de Concorrência utilizando *Timestamping* [TAN92]

Com relação às operações de leitura, na figura 2(e), não é verificada a situação de conflito. Contudo, na figura 2(f), alguma outra transação tenta escrever no objeto. O *timestamp* desta transação é menor que o da transação *beta*, o que faz com que *beta* simplesmente aguarde a operação *commit* da transação em questão para então poder ler o objeto. Na figura 2(g), a transação *gama* já escreveu no objeto e realizou um *commit*, restando a transação *beta* ser cancelada. Com relação à figura 2(h), *gama* está em processo de escrita, embora ainda não tenha realizado um *commit*. Novamente a transação *beta* será cancelada, pois possui um *timestamp* menor que a transação *gama*.

Com relação às vantagens desta abordagem, encontra-se o fato de que nenhum bloqueio é feito e, portanto, a situação de *deadlock* é inexistente, minimizando com isto os custos de implementação de algoritmos de detecção deste. O método *Timestamping* possui enfoques bem diferentes dos mecanismos de bloqueio. Quando uma transação encontra um objeto com *timestamp* maior (mais tardio que o seu), ela simplesmente é cancelada. Essa mesma situação utilizando o mecanismo de *locking*, este esperaria ou estaria habilitado a prosseguir imediatamente. Por outro lado, o método *timestamp*, conforme já descrito, está livre de *deadlock*, no qual esta característica representa uma considerável vantagem. Muitas são as variações do esquema básico de funcionamento deste método [BER80]. Duas variações serão apresentadas mais adiante.

No enfoque de sistemas distribuídos, a questão principal é que os identificadores (*timestamps*) devem ser globalmente **únicos** [DAT88], onde existem muitos nodos e estes executam diversas transações em paralelo. Não se descarta a possibilidade, entretanto, de que dois ou mais eventos discretos (por exemplo a iniciação de duas transações distintas) ocorram concorrentemente. Devido a esse fator, tais identificadores não podem ser formados por simples leituras de relógios, devendo sim, ser formados, também, por um outro atributo que corresponde a um identificador (ID) do nodo ou *site*.

Considerando dois eventos, x e y , deve-se ressaltar [DAT88]:

- se o evento x preceder y , ocorrendo ambos no mesmo nodo N , então o valor do atributo do relógio no nodo N que corresponde ao evento x terá que possuir valor menor que o correspondente ao evento y ;
- no caso do evento x caracterizar-se pelo envio de uma mensagem m do nodo $N1$ e o evento y for o recebimento de m no nodo $N2$, então o valor do relógio de $N1$ que corresponde ao evento x , será menor que o valor do relógio no nodo $N2$ que corresponde ao evento y .

4.3.1. Timestamping Conservador

Com o intuito de eliminar operações conflitantes e, conseqüentemente a reinicialização de transações, o método *timestamping conservador* [REI97] tem, como idéia principal, o fato de que, desde que se garanta que nenhuma operação sobre o banco de dados causará um conflito num tempo futuro, esta não será executada. Esta abordagem apresenta como vantagem a característica de reduzir o número de reinícios de transações conflitantes, embora apresente menor concorrência que o seu esquema básico de funcionamento.

Considerando-se que, cada transação possua um nodo originador, o qual gerencia a execução desta, direcionando aos respectivos nodos, conforme apropriado, as solicitações de acesso aos dados requeridos pela transação T , torna-se necessário, então, que cada nodo trate os processos de *commit* para as transações originadas naquele nodo na ordem do *timestamp*, pois uma transação mais nova poderia requerer atualização e ser processada por um nodo remoto, sendo executada antes da transação mais antiga, tendo esta que ser, então, reinicializada. Desse modo, se duas transações $T1$ e $T2$ originam-se nesta ordem no nodo N , este terá que realizar o processo de *commit* de $T1$ antes de $T2$, mesmo que $T2$ requeira atualização (*commit*) primeiramente [DAT88].

Cada nodo do sistema manterá uma fila de requisições de operações de leitura e outra fila de requisições de atualizações para os objetos residentes no respectivo nodo, sendo esta fila classificada pela ordem ascendente de *timestamps*. Quando um nodo N' recebe uma solicitação do nodo N , este verifica de que tipo é essa solicitação e, então, baseado em consultas nessas filas, retarda essas requisições em função de outras, caso existam, de forma a executar as operações rigidamente de acordo com a grandeza dos *timestamps*. Devido ao fato do nodo N' esperar por requisições de outros nodos, este poderá ficar bloqueado (isto é, a fila de requisições ficar vazia) caso o nodo N não tiver nenhuma solicitação genuína a ser feita. Para contornar tal problema, o nodo N deverá enviar solicitações nulas com *timestamp* nulo para aquele nodo. Segundo [DAT88], isso leva a uma considerável desvantagem deste método, que é a constante intercomunicação entre os mesmos, ressaltando-se também o fato de não fornecer muita concorrência, pois para eliminar conflitos, tende a serializar todas as operações.

4.3.2. Classes de Transação

Devido ao fato de a técnica anterior ser pessimista, pois se supõe que cada operação efetivamente conflitará com outra ainda por vir, retardando sua execução para diminuir suas vantagens, introduz-se o conceito de *classes de transação*. Estas têm por objetivo, auxiliar no reconhecimento de situações em que o conflito possivelmente não ocorrerá. Dessa forma, a aplicação do mecanismo para controlar concorrência será aplicado somente em situações que efetivamente se fizerem necessárias. De acordo com [DAT88], define-se como *readset* de uma transação o conjunto de objetos de dados (lógicos) que ela lê, e *writeset* o conjunto de objetos de dados lógicos que ela grava. Define-se, então, um conjunto máximo de *readset* e *writeset* para a classe de transação.

Quando o *readset* e o *writeset* de uma transação individual for um subconjunto respectivamente do *readset* e *writeset* de uma classe de transação diz-se, então, que a

transação individual é membro desta [LOP98]. Conseqüentemente, constata-se que duas classes são conflitantes se o *writeset* de qualquer uma intersectar o *readset* ou o *writeset* da outra.

5. CONCLUSÕES

Bancos de Dados Distribuídos representam, hoje, um grande campo de estudo na área de tecnologia de banco de dados. Em determinados tipos de aplicações, muitas são as suas vantagens em relação à abordagem de banco de dados centralizado, entre as quais destacam-se: a localidade dos dados reduzindo a demora no acesso a dados remotos, paralelismo em processamento de consultas, maior simplicidade de adaptação referente ao crescimento das bases e, principalmente, os diversos níveis de transparência implementados, dando a impressão a usuários e aplicações que os dados são armazenados e processados de forma centralizada. Tal característica de distribuição aumenta, consideravelmente, a questão do compartilhamento de recursos providos pelos vários nodos que compõem o sistema. Devido a esse fato, faz-se necessária uma compreensão cuidadosa com relação ao processamento de transações distribuídas, pois estas podem, nesses ambientes, serem executadas de forma paralela, acessando recursos compartilhados concorrentemente.

Neste artigo, buscou-se apresentar o processo de gerenciamento de transações distribuídas, enfocando seus principais problemas e demonstrando mecanismos para sincronizar os conflitos potenciais que existem nesta abordagem. Foram apresentados três métodos de controle de concorrência para gerenciar transações: bloqueio (*locking*), controle de concorrência otimista e o método de timestamping, sendo este, em particular, um método bastante utilizado para a construção de sistemas distribuídos confiáveis.

REFERÊNCIAS BIBLIOGRÁFICAS

- [BER80] BERNSTEIN, P. A. GOODMAN, N. *Timestamp-Based Algorithms for Concurrency Control in Distributed Database Systems*. Proc. 6th International Conference on Very Large Data Bases, 1980.
- [CEL97] CELKO, Joe. *Transactions and Concurrency*. PowerBuilder Developer's Journal, 1997. Disponibilidade e acesso: <<http://www.sys-com.com/pbdj/1995/7-95/transac9.htm>> (01 Mai 1999)
- [DAT88] DATE, C.J. *Banco de Dados – Tópicos Avançados*, 2ª Reimpressão, Ed. Campus, 1988
- [DEM97] DEMPSTER, Robert. *Distributed Systems: Distributed Transactions*, 1997. Disponibilidade e acesso: <<http://saturn.cs.unp.ac.za/~robd/notes/ds/distributed-data/chap14.html>> (01 Mai 1999)
- [DUR 98] DURBIN, Jason. Bobrowski, Steve. Vasterd, Peter. *Oracle8 Distributed Database Systems*. Oracle Corporation, 1998. Disponibilidade e acesso: <http://www.cs.orst.edu/~minoura/cs440/parent/ora_804_doc/doc/server.804/a58247/title.htm> (05 Mai 1999)
- [KUN81] KUNG, H.T. ROBINSON, J.T. *On Optimistic Methods for Concurrency Control*. ACM Trans. on Database Systems, vol. 6, 1981.
- [LAM90] LAMPORT, L. *Concurrent Reading and Writing of Clocks*. ACM Trans. on Computer Systems, vol.8, Nov. 1990.
- [LOP98] LÓPEZ, Jorge de Rueda. *Transaction Time Databases*, 1998. Disponibilidade e acesso: <http://www.students.cs.ruu.nl/~jrueda/TransactionTDB/TransactionTDB.html> (10 Mai 1999)

- [MAT98] MATTOSO, Marta. *Bancos de Dados Distribuídos*. Universidade Federal do Rio de Janeiro, 1998.
- [MIT95] MITCHELL, Alex. *Concurrency Control in CSCW Systems*, 1995. Disponibilidade e acesso: <<http://www.dgp.toronto.edu/~alex/unpublished/concurrency.html>> (10 Mai 1999)
- [NEU95] NEURATH, Peter. *Transaction Management in Distributed Object Environment*. Comenius University Bratislava, Faculty of Mathematics and Physics, Institute of Informatics, 1995. Disponibilidade e acesso: <http://object.dcs.fmph.uniba.sk/Diploma_theses/1995_Kobetic_Neurath/www/transaction/Transaction.htm> (02 Mai 1999)
- [NOE98] NOEL, Rick. *Scale Up in Distributed Database*. Renselaer Polytechnic Institute, 1998. Disponibilidade e acesso: <http://www.cs.rpi.edu/~noel/distr_scaleup/distributed.html> (01 Mai 1999)
- [REI97] REICH, Sigi. *Synchronisation in Distributed Systems*. Distributed Computing and Networks Center, 1997. Disponibilidade e acesso: <<http://www.mmrg.ecs.soton.ac.uk/~sr/courses/cm313/w9synch/09synch.html>> (25 Abr 1999)
- [SIL94] SILBERSCHATZ, A., and Galvin, P. *Operating System Concepts*, Don Mills: Addison-Wesley, 1994.
- [TAN92] TANENBAUM, Andrew S. *Modern Operating Systems*. Prentice Hall, New Jersey, 1992

IDÉIAS PARA O DESENVOLVIMENTO DE UM MECANISMO DE BUSCAS PARA A EMBRAPA

Marcus Vinicius Freitas Margarites¹

RESUMO

Embora a WWW seja o maior repositório de conhecimentos já criado, a sua arquitetura anárquica e a falta de instrumentos de controle dificultam muito a localização de bons documentos. Foi buscando uma alternativa para simplificar este processo que surgiram os primeiros índices que, a princípio, não passavam de grandes catálogos de URLs mantidos manualmente. Porém, logo ficou claro que a rede crescia rápido demais para que esses sistemas fossem realmente efetivos. O avanço óbvio era criar mecanismos automatizados. Ao contrário dos índices, esses exigem pouca ou nenhuma intervenção humana para localizar e catalogar documentos. Tais sistemas são, claro, muito mais eficientes para lidar com a enorme velocidade com que a rede se desenvolve. Mas mesmo este desenvolvimento não foi suficiente. Segundo os últimos estudos, a quantidade de documentos mal indexados chega à metade dos documentos disponíveis na rede. Talvez a solução seja segmentar as buscas entre sistemas especializados. Se o usuário procura por informações financeiras, deveria se dirigir a um sistema específico, enquanto que outros sistemas atenderiam outras áreas do conhecimento. Este trabalho propõe a construção de um sistema automatizado para a Embrapa, indexando todos os documentos presentes na rede da Empresa, independentemente da Unidade onde estão armazenados.

Palavras-Chave: WWW, Mecanismo de Buscas, Recuperação de Informações

ABSTRACT

Although the WWW is the largest repository of knowledge ever created, its anarchical architecture and the lack of control instruments make it very difficult to find good documents. It was searching for an alternative to simplify this process that the first indexes had appeared. At first they were nothing but big URLs catalogues kept manually. But soon it was clear that the network grew excessively fast so that these systems were really effective. The obvious advance was to create automatized mechanisms. In contrast to the indexes, these demand little or no human intervention to locate and catalogue documents. Such systems are, clearly, much more efficient to deal with the enormous speed the network develops. But even these development wasn't enough. According to the last studies, the amount of indexed documents barely reaches half of the available documents on the network. Perhaps the solution is to segment the fetchings between specialized systems. If the user is looking for financial information, he would be directed to a specific system, while other systems would take care of different areas of knowledge. This work proposes the project of an automatized system to Embrapa, indexing all documents present on the company's network, independently of the Unit where they are stored.

Keywords: WWW, Search Engine, Information Retrieval

1. INTRODUÇÃO

Encontrar informações úteis na World Wide Web (WWW) está cada vez mais difícil. Embora a web seja o maior repositório de conhecimentos já criado pela humanidade, a sua arquitetura anárquica e a falta de instrumentos de controle fazem com que localizar um bom documento em meio à enorme quantidade de arquivos disponíveis se transforme em um verdadeiro exercício de paciência.

Foi buscando uma alternativa para simplificar o processo de recuperação de informações que surgiram os primeiros catálogos de páginas HTML. Já em 1994, alguns voluntários tentavam criar listas hierárquicas dos documentos disponíveis na Internet, através de grandes índices mantidos manualmente, como ainda é o caso do Yahoo! [OVE1999].

¹ Programador da Embrapa Pecuária Sul (mvfm@mvfm.net)

Através de um formulário on-line, o próprio autor do site cadastrava-o no índice, indicando sua URL, um pequeno resumo e a categoria onde o site melhor se enquadrava. Mais tarde, aquele site era visitado por um administrador que verificava a credibilidade e importância das informações ali contidas, e somente depois disso o incluía no banco de dados a partir do qual o índice era gerado. A classificação dos sites em categorias facilitava bastante a vida daqueles que procuravam por informações na rede: bastava saber o domínio da informação desejada e navegar entre uma lista hierárquica de categorias até encontrar uma listagem dos sites mais importantes.

Com o passar do tempo, entretanto, manter manualmente este banco de dados foi ficando cada vez mais difícil, já que o número de submissões crescia rapidamente e o trabalho de visitar e classificar cada site era muito caro e demorado. Da mesma forma, o número de sites dentro de cada categoria aumentava absurdamente, forçando uma constante subdivisão das categorias para evitar o excesso de entradas, a fim de tornar o processo de recuperação e navegação menos exaustivo para o usuário.

Ainda havia outro problema: saber que um site tratava de determinado assunto na maioria das vezes não bastava, já que tudo o que se sabia sobre ele estava baseado no resumo provido pelo seu autor. O documento desejado pelo usuário poderia estar perdido no meio de um imenso emaranhado de páginas e, mesmo com a ajuda dos índices, encontrá-lo poderia levar bastante tempo, já que estes normalmente apontam para a home-page do site, raramente para as páginas internas. Uma maneira mais eficiente de indexar essas informações era necessária.

Essa demanda levou ao desenvolvimento dos primeiros sistemas de busca textuais. Ao contrário dos índices, esses “livros de endereços” da Internet eram automatizados, de forma que pouca ou nenhuma intervenção humana era necessária para que os documentos fossem localizados e catalogados. Um pequeno programa (freqüentemente chamado de Spider ou Crawler) navega ao acaso pela rede, coletando informações sobre os documentos que encontra e armazenando-as em seu banco de dados. Tais sistemas são, claro, muito mais eficientes para tratar com a enorme velocidade com que a rede se desenvolve. Mas a sua grande vantagem é que eles conseguem indexar todo o conteúdo de um documento HTML, não apenas o seu título e resumo, como ocorre no caso dos índices. Dessa forma, uma busca através de um desses sistemas reportaria mesmo àquelas páginas mais obscuras que, entretanto, tratassem do assunto pesquisado, ainda que tal informação não estivesse presente no resumo do documento disponibilizado pelo seu autor.

O problema é que mesmo os indexadores textuais têm seus problemas, mais visíveis em buscas mais abrangentes, quando a quantidade de documentos encontrados pode facilmente chegar à casa dos milhares, num clássico exemplo de quantidade versus qualidade: com uma lista de resultados muito grande, fica difícil para o usuário identificar aqueles documentos que realmente lhe são importantes. Além disso, também por causa da imensa quantidade de documentos disponíveis, novos documentos podem demorar semanas para aparecer no índice (e aqueles que não existem mais podem continuar sendo listados por meses, provocando erros freqüentemente irritantes).

Mas o pior é que mesmo esses mecanismos não garantem que documentos relevantes serão localizados. Segundo a FAPESP [FAP2000], órgão responsável pelo registro de domínios no território brasileiro, o número de domínios da porção verde-amarela da rede cresceu cerca de 430 vezes desde janeiro de 1996, e a cada dia mais de 800 novos nomes-de-domínio são solicitados. Os mecanismos de buscas precisam se reescalar rapidamente para tentar acompanhar este rápido crescimento: em 1994, um dos primeiros mecanismos de busca automatizados da Web, o World Wide Web Worm (WWW) tinha um índice de 110.000 páginas. Em 1997, o índice do WebCrawler, então o maior, já chegava aos 2

milhões de documentos e, agora, em 2000, os dois maiores índices (Google e iWon) já reportam mais de meio bilhão de documentos indexados [SEA2000].

Mas a rede cresce mais rápido que isso ... O número de documentos atualmente disponíveis na Web já ultrapassa a marca de um bilhão [SUL2000A], de forma que mesmo usando um mecanismo de buscas topo-de-linha, só se consegue acesso à metade dos documentos disponíveis.

Pior ainda, esses números consideram apenas aquelas informações publicadas sob a forma de documentos HTML. Mas é cada vez maior o número de documentos armazenados sob a forma de registros em bancos de dados, e são poucos os mecanismos capazes de indexá-los, criando a chamada “Web invisível” [SUL2000B].

O número de usuários da rede também vem aumentando rapidamente. Para citar apenas o exemplo do Brasil, os internautas brasileiros já passam de 3,3 milhões, mas ainda representam pouco mais de 2 % da população do país. Com os grandes investimentos planejados pelo governo para os próximos anos, espera-se chegar aos 20 % em 2003 [GRU2000]. É de se esperar que boa parte desses usuários tenham pequena experiência com a rede e com os computadores em geral. Eles precisarão, então, de ferramentas fáceis de usar e com uma melhor taxa de retorno.

Tudo isso leva a crer que dificilmente um sistema centralizado será capaz de algum dia indexar a totalidade da Web. Mesmo que isso fosse possível, como o número de documentos cresce diariamente, ainda assim esse sistema estaria desatualizado apenas alguns instantes depois.

A saída para isso ? Confiar nas instituições bem conhecidas. Se o usuário procura por notícias econômicas, deve se dirigir a um grande jornal especializado. Se procura por informações sobre a agropecuária, pode procurar a Embrapa.

2. A EMBRAPA NA INTERNET

A Embrapa compreende diversas unidades de execução de pesquisa e desenvolvimento, cada uma atuando em uma área diferente. A Empresa tem presença forte na Web desde meados de 1996, quando seus primeiros sites foram implantados. Embora hoje exista um padrão estabelecido pela Empresa para o lay-out dos documentos HTML publicados na rede, cada Unidade se responsabiliza pela criação, manutenção e hospedagem do seu próprio site, incluindo aí a escolha das tecnologias usadas. Enquanto algumas Unidades estão bastante avançadas neste aspecto, usando scripts CGI, Java ou servidores de banco de dados integrados à Web, outras dispõem apenas de documentos HTML estáticos, de primeira geração.

Apesar das diferenças, a Empresa é vista pelos seus clientes externos como uma entidade única: as diferentes áreas de atuação das suas unidades nem sempre são percebidas pelos visitantes dos seus diversos sites. Ocorre, entretanto, que uma consulta feita no site de uma unidade, porém numa área diferente daquela explorada por essa unidade, pode não localizar nenhum documento, mesmo que os sites de outras unidades cubram tal assunto, já que os documentos presentes em um site permanecem virtualmente invisíveis para os mecanismos de busca dos demais.

Uma análise acurada dos logs do atual mecanismo de buscas do site da Embrapa Pecuária Sul estimou que 15,32 % das consultas estariam dentro do contexto da Embrapa, mas não retornaram documentos por não estarem cobertas pelos documentos presentes no site daquela Unidade [BER2000]. Então, de cada 100 consultas no mecanismo de buscas do site, 15 não estão sendo devidamente atendidas por estarem sendo realizadas no site da unidade errada. O problema é que o cliente nem sempre percebe isso, e o fato de não estar recebendo resultados deve lhe causar muita estranheza.

A nova Política de Comunicação da Embrapa [EMB1996], que vem sendo implantada desde 1996, dá uma ênfase cada vez maior à atuação com foco nos clientes. E foi procurando por uma maneira de facilitar esse processo de atendimento que esta proposta foi elaborada.

Este projeto propõe, então, o desenvolvimento de um mecanismo de buscas distribuído capaz de indexar todos os documentos presentes nos diferentes sites da Embrapa, não importando a forma em que estão publicados (documentos HTML ou informações armazenadas em bancos de dados disponíveis na Web). Tal ferramenta poderá contribuir substancialmente para que se alcance as macroprioridades expressas na estratégia gerencial da Empresa, que estabelecia a “busca por métodos e meios de transferência inovadores” como prioridade institucional [EMB1995]. Ao mesmo tempo que se aumenta a eficácia e eficiência dos processos de atendimento aos clientes (seja através do atendimento de um maior número de consultas aos sites da empresa, seja através de mais informações e com maior qualidade, já que um maior número de documentos poderia ser consultado de uma única vez, ao invés da necessidade de se repetir a busca em diversos sites), a marca da empresa também será fortalecida, aumentando o seu reconhecimento tanto nacional quanto internacionalmente.

3. OBJETIVOS DO PROJETO

O projeto visa a atingir uma maior taxa percentual de atendimentos efetuados pelo mecanismo de buscas, seja entregando documentos locais, seja apontando para documentos presentes nos sites das demais unidades. Tal capacidade será de extrema valia para a Embrapa e seus clientes, pois tornará todo o conteúdo dos sites institucionais da Empresa visíveis de uma maneira muito mais transparente para os usuários.

Da mesma forma, o projeto também pretende fortalecer as novas orientações da Empresa no que diz respeito ao mix de marketing, especialmente no tocante a pontos-de-venda ou atuação em rede.

4. A ARQUITETURA PROPOSTA

O mecanismo de buscas proposto será composto por quatro núcleos principais [figura 1]. O primeiro, o Spider, será o responsável pela coleta de informações nos sites das unidades. Tal conjunto de programas obterá uma lista de URLs a serem indexados e fará conexões HTTP até os sites de cada uma das unidades, buscando os documentos lá disponibilizados e preparando-os para armazenamento.

O segundo núcleo, aqui chamado de StorageServer, será composto pelos programas de armazenamento no banco de dados e de indexação. Ao receber um documento recuperado pelo Spider, o servidor de armazenamento fará a sua decomposição em seus elementos componentes (título, descrição, palavras-chave, links ...) e os armazenará num banco de dados, passando mais tarde a organizá-lo de maneira a facilitar a recuperação dessas informações.

O terceiro módulo do sistema, o Searcher, fará o processo de consulta e recuperação dos documentos, a partir de uma expressão de busca especificada pelo usuário do sistema. Tal módulo efetuará conexões com o banco de dados criado anteriormente, disparará as consultas e será o responsável pela montagem da página enviada como resposta ao usuário.

Por fim, o quarto componente do sistema, o RedirectServlet direcionará o usuário à página selecionada e registrará esta ação no banco de dados, usando a seleção do usuário como uma forma de pontuar o documento e melhorar os resultados de buscas futuras.

Os quatro componentes do sistema serão implementados usando diferentes abordagens da linguagem Java. O Spider será uma aplicação Java comum, podendo existir várias instâncias espalhadas pela rede; o StorageServer será um Servlet, se comunicando com os

Spiders via o protocolo HTTP; o módulo de recuperação consistirá de Servlets e documentos JSP, fazendo acessos ao banco de dados via JDBC, enquanto que o RedirectServlet será também um Servlet, registrando todos os documentos recuperados e as escolhas feitas pelos usuários do sistema.

Para baratear os custos, todo o sistema poderá ficar baseado em um computador de pequeno porte rodando Linux / Apache HTTP Server / Apache TomCat Servlet Engine / MySQL.

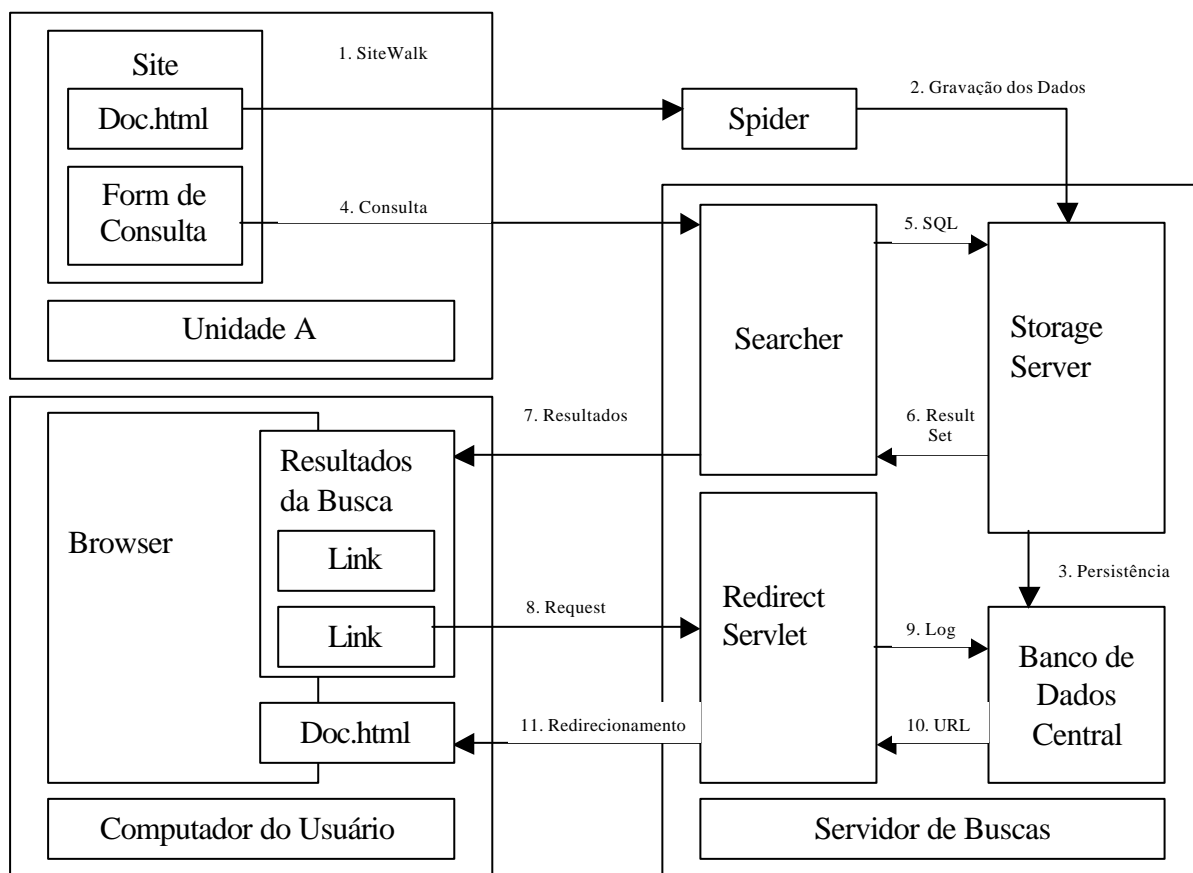


Figura 1 – A arquitetura proposta para o sistema de buscas.

O Spider

Um spider (ou robot, ou web wanderer, ou web crawler ...) “é um programa que viaja automaticamente pela estrutura de hipertexto da Web, recuperando um documento e recursivamente recuperando todos os documentos por ele referenciados” [KOS1995A].

O spider é frequentemente descrito como o “componente móvel” de um sistema de buscas, mas esta é apenas uma figura de linguagem. Na verdade ele não passa de um programa que faz conexões HTTP com os servidores Web, da mesma maneira que um browser o faria. A única diferença é que o Spider necessita de pouca ou nenhuma intervenção humana para trabalhar, varrendo a Web de maneira quase aleatória em busca de documentos. A principal vantagem dos Spiders é que eles conseguem atualizar seus bancos de dados em intervalos regulares, ao contrário do que acontece com os catálogos atualizados manualmente, onde a manutenção normalmente ocorre de forma esporádica [KOS1995B].

Por ser parte de um sistema dedicado, o Spider proposto por este projeto ficará restrito aos sites do domínio embrapa.br, indexando, assim, apenas aqueles documentos presentes nos sites da Empresa. Tecnicamente, nada impediria que o Spider ultrapassasse os limites dos

sites da Embrapa, mas esta medida visa apenas facilitar a especialização dos serviços, como será demonstrado mais adiante.

Uma das principais preocupações de quem pretende escrever um spider deve ser evitar a sobrecarga da rede, seja por exigir demais dos servidores sendo visitados, seja por tentar recuperar documentos os quais ele é incapaz de manusear [KOS1993]. Por exemplo, a maioria dos spiders só é capaz de indexar documentos HTML, então a única coisa que um spider vai conseguir ao tentar recuperar um documento PDF é gerar tráfego desnecessário na rede. Os robôs devem também seguir o padrão Robots Exclusion Protocol (REP) [KOS1994], que visa evitar que os robôs tentem indexar sites onde, por uma razão ou outra, não são bem-vindos.

O robô aqui proposto é um programa Java que pode estar em um servidor diferente daquele onde o restante do sistema de buscas reside. O robô receberá, através de uma consulta ao StorageServer, uma lista de URLs a pesquisar, e estabelecerá uma conexão HTTP para cada servidor indicado, recuperando seu arquivo robots.txt, guardando-o em cache. Depois de se certificar de que não está proibido pelo REP de recuperar os documentos indicados pelas URL, o robô tenta recuperá-los, guardando uma cópia compactada de cada documento num segundo cache, que mais tarde serão entregues ao StorageServer para armazenamento e indexação.

O StorageServer

O StorageServer é o programa responsável pelo armazenamento e indexação dos documentos HTML na base de dados do mecanismo de buscas. Será composto por um grupo de programas com diferentes funções como, por exemplo, o servidor de URL (retornando uma lista de URLs a serem consultadas pelos Spiders), o serviço de conexão com o banco de dados e os serviços de indexação e recuperação de dados.

O serviço de conexão e armazenamento se conectará com os Spiders em horários de menor tráfego na rede, solicitando os documentos recuperados. Os documentos recebidos serão descompactados e desmembrados em suas partes componentes, como título, descrição e palavras-chave, e a seguir enviados para armazenamento em cache, até que o processo de indexação ocorra. O processo de indexação percorrerá o cache, indexando e pontuando os documentos conforme o seu conteúdo. Os documentos serão pontuados num processo semelhante ao PageRank proposto por Sergey Brin e Lawrence Page, criadores do Google [BRI1998]. Tal sistema é uma maneira inovadora de estabelecer a importância ou qualidade dos documentos publicados na World Wide Web, usando para isso a própria estrutura dos documentos.

PageRank e outros sistemas de pontuação calculam o escore de uma página levando em conta os links que apontam para ela [BRI1998, RAF2000, ZHA2000]. Vamos supor que uma página A aponte para uma página B. A existência desse link deve pressupor um endosso do autor da página A em relação à página B. Se a página A é um bom documento (isto é, tem um bom escore), então isso leva a crer que a página B também é um bom documento. Ou seja, quanto mais documentos apontarem para uma página, melhor esse documento deve ser. E se uma busca na Web pretende encontrar os melhores documentos sobre um assunto, os documentos com melhores escores devem aparecer no topo da lista de resultados.

O Searcher

O Searcher é o subsistema responsável pela consulta e recuperação de documentos no banco de dados. Toda vez que o usuário usar o formulário de consulta, submetendo uma pesquisa, o Searcher será invocado, tratará os termos da consulta, efetuará uma conexão ao banco de dados, comporá uma instrução SQL, recuperará os resultados obtidos e formatará uma página de respostas ao usuário apontando os documentos mais importantes.

Durante o processo de tratamento dos termos usados, uma busca num dicionário fonético será efetuada, para evitar que erros de ortografia por parte do usuário distorçam os resultados obtidos.

O RedirectServlet

O RedirectServlet será invocado toda vez que o usuário clicar em um link apontado pelo Searcher. Essa camada serve para, progressivamente, melhorar os resultados do mecanismo de buscas, confiando na premissa de que aqueles documentos mais clicados, dependendo do contexto da pesquisa, pareceram mais interessantes para os usuários.

Para que este mecanismo seja efetivo, uma boa identificação do contexto de cada busca é necessária. Uma maneira de alcançar esse objetivo ainda está sendo estudada, mas é provável que uma consulta a um dicionário de ontologias seja suficiente para uma identificação aceitável [MEI1997].

5. ESTADO ATUAL DO PROJETO

O projeto aqui descrito foi aprovado pelo Comitê Técnico do Programa 14 da Embrapa, e estará em desenvolvimento na Embrapa Pecuária Sul entre janeiro de 2001 e junho de 2002. Os estudos preliminares se iniciaram em meados de outubro de 2000, e se espera já para junho de 2001, uma versão preliminar para fins de testes de performance e aptidão.

O novo sistema de buscas será validado através da análise dos seus logs, procurando-se identificar o número de consultas realizadas e o número de documentos obtidos como resposta, comparando seus resultados com os do mecanismo atualmente em uso no site da Embrapa Pecuária Sul.

Testes de desempenho também serão realizados, comparando o tempo de acesso do novo sistema em diferentes configurações, sempre tentando identificar os ajustes ótimos.

Uma vez pronto e testado, o mecanismo proposto (bem como a base de dados criada como resultado) será colocado à disposição das demais unidades da Embrapa, para que elas também possam usufruir das vantagens que tal ferramenta certamente proporcionará.

6. REFERÊNCIAS BIBLIOGRÁFICAS

- [BER2000] BERNARDES, Ricardo Martins. Um estudo sobre a demanda de informações em sites web: o caso de uma unidade de pesquisa de uma empresa de P&D para o agronegócio brasileiro. Projeto de dissertação para o curso de Administração da UFRGS. 1999
- [BRI1998] BRIN, Sergey; PAGE, Lawrence. The anatomy of a large-scale hypertextual web search engine. Computer Networks and ISDN Systems, volume 30, 1998. 107-117.
- [EMB1995] Estratégia gerencial da Embrapa: Gestão 95/98. Empresa Brasileira de Pesquisa Agropecuária. D.E., Brasília, 1995. 27p.
- [EMB1996] Política de comunicação. Empresa Brasileira de Pesquisa Agropecuária. D.E., Brasília, 1996. 57p.
- [FAP2000] Registro .br – Estatísticas. FAPESP, 2000. Disponível por WWW em <http://registro.br/estatisticas.html> (2-Ago-2000).
- [GRU2000] Livro verde da sociedade da informação no Brasil. Grupo de Implantação do Programa Sociedade da Informação, 2000. 90p.
- [KOS1993] KOSTER, Martijn. Guidelines for robots writers. 1993. Disponível por WWW em <http://info.webcrawler.com/mak/projects/robots/guidelines.html> (3-Out-2000).

- [KOS1994] KOSTER, Martijn. A standard for Robot Exclusion. 1994. Disponível por WWW em <http://info.webcrawler.com/mak/projects/robots/norobots.html> (3-Out-2000).
- [KOS1995A] KOSTER, Martijn. The web robots FAQ. 1995. Disponível por WWW em <http://info.webcrawler.com/mak/projects/robots/faq.html> (3-Out-2000).
- [KOS1995B] KOSTER, Martijn; Robots in the web: threat or treat? 1995. Disponível por WWW em <http://info.webcrawler.com/mak/projects/robots/threat-or-treat.html> (3-Out-2000).
- [MEI1997] MEIRA, Silvio Lemos; SALGADO, Ana Carolina; BARROS, Flávia de Almeida; ROBIN, Jacques; GONÇALVES, Pedro Falcão. BRight! Brazilian internet guide in hypertext. 1997. Disponível por WWW em <http://www.di.ufpe.br/~bright/cnpq/root.html> (28-Jul-1999).
- [OVE1999] OVERMERR, Mark A.C.J.. My personal search engine; Computer Networks, volume 31, 1999. 2271-2279.
- [RAF2000] RAFIEI, Davood; MENDELZON, Alberto O. What is this page known for? Computing Web page reputations. Computer Networks, volume 33, 2000. 823-835.
- [SUL2000A] SULLIVAN, Danny. Search engine sizes; SearchEngineWatch.com. Disponível por WWW em <http://www.searchenginewatch.com/reports/sizes.html> (2-Ago-2000)
- [SUL2000B] SULLIVAN, Danny. Invisible web gets deeper. The Search Engine Report. Disponível por WWW em <http://www.searchenginewatch.com/sereport/00/08-deepweb.html> (2-Ago-2000)
- [ZHA2000] ZHANG, Dell; DONG, Yisheng. An efficient algorithm to rank web resources. Computer Networks volume 33. 2000. 449-455.

A CRIAÇÃO DE COOPERATIVAS AUTÊNTICAS COMO ALTERNATIVA DE GERAÇÃO DE EMPREGO E RENDA

Laudes Machado da Silva¹

RESUMO

Este trabalho se propõe a levantar questionamentos pertinentes à implementação de projetos para formação de núcleos cooperativos (cooperativas singulares) sustentados na autenticidade das intenções de realização pessoal e coletiva sob a bandeira do cooperativismo de trabalho. As considerações finais tratam de dar uma interpretação a respeito do que se entende como encaminhamento ideal para viabilizar a conquista dessa pretendida e necessária autenticidade para o cooperativismo como um todo.

Palavras chave: Cooperativismo, Autenticidade, Relações e Princípios.

ABSTRACT

The purpose of this work is to rise questions about implementation of projects so as to form cooperative nuclei (single cooperatives) based on the real intentions of personal and collective accomplishment lead by the idea of cooperative work. The final considerations, try to give a clear idea of what is understood as the best way to make this intended and necessary authenticity possible for cooperativism as a whole.

key Words: Cooperativism, Authenticity, Relationships and Principles.

1. INTRODUÇÃO

Os altos índices de desemprego, marca registrada do Projeto Neoliberal, e mesmo o crescimento expressivo do contingente de pessoas na condição de subempregados vêm contribuindo, sobremaneira, para o surgimento de um número cada vez maior de cooperativas de trabalho como alternativa na geração de emprego e renda em nosso País.

Resta saber em que condições essas iniciativas têm sido geradas. A preocupação com a questão da necessária e conveniente autenticidade do processo de concepção e constituição dos núcleos cooperativos, senão observada, pode levar ao desvirtuamento dos reais propósitos desse movimento, acabando por atrasar, de maneira perniciosa, um processo em marcha que carrega potenciais significativos como proposta de combate à geração de novos excluídos e na recuperação do número de postos de trabalho.

Neste contexto, caracterizar as relações das pessoas com as organizações a que se vinculam, destacando as diferenças *sui generis* da relação cooperativada, pode representar uma forma de análise capaz de elucidar algumas questões relevantes, proporcionando um bom entendimento a respeito do quê e o como se deve fazer para a formação e administração de autênticas associações cooperativas em condições reais de produzir os resultados pretendidos na realocação e geração de novas frentes de trabalho.

Com o mesmo propósito, revisar os Princípios Doutrinários do Cooperativismo, à luz das interpretações do seu órgão maior de representação - a Aliança Cooperativa Internacional - ACI, levadas a efeito no Congresso de Manchester, realizado no ano de 1995, se traduz com certeza numa interessante alternativa para o alcance dos objetivos pretendidos nesta abordagem.

¹ Bacharel em Administração, Mestrando em Administração PPGA da UFRGS, Professor do CCEI - URCAMP

Campus Universitário de Sant'Ana do Livramento, e-mail: laudes@v-expressa.com.br

2. CARACTERÍSTICAS DAS RELAÇÕES HOMEM X ORGANIZAÇÃO EMPRESARIAL

No mundo atual, vigoram três formas genéricas de organização da economia: duas bem conhecidas e a terceira, com conceito formulado há mais de cem anos, mas de conhecimento restrito. São elas: o Capitalismo (privilegia o capital) o Socialismo (privilegia o Estado) e a terceira (privilegia o indivíduo) e é chamada de Economia Social. Neste particular, advoga IRION (1997) que: “a Economia Social é a organização do processo econômico que tem por centro a pessoa. É uma forma de economia empresarial de natureza associativa, que cria atividades autônomas (autogestionada) com objetivos baseados na solidariedade e democracia e que dá primazia às pessoas e ao trabalho sobre o capital na distribuição dos benefícios”, aí vamos encontrar o cooperativismo formal².

Com a preocupação de começar a dirigir a abordagem deste tema para as especificidades do cooperativismo de trabalho e as suas potencialidades como instrumento de combate ao desemprego e geração de renda, passamos diretamente à descrição das relações havidas entre elemento humano e organização empresarial. Nessa relação, a posição do quadro social frente à cooperativa a que pertence, apresenta-se de forma singularmente diferenciada, se comparada a todas as demais configurações construídas nas relações das pessoas com suas organizações. Em primeiro lugar, a condição de homem-sócio garante ao indivíduo uma posição de dono da empresa, fato que, mesmo existindo em outras relações, que não a de cooperado x cooperativa, não tem a mesma conotação materializada nesta relação, ou seja, a condição de dono da cooperativa é garantida ao associado pelas prerrogativas próprias do sistema cooperativo como, por exemplo, o direito e o peso de voto igual para todos, independente do capital ou volume de produção aportado na cooperativa. Por outro lado, o associado assume, também, a condição de usuário da cooperativa, uma vez que o grande propósito das sociedades cooperativas é o de prestar serviços aos seus associados. Dependendo do tipo de cooperativa, o associado poderá ainda assumir a condição de empregado da sociedade-empresa, como ocorre muitas vezes no segmento cooperativo agropecuário.

Quando se trata de caracterizar o quadro social das cooperativas de trabalho, temos uma diferenciação ainda mais acentuada neste aspecto pois, em sendo sócio, será dono, usuário, por vezes dirigente, e sempre a “força de trabalho” da cooperativa, mas nunca empregado desta. Situação que fica bem clara pelo aditamento da CLT com um parágrafo ao seu art. 442, através da Lei n.º 8.949, de 09.12.94, com os seguintes termos: “qualquer que seja o ramo de atividade da sociedade cooperativa, não existe vínculo empregatício entre ela e seus associados, nem entre estes e os tomadores de serviços daquela”.

Essa caracterização propõe-se a descortinar aspectos que se entrelaçam nas análises dos conteúdos que tratam da questão da autenticidade do sistema cooperativo, cuja proposta original, demonstra preocupações reais de combater a exclusão social, eliminar o lucro mercantil e humanizar as relações de comércio e trabalho .

3. OS PRINCÍPIOS DO COOPERATIVISMO

Formatados pelos Pioneiros de Rochdale e revisados ultimamente no Congresso de Manchester, realizado em 1995, na Inglaterra, são os Princípios Doutrinários do Cooperativismo os caracterizadores máximos da singularidade desta modalidade associativa e empresarial.

Analisando-se a proposta da última revisão dos princípios temos: **1º Princípio** – Adesão livre e voluntária – Cooperativas são organizações voluntárias abertas a todas as pessoas

² Cooperativas organizadas com base legal. No Brasil são definidas conforme a Lei 5.764 de 16 de dezembro de 1971

aptas a usar seus serviços e dispostas a aceitar responsabilidades de sócio, sem discriminação social, racial, política ou religiosa e de gênero. **2º Princípio** – Controle democrático pelos sócios – As cooperativas são organizações democraticamente controladas por seus sócios, os quais participam ativamente no estabelecimento de suas políticas e tomadas de decisões. Homens e mulheres eleitos como representantes são responsáveis para com os sócios. Nas cooperativas singulares, os sócios têm igualdade de votação (um sócio, um voto); as cooperativas de outros graus também são organizadas de maneira democrática. **3º Princípio** – Participação econômica dos sócios – Os sócios contribuem de forma equitativa e controlam, democraticamente, o capital de suas cooperativas. Parte desse capital é propriedade comum das cooperativas. Geralmente, os sócios recebem juros limitados (se houver algum) sobre o capital, como condição da sociedade. Os sócios destinam as sobras aos seguintes propósitos: desenvolvimento das cooperativas, possibilitando a formação de reservas, parte dessas podendo ser indivisíveis, retorno aos sócios na proporção de suas transações com a cooperativa e apoio a outras atividades que forem aprovadas pelos sócios. **4º Princípio** – Autonomia e independência – As cooperativas são organizações autônomas para ajuda mútua, controladas por seus membros. Entrando em acordo operacional com outras entidades, inclusive governamentais, ou recebendo capital de origem externa, elas devem fazê-lo em termos que preservem o seu controle democrático pelos sócios e mantenham sua autonomia. **5º Princípio** – Educação, treinamento e formação – As cooperativas proporcionam educação e treinamento para os sócios, dirigentes eleitos, administradores e funcionários, de modo a contribuir efetivamente para o seu desenvolvimento. Eles deverão informar o público em geral, particularmente os jovens e os líderes formadores de opinião, sobre a natureza e os benefícios da cooperação. **6º Princípio** – Cooperação entre cooperativas – As cooperativas atendem seus sócios mais efetivamente e fortalecem o movimento cooperativo, trabalhando juntas através de estruturas locais, regionais e internacionais. **7º Princípio** – As cooperativas trabalham pelo desenvolvimento sustentável de suas comunidades, através de políticas aprovadas por seus membros.

A sobreposição das bases teóricas contidas no contexto das relações do cooperado com sua sociedade-empresa e nos preceitos doutrinários do sistema servem de referencial para uma perfeita interpretação do ideário cooperativista e da distinção fundamental a ser grifada quando da comparação entre estas organizações e aquelas que não as da iniciativa cooperativada.

4. A FORMAÇÃO DE AUTÊNTICAS COOPERATIVAS

Na justaposição dos fatos que geram, muitas vezes, toda sorte de interpretações a respeito da proposta e das intenções do movimento cooperativista, de maneira especial, do movimento cooperativo de trabalho, procede colocar em evidência alguns dos fatores que mais têm contribuído para esse estado de coisas.

Em primeiro lugar, a cooperativa se mostra como uma modalidade empresarial de difícil conceituação, isto pela possibilidade que existe de mover-se num leque de opções bastante amplo, no que diz respeito ao objeto que pode adotar para consecução de suas finalidades. Fator que fica vinculado ao vislumbamento das ações que devem ser implementadas para o atendimento das necessidades particularizadas nos interesses dos seus cooperados.

Soma-se a esta dificuldade, o fato constatado de existirem algumas interpretações equivocadas a respeito do sistema, como por exemplo: entender as cooperativas como um fim em si mesmas, logo alijadas do contexto de mercado competitivo ou, na pior das hipóteses, imaginá-las como entidades filantrópicas. Explica-se que estas interpretações são frutos de uma predisposição intencional dos inimigos do movimento (segmentos não cooperativados) que, em verdade, ainda que expressem de maneira diversa, entendem as organizações cooperativas como uma proposta que representa, verdadeiramente, uma

alternativa consistente para a solução de alguns dos problemas mais graves no tocante à conquista de um equilíbrio desejável entre as necessidades das pessoas e os interesses das organizações. Por isso mesmo reage à concorrência com todo o ardil que está ao seu alcance.

Também faz parte desse conjunto de barreiras impostas ao crescimento oportuno e adequado do movimento cooperativo de trabalho no Brasil, a equivocada matriz tributária do governo federal, que orientada pela necessidade imediatista de gerar receitas a qualquer custo, acaba permitindo tomadas de decisões que resultam em atrasos danosos para o segmento cooperativo como um todo.

Para combater ou fazer frente a este conjunto de fatores que procuram inibir ou desarticular a organização do sistema cooperativo no país, será necessário que se adote como única proposta viável para o cooperativismo brasileiro, aquelas iniciativas que privilegiem a integridade do movimento, tal qual foi concebido pelos Probos Pioneiros de Rochdale³, fundamentalmente adotando, na prática, os princípios que norteiam o sistema, representando, em última análise, a melhor resposta na defesa do seu ideário e condição *sine Qua non* para lograrem o respeito e a renição da sociedade econômica aos seus dotes de instrumento capaz de viabilizar uma saída mais digna para as classes menos favorecidas.

Salienta-se que a origem de formação das cooperativas singulares (núcleos cooperativados) e a forma de condução desses processos poderá assegurar ou, como resultado indesejado, distanciar a iniciativa do atingimento dos resultados almejados.

Portanto, é recomendável que a fase de concepção e criação propriamente dita da sociedade cooperativa seja acompanhada das discussões de todos os temas que se propõem a identificar o perfil dos futuros cooperativados, bem como os objetivos que estes buscarão conquistar sob a bandeira do cooperativismo. A democracia, signo do movimento cooperativista, deve acompanhar a sociedade-empresa desde o seu nascedouro, o que prevê uma participação ampla e aberta de todos os candidatos a sócio nos encontros promovidos pela futura sociedade. Não deve, no entanto, representar esta participação a impossibilidade de se buscar uma conveniente depuração do futuro quadro social, ou seja, interessante seria que viessem a pertencer à cooperativa somente aqueles que realmente estejam em sintonia com o ideário cooperativista de ajuda mútua e entreajuda.

5. OS NÚMEROS EXPRESSAM UM REALIDADE INCONTESTÁVEL

Os números podem nos dizer, além das palavras, quando se busca conhecer um pouco mais a respeito do avanço do movimento cooperativo de trabalho em nosso país. Esta realidade está demonstrada na tabela 1.

**Tabela 1 - COOPERATIVAS, COOPERADOS E EMPREGADOS,
RAMO TRABALHO, POR REGIÃO**

SEGMENTO	Região	Cooperativas	Cooperados	Empregados
TRABALHO	CENTRO OESTE	73	5.429	73
	NORDESTE	374	30.710	1.409
	NORTE	41	12.252	72
	SUDESTE	919	207.210	3.773
	SUL	254	37.898	1.095
	TOTAL	1.661	293.499	6.422

Fonte: Núcleo de Banco de Dados da OCB⁴

Elaboração: Núcleo de Banco de Dados – dez/99

³ Ingleses idealizadores do Movimento Cooperativo Formal, cujo marco referencial deu-se no ano de 1844.

⁴ Organização das Cooperativas Brasileiras.

6. CONCLUSÕES

Embora alguns descaminhos tenham penalizado o sistema cooperativo no seu processo de evolução em nosso país, fruto de adoções equivocadas, ora por desavisados, ora por mal intencionados, exploradores da força de trabalho alheia e, naturalmente, sem querer entendê-lo como uma panacéia, pois os tempos atuais curvam-se diante da racionalidade econômica, temos que: a) uma vez observadas as orientações e os caminhos a serem percorridos para a constituição de sociedades cooperativas autênticas, com administrações profissionais que privilegiem métodos de gestão ágeis e atualizados, capazes de oferecer maior eficiência às atividades da empresa; b) adotando-se, “na prática”, os princípios doutrinários norteadores do sistema cooperativista; c) finalmente, conjugando-se esses recursos a uma perfeita interpretação da forma diferenciada em que se traduzem as relações do homem-sócio com sua cooperativa, estarão garantidos os fatores que poderão, de forma definitiva, sustentar o mérito atribuído a essa modalidade sócio-empresarial como organização humanizadora das relações de comércio e trabalho, com potencialidades reais no combate aos altos índices de desemprego vigentes no país e, na melhor das hipóteses, contribuir na geração de renda.

Quanto à responsabilidade dos governantes neste processo, é de se esperar que não se deixem contaminar pela visão distorcida da realidade atual, contida no modelo estatutário, ou seja, relações trabalhistas determinadas por instrumentos estatais de controle (leis) e possam ser sensibilizados pela evolução social e pelas regras impostas pela globalização, dando oportunidade ao moderno sistema negocial. Sem, no entanto, exigir em contra partida uma penalização ao sistema com a fúria arrecadatória do Estado.

7. BIBLIOGRAFIA

BARELLI, Walter. Ajudando a criar a empresa do futuro. [online] Disponível por WWW em <<http://www.cooperativa.com.br/revista/entrevista.htm>>. Arquivos capturados em novembro de 2000.

IRION, João Eduardo Oliveira. **Cooperativismo e economia social**. São Paulo: Atlas, 1997.

NETO, Sigismundo Bialoskorski. **O que é a nova geração de cooperativas –NGC**. Revista Gestão Cooperativa. Minas Gerais: ANO 02 – Nº 04 – agosto/1998, p.46.

OCB – Organização das Cooperativas Brasileiras. [online] Disponível na Internet: <<http://www.ocb.org.br/>>. Arquivos capturados em novembro de 2000.

PEDROZO, Eugenio Avila. **Cadre conceptuel pour le management des cooperatives agricoles: l'économie des conventions, la systémique et la complexité**. Tese de Doutorado. Institut National Polytechnique de Lorraine. Nancy, France, 1995.

POLONIO, Wilson Alves. **Manual das sociedades cooperativas**. São Paulo: Atlas, 1999.

SCHNEIDER, José Odelso. Democracia, participação e autonomia cooperativa. 2. Ed. São Leopoldo: UNISINOS, 1999.

DESENVOLVIMENTO DE UM ESQUEMA XML PARA BANCO DE DADOS SOBRE OVINOS

João Abelar Martins Costa¹

Antônio Carlos da Rocha Costa²

RESUMO

O maior patrimônio de uma empresa que trabalha com banco de dados são as informações contidas no mesmo. Muitas empresas precisam disponibilizar essas informações e deparam-se com um problema: como disponibilizar a um custo baixo, evitando o uso de papel, dispositivos de armazenamento “ZIP-DRIVE” e até mesmo o deslocamento do usuário até a empresa para fazer as devidas consultas. A Internet está surgindo como uma solução para esse problema, disponibilizando novas tecnologias no tratamento de informações, tecnologia esta conhecida como XML, que possibilita armazenar informações em um banco de dados (padrão XML), e a partir daí o usuário acessá-las via Internet, sem que o mesmo precise abalar-se até as dependências da empresa. Junto ao XML são incorporadas outras tecnologias, tais como ESQUEMAS e XHTML, que têm por função criar regras para esse banco de dados e permitir a visualização das informações.

Palavra-chave: XML.

ABSTRACT

The greatest asset of a business working with a data bank is the information contained within it. Many businesses need to avail readily such information and they face a problem: how to avail same at low cost, avoiding the use of paper, storage devices, zip-drive, and even the user's physical transportation to the business site so as to perform the necessary consultations. Internet looms as a solution for such a problem, in that it renders available new technologies for the handling of information, like the technology known as XML, which makes possible the storage of information in a data bank (standard XML) which the user can then access via internet without his/her physically moving to the business premises. Along with XML, other technologies are incorporated like esquemas and XHTML, whose function is to generate rules for the data bank and also to allow for the visualization of stored information.

Keyword: XML.

1. INTRODUÇÃO

Percebendo-se os problemas encontrados no setor primário, devido à dificuldade na obtenção de informações e a falta de incentivos a esse setor, optou-se por elaborar um projeto que viesse beneficiá-lo, bem como propiciar um maior desenvolvimento.

As ferramentas disponíveis hoje em dia, pela Internet (como por exemplo, XML), serão os recursos utilizados na formação deste projeto, o que revolucionará a forma como hoje são elaborados os trabalhos nesse setor, trazendo maior segurança e confiabilidade nos resultados obtidos.

Com isto pretende-se tornar o setor primário mais competitivo, fazendo uso de um banco de dados já existente na A.R.C.O. (Associação Brasileira de Criadores de Ovinos), o que permitirá que o usuário tenha acesso aos dados referentes a ovinos utilizando a Internet como meio de comunicação.

¹ Professor do CCEI – URCAMP – Mestrando em Ciência da Computação PPGCC/UFRGS – Email:jamc@urcamp.tche.br

² Professor da Escola de Informática – UCPel – PPGCC/UFRGS – Email:rocha@atlas.ucpel.tche.br

Hoje, o pecuarista precisa deslocar-se até a entidade (A.R.C.O.) para tomar posse dessas informações, o que muitas vezes é visto como um prejuízo pelo mesmo, desta forma enfraquecendo o vínculo empresa/usuário.

Com o uso desta tecnologia (XML), o prejuízo decorrente da locomoção será sanado, pois o acesso via Internet permitirá que o usuário interaja com o banco de dados, trabalhando com as informações sem precisar sair de casa. Em contrapartida, a entidade (A.R.C.O.) ficará mais próxima de seus usuários.

Tem, também, como objetivo, estimular por parte do criador a elaboração de banco de dados de acordo com o padrão estipulado pela A.R.C.O..

2. XML - (EXTENSIBLE MARKUP LANGUAGE)

XML surgiu de um refinamento da SGML(Standart Generalized Markup Language), que por sua vez é uma linguagem poderosa, mas de difícil compreensão . O XML incorpora as principais características da SGML no que diz respeito à distribuição de informações (Banco de Dados) via WWW, com a vantagem de ser de fácil compreensão por parte do desenvolvedor.

O XML e o HTML possuem uma origem comum, o SGML, e servem para propósitos distintos, sendo assim é errôneo dizer que o XML veio para substituir o HTML, pois com o auxílio do HTML podemos trabalhar informações dispostas em XML.

Em XML, projetistas podem criar seus próprios elementos de acordo com a aplicação que está sendo modelada, dando importância ao conteúdo e à estrutura da informação, sem se preocupar com a apresentação. Para que o parser XML verifique se um documento está correto ou não (parser de validação), ele processa inicialmente seu DTD (Document Type Definitions) correspondente para verificar a estrutura do documento . Vários parsers, disponíveis gratuitamente no WWW, verificam (análise léxica) e validam (análise sintática) documentos XML de acordo com seu DTD associado. Além desses, há também parsers que não exigem a presença de um DTD e que, portanto, somente validam seus elementos.

3. ESQUEMA XML

A necessidade de se criar regras bem formadas para receber e enviar informações de/para uma base de dados originou o surgimento de esquemas.

Os esquemas XML são documentos controladores, controle este executado por regras estabelecidas pelas normas de segurança e integridade do referido banco de dados. De posse dessas regras, o programador constrói o esquema XML que irá interagir junto ao banco de dados XML e o usuário, não permitindo dados indesejáveis e ajudando o mesmo na tomada de decisões.

Um esquema XML possui uma sintaxe baseada na XML, por definição usa a forma de marcação de um documento XML.

4. APLICAÇÃO DE UM ESQUEMA XML EM UM ARQUIVO XML

Tendo por objetivo construir um banco de dados XML protegido por regras elaboradas por um esquema XML, é preciso primeiramente criar um banco de dados compatível com o existente na ARCO (Associação Brasileira de Criadores de Ovinos) e, posteriormente, estudar uma forma de formatar esses dados, para isso utilizar uma tecnologia conhecida como esquema XML, que nada mais é do que uma linguagem de formatação.

Para criar um banco de dados XML compatível com o existente deverá ser feito um levantamento dos dados e, após, das regras de formação dos mesmos (tipo de dado, obrigatoriedade de existência do mesmo, etc). Com base nos dados coletados, criar uma

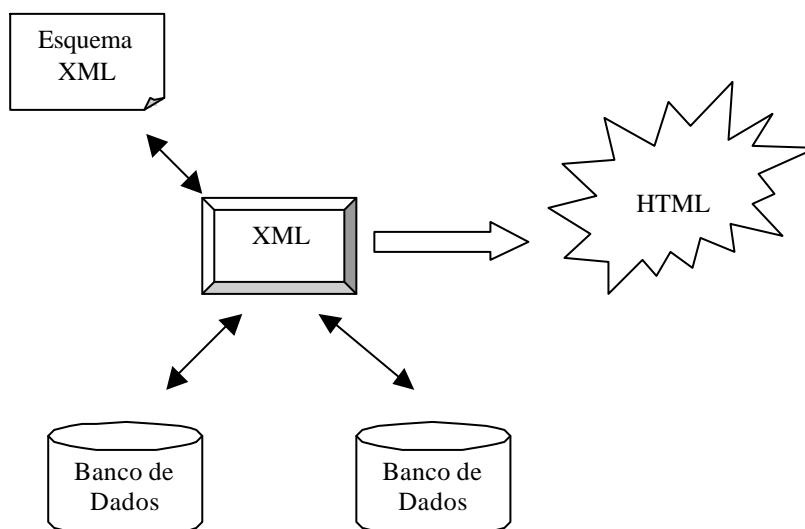
versão preliminar de um esquema XML que atuará junto ao banco de dados XML proposto.

A seguir será exposto um exemplo de **Esquema XML** para um **banco de dados XML** de cadastro.

```
<?xml version = "1.0"?>
<!--Generated by XML Authority. Conforms to XML Data subset for IE 5-->
<Schema name = "Fichario.biz"
  xmlns = "urn:schemas-microsoft-com:xml-data"
  xmlns:dt = "urn:schemas-microsoft-com:datatypes">
  <ElementType name = "Fichario" content = "eltOnly" order = "seq">
    <element type = "Socio"/>
    <element type = "Nome"/>
    <element type = "CXP" minOccurs = "1" maxOccurs = "*/>
    .
    .
  <ElementType name = "Socio" content = "textOnly" dt:type =
"string">
    <AttributeType name = "Codigo" dt:type = "id"/>
    <attribute type = "Codigo"/>
  </ElementType>

  <ElementType name = "Nome" content = "textOnly" dt:type =
"string"/>
  <ElementType name = "CXP" content = "textOnly" dt:type = "string"/>
  .
  .
  <ElementType name = "ARCO" content = "eltOnly" order = "seq">
    <element type = "Fichario" minOccurs = "0" maxOccurs = "*/>
  </ElementType>
</Schema>
```

O resultado será apresentado via WWW com auxílio do browser (Internet Explorer 5), acionando uma página HTML que por sua vez busca os dados do arquivo XML e apresenta no vídeo, sendo este arquivo XML protegido por um esquema XML.



5- CONCLUSÃO

Conscientes da importância do desenvolvimento do setor primário e das dificuldades encontradas pelo mesmo em adquirir tecnologia de informações procurou-se fazer uso de uma ferramenta que traria um avanço tecnológico para esse setor.

Fazendo uso da Internet e de recursos tais como XML e Esquemas, disponibilizar informações (com um padrão definido) para acesso independente de linguagem de programação e/ou plataforma, permitindo que a entidade (ARCO) se aproxime de seus associados a um baixo custo, pois os mesmos não precisarão deslocar-se até a entidade para obtenção de informações sobre cadastros tais como: Associados e Registro de Ovinos.

6- REFERÊNCIAS BIBLIOGRÁFICAS

Layman, Andrew. XML Data. janeiro 1998.

<http://www.w3.org/tr/1998/note-xml-data-0105/>

Malhotra, Ashok. XML Schema Requirements. fevereiro 1999.

<http://www.w3.org/tr/note-xml-schema-req>

Alshuler, Liora. Schema Repositories. janeiro 2000.

<http://xml.com/pub/2000/01/26/feature/index.html>

Bray, T. e DeRose, S. Extensible Markup Language (XML) Part 2: Linking. "XML Principles, Tools and Techniques". Setembro 1997, V.2,n.4,pp.219-227

Richard Light. Tradução, Neilande de Moraes. "Iniciando em XML". MAKRON Books. São Paulo. Junho 1997

Prentice Hall, Inc. Tradução, Victor Hugo da Paixão Alves. "XML Aplicações Práticas". Ed. Campus. Rio de Janeiro, 1997

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 feita em três vias impressas, sendo duas sem a identificação dos autores.

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: 15/06/2001
Publicação: Agosto/2001

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.ccei.urcamp.tche.br/revista/>