

Projetos e Pesquisas em Ciência da Computação no DCOMP/PROCC/UFS



Maria Augusta Silveira Netto Nunes
Adicinéia Aparecida de Oliveira
Edward David Moreno Ordonez

Organizadores



Editora UFS

Cidade Universitária José Aloísio de Campos

CEP - 490100-00 - São Cristóvão - SE

Telefone: 079 2105-6920 / 2105-6922 / 2105-6923

Email: editora@ufs.br

www.ufs.br/editora

Maria Augusta Silveira Netto Nunes
Adicinéia Aparecida de Oliveira
Edward David Moreno Ordonez

Organizadores

Projetos e Pesquisas em Ciência da Computação no DCOMP/PROCC/UFS

Adicinéia Aparecida de Oliveira
Admilson de Ribamar Lima Ribeiro
Antônio Monteiro Freire
Carlos Alberto Estombelo-Montesco
Debora Maria Coelho Nascimento
Edward David Moreno Ordonez
Kênia Kodel Cox
Leila Maciel de Almeida e Silva
Marco Túlio Chella
Maria Augusta Silveira Netto Nunes
Rogério Patrício Chagas do Nascimento
Sérgio Queiroz de Medeiros
Tarcísio da Rocha

São Cristóvão, 2012



© 2012, Editora UFS.

Todos os direitos reservados e protegidos pela Lei 5988 de 14/12/1973.

Nenhuma parte deste livro, sem autorização prévia por escrito da editora, poderá ser reproduzida ou transmitida sejam quais forem os meios empregados: eletrônicos, mecânicos, fotográficos, gravação ou quaisquer outros.

Capa

Meiry Ane Acciole Gomes

Supervisão Editorial

Adicinéia Aparecida de Oliveira

Editoração Eletrônica

Meiry Ane Acciole

**FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA CENTRAL
UNIVERSIDADE FEDERAL DE SERGIPE**

P964p

Projetos e Pesquisas em Ciência da Computação no DCOMP/PROCC/
UFS / Maria Augusta Silveira Netto Nunes, Adicinéia Aparecida
de Oliveira, Edward David Moreno Ordonez, Organizadores – São
Cristóvão: Editora UFS, 2012.

241 p.: il.

ISBN: 978-85-7822-240-6

1. Engenharia de *Software*. 2. *Software* - Mediação. 3. Ciência da
Computação - Pesquisa. I. Nunes, Maria Augusta Silveira Netto.
II. Oliveira, Adicinéia Aparecida de. III. Ordonez, Edward David
Moreno.

CDU 004.41-047.3

Sumário

Prefácio, 7

Adicinéia Aparecida de Oliveira e Antônio Monteiro Freire

Introdução, 9

Leila Maciel de Almeida e Silva

Seção 1 - Engenharia de *Software*

1. EXPERIÊNCIAS NA UTILIZAÇÃO DE MÉTODOS ÁGEIS NO DESENVOLVIMENTO DE TRABALHOS ACADÊMICOS, 15
Debora Maria Coelho Nascimento
2. COMPUTAÇÃO E ENSINO: “ENSINO DE COMPUTAÇÃO” E “COMPUTAÇÃO NO ENSINO”, 33
Kênia Kodel Cox
3. METODOLOGIA PETIC – PLANEJAMENTO ESTRATÉGICO DE TIC, 55
Rogério Patrício Chagas do Nascimento
4. CASAMENTO DE PADRÕES USANDO UMA MÁQUINA VIRTUAL DE PARSING PARA PEGS, 79
Sérgio Queiroz de Medeiros

Seção 2 - Computação Inteligente

5. MÉTODOS COMPUTACIONAIS PARA A DETECÇÃO, CARACTERIZAÇÃO E EXTRAÇÃO DE PADRÕES DE ATIVIDADE A PARTIR DE DADOS BIOMÉDICOS, 95
Carlos Alberto Estombelo-Montesco
6. COMPUTAÇÃO AFETIVA PERSONALIZANDO INTERFACES, INTERAÇÕES E RECOMENDAÇÕES DE PRODUTOS, SERVIÇOS E PESSOAS EM AMBIENTES COMPUTACIONAIS, 113
Maria Augusta Silveira Netto Nunes

Seção 3 - Sistemas Distribuídos e Arquitetura de Computadores

7. NOVAS DIREÇÕES EM ENGENHARIA *WEB*, 153
Admilson de Ribamar Lima Ribeiro
8. P&D EM ARQUITETURA DE SISTEMAS COMPUTACIONAIS E SISTEMAS EMBARCADOS, 165
Edward David Moreno Ordonez
9. O ENCONTRO DA *INTERNET* DAS PESSOAS COM A *INTERNET* DOS OBJETOS, 187
Marco Túlio Chella
10. PESQUISA EM SERVIÇOS DE TRANSAÇÕES PARA COMPUTAÇÃO MÓVEL E MIDDLEWARES ABERTOS, 211
Tarcísio da Rocha

Prefácio

Um longo tempo transcorreu entre a invenção do Ábaco e a popularização da *Internet*. O impacto das mudanças promovidas pelas tecnologias ao longo da história, em especial da computação, mudou a maneira como a sociedade se comunica, diverte-se, trabalha ou mesmo estuda. É inegável a evolução e a penetração das tecnologias em todos os extratos sociais.

Nas universidades e laboratórios, pesquisadores têm pensado muito mais sobre o futuro e refletido sobre o desenvolvimento e o papel das tecnologias na sociedade. Esse é o papel da universidade: ensinar, estudar, debater, questionar e divulgar as tecnologias.

Com uma experiência de 22 anos no ensino de Computação, a missão do Departamento de Computação (DCOMP) enfatiza a formação de profissionais capazes de assumir um papel de agente transformador da sociedade e aptos para provocar mudanças através da incorporação de novas tecnologias na solução de problemas.

Têm-se ciência e a consciência de que o trabalho acadêmico nunca está acabado, completo. Pelo contrário, é uma construção diária, constante e dedicada. O crescimento é inevitável e o Programa de Pós-Graduação em Ciência de Computação (PROCC) reflete a preocupação do Departamento e a necessidade da sociedade por pesquisadores altamente qualificados.

A nossa ideia inicial era comemorar os 20 anos do DCOMP e qual melhor presente senão mostrar os trabalhos desenvolvidos pelos professores do DCOMP e do PROCC? Quando começamos a planejar este livro, havia preocupação se ele ficaria perdido nas centenas de volumes que já foram escritos sobre pesquisas nos departamentos de computação de todo Brasil. Porém, relevando essa preocupação, concluímos que era necessária a divulgação para a comunidade acadêmica e sociedade, do que estávamos produzindo. Afinal, este é o papel da universidade.

Este livro é uma coletânea de artigos que descrevem pesquisas desenvolvidas por professores do DCOMP e PROCC. Os artigos estão organizados por áreas de conhecimento: Engenharia de *Software* com 4 artigos, Computação Inteligente com 2 artigos e Sistemas Distribuídos, Redes e Arquitetura de Computadores com 4 artigos.

O presente livro pode ser visto como a reflexão da nossa história, muito bem descrita pela Profa. Leila na Introdução, do que fizemos nos últimos anos e, principalmente, a via que estamos construindo para o futuro nos capítulos subsequentes.

Na autoria dos vários artigos aqui apresentados vocês encontrarão a maior parte do nosso quadro docente. A eles o DCOMP e o PROCC agradecem o carinho pela idealização desta obra a qual esperamos seja a primeira desta nova era que se inicia com a aprovação do Mestrado.

Também gostaríamos de expressar os agradecimentos aos nossos alunos, colegas professores e técnicos, afinal todos contribuíram com alguma parcela na construção deste livro.

Boa leitura a todos!

Adicinéia Aparecida de Oliveira
Antônio Monteiro Freire

Introdução

Leila Maciel de Almeida e Silva

O Departamento de Computação originou-se do Departamento de Estatística e Informática, formado em 1979. O primeiro curso oferecido pelo departamento foi o Bacharelado em Informática, cujo projeto foi aprovado na UFS em 1990. A primeira turma ingressou em 1991 e era composta de 30 alunos. Nesta ocasião, compunham o núcleo de Informática do departamento os mentores do curso, Profa. Ana Rosimeire Soares, Profa. Ângela Maria Carvalho Souza, Prof. Antônio Monteiro Freire e Prof. Raimundo Machado Costa. Em 1992, dois novos professores foram contratados, Prof. Henrique Nou Schneider e Profa. Leila Maciel de Almeida e Silva. O Brasil atravessou de 1993 a 2002, um período de retração na contratação de professores para as universidades e, apesar do curso ter sido criado com a proposta de contratação de dez novos professores, o fato não se concretizou, o que impediu a consolidação mais rápida do núcleo de Informática do departamento. Em 1998, quando o curso recebeu a primeira visita da comissão de especialistas do Ministério de Educação (MEC), para efeito de reconhecimento, o núcleo de Informática contava com 8 professores efetivos, pela contratação dos professores Evandro Curvelo Hora, Giovanny Fernando Lucero Palma, Leonardo Nogueira Matos e Luiz Brunelli e pelas aposentadorias dos professores Ângela Maria Carvalho Souza e Raimundo Machado Costa.

Dado ao exíguo número de professores, todos eles na ocasião mestres, e da deficiente estrutura física e laboratorial do curso, o Curso de Bacharelado em Informática foi reconhecido com conceito C em 1998 e foi sugerido pelo MEC o enquadramento da sua denominação como Bacharelado em Ciência da Computação (BCC). A troca do nome foi realizada em 1999 e na ocasião, por sugestão do Prof. Antônio Monteiro Freire, aprovou-se a troca do nome do departamento para Departamento de Ciência da Computação e Estatística, denominação que vigorou até 2006, quando o departamento foi fragmentado em Departamento de Computação (DCOMP) e Núcleo de Graduação em Estatística (NES).

Apesar do exíguo número de professores do departamento, em consonância com o programa de capacitação docente da UFS, implantou uma estratégia de qualificação do seu corpo docente. A Profa. Leila foi a primeira docente a sair para a qualificação em 1996 e, por conseguinte, foi a primeira professora doutora do departamento, no ano de 2000. Curiosamente, foi também a primeira professora doutora em Ciência da Computação a atuar no Estado de Sergipe, um reflexo

da incipiência da área no estado, nessa época o departamento chegou a ter 40% do seu quadro efetivo de Computação fazendo doutorado.

O ano de 2000 foi um marco no departamento por duas razões. Por um lado, o departamento ganhava então seu primeiro docente doutor e a formação do primeiro grupo de pesquisa, o de Engenharia de *Software*, inicialmente denominado de Métodos Formais. Por outro lado, recebia a segunda visita do MEC para efeitos de revalidação do curso de Bacharelado em Ciência da Computação. A visita encontrou o departamento com a mesma estrutura física e corpo docente da época do reconhecimento de curso, a qual já era julgada como insuficiente. Dado que a UFS não havia envidado melhorias no curso, a comissão de especialistas do MEC colocou o curso em Diligência e deu um prazo de um ano para que algumas melhorias fossem realizadas e nova visita fosse realizada.

Apesar do impacto negativo do novo conceito, o fato gerou uma corrente de transformações mais rápidas no BCC. Em 2001 o curso de BCC sofreu uma grande reforma curricular, abandonando-se então a grade com a qual tinha sido criado e evoluindo para uma grade alinhada com padrões de qualidade nacionais e internacionais. Esta reforma gerou excelentes frutos em anos posteriores, quando o curso, desde a primeira avaliação do ENADE obteve conceito 4, numa escala de 1 a 5. Em 2003, após intensa negociação com a administração da UFS, o departamento recebeu pela primeira vez uma instalação com identidade própria, uma área de cerca de 500 m² destinada exclusivamente às atividades do núcleo de Computação do departamento, passando a ocupar essa área em janeiro de 2004. O corpo docente de Computação manteve o permanente zelo pela evolução da qualidade do curso de BCC e micro-reformas na grade aprovada de 2001 foram realizadas em 2005 e 2007. O conceito do ENADE, no entanto se manteve em 4, pois para a obtenção da nota máxima fazia-se necessário a expansão do corpo docente e da infra-estrutura departamental, ações que dependiam da administração da UFS.

A partir de 2003, no âmbito do programa de crescimento do ensino superior no país, as universidades puderam contratar mais professores, especialmente para suprir a oferta de novos cursos à comunidade. Seguindo a onda de expansão que se instaurou na UFS, em 2006 o departamento aprovou o curso de Bacharelado em Sistemas da Informação, cuja primeira turma iniciou em 2007. Em 2008, foi aprovado o curso de Engenharia da Computação, numa proposta conjunta com o Departamento de Engenharia Elétrica e cuja primeira turma iniciou em 2009.

A abertura de cursos novos pelo então DCOMP foi uma estratégia departamental para captação de novos professores, o que se deu de forma mais acelerada a partir de 2005, atingindo atualmente 24 professores, sendo 18 doutores e 6 mestres.

O advento de mais professores consolidou a pesquisa departamental, culminando na criação do Mestrado em Ciência da Computação em maio de 2010, cuja primeira turma iniciou em agosto de 2010. A criação do Mestrado marcou então os 20 anos do DCOMP, os quais comemoramos com a edição deste livro e consolidou a pós-graduação oferecida pelo departamento, que até então tinha apenas podido ofertar quatro cursos de especialização.

Atualmente o DCOMP conta com três grupos de pesquisa, responsáveis por diversos projetos em andamento aprovados por agências de fomento. Os discentes, além do ótimo desempenho no antigo Provão e atual ENADE, também obtiveram premiações. O DCOMP possui três trabalhos selecionados entre os dez melhores do Concurso de Trabalhos de Iniciação Científica (CTIC) da Sociedade Brasileira de Computação (SBC), sendo um deles premiado em terceiro lugar. Além disso, o DCOMP tem quatro trabalhos premiados no Workshop de Iniciação Científica e Trabalhos de Graduação da Regional Bahia, Sergipe e Alagoas (WTICG-BASE). A equipe “Anuncie Aqui”, composta por discentes da Engenharia Elétrica e Computação, foi premiada na Maratona de Programação, tendo sido medalha de bronze em 2007 e medalha de ouro em 2009, fato que proporcionou que a UFS integrasse a delegação brasileira no mundial de 2010 realizado

na cidade de Harbin, China.

O DCOMP atual agrega, de forma harmônica, o dinamismo e esperança dos seus jovens pesquisadores, à maturidade dos professores que o viram nascer, como o Prof. Antônio Monteiro Freire, último fundador do departamento e hoje atuando como seu chefe. O departamento continua em expansão constante e para poder atender aos seus mil e poucos alunos de graduação e pós-graduação em 2012, o departamento deverá pelo menos dobrar seu corpo de docentes e técnicos nos próximos cinco anos e expandir sua infra-estrutura física. Um prédio com 2000 m² está previsto para acrescer o então espaço do atual DCOMP em 2012. O número de grupos de pesquisa deve crescer e a instalação do doutorado na próxima década é a meta principal do PROCC e DCOMP.

Seção 1

Engenharia de *Software*



Experiências na Utilização de Métodos Ágeis no Desenvolvimento de Trabalhos Acadêmicos

Debora Maria Coelho Nascimento

Resumo. Ao longo do tempo, têm-se buscado melhorar a qualidade de projetos de *software* a partir da utilização de processos de desenvolvimento. Por outro lado, além da qualidade, o processo de desenvolvimento precisa entregar valor agregado ao cliente o mais rápido possível, exatamente como proposto pelos métodos ditos Ágeis. *Softwares* desenvolvidos como resultado de trabalhos acadêmicos assemelham-se a outros tipos de projetos de *software*, contudo apresentam um conjunto de restrições, que usualmente ocorrem simultaneamente, o que ocasiona o aumento dos riscos envolvidos. Neste capítulo, relatam-se as lições aprendidas na aplicação de práticas ágeis definidas pelos métodos *Scrum* e *Extreme Programming*, na realização de trabalhos acadêmicos, servindo de reflexão e posterior direcionamento de aplicação desses métodos em trabalhos a serem desenvolvidos no Departamento de Computação da Universidade Federal de Sergipe.

Palavras-chave: Trabalhos Acadêmicos. Métodos Ágeis. *Scrum*. *Extreme Programming*.

1 Introdução

O desenvolvimento de *software* é um processo complexo. Schawber (2004) enumerou três dimensões principais para explicar esta complexidade, são elas: requisitos, tecnologias e pessoas.

Requisitos definem as necessidades que o *software* deve atender, entretanto, como são muitos os envolvidos em um projeto, há diferentes expectativas em jogo, gerando diversos requisitos. Tais requisitos muitas vezes são difíceis de serem compreendidos, às vezes, conflitantes e difíceis de serem articulados. Além de que, muitos requisitos sofrem modificações ao longo do tempo.

A evolução contínua da tecnologia faz com que, em diversas ocasiões, a fim de obter vantagens competitivas, sejam empregadas tecnologias não confiáveis, isto é, ainda não estabilizadas. Em outros momentos, a dificuldade provem da necessidade de integrar diversas tecnologias para atender um único objetivo.

Enfim, as equipes de desenvolvimento são formadas por pessoas que têm diferentes habilidades, experiências, pontos de vista e atitudes, que a cada dia podem ter comportamento diverso, a depender de seus problemas pessoais.

Um dos grandes desafios da Engenharia de *Software* é mesmo com estas dificuldades, ser capaz de gerar produtos de qualidade. Processos de Desenvolvimento buscam melhorar esta qualidade, a partir da definição e organização das atividades de construção do *software*, de modo que a complexidade envolvida diminua, em virtude do maior controle e acompanhamento das tarefas.

Entretanto, segundo Sommerville (2007), modelos como o Cascata convencional ou processos pesados baseados fortemente em especificações demoram muito tempo para apresentar resultados. Há alguns anos, clientes demandam experimentação e inovação contínua, que de acordo com Highsmith (2004), força a substituição de processos prescritivos para modelos adaptativos de desenvolvimento. Sendo assim, os processos de *software* precisam ser evolutivos e adaptativos, ao invés de planejados e otimizados.

Foi com esta ideia que ainda em 2001, dezessete desenvolvedores, a partir de práticas aplicadas em diversos projetos, lançaram um conjunto de doze princípios, chamado de Manifesto Ágil (MANIFESTO, 2001). Estes princípios estão fundamentados nas seguintes prioridades: *indivíduos e interações a processos e ferramentas; software funcionando é mais importante do que documentação completa e detalhada; colaboração com o cliente é mais importante do que negociação de contratos, enfim, adaptação a mudanças é mais importante do que seguir o plano inicial* (MANIFESTO, 2001). Desde então, as abordagens que seguem esses princípios, eram denominadas de Métodos Ágeis. Duas destas abordagens são o *Extreme Programming* (XP) e o *Scrum*.

O *Extreme Programming* é definido por Beck e Andres (2004) como um estilo de desenvolvimento de *software* focado na aplicação de boas técnicas de programação, na comunicação clara e na equipe do projeto. Inclui uma filosofia de desenvolvimento baseada em valores e princípios, que direcionam a aplicação de práticas de construção do *software*.

Segundo Schawber e Sutherland (2010), o *Scrum* não é um processo ou técnica para a construção de produtos, preferencialmente, é um *framework*, onde vários processos e técnicas podem ser aplicados, de modo que produtos complexos possam ser construídos.

Desse modo, muitos trabalhos sugerem o uso conjunto de *Scrum* e XP, citando como exemplos Kniberg (2007) e Vriens (2003). Sendo o *Scrum* utilizado para o planejamento e acompanhamento do desenvolvimento, enquanto que, as práticas XP são utilizadas como forma de garantir a qualidade final do produto.

Usualmente diversos *softwares* são desenvolvidos ao longo dos cursos de computação. Estes *softwares* são frutos da realização de pesquisas, projetos de extensão, ou mesmo, trabalhos de conclusão de curso. Considerando que todo trabalho acadêmico é desenvolvido por professores e alunos, contando algumas vezes com a participação de especialistas, podem-se listar algumas características importantes para este tipo de projeto, são elas:

- As equipes de desenvolvimento geralmente envolvem poucas pessoas, sendo que algumas vezes apenas um desenvolvedor é responsável pela construção do *software*;
- O tempo de dedicação ao projeto é apenas parcial, uma vez que, a equipe, devido ao perfil de seus membros, geralmente está envolvida em várias outras atividades de mesma prioridade. Tal característica também dificulta a proximidade da equipe, de modo que, cada membro dedica-se ao trabalho em horários e locais diversos, caracterizando um desenvolvimento distribuído do *software*;
- Muitas vezes, falta maturidade à equipe com relação à gerência de projetos, dificultando o planejamento, monitoramento e controle do projeto;
- Raramente, os requisitos do *software* podem ser claramente elicitados no início do projeto, em virtude da incerteza sempre presente em trabalhos de pesquisa;
- Por fim, a natureza de inovação frequentemente faz com que sejam utilizadas tecnologias novas, ainda não estáveis, aumentando os riscos técnicos relacionados ao projeto.

Cabe ressaltar, que estas características podem estar presentes em outros projetos de *software*. Entretanto, a presença combinada das mesmas, comum aos projetos acadêmicos, aumentam os riscos do projeto. Não são raros os casos que os trabalhos ficam inacabados, sem que todas as funcionalidades sejam implementadas, ou são finalizados às pressas, afetando sensivelmente a qualidade do produto final.

Costa, Oliveira e Soares (2007) relataram a experiência na aplicação das práticas XP (programação em pares, desenvolvimento dirigido por testes, refatoração e reunião em pé) em conjunto com uma planilha para a gestão de riscos, em um ambiente acadêmico. Quanto às práticas, a única que não obteve sucesso, foi a relacionada aos testes, enquanto que, com relação à gestão de riscos, houve dificuldade no preenchimento da planilha.

Contudo, dada à simplicidade do *Scrum*, acredita-se que o mesmo é adequado à redução dos riscos provenientes das características dos projetos acadêmicos listadas. Já que, de acordo com Schwaber (2004), além de serem poucas práticas, artefatos e regras definidos, são fáceis de aprender. Portanto, professores e alunos podem gerenciar seus projetos, sem que haja desperdício de tempo, e aplicar as práticas XP nas atividades de engenharia.

O objetivo deste capítulo é relatar as experiências vivenciadas com a aplicação das práticas e artefatos definidos pelos Métodos Ágeis, *Scrum* e XP, em projetos acadêmicos, realizados no Departamento de Computação (DCOMP) da Universidade Federal de Sergipe (UFS), a fim de suprir as dificuldades provenientes desse tipo de projeto.

O capítulo está organizado da seguinte forma: na segunda seção, há uma breve descrição dos métodos *Scrum* e *Extreme Programming*; na terceira seção, são apresentados os trabalhos realizados empregando um e/ou outro método, relatando os resultados obtidos, e enfim, tem-se a conclusão, com o direcionamento de trabalhos futuros.

2 Métodos Ágeis

Como reação aos processos pesados, os Métodos Ágeis buscam entregar valor ao cliente de maneira rápida, por meio da disponibilização de versões cada vez mais completas do *software*, geradas a partir de iterações curtas, adaptando a solução às necessidades, a cada momento.

A construção do *software* de modo evolutivo e adaptativo proposto pelos Métodos Ágeis encaixa-se perfeitamente na condução de projetos acadêmicos, uma vez que, os requisitos do *software* são definidos ao longo das pesquisas realizadas. Corroborando, propõem equipes pequenas e estabelecem poucos artefatos, prezando pela simplicidade, facilitando o aprendizado, itens importantes, quando indivíduos não têm experiência na condução de projetos.

A seguir os métodos *Scrum* e *Extreme Programming* são descritos de modo a permitir uma melhor compreensão do relato dos trabalhos acadêmicos realizados, apresentados na próxima seção.

2.1 Scrum

Como dito anteriormente, o *Scrum* corresponde a um *framework*, onde se define uma série de passos a percorrer, independentemente das técnicas utilizadas na construção do *software*. O foco está na disponibilização, ao final de cada iteração, das funcionalidades priorizadas.

Segundo Schawber (2004), o *Scrum* trata problemas complexos, controlando o processo por meio de três fundamentos: visibilidade, inspeção e adaptação. A visibilidade é dada pela transparência e veracidade dos aspectos do processo que afetam o resultado. A inspeção avalia tais aspectos, periodicamente, para a identificação de variações. Adaptações são realizadas, caso estas variações estejam fora dos limites aceitáveis.

Basicamente, está sendo usado o Guia do *Scrum*, elaborado por Schawber e Sutherland

(2010), para explicar o *framework*, por se tratar de um documento oficial e atualizado. Assim, são apresentados os papéis, eventos e artefatos estabelecidos pelo processo.

A equipe *Scrum* é formada por três papéis: *ScrumMaster*, responsável por garantir que o processo é completamente compreendido e seguido; *Product Owner*, responsável por maximizar o valor do trabalho executado, e *Team*, que executa o trabalho necessário.

Detalhando um pouco mais, o *ScrumMaster* deve remover impedimentos, ou seja, deve garantir que o time seja produtivo, buscando a cooperação entre os papéis e funções, evitando interferências externas e ajudando o time a auto organizar-se, essencialmente seria um líder que serve ao time, ao invés de conduzi-lo. Por outro lado, o *Product Owner* é o indivíduo que deve representar os interesses de todos os usuários, por meio da definição e priorização do que deve ser realizado no projeto.

Enfim, o time, cujos membros devem ter todas as habilidades necessárias para gerar cada incremento do *software*, ressaltando que não devem existir definições de papéis por especialidade dentro do mesmo, isto é, todos são responsáveis por todas as atividades. Duas outras exigências para o time são a auto-organização e o auto-gerenciamento, importantes para a obtenção do comprometimento no cumprimento dos objetivos traçados.

Os seguintes eventos são definidos pelo *framework Scrum*: Reunião de Planejamento da *Release*, *Sprint*, Reunião de Planejamento da *Sprint*, Reunião Diária *Scrum*, Reunião de Revisão da *Sprint* e a Reunião de Retrospectiva da *Sprint*.

A Reunião de Planejamento da *Release* tem como alvo transformar uma visão ou ideia em um produto, que venha atender ou exceder tanto a satisfação do cliente quanto o retorno de investimento. Por conseguinte, a reunião estabelece o objetivo da *release* a ser entregue, os riscos críticos envolvidos, a data de entrega provável, custos, se nada mudar, e de modo não detalhado, as características e funcionalidades que a mesma deve conter.

Uma *Sprint* corresponde a uma iteração, com a duração de um mês ou menos, durante a qual ocorre todo o esforço de desenvolvimento, produzindo um incremento do produto que seja potencialmente utilizável pelos usuários. É o ponto central do *framework*, todas as outras reuniões acontecem dentro de uma *Sprint*.

Como artefatos, são definidos o *Backlog* do Produto, que representa a lista priorizada de tudo que precisa ser feito para o produto, funcionalidades, características, correções de erros, dentre outros; o *Backlog* da *Sprint*, que corresponde ao objetivo da *Sprint*, ou seja, ao desdobramento em tarefas dos itens selecionados do *Backlog* do Produto e os Gráficos de *Burndown*, que ilustram quanto de trabalho ainda resta a ser executado, tanto para a *Release*, quanto para *Sprint*, denominados de *Burndown* da *Release* e *Burndown* da *Sprint*, respectivamente.

O *Backlog* do Produto é geralmente representado no formato de uma tabela, com as informações de descrição, prioridade e estimativa, para cada item. Sendo o *Product Owner* responsável pelo seu conteúdo, disponibilização e priorização. Salienta-se, que o artefato nunca está completo, à medida que o produto evolui, o seu *Backlog* também evolui. Quanto ao *Backlog* da *Sprint*, tem-se usado preferencialmente Quadros de Tarefas (*Taskboards*) para representá-lo, dada a visibilidade permitida. Durante uma *Sprint*, somente o time pode atualizar este artefato, adicionando tarefas, removendo tarefas, atualizando o tempo restante para uma tarefa, etc.

A Figura 1 resume o fluxo do processo definido pelo *Scrum*. A partir da Reunião de Planejamento da *Release*, o *Backlog* do Produto é gerado e priorizado pelo *Product Owner*. Na Reunião de Planejamento da *Sprint*, considerando-se os itens priorizados, o time define o objetivo da *Sprint*, gerando o seu *Backlog*. Ao longo da iteração, o time realiza as Reuniões Diárias. Estas reuniões devem ser curtas e têm a intenção de analisar o progresso com relação ao alvo traçado para a *Sprint*, através do relato por todos os membros do time, do que foi realizado no dia anterior, o que pretende-se fazer no dia em questão e quais foram os obstáculos que surgiram, para que assim,

seja possível realizar adaptações, quando for o caso. Todos são convidados, porém somente o time, o *Product Owner* e *ScrumMaster* podem falar.

Ao final da *Sprint*, na Reunião de Revisão, as funcionalidades incrementadas ao produto são apresentadas a todos os envolvidos no projeto, sendo que desta vez, qualquer pessoa pode expor seus comentários. O resultado desta reunião pode gerar atualizações para o *Backlog* do Produto. Enfim, tem-se a Reunião de Retrospectiva, onde se faz uma reflexão sobre o uso do processo e suas práticas, a fim de melhorá-lo.

2.2 Extreme Programming

A primeira vista, com uma visão bastante superficial, imagina-se que na programação extrema só há codificação, nada mais é necessário, ou seja, basta programar uma solução para o problema. Todavia, aprofundando-se o estudo, verifica-se que representa um conjunto de práticas que se aplicadas, principalmente em conjunto, garantem a qualidade dos produtos gerados.

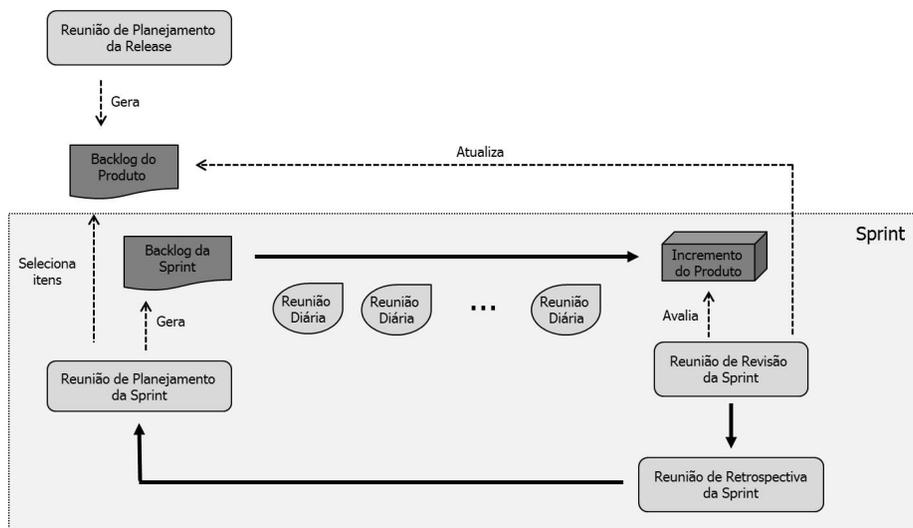


Figura 1: Processo Scrum.

Beck e Andres (2004) distinguem o XP de outras metodologias, em virtude de:

- estabelecer ciclos curtos de desenvolvimento, que resultam em um *feedback* mais cedo, concreto e contínuo;
- possuir uma abordagem de planejamento incremental;
- possibilitar mudanças de acordo com a necessidade do negócio;
- apostar nos testes automatizados escritos seja para identificar defeitos mais cedo, seja para monitorar o progresso do desenvolvimento;
- confiar na comunicação oral, nos testes e no próprio código fonte como documentação do objetivo e estrutura do projeto;
- determinar que a evolução do *design* do projeto deve ser realizada quando necessário ao longo do projeto;
- engajar ativamente os indivíduos por meio da colaboração entre seus participantes;
- basear-se em práticas que funcionam tanto no escopo dos membros do time, quanto no escopo dos interesses do projeto.

Tais práticas, segundo Beck e Andres (2004), são divididas em práticas primárias, úteis em qualquer situação, traduzem-se em bons resultados logo que aplicadas; e práticas corolárias, que

na inexistência das primárias, são difíceis de serem implantadas. As Tabelas 1 e 2 resumem as práticas definidas pelo XP, apresentando sua denominação e uma breve descrição.

Tabela 1: Resumo das Práticas Primárias.

| Prática | Descrição |
|--------------------------------|---|
| Sentar-se Junto | A equipe procura trabalhar junto, em uma mesma sala, a fim de facilitar a comunicação. |
| Equipe Completa | A equipe deve ser composta de pessoas com todas as habilidades e perspectivas necessárias para o sucesso do projeto. É importante que estes profissionais criem o senso de time. |
| Ambiente informativo | O ambiente de trabalho deve dar uma ideia geral sobre o projeto de modo rápido para qualquer observador. |
| Trabalho energizado | Cada membro da equipe deve trabalhar apenas a quantidade de horas suportadas, isto é, nas quais é produtivo. |
| Programação em Par | Os programas devem ser desenvolvidos em dupla, utilizando apenas um computador. |
| Estórias | As funcionalidades da aplicação devem ser descritas por meio de estórias (unidades de funcionalidades descritas em um nível de abstração perfeitamente compreensível pelos usuários e estimável pela equipe). |
| Ciclo Semanal | O trabalho será planejado e executado a cada semana, por meio de reuniões no início da semana. |
| Ciclo trimestral | O planejamento geral do projeto XP é dividido em trimestres. Esse planejamento consiste em estabelecer com o cliente o tema ou temas que serão implementados. |
| Folga | No planejamento devem ser incluídas algumas tarefas não muito importantes, que podem ser descartadas no caso de ocorrer algum atraso. |
| <i>Build</i> de dez minutos | O <i>build</i> do sistema e todos os testes automatizados devem ser executados em dez minutos, como forma de assegurar a sua execução constantemente. |
| Integração Contínua | O trabalho deve ser integrado diversas vezes ao dia, assegurando através de teste principalmente, que a base de código permaneça consistente ao final de cada integração. |
| <i>Design</i> incremental | O <i>design</i> da aplicação deve ser simples e feito a fim de atender a necessidade de cada dia. |
| Codificação Orientada a Testes | Os testes devem ser escritos antes do próprio código do <i>software</i> , servindo de base para a criação deste. |

Tabela 2: Resumo das Práticas Corolárias.

| Prática | Descrição |
|-------------------------------|--|
| Envolvimento real do cliente | Indivíduos, cujas necessidades o projeto deverá atender, devem fazer parte da equipe, sendo envolvidos durante todo o processo de desenvolvimento. |
| Distribuição Incremental | A substituição de um sistema legado deve ocorrer de modo gradual desde o início do projeto. |
| Continuidade da Equipe | Uma equipe eficaz deve ser mantida, contudo, uma razoável rotatividade também é importante para a troca de conhecimento e experiências. |
| Encolher equipe | Quando uma equipe aumenta a sua capacidade de produção é interessante que a carga de trabalho mantenha-se constante, para isso é necessário reduzir o tamanho da equipe. Quando esta estiver com poucas pessoas, junte com outra equipe que também seja pequena. |
| Análise da Raiz do Problema | Quando ocorrer um defeito, tanto o defeito quanto a causa do mesmo devem ser eliminados. |
| Código compartilhado | A propriedade do código é coletiva e todos são igualmente responsáveis pela manutenção de todas as partes |
| Código e Testes | Os únicos artefatos permanentes são o código e os testes (que precisam ser mantidos) |
| Base de código única | Apenas uma versão do código deve existir, independentemente do número de clientes. Ramificações devem ser temporárias. Diferentes regras devem ser tratadas por arquivos de configuração preferencialmente. |
| Distribuição Diária | Novas implementações do sistema devem ser distribuídas diariamente, evitando a diferença entre o código desenvolvido pela equipe e o disponível aos usuários. |
| Contrato de Escopo Negociável | Os contratos devem fixar tempo, custo e qualidade, mas não o escopo. Os riscos são reduzidos em virtude da assinatura de uma sequência de pequenos contratos, ao invés de um contrato extenso e único. |
| Pagamento por uso | Cobrar por cada vez que o sistema é utilizado ao invés de ser pela disponibilização da <i>release</i> . |

3 Trabalhos Realizados

Esta seção mostra a síntese de seis trabalhos de conclusão de curso realizados no DCOMP, sob a orientação da Profa. Debora Maria Coelho Nascimento, nos quais de alguma forma tentou-se a utilização dos métodos ágeis brevemente descritos na seção anterior.

O propósito é apresentar o objetivo de cada trabalho, como o método ágil foi aplicado, os resultados do trabalho em si e por fim, a análise da aplicação do método ágil na execução do trabalho acadêmico.

3.1 Gestão de Conteúdo como Apoio à Produção Jornalística: a Construção do Sistema Unificado de Reportagem SURF

O primeiro trabalho, desenvolvido por Correia, Soares e Silva (2008), trata-se de um projeto interdisciplinar, envolvendo duas áreas de conhecimento, Jornalismo e Computação. O objetivo foi a construção de uma aplicação, denominada posteriormente de SURF, que fornecesse suporte à produção jornalística, nas fases de apuração e produção. Os conceitos da produção jornalística, ciberjornalismo, fluxo de trabalho, trabalho colaborativo e gestão de conteúdo foram abordados na construção da ferramenta.

Durante a realização do trabalho foi analisada a adoção do *Extreme Programming*, contudo, a impossibilidade de aplicação de algumas práticas importantes, como por exemplo “Sentar-se junto”, “Programação em pares”, “Desenvolvimento Dirigido por Testes”, entre outras, fez com que a equipe descartasse o seu uso. As duas primeiras práticas não puderam ser aplicadas, em virtude dos integrantes da equipe estarem envolvidos em outras atividades diversas, de modo que não puderam estabelecer um horário e local comuns para o desenvolvimento do projeto. Enquanto que, a terceira prática citada não pode ser aplicada, uma vez que, para ser viável, seria necessária a utilização de ferramentas que possibilitassem a automatização dos testes.

O *Scrum* foi empregado para a condução do projeto. Desse modo, foram realizadas nove *Sprints* durante os dois meses do projeto. Devido a esta restrição de prazo, inicialmente as *Sprints* tiveram a duração de dez dias, mas a partir do amadurecimento da equipe, quanto ao uso da tecnologia, este tempo foi reduzido para uma semana.

A equipe foi formada pelos autores do trabalho, sendo que o estudante de jornalismo assumiu o papel de *Product Owner*, enquanto que, o time foi composto pelos alunos de computação. O papel de *ScrumMaster* não foi atribuído a nenhum indivíduo em específico, ficando suas tarefas sob responsabilidade do próprio time.

Durante as *Sprints* foram realizados os eventos definidos pelo *Scrum*: a Reunião de Planejamento da *Sprint*, as Reuniões Diárias e a Reunião de Revisão. Durante a Reunião de Planejamento da *Sprint*, ocorria a atualização do *Backlog* do Produto, pois ao final de cada iteração surgiam novos requisitos, a definição do que seria implementado durante a *Sprint* em questão e a estimativa de quanto tempo seria necessário para executar cada tarefa, gerando assim o *Backlog* da *Sprint*.

As Reuniões Diárias eram realizadas por telefone, já que, não havia um local específico para o desenvolvimento do projeto, o *Product Owner* não participava e o time era composto por apenas duas pessoas. Enfim, na Reunião de Revisão, a nova versão da aplicação era apresentada ao *Product Owner*. A Reunião de Revisão da *Sprint* anterior e a Reunião Planejamento da próxima *Sprint* ocorriam uma após a outra, a fim de otimizar o uso do tempo, uma vez que, os participantes das reuniões eram os mesmos. Como não houve um *ScrumMaster* definido, a reflexão sobre o andamento do processo de desenvolvimento foi realizada durante as Reuniões Diárias, desse modo não ocorreram as Reuniões de Retrospectiva.

Para o registro do *Backlog* do Produto foi definida uma planilha, a qual ficou armazenada na *web* utilizando-se da ferramenta *googledocs* (disponível em: <http://docs.google.com/>), de maneira que todos os participantes poderiam visualiza-la ou mesmo altera-la, caso fosse necessário.

Ao final do prazo para a realização do trabalho de conclusão de curso (TCC), as principais funcionalidades do SURF estavam implementadas, contudo o *Backlog* do Produto não foi totalmente executado. Além do tempo exíguo para a realização das atividades, tem-se como justificativa, as solicitações de modificação por parte do *Product Owner* praticamente ao final da cada *Sprint*, pois ao ver o funcionamento da aplicação, tinha uma nova ideia de como realizar a tarefa já implementada, de outra forma.

Analisando a aplicação do *Scrum*, visualiza-se que foi empregado com sucesso, principalmente por servir de guia na execução das atividades, já que os estudantes de computação não tinham

tido até então nenhuma experiência real no planejamento e execução de projeto de *software* que atendesse as necessidades de um cliente específico.

A falta de definição de um *ScrumMaster*, neste caso, não prejudicou o projeto devido essencialmente à motivação e comprometimento da equipe, tanto com o desenvolvimento do produto, quanto com o uso do processo, bem como, devido ao excelente relacionamento entre seus membros. Todavia, o SURF não foi utilizado em produção. Entende-se que a razão foi a não participação dos demais usuários nas Reuniões de Revisão, de modo que a aplicação atendeu somente a visão individual do *Product Owner*.

3.2 Utilização de Scrum e XP no Desenvolvimento de uma Aplicação Web - AFERO

Este trabalho foi gerado a partir da conclusão por parte de Poletti e Nascimento (2008) do curso de especialização em Engenharia de *Software*. Como seus autores tinham experiência nos modelos tradicionais de desenvolvimento de *software*, nas empresas em que trabalhavam, buscou-se por meio deste projeto, que os mesmos experimentassem processos alternativos, a partir do uso dos métodos ágeis *Scrum* e XP.

Portanto, o trabalho consistiu no desenvolvimento de uma aplicação *web*, a AFERO, aplicando os métodos mencionados. A AFERO tinha como objetivo atender uma empresa de pequeno porte do ramo alimentício, que fabrica e vende doces e salgados aos mais diversos tipos de clientes, na sociedade sergipana.

Foram necessárias quatro *Sprints*, com a duração de quatorze dias corridos cada uma, para a construção da aplicação. Os papéis *Scrum* foram assumidos da seguinte forma: o dono da empresa foi o *Product Owner*, o time foi composto pelos autores do trabalho, sendo que desta vez, o papel de *ScrumMaster* foi assumido pela orientadora do trabalho.

Do mesmo modo que no trabalho anterior, durante as *Sprints* foram realizadas as reuniões de Planejamento da *Sprint*, as Diárias e de Revisão. Neste caso, existindo o *ScrumMaster*, ocorreram ainda as reuniões de Retrospectiva, ao final de cada *Sprint*. Contudo, dado o caráter comercial da aplicação, o *ScrumMaster* não participou das reuniões de Planejamento e Revisão.

Quanto às reuniões Diárias, também precisaram ser adaptadas, de modo que, aconteciam virtualmente, por meio da ferramenta Skype, pois os membros do time trabalhavam em empresas diferentes, sendo que, nem o *ScrumMaster*, nem o *Product Owner* participavam, por estarem envolvidos em outras atividades. Todavia, eventualmente ocorriam encontros entre o time e o *ScrumMaster* a fim de resolver conflitos e analisar o andamento do projeto.

Foram utilizados como artefatos o *Backlog* do Produto, o *Backlog* de cada *Sprint*, o Gráfico de *Burndown* e o *Taskboard*. Os dois primeiros foram construídos usando planilhas eletrônicas, sendo o *Burndown* gerado automaticamente a partir das mesmas. O *Taskboard* foi montado na empresa onde um dos autores trabalhava, que permitiu inclusive que as reuniões com o *Product Owner* acontecessem em alguma de suas salas.

A partir das descrições de Martins (2007), foram selecionadas as práticas XP a serem aplicadas, são elas: Estórias, Pequenos *Releases*, Projeto Técnico Incremental, Programação em Par, Código Compartilhado, Integração Contínua, Padronização de Codificação, Refatoração e Propriedade Coletiva. Convém ressaltar, que a prática Programação em Par foi aplicada somente nos casos de maior complexidade e quando era necessário transmitir conhecimento.

Com o término do trabalho, obteve-se um resultado plenamente convincente, uma vez que, a AFERO foi finalizada no prazo previsto e em conformidade às necessidades do cliente; os desenvolvedores ficaram satisfeitos com o processo de desenvolvimento aplicado, visualizando um maior compartilhamento de conhecimento, integração, motivação, comprometimento e

produtividade da equipe. Os autores destacaram que ao longo do processo, houve o ganho de experiência, de modo a melhor estimar as atividades a serem realizadas em cada *Sprint*; que a principal dificuldade foi implantar a prática das Reuniões Diárias, dada a inexistência de local e horário para o desenvolvimento comum e por fim, que é essencial uma maior participação do *Product Owner*, durante o andamento da *Sprint*, com o intuito de possibilitar o esclarecimento de dúvidas.

É importante salientar que este trabalho não foi puramente acadêmico, já que seus autores possuíam a demanda do serviço aliada a exigência de elaboração do TCC para a conclusão do curso. Entretanto, acredita-se que cada experiência gera aprendizado, por isso a análise da aplicação dos Métodos Ágeis também foi realizada.

Como características deste trabalho, diferentemente do trabalho anterior, tem-se que seus autores possuíam experiência profissional, trabalhando no mercado há alguns anos, e que os mesmos utilizaram na implementação do projeto uma tecnologia de seu conhecimento. Com isso, reduziram-se tanto riscos gerenciais, os membros já participaram ou foram responsáveis por outros projetos anteriormente; quanto riscos técnicos, o domínio da tecnologia evita o surgimento de grandes surpresas no seu funcionamento.

Na implementação do processo identificou-se um problema, decorrente da forma escolhida para gerar e manter os artefatos propostos pelo *Scrum*, por meio de planilhas eletrônicas que não eram editadas colaborativamente ou pelo *Taskboard* existente na empresa de um dos membros da equipe. Com isso, apenas um membro da equipe ficou responsável por mantê-los, quase como um gerente de projetos, papel não existente nos Métodos Ágeis, além da diminuição da visibilidade do andamento do processo.

3.3 Construção do Portal Departamental: Definição e Validação da Arquitetura com a Implementação da Aplicação “Avaliação de Curso”

No trabalho desenvolvido por Santana, Santos e Osses (2009) houve a tentativa de utilização do *Scrum*, contudo ao longo do desenvolvimento, o processo foi abandonado. A intenção de descrevê-lo neste capítulo, é servir de reflexão sobre as possíveis causas.

O objetivo do trabalho era definir e implementar uma arquitetura de *software* que servisse de suporte para a construção de um Portal Departamental, isto é, permitisse que a partir da utilização desta arquitetura, várias aplicações fossem construídas de modo a atender corporativamente aos departamentos acadêmicos e administrativos da UFS. A arquitetura de referência seria validada pela implementação de uma dessas aplicações, denominada de Avaliação de Curso.

Uma das restrições do projeto foi que somente fossem utilizadas ferramentas *free*, ou seja, que não precisassem ser licenciadas. Deste modo, estabeleceu-se que a arquitetura fosse implementada sob a plataforma Java *Enterprise Edition*, gerando a necessidade de estudo da tecnologia envolvida, por parte dos autores do projeto.

A decisão de suspender o uso do *Scrum* foi consequência do não cumprimento dos *Backlogs* das *Sprints*, ou seja, ao final das *Sprints* as metas traçadas não eram atendidas. As justificativas por parte do time foram a dificuldade de configuração da tecnologia empregada, isto é, identificar e instalar as bibliotecas ou *plug-ins* adequados; a instabilidade da tecnologia, quando do seu uso; a falta de domínio da linguagem Java e APIs (*Application Programming Interfaces*) necessárias; a falta de experiência com a metodologia; a ausência de indicação no *Scrum* de como atuar, quando da adoção de novas tecnologias; a complexidade das regras de negócio da aplicação, o tempo disponível por parte dos membros da equipe e enfim, a dificuldade de auto-organização do time.

Entende-se que todas essas considerações são extremamente válidas, contudo acredita-se que

também faltou um maior comprometimento ou priorização do projeto pelo lado do time, uma vez que, no trabalho descrito na seção 3.1, muitos desses fatos também ocorreram, principalmente, com relação à adoção de novas tecnologias.

O problema da dificuldade de auto-organização da equipe foi uma questão inexistente nos trabalhos anteriores e que realmente comprometeu o andamento do projeto, resultando inclusive em uma aplicação não totalmente integrada. Todavia, acredita-se que este problema ampliou-se em virtude da não utilização de ferramentas que permitissem a execução e acompanhamento do trabalho de modo distribuído.

3.4 Aplicação de Metodologias Ágeis no Desenvolvimento Distribuído de Software

Atualmente, com a evolução dos meios de comunicação e a necessidade de atender a demanda de *software*, muitos projetos podem ser desenvolvidos de modo distribuído, ou seja, com os membros da equipe trabalhando em locais e horários diversos. Entretanto, uma das principais características defendidas pelos Métodos Ágeis é a comunicação direta, possibilitada por desenvolvedores e usuários trabalhando no mesmo local. Por conseguinte, Martins e Neri (2010) definiram uma proposta para a integração do Desenvolvimento Distribuído de *Software* (DDS) com práticas propostas pelos Métodos Ágeis, *Scrum* e *Extreme Programming*, em específico.

Inicialmente realizaram uma análise do impacto do DDS sobre as práticas *Scrum* e XP. Dentre os eventos propostos pelo *Scrum*, estão as reuniões, essas seriam afetadas dada a distância existente entre os envolvidos no desenvolvimento. Quanto ao *Extreme Programming*, as práticas afetadas seriam: Sentar-se Junto, Equipe Completa, Ambiente Informativo, Programação em Par, Integração Contínua, Codificação Orientada a Testes, Envolvimento real do cliente, Distribuição Incremental, Código compartilhado, Código e Testes, Base de código única e Distribuição Diária.

Após a análise, elaboraram uma proposta de integração, sem, contudo, fixar quais práticas deveriam ser aplicadas, uma vez que, segundo os autores, estas dependem de projeto para projeto. No entanto, duas atividades foram definidas como essenciais para a integração dos Métodos Ágeis ao DDS, seriam a execução de reuniões e a realização de Gerência de Configuração ao longo de todo o desenvolvimento.

Com relação às reuniões, seriam realizadas as definidas pelo *Scrum*, sendo que para minimizar perdas na comunicação, propuseram o emprego de ferramentas que permitissem a comunicação por voz e vídeo, além da troca de informações em um ambiente compartilhado.

A Gerência de Configuração deveria ser executada, com o uso de uma ferramenta apropriada, para o código e todos os outros artefatos criados, que não fossem supridos por outra aplicação que permitisse a edição de modo colaborativo.

O trabalho sugeriu, dentre outras ferramentas, a utilização do correio eletrônico, para a comunicação assíncrona; *Skype*, para a comunicação síncrona; *Subversion* também conhecido como SVN, para a gerência de configuração e o *FireScrum*, aplicação *web* que permite a simulação do processo *Scrum*, a partir da edição colaborativa de seus artefatos: *Backlog* do Produto, *Backlog* da *Sprint*, *Taskboard*, *Burndown* da *Release* e da *Sprint*.

A contribuição deste trabalho para a análise de aplicação de Métodos Ágeis em trabalhos acadêmicos foi a identificação do *FireScrum*, como ferramenta para dar suporte aos artefatos *Scrum* e a importância da definição de uma ferramenta de Gerência de Configuração, para projetos desenvolvidos por equipes distribuídas, caso comum em projetos acadêmicos. Tais ferramentas, com certeza, teriam ajudado no trabalho descrito na seção 3.3.

3.5 Sistema de Gestão da Produção Jornalística

O trabalho desenvolvido por Santos, Sobrinho e Bernardo (2010), similarmente ao trabalho apresentado na seção 3.1, também se tratava de um projeto interdisciplinar, só que desta vez, além de envolver Computação e Jornalismo, também envolveu Engenharia de Produção. O projeto tinha como objetivo construir o *software* denominado de Sistema de Gestão de Produção Jornalística, que com a implementação da Gestão de Processo e Gestão de Conhecimento, auxiliasse a produção jornalística.

A aplicação seria fruto da pesquisa intitulada “Matriz de Cobertura e Programa de Cobertura Jornalística: pesquisa aplicada para o desenvolvimento de ferramentas de gestão da produção jornalística”, financiada pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) e coordenada pelo prof. Josenildo Luiz Guerra. O propósito da pesquisa é tentar sistematizar o processo de produção jornalística, de maneira que possa ser monitorado, otimizado, por conseguinte, melhores resultados sejam obtidos. Cabe salientar, que o projeto ainda se encontra em andamento.

Como o desenvolvimento do *software* fazia parte do projeto de pesquisa, foi dividido em duas fases. Durante a primeira fase, o *Scrum* foi aplicado para o planejamento e acompanhamento das atividades de estudo, identificação de conceitos, modelagem de processos, análise de tecnologias, configuração de equipamentos, dentre outras. Nesta fase os *Sprints* tiveram a duração de duas semanas, não aconteciam as reuniões diárias, nem as de retrospectiva, as reuniões de revisão serviam para o esclarecimento de dúvidas e modelagem conceitual a partir da visão integrada: Jornalismo, Engenharia de Produção e Computação. As reuniões de Planejamento ocorriam logo após as reuniões de Revisão. Esta fase finalizou com a decisão do time em utilizar a plataforma de desenvolvimento que seus componentes possuíam mais experiência, em razão de não terem conseguido suporte técnico para empregar a tecnologia específica para o tema do trabalho.

Na segunda fase, ocorreu o desenvolvimento das funcionalidades propriamente ditas. Os *Sprints* passaram a ter a duração de uma semana, ocasionalmente, duas semanas, em virtude da necessidade de aumentar a interação com o *Product Owner*. Durante as reuniões de Revisão as funcionalidades eram validadas e modificações eram sugeridas, sendo que usualmente concentravam-se na navegabilidade e disposição dos elementos na interface. O time, formado pelos autores do trabalho descrito, realizava as reuniões diárias às vezes virtualmente, às vezes presencialmente. Os encontros virtuais geralmente ocorriam por meio de videoconferência, complementados por telefone ou envio de mensagens eletrônicas. As reuniões presenciais aconteciam no laboratório de pesquisa do DComp. Frequentemente eram seguidas por sessões de trabalho conjunto, onde dúvidas e problemas técnicos eram resolvidos em equipe. Quando necessário, tanto o *ScrumMaster* quanto o *Product Owner* eram acionados via correio eletrônico, sendo que estes retornavam em poucas horas. Do ponto de vista do time, este procedimento não prejudicou o progresso das *Sprints*. As reuniões de Planejamento continuaram sucedendo as reuniões de Revisão, enquanto que, as reuniões de Retrospectiva dar-se-iam ao final do mesmo dia, somente com a participação do *ScrumMaster* e time.

O papel de *Product Owner* foi assumido pelo professor de jornalismo responsável pela pesquisa, enquanto que a professora orientadora do desenvolvimento do SGPJ ficou com o papel de *ScrumMaster*.

As ferramentas SVN e *FireScrum* foram usadas para a gerência de configuração e o planejamento e acompanhamento do projeto, respectivamente.

Como dito anteriormente, o projeto como um todo ainda está em andamento, todavia, os resultados apresentados, até a escrita do TCC associado, foram satisfatórios.

Avaliando a utilização do *Scrum*, relata-se que houve uma certa resistência por parte do time no preenchimento dos artefatos, *Backlog* do Produto e *Backlog* da *Sprint*, de modo que os objetivos

da iteração não ficavam totalmente claros, dificultando assim a análise do seu progresso; as tarefas eram assumidas por cada membro, mas não havia a estimativa do tempo exigido para a sua execução e enfim, muitas vezes esperavam que o *ScrumMaster* tomasse decisões, conduzindo e direcionando o andamento dos trabalhos. A questão do direcionamento justifica-se pela simples presença do *ScrumMaster* em todas as reuniões de Revisão e Planejamento, tal comparecimento pode ter inibido a atuação do time, pois o primeiro papel foi assumido por uma professora. Quanto às estimativas, acredita-se que faltou experiência, neste caso, o *ScrumMaster* poderia ter auxiliado, principalmente no início do projeto.

3.6 Aplicando Métodos Ágeis em Desenvolvimento Solo

Paralelamente ao trabalho anterior, Júnior (2010) analisou e propôs adaptações para o emprego de Métodos Ágeis, especificamente o *Scrum* e o *Extreme Programming*, em projetos onde apenas um desenvolvedor é envolvido, isto é, no Desenvolvimento Solo. Ressalta-se, que o Desenvolvimento Solo é muito comum em projetos acadêmicos, onde participam o orientador e o orientando, sendo que apenas este último é responsável pela execução das tarefas.

Na análise do *Scrum*, concluiu-se que o evento mais afetado seria a Reunião Diária, dada a ausência de outros membros no time. A fim de permitir a inspeção e adaptação diárias, características essenciais ao *Scrum*, o autor propôs o uso de um Diário, onde seriam anotadas decisões tomadas, mudanças no *design* e codificação, dificuldades e soluções encontradas. Tais registros seriam apresentados ao *ScrumMaster* semanalmente. Lembrando que, no Desenvolvimento Solo os papéis de *ScrumMaster* e *Product Owner*, continuam existindo.

Examinando as práticas XP, identificaram-se as que possuíam aplicabilidade no Desenvolvimento Solo em conjunto com o *Scrum*, definindo-se, quando convinha, as mudanças necessárias para uma melhor aplicação das mesmas ao processo. Por conseguinte, as práticas possíveis de serem aplicadas seriam: Envolvimento Real do Cliente; Estória, Código e Testes, Codificação Dirigida a Testes, *Build* de Dez Minutos, *Design* Incremental, Trabalho Energizado, Folgas, Código Compartilhado, Ambiente Informativo, Integração Contínua, Continuidade da Equipe, Análise da Raiz do Problema, Distribuição Incremental, Pagamento por Uso e Contrato com Escopo Negociável.

Com a intenção de suprir, de certo modo, as deficiências causadas pela impossibilidade de aplicação das práticas XP, Programação em Par e Equipe Completa, o autor propôs o emprego da técnica de codificação Desenvolvimento Dirigido por Testes (DDT), definida por Beck (2002). Corroborando com seu emprego, a DDT está completamente alinhada às práticas: Codificação Orientada a Testes, *Design* incremental e Integração Contínua.

Para validar a proposta foi escolhido, como estudo de caso, o desenvolvimento de um sistema para apoio a um projeto de pesquisa do Departamento de Biologia (DBI) da Universidade Federal de Sergipe (UFS), denominado *Serapilheira*. O autor do trabalho compôs o time, a professora de Biologia, Myrna F. Landim, responsável pela pesquisa assumiu o papel de *Product Owner*, enquanto que, a orientadora do TCC, correspondeu ao *ScrumMaster*.

A aplicação foi desenvolvida ao longo de quatro *sprints*, com a duração de quatorze dias cada uma. Os demais eventos da *Sprint* ocorreram de modo semelhante aos demais projetos, com exceção para a Reunião Diária, em razão da adaptação definida. Todavia, as reuniões semanais estabelecidas aconteceram apenas nas duas primeiras *sprints*, pois se percebeu, durante as reuniões de Retrospectiva, uma perda de produtividade devido ao tempo gasto com deslocamento e realização da reunião. Desse modo, buscou-se a otimização do processo, com o envio semanal, através do correio eletrônico, dos diários para o *ScrumMaster*, possibilitando que esse verificasse se seu preenchimento estava adequado.

Visualizando a aplicação das práticas XP, as selecionadas para o estudo de caso foram: Trabalho

Energizado, Folgas, Ambiente Informativo, Codificação Dirigida a Testes, Integração Contínua e o *Design Incremental*, Build de Dez Minutos, Envolvimento Real do Cliente, Análise da Raiz do Problema e Continuidade da Equipe.

O *FireScrum* foi empregado para a elaboração dos artefatos *Scrum*, concedendo visibilidade ao processo, mesmo com poucas pessoas envolvidas. Como o código estava sendo desenvolvido por uma única pessoa, não houve preocupação na adoção de uma ferramenta para Gerência de Configuração.

Com o término da implementação do Serapilheira, verificou-se que as dificuldades provenientes do desenvolvimento solo foram contornadas, através das adaptações às práticas ágeis sugeridas e que a qualidade do produto de certo modo foi obtida, uma vez que o *Product Owner* não identificou nenhum erro com relação às funcionalidades, os problemas levantados foram somente quanto à usabilidade das mesmas.

Entende-se que o resultado na aplicação das práticas ágeis foi totalmente convincente, pois houve motivação e comprometimento do aluno na sua aplicação. Destaca-se que nesse caso, havia também domínio do aluno com relação à tecnologia utilizada, reduzindo os riscos técnicos relacionados. Porém, do mesmo modo do primeiro trabalho apresentado, houve certa deficiência no envolvimento real do cliente.

4 Conclusão

Trabalhos acadêmicos possuem características que combinadas, aumentam os riscos do desenvolvimento de *software*, conforme descrito anteriormente. Com o objetivo de melhorar a qualidade dos projetos desenvolvidos, práticas ágeis definidas pelo *Scrum* e *Extreme Programming* estão sendo aplicadas. Este capítulo visou apresentar o resultado e evolução da aplicação destas práticas em alguns trabalhos desenvolvidos no DCOMP / UFS.

A aplicação do *Scrum* para o planejamento, condução e acompanhamento das atividades mostrou-se plenamente satisfatório. Porém, apesar do *framework* ser simples e com poucos artefatos, encontrou-se certa resistência, principalmente quando o objetivo do trabalho não estava diretamente ligado à aplicação do processo. Usualmente, os estudantes preferem dedicar-se a implementação propriamente dita da solução. Por conseguinte, é preciso conscientizar os participantes da importância da sua utilização, do comprometimento e da priorização.

A indisponibilidade de tempo, dado o envolvimento do time em atividades de superior ou mesma prioridade, dificultou a realização das reuniões diárias. Como solução, utilizaram-se os mecanismos de comunicação possíveis.

Com relação à atribuição dos papéis, identificou-se que é importante a figura do *ScrumMaster*, exigindo a aplicação do processo e atuando como mentor quando necessário. A correta identificação do *Product Owner* e a participação dos demais usuários durante as reuniões de Revisão são extremamente importantes para a obtenção de uma solução adequada. Contudo, cabe lembrar, que a participação do *Product Owner* não deve restringir-se a definição, esclarecimento de dúvidas e priorização dos itens do *Backlog* do Produto. Deve essencialmente acontecer a cada disponibilização de um novo incremento do *software*, ao final da *Sprint*, a fim de obter o resultado da iteração efetiva dos usuários com o *software*. Esta é a única forma de garantir que o projeto está indo na direção certa.

O *FireScrum* apresentou-se como uma ferramenta bastante interessante para dar suporte aos artefatos, por se tratar de uma aplicação *web*, que permite a visualização e edição dos documentos por todos os membros do projeto.

A aplicação do *Extreme Programming* é um pouco mais complexa, uma vez que requer a análise de cada uma das vinte e quatro práticas e definir quais podem ser aplicadas em cada projeto. Algumas são difíceis de serem adotadas, “Sentar-se Junto”, “Equipe Completa”; algumas requerem

experiência dos desenvolvedores, “*Design Incremental*”; outras exigem o uso de ferramentas, “*Codificação Orientada a Testes*”; outras vão de encontro a práticas *Scrum*, “*Distribuição Diária*”, “*Ciclo Trimestral*”; enfim, outras não tem sentido para trabalhos acadêmicos, “*Pagamento por uso*”. A aprendizagem com os trabalhos apresentados mostrou que nos poucos trabalhos onde foram empregadas os resultados foram bons, uma vez que as aplicações foram concluídas e em conformidade com as necessidades dos usuários.

Uma característica essencial dos Métodos Ágeis é a exigência da equipe ser auto gerenciada, para isso, são essenciais dentre outras atitudes, o comprometimento, a organização e a maturidade. A imaturidade, comum entre os estudantes, provém da falta de experiência na participação em projetos anteriores. Entretanto, como comprovado em alguns trabalhos, esta pode ser compensada pelo comprometimento e organização da equipe. O comprometimento depende da motivação. Para a organização, é essencial o uso de ferramentas, tanto para o acompanhamento do projeto, exemplificada pelo *FireScrum* – já citado, quanto para o gerenciamento de configuração, principalmente quando o time for composto por mais de uma pessoa.

A adoção de novas tecnologias traz grandes riscos a qualquer tipo de projeto. Em trabalhos acadêmicos, seja em razão de seu caráter inovador, seja em razão da falta de conhecimento dos alunos, o fato ocorre com bastante frequência. Por outro lado, não foram encontradas nos métodos ágeis empregados, quaisquer indicações de como proceder neste caso. Portanto, nos trabalhos mencionados, tentou-se fugir desse risco, mas nem sempre foi possível.

Como trabalhos futuros, pretende-se formalizar um *framework* de processo, englobando as práticas mais adequadas, que sirva de orientação para a aplicação em outros projetos, inclusive sob outras orientações; pesquisar outros trabalhos que tratem da adoção de novas tecnologias em projetos conduzidos de modo ágil; estudar e analisar práticas ágeis de estimativa; por fim, avaliar a adequação desta proposta a modelos de qualidade.

É importante destacar que seguindo a ideia de melhoria contínua, nenhum processo está completamente definido. Todo projeto é diferente, de modo que novas experiências são sempre válidas e lições são aprendidas, conseqüentemente, o processo pode ser melhorado.

É importante destacar que seguindo a ideia de melhoria contínua, nenhum processo está completamente definido. Todo projeto é diferente, de modo que novas experiências são sempre válidas e lições são aprendidas. A partir do conhecimento adquirido em outras áreas como arquitetura e testes de *software* e tentando alinhar teoria e prática, com foco na qualidade do produto e processo de *software*, pretende-se ainda conduzir pesquisas voltadas para Engenharia de *Software* Experimental, Visualização de *Software* e Linha de Produto.

Referências

- BECK, Kent. Test-driven development: by example. Boston: Addison - Wesley, 2002.
- BECK, Kent; ANDRES, Cynthia. Extreme Programming Explained: Embrace Change. 2. ed. Addison Wesley Professional, 2004.
- CORREIA, B. de A.; SOARES, G. A.; SILVA, W. A. Gestão De Conteúdo Como Apoio À Produção Jornalística: A Construção Do Sistema Unificado De Reportagem Surf. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) – Departamento de Computação. São Cristóvão: Universidade Federal de Sergipe, 2008.
- COSTA, G.R.; OLIVEIRA, S.D.; SOARES, A.C. Desenvolvimento de Projetos de *Software*, em um Ambiente Acadêmico, Utilizando Parcialmente o Extreme Programming – XP. In: Anais XIX INCITEL 2007. Disponível em: <http://ic.inatel.br/ic/component/docman/cat_view/4-documentos/5-incitel?limit=5&limitstart=0&order=hits&dir=DESC>. Acesso em: 10 de agosto de 2010.
- HIGHSMITH, Jim. Agile Project Management: Create Innovative Products. Boston: Addison-Wesley, 2004.
- JÚNIOR, A. S. da P. Aplicando Métodos Ágeis Em Desenvolvimento Solo. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) – Departamento de Computação. São Cristóvão:

- Universidade Federal de Sergipe, 2010.
- KNIBERG, Henrik. Scrum and XP from the Trenches. How we do Scrum. C4Media, Publisher of InfoQ.com. 2007. Disponível em: <<http://www.infoq.com/minibooks/scrum-xp-from-the-trenches>>. Acesso em: 9 de fevereiro de 2011.
- MANIFESTO para Desenvolvimento Ágil de *Software*. 2001. Disponível em: <<http://agilemanifesto.org/iso/ptbr/>>. Acesso em: 3 fevereiro de 2011.
- MARTINS, D. S.; NERI, L. E. C. Aplicação De Metodologias Ágeis No Desenvolvimento Distribuído De *Software*. Trabalho de Conclusão de Curso (Especialização Lato Sensu em Gestão de Projetos de Tecnologia da Informação) – Departamento de Computação. São Cristovão: Universidade Federal de Sergipe, 2010.
- MARTINS, J. C. C. Técnicas para Gerenciamento de Projetos de *Software*. 1ª Edição. Rio de Janeiro: Brasport, 2007.
- POLETTI, M. V. S.; NASCIMENTO, S. C. Utilização de SCRUM e XP no Desenvolvimento de uma Aplicação Web – AFERO. Trabalho de Conclusão de Curso (Especialização Lato Sensu em Engenharia de *Software*) – Departamento de Computação. São Cristovão: Universidade Federal de Sergipe, 2008.
- SANTANA, A. S.; SANTOS, M. A.; OSSES, P. E. F. Construção Do Portal Departamental: Definição E Validação Da Arquitetura Com A Implementação Da Aplicação “Avaliação De Curso”. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) – Departamento de Computação. São Cristovão: Universidade Federal de Sergipe, 2009.
- SANTOS, E. M.; SOBRINHO, L.P.S.; BERNARDO, L. de J. Sistema De Gestão Da Produção Jornalística. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) – Departamento de Computação. São Cristovão: Universidade Federal de Sergipe, 2010.
- SCHAWBER, Ken. Agile Project Management with Scrum. Redmond: Microsoft Press, 2004.
- SCHWABER, Ken; SUTHERLAND, Jeff. The Scrum Guide. 2010. Disponível em: <<http://www.scrum.org/scrumguides/>>. Acesso em 2 fevereiro de 2011.
- SOMMERVILLE, Ian. Engenharia de *Software*. Tradução Selma Shin Shimizu Melnikoff, Reginaldo Arakaki, Edilson de Andrade Barbosa; Revisão Técnica : Kechi Kirama.-- 8.ed de *Software Engineering*. São Paulo: Pearson Addison-Wesley, 2007.
- VRIENS, Christ. *Certifying for CMM Level 2 and ISO9001 with XP@Scrum*. In: Agile Development Conference, 2003, Salt Lake City, Utah. Disponível em: <<http://agile2003.agilealliance.org/files/R8Paper.pdf>>. Acesso em: 9 de fevereiro de 2011.

Computação e Ensino: “Ensino de Computação” e “Computação no Ensino”

Kênia Kodel Cox

Resumo. Neste capítulo são relatados trabalhos onde se observa a consolidação da computação e do ensino, sejam de “ensino de computação”, ou de “computação no ensino”, executadas no âmbito do Departamento de Computação, da Universidade Federal de Sergipe, (DComp/UFS). O “ensino de computação” é neste capítulo compreendido como iniciativas em que, para o alcance dos alvos da computação são usados ferramentas, técnicas e métodos de ensino; seja para a inclusão digital de jovens e adultos, ou até para formar cientistas ou engenheiros da computação visando o desenvolvimento da ciência. Já a denominação “computação no ensino” é dada à consolidação dos objetivos do ensino com o apoio dos recursos da computação; quando então são aplicados ferramentas, técnicas e métodos da computação para promover o alcance dos alvos almejados pelo ensino; seja para treinar, instruir e/ou construir competências. Como ações futuras intenta-se dar continuidade a projetos de consolidação da computação e do ensino, em especial, relativas ao ensino de programação, com aplicação de *design* instrucional e/ou a partir da aplicação de análises de registros semióticos.

Palavras-Chave: Computação. Ensino. Ensino de Computação. Computação no Ensino.

1 Introdução

A computação e o ensino são áreas autônomas de ação humana; entretanto, quando consolidadas, podem ter seus efeitos centuplicados.

Ao consolidar os objetivos da computação com o apoio dos recursos do ensino, obtém-se o que aqui se denomina “ensino de computação”. Neste caso, para o alcance dos alvos da computação são usados ferramentas, técnicas e métodos de ensino; seja para iniciativas de alfabetização tecnológica, bem como para formar cientistas ou engenheiros da computação com vistas à promoção do desenvolvimento científico.

Já a denominação “computação no ensino”, aqui é dada à consolidação dos objetivos do ensino com o apoio dos recursos da computação. Então, são aplicados ferramentas, técnicas e métodos da computação para promover o alcance dos alvos almejados pelo ensino; seja para treinar, instruir,

construir competências; através de construção de *software* ou de cursos à distância, páginas *web*, ou aplicação de listas de discussão e “bate-papo” para apoiar as atividades educacionais clássicas desenvolvidas em sala de aula.

Tanto no “ensino da computação” como na “computação no ensino” são inúmeras as iniciativas que podem ser desenvolvidas e, a partir destas promover o desenvolvimento científico social almejados pelas áreas envolvidas.

São múltiplos os “olhares” que podem nortear projetos e pesquisas acerca de ensino de computação: voltados para ambientes e ferramentas de apoio ao aprendizado, de *e-learning*, com estudos de caso, explorando aprendizagem colaborativa, aplicando o ensino por projeto, apoiados no protagonismo juvenil, sustentadas nas diversas concepções de aprendizagem – significativa, proposta por Ausubel; de adaptação e equilíbrio, de Piaget; mediada, Vygotsky; e/ou afetiva, Wallon.

Pereira *et al* (2010), por exemplo, no artigo intitulado “Ateliê de Objetos de Aprendizagem: Uma Abordagem para o Ensino de Computação em Cursos Técnicos”, apresentam estudo sobre quatro abordagens de ensino de computação: simplificação do conteúdo, projetos reais, aprendizagem colaborativa e modelos de apoio à colaboração, e sobre estas conclui:

“Todas estas estratégias, quando aplicadas nas pesquisas [...], surtiram resultado positivo quanto ao ensino de computação. No entanto, sua aplicação isolada não provê todo o benefício que seria possível ao usar as abordagens em conjunto[...]” (PEREIRA *et al*, 2010).

Este evidencia a diversidade de possibilidades de estudos de ensino de computação, os quais podem ser norteados pelos múltiplos supracitados “olhares” e/ou combinações destes, como no artigo em foco.

Ainda Pereira *et al* (2010), afirmam sobre objetos de aprendizagem (OA): “O nome Ateliê foi escolhido para evidenciar a criatividade e liberdade envolvidas no processo de desenvolvimento dos OA[...]”. Esta afirmação reflete com fidelidade a multiplicidade de projetos e pesquisas voltadas pela ensino de computação, onde criatividade e liberdade conduzem o processo de desenvolvimento.

Quanto às pesquisas de “computação no ensino”, estas podem voltar-se, por exemplo, para a exploração e/ou desenvolvimento de ferramentas e recursos informáticos: jogos, aplicações para *internet*, realidade aumentada, *software*, portais *web*; para a exploração de determinadas estratégias de ensino neste contexto; para um público específico: adultos, portadores de necessidades especiais, estrangeiros; bem como para observar os efeitos desta no desempenho discente.

Silva (2010), em artigo sobre desafios aos usos das TIC, esclarece que:

“Na busca pela aprendizagem efetiva, tecnologias são colocadas a serviço do fazer educativo. Porém, é preciso se ter clareza do que é importante do ponto de vista pedagógico na utilização de qualquer tecnologia no ambiente escolar. ...Neste sentido, a escola é pensada enquanto espaço de aprendizagem e exercício da cidadania onde o sujeito possa adquirir condições de atuar no manejo e produção de conhecimentos que lhes seja útil não somente para o seu próprio desenvolvimento, mas também para o desenvolvimento da coletividade.” (SILVA, 2010).

É neste contexto, de busca de “clareza do que é importante do ponto de vista pedagógico na utilização de qualquer tecnologia no ambiente escolar”, que surgem as pesquisas e projetos acerca da informática no ensino, os quais, assim como as iniciativas que se voltam para o ensino de computação, são inúmeros, podendo explorar múltiplos recursos computacionais e pedagógicos. Neste capítulo são relatadas iniciativas de consolidação da computação e do ensino executadas no âmbito do Departamento de Computação (DComp), da Universidade Federal de Sergipe (UFS).

2 Ensino de Computação

Segundo Delors (2003), em relatório da Comissão Internacional sobre Educação para o Século XXI, para a UNESCO, as ações educacionais hodiernas devem apoiar-se em quatro pilares: (1) aprender a conhecer – aprender a aprender para beneficiar-se das oportunidades oferecidas pela educação ao longo de toda a vida; (2) aprender a fazer – construir competências que tornem a pessoa apta a enfrentar as diversas situações e desafios, que permeiam seu entorno; (3) aprender a viver juntos – aprender a participar e cooperar com os outros em todas as atividades humanas; (4) aprender a ser – estar à altura de agir cada vez com maior capacidade de autonomia, de discernimento e de responsabilidade.

Contudo, a educação formal, inclusive de computação, ainda volta suas ações prioritariamente para a transmissão de informações, assim desprezando três dos pilares apresentados: aprender a fazer, a viver junto e a ser. O que impossibilita que o educando desenvolva, de forma plena, sua capacidade de interferir ativa e construtivamente em seu contexto escolar e social comunitário e, conseqüentemente, compromete o alcance das missões do ensino.

Assim sendo, é preciso haver iniciativas para preencher esta lacuna. A academia deve criar oportunidades para que o educando possa aprender, fazer, viver coletivamente, ser, despojar-se da postura de mero usufrutuário, de sujeito acrítico, para assumir seu papel ativo e empreendedor frente a sua própria formação, à sociedade e à ciência.

Neste contexto, diversas são as possibilidades de iniciativas, dentre as quais destacam-se as de ensino de computação, cujos alvos podem ser os mais diversos objetos de ensino da área: de orientação a objetos, de inteligência artificial, de redes de computadores, e outros.

2.1 De Orientação a Objetos (CORRÊA e COX, 2010)

Segundo Kölling (1999), a programação orientada a objetos (OO) vem sendo cada vez mais utilizada. Este crescimento é notado inclusive no mundo comercial em que as aplicações são desenvolvidas de forma distribuída e voltadas para a realidade *Web*; o que, por sua vez, requer um desenvolvimento fácil e ágil para atender a dinâmica própria do mercado, cada vez mais exigente. Para tanto as empresas ao redor do mundo têm tirado proveito dos efeitos da adoção do paradigma orientado a objetos.

Segundo Corrêa e Cox (2010), na academia esse crescimento também é notado, já que além de ter que acompanhar, esta deve principalmente, contribuir com os avanços na tecnologia de uma forma geral. As iniciativas acadêmicas têm o dever de preparar seus alunos para o mundo além da universidade. Hoje se pode dizer que em todos os cursos relacionados à Computação, de todas as instituições de ensino superior, o paradigma orientado a objetos é abordado.

Porém, no processo de ensino e aprendizagem do citado paradigma percebe-se uma significativa deficiência na assimilação do conteúdo abordado. Nota-se que a maior parte dos egressos dos cursos voltados para OO não é capaz de construir *software* utilizando os conceitos da orientação a objetos; estes codificam de forma estruturada em uma linguagem desenhada para o uso de objetos. Essa deficiência no aprendizado decorre de problemas metodológicos e, como afirma Kölling (1999), de ferramentas disponíveis para o ensino.

Observa-se que a grande parte das ferramentas de OO que surgem visam suprir necessidades do meio comercial, e, apesar de ricas em funcionalidades, não contemplam todas as necessidades do meio acadêmico.

Visando superar as supracitadas deficiências, Kölling (1999) propõe o ambiente e linguagem *Blue*. E inspirado nesta proposta, o aluno Daniel Regis dedicou seu trabalho de conclusão de curso ao objetivo de prover um ambiente para desenvolvimento OO que fornecesse, de uma forma integrada, ferramentas e facilidades para o ensino e o aprendizado do respectivo paradigma. A

linguagem usada foi Java devido a sua disponibilidade, pelo fato desta ser de código livre, e pelo grande número de empresas e pessoas que a utilizam.

Quando propôs o *Blue*, Kölling (1999) definiu requisitos que um ambiente deve apresentar para adequar-se à tarefa do ensino do paradigma OO: facilidade de uso, ferramentas integradas, suporte a reutilização de código, suporte a objetos, suporte a trabalho coletivo, disponibilidade e suporte ao aprendizado.

É fácil identificar cada um dos requisitos supracitados em várias ferramentas e ambientes de desenvolvimento de *software* disponíveis no mercado, mas não há nenhuma em que todos eles estejam presentes. Entretanto duas dessas ferramentas, Eclipse e o Netbeans, se destacam. Ambas são de código fonte aberto e possibilitam que seus usuários componham recursos adequando o ambiente computacional às suas necessidades. Isso é possível através de *plugins* desenvolvidos tanto por grandes corporações mundiais de tecnologia da informação quanto por desenvolvedores solitários.

No projeto de Daniel Regis foi criado um *plugin*, denominado *OOTeacher* (CORRÊA e COX, 2011), para adequar o Eclipse aos requisitos apontados por Kölling (1999) como indispensáveis a uma ferramenta de apoio ao ensino de OO. Com a aplicação deste, o estudante de OO pode ter melhor interação com os objetos em construção e/ou estudo. Na Figura 1 é apresentada a tela obtida a partir do uso do *OOTeacher* através da qual o estudante pode acompanhar os efeitos resultantes da chamada de um método em estudo, ou construção.

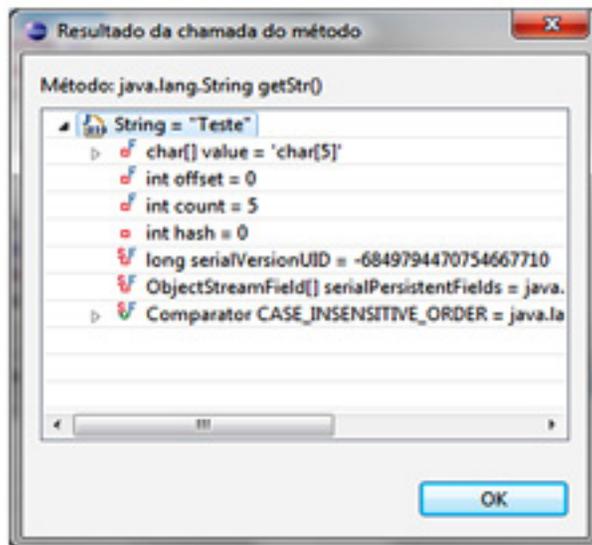


Figura 1: Interface do *OOTeacher*, de exibição dos efeitos da chamada de um método.

Fonte: Corrêa e Cox (2010).

No contexto do *OOTeacher*, ainda há espaço para continuidade. Por exemplo, a partir da observação dos efeitos deste em sala de aula, é possível ajustá-lo de forma a melhor atender ao processo de ensino; bem como identificar novas funcionalidades para o mesmo.

Já no ensino da Orientação à Objetos, o número de possibilidades de estudos é infinito: construção de curso à distância, desenvolvimento de material instrucional voltado para crianças e/ou adolescentes, acompanhamento da aplicação dos objetos de ensino em sala de aula. Sendo estes também válidos para o ensino de outros campos da computação, como o de Introdução à Programação.

2.2 De Introdução à Programação (PORTO e COX, 2011)

A programação de computadores é um conhecimento indispensável para alunos de cursos superiores de computação e de engenharia; nos quais muitas matérias envolvem direta ou indiretamente a construção de programas computacionais. Por isso, o ensino de programação é inserido nas disciplinas iniciais dos cursos profissionalizantes de computação, sejam de nível técnico ou superior.

Segundo Silva (2010):

“As técnicas de construção de algoritmos e programação constituem o principal núcleo de conhecimento no contexto das disciplinas que compõem os cursos de informática de nível superior. O entendimento conceitual e a aquisição das habilidades de utilização das técnicas básicas de resolução de problemas por métodos algorítmicos são essenciais para a compreensão da maioria dos tópicos abordados no ensino de computação.” (SILVA, 2010)

Na UFS, por exemplo, introdução à programação é objetivo da disciplina “Programação Imperativa” e esta compõe o elenco de disciplinas obrigatórias do primeiro período dos cursos de Ciência da Computação, Sistemas de Informação e Engenharia da Computação; bem como de Engenharia Elétrica, e outros.

Ou seja, a introdução à programação é ministrada aos egressos do ensino médio, na primeira etapa da educação superior. Porém, segundo Silva (2010), no ensino fundamental e médio, os alunos não são treinados para resolver problemas de maneira lógica ou algorítmica; o que é indispensável para o processo de programar computadores.

Ainda Silva (2010) afirma que as matérias exatas não são ensinadas de maneira que haja uma ligação destas com o mundo real e eventos do cotidiano; outro requisito da construção de programas computacionais.

Para construir um programa, o aluno deve elaborar o passo-a-passo de resolução de um problema real, a partir da aplicação de recursos computacionais, o que requer que haja uma compreensão da problemática, e a articulação desta às específicas soluções computacionais.

Assim, para aprender a propor soluções algorítmicas, o aluno precisa dominar conhecimentos complexos e articulados, o que não se consegue com facilidade, devido à citada dissociação entre escola e mundo real observada no ensino brasileiro, bem como a falta de estruturação lógica para resolução de problemas.

E ainda observa-se a propagação de deficiências no aprendizado durante todo curso profissionalizante, em especial nas matérias que dependem diretamente da programação.

O problema torna-se ainda maior quando os alunos dos cursos de computação com dificuldades em programação de computadores acha que é inapto para a área, e assim, acaba desistindo de continuar sua graduação ou, quando continua, tem um índice de reprovação muito alto, sobrecarregando, assim, as matérias iniciais dos cursos de computação.

Conforme Mota (2009), “O mínimo que se espera é que o estudante seja capaz de desenvolver programas para resolver problemas reais simples, mas apesar disso, são altos os níveis de reprovação em disciplinas de programação.”

Por ser um problema bem claro, essa dificuldade inicial e o processo de aprendizado das linguagens de programação tem se tornado temas que são alvos de muitas pesquisas na área de educação.

“A determinação das causas das dificuldades encontradas pelos alunos no processo de ensino/aprendizagem de linguagens de programação tem sido uma tarefa difícil e suscita toda ordem de investigações. Um número significativo de reprovações nestas disciplinas é uma ocorrência preocupante e bastante comum.” (SILVA, 2010).

Assim, visando validar propostas de solução para as dificuldades discentes relatadas, em

trabalho de conclusão de curso, o aluno Danilo Mont'Alegre desenvolveu um curso à distância de introdução à programação explorando recursos de apoio ao ensino (PORTO e COX, 2011), com o cuidado de garantir que este apresente as características apontadas como ideais pelos estudos acerca de ensino de programação.

A construção de um curso envolve diversas nuances: seleção das aulas, organização do conteúdo, projeto de ilustrações, elaboração de exercícios e formas de avaliação; o que requer esforço significativo e múltiplas escolhas podendo ser objeto de diversos estudos, com distintos objetivos e formatações. Dentre estes objetivos, é possível destacar o de motivar o aluno a protagonizar seu próprio processo educativo – o ensino com protagonismo juvenil.

2.3 Com Protagonismo Juvenil (COSTA NETO e COX, 2004)

O protagonismo juvenil estimula o educando a tomar para si os rumos de seu aprendizado em ações de monitoria, de liderança, em grupos de trabalho e estudo, em parcerias com professores, no auxílio a colegas em atividades, no cuidado dos espaços, nas pesquisas – é protagonista.

O Curso de Ciência da Computação da Universidade Federal de Sergipe (UFS), em 2003 e 2004, aposta no protagonismo juvenil e propõe projetos como o ProLabinf - Projeto de Revitalização dos Laboratórios de Informática do Departamento de Computação (DComp), o qual previu a cooperação entre docentes e discentes, voltados à busca de soluções para os problemas dos laboratórios (COSTA NETO e COX, 2004).

O número de equipamentos (microcomputadores, servidores, dispositivos de rede e multimídia) disponíveis no curso de Ciência da Computação da UFS tem crescido significativamente nos últimos anos. Entretanto, houve momento na história do DComp que essa elevação não foi acompanhada pelo aumento de recursos humanos para gerenciá-los. Nesta oportunidade, professores revezavam-se na gerência dos laboratórios e atividades relacionadas.

Foram feitas tentativas de recrutar alunos para trabalhar nos laboratórios, mas como a participação consistia em executar atividades especificadas e controladas pelos professores, os alunos não se sentiam motivados já que não havia liberdade, por exemplo, para definir metodologia de trabalho e participar das decisões. Consequentemente, houve pouca cooperação destes.

Sendo assim, identificou-se uma oportunidade de aplicar o Protagonismo Juvenil em um projeto que unisse alunos e professores visando aperfeiçoar os serviços dos laboratórios e ao mesmo tempo favorecer o processo de formação profissional discente.

Dessa forma, segundo Costa Neto e Cox (2004), surgiu o ProLabinf, sob a coordenação técnica do professor Alberto Costa Neto, cuja proposta foi discutida com alunos, outros professores, servidores e membros da Empresa Júnior de Informática da UFS (Softeam), sendo este aprovado em Conselho Departamental e posteriormente implantado. O ProLabInf previu a cooperação entre docentes e discentes, buscando soluções para os problemas dos laboratórios.

No desenvolvimento do projeto foi prevista a formação de grupos de estudantes, veteranos e novatos, orientados por professores, e co-orientados por alunos com mais experiência no assunto, para realização de atividades inerentes ao gerenciamento dos laboratórios.

Os educandos foram incentivados a realizar trabalhos de conclusão de curso, estágios, projetos de disciplinas e atividades de extensão aplicados aos laboratórios de informática, bem como apresentar seminários socializando as atividades e estudos realizados. Foram criados cinco grupos de trabalho: (1) de Infra-Estrutura, (2) de Desenvolvimento de Sistemas, (3) de Banco de Dados, (4) de Rede e (5) de Segurança; cada um com suas funções específicas.

A ideia do projeto foi apresentada ao corpo docente e discente por meio de uma palestra, seguida por divulgação em turmas isoladas e em grupos de discussão, e assim diversos alunos foram motivados a participarem do ProLabinf.

As atividades do ProLabinf foram iniciadas pelo grupo de Infra-Estrutura, composto principalmente por alunos do primeiro período. Em 20 dias, estes recuperaram os microcomputadores de um dos laboratórios, ocioso, por 10 meses, por falta de manutenção, identificaram e trocaram componentes defeituosos, instalaram os *software* e testaram, deixando tudo em condições de uso.

A cooperação entre os alunos aflorou naturalmente. Organizaram seus horários de dedicação de forma a sistematizar as atividades; participaram ativamente da decisão, do planejamento, e mantiveram-se dispostos a atuar na execução e avaliação dessas.

No desenvolvimento do ProLabinf, houve ganhos para todos os envolvidos. Os alunos tiveram a oportunidade de colocar em prática, desenvolver e testar seus conhecimentos e habilidades em situações reais. Os professores encontraram fonte inesgotável de questões desafiadoras para instigar a aplicação dos programas curriculares trabalhados. E o curso teve o gerenciamento de seus recursos efetuado com qualidade e eficiência.

Outro ponto positivo foi a interação entre alunos de períodos diferentes, o que normalmente não ocorre, já que dificilmente frequentam disciplinas em comum. Dessa forma, eles puderam compartilhar experiências, livros, artigos, apostilas, fontes de pesquisa e socializar saberes e práticas de uso dos recursos tecnológicos explorados ao longo do curso.

Outros projetos de extensão do DComp devem explorar o protagonismo juvenil já que a iniciativa então apresentada sinalizou que são inúmeros os benefícios obtidos a partir deste. Também os citados benefícios podem ser estendidos à sociedade envolvendo alunos em projetos, por exemplo, de inclusão digital.

2.4 Para Inclusão Digital (ANDRADE, COX e SILVA, 2007)

Em projeto de extensão do DComp, em 2007, envolvendo as alunas do curso de computação Jéssica Caroso e Luana Barreto como instrutoras, foram capacitados: 19 jovens, 4 professores e 7 funcionários da Escola Estadual “Armando Guarana”, muitos dos quais tiveram o primeiro contato direto com um computador através deste (ANDRADE, COX e SILVA, 2007).

Os assistidos por este projeto, hoje encontram-se habilitados a: editar e formatar textos e planilhas eletrônicas; elaborar apresentações com efeitos, ilustrações e gráficos; navegar e pesquisar na rede mundial de computadores; e trocar mensagens eletrônicas por meio de gerenciadores de e-mail.

Os professores assistidos também tiveram contato com formas, vantagens, desvantagens e requisitos de aplicação dos recursos computacionais a serviço de práticas educacionais escolares. Podendo, quando a escola dispor de um laboratório próprio, aplicar criticamente recursos computacionais em suas atividades.

As alunas do curso de computação, instrutoras do projeto, complementaram sua formação acadêmica ao: estudar as ferramentas de microinformática; pesquisar e propor aplicações dos recursos estudados; vivenciar experiência de ensino: planejamento de aulas, elaboração e correção de provas, interação com alunos, estudo de alternativas metodológicas de ensino; e elaborar artigo para participação em evento científico.

A proposta inicial do projeto não vislumbrava a possibilidade de capacitar funcionários da escola; mas estes demonstraram interesse e foram acolhidos, de bom grado, junto à turma de professores. A escola dispõe de um computador na secretaria; o qual, antes da capacitação aqui descrita, era operado por um único funcionário. Assim, muitas vezes era preciso esperar o turno de trabalho deste para cumprimento das atividades que envolvessem o uso do computador. Hoje, outros sete servidores encontram-se habilitados para efetuar tais funções.

Dado o exposto, percebe-se que os resultados obtidos justificam a preservação de esforços neste sentido – da inclusão digital. Houve: capacitação de jovens, desmistificação de tecnologia,

qualificação de professores, extensão do alcance da atuação da universidade, complementação de formação acadêmica; e, ainda, possivelmente o ensino na Escola Estadual “Armando Guarani” pode ter melhorias se for oportunizado aos professores capacitados aplicarem os recursos computacionais estudados a serviço de suas atividades educacionais.

Segundo Andrade, Cox e Silva (2007), trabalhou-se com professores e alunos, recursos de: sistema operacional (10 horas), *internet* (10 horas), editor de textos (16 horas), planilha eletrônica (12 horas), *software* de apresentação (18 horas) através de projetos de aplicação: textos significativos, planilhas de utilidade para a escola, apresentações de interesse dos alunos assistidos, trocas efetivas de e-mail e pesquisas diversas na *internet*.

O curso totalizou 66 horas de aulas ministradas nos laboratórios do DComp. Os assuntos foram expostos através de slides em TVs existentes nos laboratórios. Inicialmente, para ajudar aos alunos a terem o primeiro contato com o computador, foram introduzidos jogos educacionais às aulas. Jogos estes que desenvolveram as habilidades de uso do mouse e teclado. As turmas foram compostas pela escola e muitos dos alunos eram adultos, o que resultou no primeiro desafio para as instrutoras.

É muito comum ao se trabalhar informática com alunos adultos identificarem-se indivíduos com expressa dificuldade no manuseio de teclado e mouse. Este entrave, ainda que natural, já que cada idade conta com suas características, precisa ser vencido, pois clicar, teclar, abrir, fechar, arrastar são operações básicas das quais dependem todas as atividades de interação humano-computador. Então surge o questionamento de como tratar esta situação. Demonstrar? Deixar o tempo encarregar-se de tal treinamento? Passar exercícios estritamente mecânicos? Segurar na mão do aluno sempre que necessário?

Para crianças e jovens tal treinamento poderia ser efetuado por meio de jogos – atrativos, prazerosos, adaptáveis ao ritmo de cada aluno. Entretanto, como o adulto responderiam a esta proposta? Na busca de respostas às questões surgiu a necessidade de “ensino de computação” na andragogia.

2.5 Na Andragogia (ANDRADE, COX e SILVA, 2010)

Os adultos, diferente das crianças, são aprendizes que trazem consigo as experiências de vida que foram acumuladas ao longo de sua existência e estas devem ser consideradas no processo de aprendizagem de maneira a despertar aplicações dos novos conhecimentos e assim estimular a compreensão destes. O papel de educar, desta maneira, consiste em facilitar o aprendizado do aluno, considerando seus conhecimentos.

Larrosa (1999), na apresentação da coleção Educação: Experiência e Sentido, acentua a importância da experiência do aprendizado, afirmando que a experiência, e não a verdade é o que dá sentido à educação. “A andragogia foca-se no aprendizado do aluno em detrimento do modelo centrado no professor que atribui ao aprendiz o papel de submisso, com o encargo de acatar as instruções dadas.”

Segundo Goecks (2009), em seu artigo Educação de Adultos, a andragogia volta-se para o ensino de adultos. Um caminho educacional que busca compreender o adulto com os componentes humanos: psicológico, biológico e social. Assim, a atividade educacional do adulto deve ser centrada na aprendizagem e não no ensino. Os adultos são motivados a aprender à medida que percebem que o aprendizado irá ajudá-los a resolver tarefas ou lidar com problemas com os quais eles se confrontam em situações concretas. Em outras palavras, adquirem novos conhecimentos quando estes são apresentados no contexto de vida real.

O modelo andragógico de educação se apóia no princípio de que o ator principal no processo de ensino-aprendizagem é o aluno adulto. No processo de educar proposto por tal modelo, o professor deixa de ser o único canal de transmissão do conhecimento e torna-se um guia, um

facilitador, que também participa na aprendizagem na proporção da vitalidade e relevância de suas experiências.

Neste contexto, uma estratégia de ensino que se destaca são as denominadas atividades desafiadoras – as quais são programadas, com objetivos pré-determinados, e correspondem à proposição de situações problemas que desafiem o educando a criar, a protagonizar, a propor uma solução, e assim auxiliam no processo educativo, por serem extremamente envolventes, tornando o ensino-aprendizagem mais significativo.

Com essas atividades almeja-se fazer com que o aluno despoje-se da postura de sujeito e assumo o papel de artífice, assimile diferentes representações sobre o mundo e desenvolva inúmeras formas de se comunicar, vivenciar suas emoções, interagir com os colegas, melhorar seu desempenho, nível linguístico e formação cultural.

As atividades desafiadoras podem, e são aplicadas a crianças e jovens, mas para o aprendiz adulto há um diferencial significativo: a maturidade. Como para a aplicação destas, o professor deve encontrar formas para que o aluno se sinta motivado a aprender, para que o aluno se sinta desafiado, tais estímulos no adulto devem considerar a citada maturidade, ou devem ser compatíveis com as necessidades e expectativas destes.

A aprendizagem é entendida, então, como um processo ativo, participativo e construtivo, portanto, o aprendiz adulto deve ser conduzido ao seu próprio desenvolvimento.

Em projeto de extensão, as alunas Jéssica Caroso e Luana Barreto, como instrutoras de uma capacitação voltada para inclusão digital, experimentaram a aplicação das citadas atividades desafiadoras no processo educacional andragógico para o ensino de informática. Estas ministraram curso promovido pela Softeam, Empresa Júnior de Informática da Universidade Federal de Sergipe para funcionários de um hospital particular do estado. (ANDRADE, COX e SILVA, 2010)

A turma era composta por 20 alunos com idade que variava entre 25 e 50 anos, todos dispostos a familiarizar-se com computadores – os quais, para a maioria, eram equipamentos totalmente nunca antes manuseados. Visando trabalhar com atividades desafiadoras foram explorados inicialmente, jogos para estimular o manuseio do teclado e do mouse.

Um dos jogos aplicados foi de palavras cruzadas, em que o jogador era desafiado a completar as lacunas existentes com a palavra adequada. Ver Figura 2.

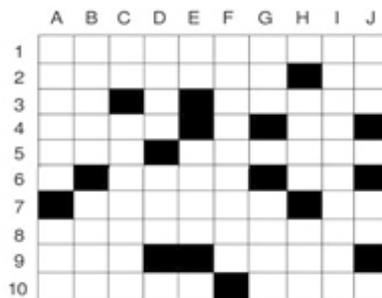


Figura 2: Jogo Palavras-Cruzadas, usado em atividade desafiadora de projeto andragógico.

Fonte: Andrade, Cox e Silva (2010).

Assim, contando com o estímulo complementar da ludicidade, os educandos foram desafiados a manusear o teclado. Através deste ocorreu o reconhecimento do teclado, sua composição e funcionalidades sem necessidades de preleções prévias acerca destas, mas por meio da exploração do mesmo e descobertas.

A maioria dos educandos se entusiasmou com a possibilidade de brincarem aprendendo. A euforia era nítida. Envolvidos com o propósito de “vencer” o jogo, esqueceram que estavam

explorando o “temido” teclado. Esta primeira experiência já sinalizou que as atividades desafiadoras atuam como bons aliados no processo de ensino de adultos.

Um outro jogo aplicado com o intuito de aproximar fisicamente o estudante das máquinas, e assim favorecer o manuseio desta, foi um caça palavras o qual estimulava o manuseio do mouse de forma prazerosa e divertida. Neste o jogador deveria circular com o uso do mouse as palavras apresentadas. Ver Figura 3.



Figura 3: Jogo Caça-Palavras, usado em atividade desafiadora de projeto andragógico.
Fonte: Andrade, Cox e Silva (2010).

Em seguida foram propostas a participação em chats, e criação de blogs tratando dos mais diversos assuntos de interesse da turma. O objetivo era aproximar o aprendiz dos recursos de edição de texto.

O curso completo de capacitação em informática, voltada para os funcionários do hospital, foi composto pela exploração de recursos voltados para: gerenciamento de e-mails, uso de browser, editor de texto e planilha eletrônica; além de conceitos básicos de computação apresentados à medida que foram sendo necessários.

Na exploração de edição de texto, foram trabalhados: ofícios, circulares internos, avisos, cartazes e e-mails com conteúdos voltados para o contexto hospitalar.

A grande maioria dos alunos quis construir documentos úteis ao cumprimento de suas funções na instituição e buscaram saber dos recursos do editor antes mesmo destes serem apresentados, o que sinaliza que o aprendizado estava ocorrendo de forma ativa, criativa, funcional – autônoma, como esperado. Para trabalhar os recursos de planilha eletrônica foram também exploradas situações problemas presentes em ambientes hospitalares; a exemplo de planilha de controle de materiais. Ver Figura 4.

| CONTROLE DE MATERIAIS | | | | |
|--|-------|-------|-------------|-------------|
| Descrição | Unid. | Qtde. | Valor Unid. | Valor Total |
| Água sanitária 1000ml | | 20 | R\$ 2,00 | R\$ 40,00 |
| Flanela amarela 30x60 | | | | |
| Hipoclorito de sódio - Galão 5 L | | | | |
| Lã de aço - pacote com 8 unidades | | | | |
| Lixeira 25L | | | | |
| Lixeira 80L, branca - c/ tampa | | | | |
| Lixeira 80L, cinza - c/ tampa | | | | |
| Luva látex Mucambo - grande - (par) | | | | |
| Luva látex Mucambo - média - (par) | | | | |
| Pano de prato | | 10 | R\$ 2,00 | R\$ 20,00 |
| Papel higiênico - folha dupla neutro - 30m | | | | |
| Papel toalha 23x23cm - 1000 folhas | | | | |
| Sabão barra 200g | | | | |
| Sabonete líquido - 5 litros | | | | |

Figura 4: Planilha usada em atividade desafiadora de projeto andragógico.

Fonte: Andrade, Cox e Silva (2010).

O que motivava os alunos a frequentar as aulas era o fato de que estas não eram unicamente expositivas, mas sim dinâmicas, tornando-as prazerosas e divertidas. A cada encontro novos desafios eram lançados e cumpridos das mais diversas formas possíveis, uma vez que estimulavam a criatividade, e o ritmo e perfil de cada um era respeitado.

Ao final do curso, as ferramentas digitais que pareciam complexas passaram a ser apreciadas, não mais amedrontavam os aprendizes. Uma fração de 99% da turma passou a apresentar absoluta desenvoltura no manuseio do equipamento. Assim, as resistências observadas na maturidade foram superadas. Como também superadas (atendidas) podem ser as necessidades especiais apresentadas por uma significativa parcela da população mundial, a exemplo dos invisuais, conforme relata a seção que segue.

2.6 Para Invisuais (PALMEIRA, HIROSE, MESQUITA e COX, 2005)

Segundo Palmeira, Hirose, Mesquita e Cox (2005), no âmbito do Departamento de Computação, da Universidade Federal de Sergipe, foi construído o *software* de apoio ao ensino denominado *ClassVision*, voltado para portadores de necessidades visuais (PNV).

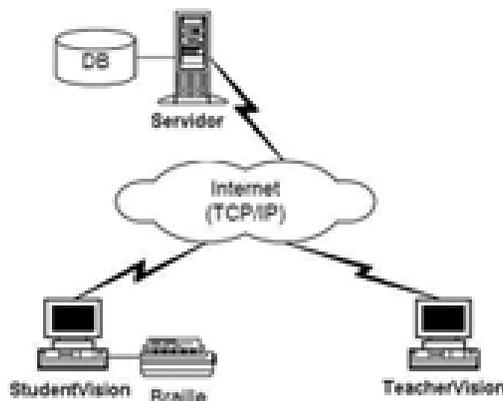


Figura 5: Arquitetura da *ClassVision* – sistema educativo voltado para invisuais.

Fonte: Palmeira, Hirose, Mesquita e Cox (2005).

O *ClassVision* visa garantir acesso à informação para o PNV através da utilização de tecnologia de síntese de voz. Sua arquitetura pode ser ilustrada na figura 5. Nesta arquitetura, existem três elementos intercomunicantes: o *TeacherVision*, o *StudentVision* e o Servidor.

O *TeacherVision* é o módulo voltado para os professores. Possibilita a elaboração e o monitoramento de atividades educacionais, de forma clara, rápida e objetiva.

Para atender aos requisitos a que se propõe, o *TeacherVision* apresenta como características: facilidade de uso, visto que a maioria dos professores não tem conhecimento avançado em informática; interface visual; disponibilidade de recursos que tornam as aulas atraentes e que interajam com os alunos através de recursos sonoros; e possibilidade de confecção de provas e exercícios.

Sendo assim, o *TeacherVision* atua como fomentador da adoção de uma postura de inclusão do PNV na instituição de ensino, postura essa que implica na correta formação do professorado.

O módulo *StudentVision* volta-se para o aluno portador de necessidades visuais, na medida em que disponibiliza as aulas redigidas por professores, através da exploração da síntese de voz, que é o principal canal de interação entre o aluno invisual e a aplicação, visando aproximá-lo cada vez mais do conhecimento. A partir do contato com o saber, o invisual poderá desenvolver seu potencial e, conseqüentemente, assumir papéis junto à sociedade, atingindo a almejada integração social.

A comunicação entre esses dois módulos foi destinado a projetos futuros, através de um servidor que atenderá às requisições desses elementos, provendo os serviços básicos de integração a fim de que o conteúdo seja disponibilizado ao aluno. O servidor possui um papel muito importante relacionado à centralização das informações no que diz respeito às aulas disponibilizadas, atuando como um gerenciador não somente do conteúdo educacional, mas também provendo mecanismos de integração de novas funcionalidades.

No *StudentVision*, a comunicação entre o aluno e o computador é feita via teclas de atalho ou o mouse; o qual ao ser passado pelos botões que estão estrategicamente posicionados por toda a tela, dispara um evento de síntese de voz que lê o respectivo texto. A Figura 6 mostra a aparência da tela principal do *StudentVision*.



Figura 6: Tela principal do *StudentVision* – sistema educativo voltado para invisuais.

Fonte: Palmeira, Hirose, Mesquita e Cox (2005).

Foi utilizada toda a área da tela do computador para apresentação das funcionalidades do sistema para evitar que o mouse seja posicionado sobre coordenadas sem funcionalidades correspondentes. Com este mesmo intuito há o destaque sonoro, através da leitura do texto correspondente aos botões; e o destaque visual aos botões por contraste de cores quando o mouse estaciona em um deles, para atender também aos portadores de visão subnormal, ou seja, que conseguem enxergar sob certas condições.



Figura 7: Tela inicial do *TeacherVision* – sistema educativo voltado para invisuais.

Fonte: Palmeira, Hirose, Mesquita e Cox (2005).

O *TeacherVision* apresenta como funcionalidades: Elaborar Aula e Cadastrar Professor. Ver Figura 7. Este foi projetado de forma a facilitar a adição de novas funcionalidades, as quais aparecerão como opções a mais nessa mesma tela. Na Figura 8 é ilustrada a interface da funcionalidade que permite a elaboração de aulas.



Figura 8: Tela principal do *TeacherVision* – sistema educativo voltado para invisuais.

Fonte: Palmeira, Hirose, Mesquita e Cox (2005).

Através do *TeacherVision* é possível elaborar, consultar, alterar e excluir aulas, e o cadastramento de disciplinas. Na tela apresentada na Figura 8 observa-se o nome completo do professor que está usando o *software* – campo 1; nome da aula e da disciplina – campo 2; itens disponíveis para composição da aula – região 3; na região 4 há as ferramentas necessárias à edição do texto de conteúdo da aula; o Conteúdo da Aula é apresentado na região 5; e na região 6 são apresentados os textos referentes ao conteúdo da aula, juntamente com seus tópicos, que aparecem em negrito, e os itens de aula. Vale esclarecer que o *ClassVision* foi projetado de forma a tornar possível acréscimos de funcionalidades. O *ClassVision* foi testado e avaliado por docentes de ensino médio e superior, profissionais do Centro de Apoio Pedagógico do Estado de Sergipe – CAP/SE – e alunos invisuais desse centro. Segundo Palmeira, Hirose, Mesquita e Cos (2005), o resultado da avaliação sinaliza que vale o esforço de continuidade deste.

Até este ponto foram apresentadas iniciativas onde ocorreram o consolidar de objetivos da

computação – orientação à objetos, introdução à programação, inclusão digital – com o apoio dos recursos do ensino, obtendo o que aqui se denomina “ensino de computação”. Em seguida são apresentados projetos onde os alvos considerados são de ensino e a computação oferta ferramentas, técnicas e métodos – a “computação no ensino”.

3 Computação no Ensino

São, dentre outras, vantagens de uso da computação no ensino: estímulo a mudanças na educação escolar, favorecimento do desenvolvimento da cidadania, favorecimento da interdisciplinaridade e preparação para o mundo do trabalho.

Há estímulo a mudanças na educação escolar porque, para adoção da computação no ambiente escolar é preciso avaliar quais os objetivos da educação, então missões são rediscutidas, posturas reavaliadas, instrumentos remodelados.

Também ocorre favorecimento do desenvolvimento da cidadania devido ao fato de que os recursos da computação facilitam a interação de pessoas entre si e de pessoas com informações. Sendo cidadania esta capacidade humana de interagir com os elementos do entorno de forma ativa, esta é favorecida com a adoção da computação na vida escolar.

Já a interdisciplinaridade é favorecida porque com a computação o desenvolvimento de projetos de pesquisa nas escolas é otimizado através da aplicação de banco de dados, facilidades de consulta, beneficiamento de interações entre usuários e usuário-informação. E muitas pesquisas, em especial as aplicadas, apresentam natureza interdisciplinar.

Quanto à preparação para o mundo do trabalho, basta considerar o hodierno entorno informatizado para facilmente concluir que conhecimentos básicos em informática são indispensáveis a todo cidadão.

Dado o exposto, a “computação no ensino” faz-se necessária; e, assim como o “ensino da computação” pode ser desenvolvida por meio de inúmeras iniciativas, embasadas em múltiplas abordagens e voltadas para diversos objetos de estudo: treinamento de pilotos, formação de professores, para educação musical, para educação financeira, ensino de línguas, e muitos outros; tantos quantos forem os campos de estudo de interesse da humanidade.

3.1 Na Educação Musical (COSTA-SILVA, LEÃO e COX, 2010)

No Brasil, desde o dia 18 de agosto de 2008, o ensino de música na educação básica passou a ser obrigatório. É o que determina o governo federal através da Lei n.º 11.769/2008.

Muitas são as dificuldades enfrentadas pelas escolas ao se implantar o ensino de música, das quais pode-se destacar o alto custo de aquisição de instrumentos musicais, bem como a manutenção destes. Ressalta-se ainda a falta de exigência legal de que o professor de música seja um profissional com formação específica na área; o que torna necessário o uso de ferramentas que o auxilie durante suas aulas, a exemplo de *software* voltado pela educação musical.

Segundo Costa-Silva, Leão e Cox (2010), cada vez mais, são reconhecidos os efeitos positivos resultantes da utilização de computadores por parte dos docents, em salas de aula. Nas aulas de música, estes possibilitam simular situações musicais, reproduzir com qualidade e fidelidade os diversos sons dos instrumentos, reduzindo assim os altos custos que a escola teria com aquisição e manutenção de instrumentos, o que tornaria inviável para muitas escolas o cumprimento da legislação. Portanto, a utilização de computadores e *software* como ferramentas de apoio ao professor no processo educacional certamente minimizará as dificuldades existentes na implantação do ensino de música nas escolas.

Considerando então as exigências curriculares e regulamentares da atualidade, os inúmeros benefícios da música na educação, bem como a necessidade de uma ferramenta que auxilie no

processo de implantação do ensino de música, no trabalho de conclusão de curso dos alunos Danielle de O. Costa-Silva e Raphael P. S. Panta Leão foi construído o *software* “Percep”, visando desenvolver a percepção musical de crianças através da experimentação sonora. (COSTA-SILVA, LEÃO e COX, 2010)

Existem muitas ferramentas voltadas para o desenvolvimento da percepção musical, como o Zorelha, *GNU Solfège* e Sistema de Treinamento Rítmico, porém estas trabalham com ensino de conceitos e não apresentam amigabilidade voltada para o público infantil. Por esta razão, o “Percep” preocupou-se em oferecer uma interface interativa, atrativa, colorida, dinâmica e intuitiva ao máximo. Ver Figura 9.



Figura 9: Exemplo de interface do *software* de Educação Infantil “Percep”.

Fonte: Costa-Silva, Leão e Cox (2010).

Vale esclarecer que a percepção musical é a arte de perceber as ondas sonoras como parte de uma linguagem musical. É a capacidade de analisar e entender auditivamente elementos musicais, como melodia e ritmo, bem como propriedades físicas do som. Além da educação musical, outro possível campo de ação da “computação no ensino” é na Educação Financeira.

3.2 Na Educação Financeira (CAROSO e COX, 2010)

Desde as noções básicas de vida, como as necessidades fisiológicas, até os padrões de bom comportamento e cultura, tudo que constrói o ser humano passa pela educação. Não pode ser diferente com as noções relacionadas a dinheiro. Assim como as noções básicas, a educação financeira também deve ser iniciada nos primeiros meses de vida.

Segundo D’Aquino (2008), exercitar a paciência dos pequenos é um dos pontos fundamentais no processo de educação financeira; pois a ansiedade muitas vezes conduz o adulto ao endividamento devido a incapacidade de esperar angariar os recursos necessários à obtenção dos bens e serviços que almejam.

Desde cedo também, é necessário que as crianças entendam e materializem o conceito de poupar. Com o aprendizado do economizar, pode-se mostrar às crianças como adquirir produtos com valor superior ao que se dispõe.

A educação financeira ainda é um tema pouco explorado por *software* educativos. Antes ignorada pelo Brasil, este tipo de educação vem sendo atualmente enfatizada pelo Governo Federal, em parceria com outras instituições como o Banco Central do Brasil, através de recentes iniciativas como a “Vida e Dinheiro”, as quais visam aproximar as diversas classes sociais das informações que lhes permitam compreender o funcionamento dos produtos financeiros, e pretendem alcançar todas as faixas etárias.

Embasada por estas evidências, Caroso e Cox (2010) concretizou-se a escolha do tema

“Educação Financeira para Crianças” como conteúdo do *software* educativo (SE), “Max Descobrimo o Mundo Financeiro”, desenvolvido no trabalho de conclusão de curso da aluna Jéssica Caroso.

Os pontos chaves abordados no SE foram: a importância de elaborar uma lista de compras; o conceito de caro e barato; a diferença entre supérfluo e necessário; orientações sobre como administrar o próprio dinheiro; a diferença entre pagamento à vista e a prazo; a importância de economizar e esperar.

Em “Max Descobrimo o Mundo Financeiro”, a missão do aluno é participar das finanças da casa e aprender com elas os principais ensinamentos da educação financeira.

A tela inicial apresenta o garoto – Max – dando “giros” sobre uma moeda. Esta explora cores e animação visando atrair o usuário infantil. Ver figura 10.



Figura 10: Tela inicial do SE “Max Descobrimo o Mundo Financeiro”.

Fonte: Caroso e Cox (2010).

Na tela seguinte o aluno é convidado a aprender sobre as finanças junto com o personagem principal do SE: Max. Ao clique positivo do aluno, o SE traz o menu com 6 fases que representam, cada uma, um ensinamento da educação financeira. As fases podem ser acessadas independentemente umas das outras, ou sequencialmente. Em cada fase é apresentada inicialmente uma história com Max e os pais. Ver exemplo em Figura 11.



Figura 11: Exemplo de tela do “Max Descobrimo o Mundo Financeiro”.

Fonte: Caroso e Cox (2010).

Considerando o mesmo objetivo geral, de educação financeira, a aluna Cláudia Carvalho, em trabalho de conclusão de curso em andamento, encontra-se desenvolvendo jogo que visa o ensino de princípios básicos de educação financeira, como: diferença entre o que é um ativo e

um passivo; valor do dinheiro no tempo; efeito da incidência de juros; custo de financiamento; orçamento e risco, através de exemplos que permitem que as crianças visualizem a aplicação deste em situações reais.

O jogo é composto de um tabuleiro, uma roleta e um dado. A medida que o personagem percorre o tabuleiro ele vivencia situações semelhantes à realidade. Neste o usuário poderá efetuar uma compra com pagamento parcelado ou à vista, empréstimo bancário, investimento em ações e outras ações financeiras.

Ao final do jogo, não ganha quem percorrer o tabuleiro mais rápido, mas sim aquele que chegar ao final do tabuleiro com mais recursos.

Iniciativas de “computação no ensino” podem ainda voltar-se para o desenvolvimento de inteligências múltiplas.

3.3 No Desenvolvimento de Inteligências Múltiplas (GALVÃO e COX, 2011)

Howard Gardner, autor da teoria das inteligências múltiplas, questionou a tradicional visão da inteligência que enfatiza apenas as habilidades linguística e lógico-matemática. A teoria de Inteligências Múltiplas de Gardner identifica as dimensões: linguística, lógico-matemática, espacial, musical, cinestésica, interpessoal e intrapessoal. É argumentado que essas competências intelectuais são relativamente independentes.

Para Gardner, os seres humanos dispõem de graus variados de cada uma das inteligências. Estas apresentam maneiras diferentes de se combinarem e se organizarem. Além disso, o indivíduo utiliza-se dessas capacidades intelectuais para resolver problemas e criar produtos.

A teoria das inteligências múltiplas propõe que todos os indivíduos possuem habilidade de questionar e procurar respostas usando todas as inteligências. Já o desenvolvimento de cada inteligência, no entanto, é determinado tanto por fatores genéticos e neurobiológicos quanto por condições ambientais. O ambiente e a educação são essenciais para os estímulos das inteligências.

Um meio eficiente de estimular as inteligências são os jogos. Estes provocam um fascínio nas mentes dos indivíduos ao permitir que realizem tudo quanto deseja. Criam, também, um atalho entre o mundo da fantasia e o mundo real. Nesse caso, não são empregados no sentido de competição entre pessoas implicando em vitória ou derrota, e sim, de modo a contribuírem de forma satisfatória como estímulo ao crescimento e ao desenvolvimento cognitivo dos indivíduos, e também, como preparação para os desafios da vida.

Para enfatizar essa relevância dos jogos, Gardner (1987) ressaltou que as atividades lúdicas têm cada vez mais o seu lugar garantido no processo de ensino-aprendizagem, pois estas aliam o lazer ao desafio, operando com todos os tipos de inteligência.

Considerando o exposto, o aluno Lauro Galvão, em trabalho de conclusão de curso em desenvolvimento, objetiva criar jogos computacionais para o desenvolvimento de inteligências múltiplas. (GALVÃO e COX, 2011)

Quando os jogos fazem parte do processo de aprendizagem como motivadores eles são definidos como jogos educacionais. Como ferramenta tecnológica, o computador pode (e deve) ser bastante utilizado com propósitos educacionais, principalmente através de jogos educativos. A sua importância vai além do impacto na nossa sociedade e das novas exigências sociais e culturais que se impõe a, até mesmo, o uso das novas tecnologias na educação. O professor tem o direito de usufruir desta tecnologia para angariar resultados eficientes no processo de ensino-aprendizagem. Já seus alunos ganham mais um aliado para o aprendizado, que são os jogos para computador.

Segundo Tarouco (2011), a utilização de jogos educacionais para computador motiva o aluno, além de desenvolver hábitos de persistência no desenvolvimento de desafios e tarefas.

Neste contexto, são múltiplas as possibilidades de iniciativas. Podem-se ter jogos voltados para inteligências específicas, ou para grupos de inteligência, ou para todas as inteligências; e estes ainda podem ser voltados para faixas etárias e/ou grupos específicos de usuários. Ainda tratando sobre “computação no ensino”, é possível citar trabalhos que se voltam para informática médica, com é o caso do ensino de Patologia, citado a seguir.

3.4 De Patologia (SOUZA, COX, SILVA e FAKHOURI, 2004)

O estudo de Processos Patológicos Gerais é uma área crítica do conhecimento nos cursos das Ciências da Saúde, pois envolve a assimilação de quadros pictóricos microscópicos dos processos de doenças nos tecidos, o que implica em alguns obstáculos ao aprendizado, tais como dificuldade em distinguir imagens bastante similares, utilização de microscópio e outros.

Considerando o exposto, a aluna Paula Costa, em trabalho de iniciação científica, sob a orientação dos professores Kenia Kodel, Leila Silva e Ricardo Fakhouri, dedicou-se ao desenvolvimento de *software* educativo de exercício e prática aplicado à distância, junto às atividades docentes tradicionais, visando suprir as citadas necessidades. (SOUZA, COX, SILVA e FAKHOURI, 2004)

Na educação médica, a educação à distância apresenta resultados positivos de aprendizado, com várias iniciativas neste sentido. No entanto, na oportunidade estas não apresentavam exercícios para a prática e construção do conhecimento.

Então surgiu a ideia de construir um *software* para a disciplina Processos Patológicos Gerais. Esta última é fundamental para a formação básica dos alunos da área de saúde.

As aulas do *software* são compostas de conteúdo teórico e de exercícios. Nos exercícios são disponibilizados quatro opções de resposta, ilustrada na Figura 12.

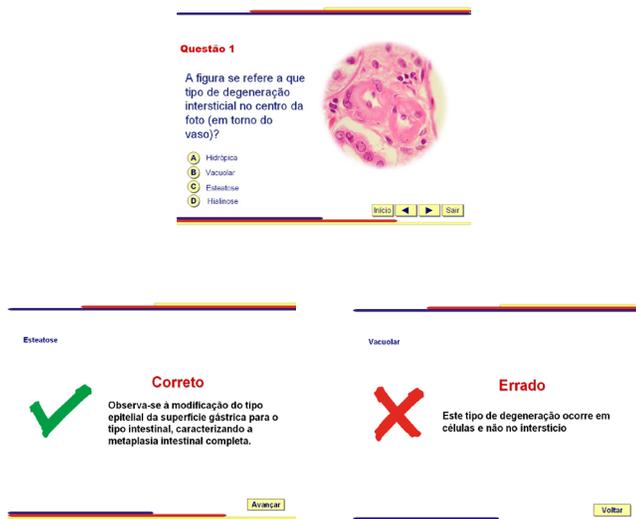


Figura 12: Exemplos de interfaces do Curso à Distância de Patologia.
Fonte: Souza, Cox, Silva e Fakhouri, 2004.

Se o aluno selecionar uma resposta errada, uma tela explicando a razão do erro cometido é exibida. Caso contrário, uma explicação enfatizando a resposta é apresentada. Assim observa-se que os campos de emprego da “computação no ensino” podem estender-se para tantos quantos sejam os diferentes campos de aplicação do ensino. Em outras palavras, são, infínitos.

4 Conclusão

São inúmeras, e de diversos matizes, as iniciativas que podem ser desenvolvidas a partir da consolidação da computação com o ensino, seja no “ensino da computação” – voltadas, por exemplo, para ambientes e ferramentas de apoio ao aprendizado, de *e-learning*, com estudos de caso, explorando aprendizagem colaborativa, aplicando o ensino por projeto; assim como na “computação no ensino” – para a exploração e/ou desenvolvimento de ferramentas e recursos informáticos: jogos, *internet*, realidade aumentada, *software*, portais *web*. E estas merecem atenção da academia porque promovem o desenvolvimento científico e social almejados pelas duas áreas envolvidas. Um nicho que se destaca neste entorno é o ensino de programação.

Hodiernamente são inúmeros os papéis desempenhados pelos computadores no cotidiano humano. E para cumprimento destas inúmeras funções, apesar da diversidade destas, uma única máquina de processamento é suficiente; isto porque os computadores são modeláveis, são plásticos. E esta plasticidade é garantida pela programação.

Os programas, por sua vez, são compostos por instruções, elaboradas pelo homem, e efetuados pela máquina, para execução de uma determinada tarefa. Assim, estes devem ser legíveis para os homens – seus construtores; bem como para as máquinas de processamento – suas executoras.

Então entram em cena os algoritmos e linguagens de programação. Nem “naturais” como as linguagens humanas; nem compostas por zeros e uns como as compreendidas pelas máquinas de processamento. Para o computador atender a multiplicidade de funções atribuídas a este, faz-se necessário múltiplos algoritmos, o que requer profissionais capacitados para a construção destes; profissionais com as habilidades técnicas necessárias, eficientes e criativos.

Entretanto, o que se observa nos cursos de introdução a algoritmos é um número considerável de alunos com dificuldades de aprendizagem, o que requer providências, requer a busca por recursos de ensino adequados – suficientes à superação destas dificuldades discentes. Então, faz-se necessário desenvolver estudos visando a superação destes obstáculos ao aprendizado e, havendo êxito, serão favorecidos os processos de construção de programas computacionais, o alcance dos benefícios da computação, a formação de mão-de-obra; o que é indubitavelmente relevante em termos cultural, social e político.

No ensino de programação destacam-se: (a) a abstração necessária à construção dos algoritmos, pois os itens concretos pertinentes aos problemas, bem como as relações entres estes, devem ser representados nos algoritmos a partir da aplicação dos recursos disponibilizados pela linguagem algorítmica; e (b) a dinâmica em que os algoritmos são traduzidos, já que estes são sequencias de ações, cuja execução deve resultar na resolução do problema inicial.

Segundo Martins (2010):

”O estudante vivencia várias atividades: leitura e entendimento da proposta do problema, articulação e busca de conhecimentos, abstração, criatividade, organização lógica e descrição. A vivência em sala de aula permite afirmar que a conjugação destas várias atividades é um dos fatores que contribuem para as dificuldades experimentadas pelos alunos.” (MARTINS, 2010)

Alicerçado na teoria dos Registros de Representação Semiótica, na Dialética Ferramenta-Objeto e na Engenharia Didática, Martins (2010) concluiu que houve favorecimento ao entendimento e domínio da noção da dinâmica dos algoritmos, o que afirma ter contribuído para o aprendizado do entendimento e elaboração de algoritmos e programas. Isto através da aplicação de quatro experimentos que exploraram os conceitos iniciais trabalhados nas disciplinas de introdução a algoritmos.

Assim, considerando como alicerce teórico-metodológico a teoria dos Registros de Representação Semiótica e o *Design* Instrucional, pode-se dar continuidade às investigações de como o estudante revela, trata e domina a noção de processo dinâmico inerente a um algoritmo ou

programa, e o reflexo deste domínio nas atividades de compreensão e construção de algoritmos.

Segundo Duval (2008), a teoria dos Registros de Representação Semiótica foi concebida para buscar o entendimento do processo de aprendizagem de objetos matemáticos. Duas categorias de registros de representação são empregadas no desenvolvimento de algoritmos: (1) registros em linguagem de programação, ou algorítmica – que visam a elaboração do resultado final (algoritmo, ou programa) e (2) registros em língua natural, algébricos e informais – usados na compreensão da situação problema a ser solucionada e na concepção do algoritmo-solução.

A partir da aplicação da teoria de Registros de Representação Semiótica, observando o aprendizado dos alunos e analisando tais registros; pressupõe-se que é possível aplicar recursos do *design* instrucional com vista a projetar materiais de ensino adequados às citadas disciplinas visando à superação das supracitadas dificuldades.

O *design* instrucional se distingue de outras formas de criação de materiais e projetos educacionais pela aplicação metódica e acurada de processos de análise, planejamento, desenvolvimento e avaliação. Trata-se de uma abordagem sistêmica que pondera múltiplos fatores que afetam e/ou podem ser afetados pela implementação de uma iniciativa de educação ou treinamento.

Segundo Gellervij (2001), as vantagens do *Design* Instrucional incluem: situar o foco do processo de ensino-aprendizagem no aluno; facilitar o desenvolvimento de soluções alternativas às práticas usuais em um determinado campo de ensino; levar à convergência dos objetivos, atividades e avaliações.

Referências Bibliográficas

- ANDRADE J., COX, K. e SILVA, L. Andragogia e Atividades Desafiadoras. Escola Regional Bahia Alagoas Sergipe (ERBASE), Maceió – AL, 2010.
- ANDRADE J., COX, K. e SILVA, L. Duplo Clique? Preciso Treinar!. Escola Regional Bahia Alagoas Sergipe (ERBASE), Maceió – AL, 2007.
- ANTUNES, C. Jogos para a estimulação de múltiplas inteligências, Rio de Janeiro: Vozes, 1998.
- CAROSO J. e COX, K. Validação de Modelo de Desenvolvimento de Software Educativo através de Estudo de Caso Voltado para Educação Financeira, Trabalho de Conclusão de Curso, Universidade Federal de Sergipe, 2010.
- CORRÊA, D. e COX, K. Adaptação do Ambiente de Desenvolvimento Eclipse para o Ensino de Programação Orientada a Objetos. Trabalho de Conclusão de Curso, Departamento de Computação, Universidade Federal de Sergipe, 2010.
- COSTA, A. O Adolescente como Protagonista. Disponível na *Internet*. http://escola2000.globo.com/eac/protagonismojuvenil_agosto em set. 2003
- COSTA NETO, A. e COX, K. Protagonismo Juvenil no Ensino da Computação da Universidade Federal de Sergipe, Escola Regional Bahia Alagoas Sergipe (ERBASE), Salvador – BA, 2004.
- COSTA-SILVA D., LEÃO, R. e COX, K. “*Percep*”: Desenvolvendo a Percepção Musical de Crianças Iniciantes no Processo de Musicalização, Escola Regional Bahia, Alagoas e Sergipe, 2010.
- D’ÁQUINO, Cássia. “Educação Financeira: Como educar seu filho.” 1ª ed. Rio de Janeiro: Elsevier, 2008.
- DELORS, J. Os Quatro Pilares da Educação. Disponível na *Internet*. http://escola2000.globo.com/eac/protagonismojuvenil_agosto, em set. 2003.
- DUVAL, R. Registros de Representações Semióticas e Funcionamento Cognitivo da Compreensão em Matemática. In: MACHACO, S. D. A. (org.). *Aprendizagem em Matemática: Registros de Representação Semiótica*. Campinas, SP: Papirus, 2008
- FREIRE, P. “Medo e Ousadia: O Cotidiano do Professor”. Rio de Janeiro: Paz e Terra, 1986.
- GALVÃO, L. e COX, K. Jogos Educativos de Computador para Estimulação de Inteligências

- Múltiplas. Trabalho de Conclusão de Curso (*Em construção*). Universidade Federal de Sergipe, 2011.
- GARDNER, H. *The mind's new science*. New York, Basic Books Inc., 1987.
- GELLEVIJ, M. *Disciplina 'Principles of Learning and Instructional Design'* Universidade de Twente, 2001.
- GOECKS, R. “Educação de Adultos – Uma Abordagem Andragógica.” Disponível em: < www.andragogia.com.br > Acesso em: 17 nov. 2009.
- KÖLLING, M. *The Design of an Object-Oriented Environment and Language for Teaching*. University of Sydney. Austrália: 1999.
- LARROSSA, J. “Educação, Experiência e Sentido.” Belo Horizonte: Autêntica, 1999.
- MARTINS, C. T. K. *Uma Engenharia Didática para Explorar o Aspecto de Processo Dinâmico Presente nos Algoritmos*. Tese de Doutorado PUC-SP, 2010.
- MENDES, C. L. *Jogos eletrônicos: Diversão, poder e subjetivação*. Papirus. Campinas, SP. 2006.
- MILETTO, E. *Introdução à Computação Musical*. Congresso Brasileiro de Computação, 2004. Itajaí: Acessado em: <<http://www.inf.ufrgs.br/lcm/web/textos/aula1/CMintro2.pdf>>, 2009.
- MOTA, M. *Ambiente Integrado à Plataforma Moodle para Apoio ao Desenvolvimento das Habilidades Iniciais de Programação* Introdução à Computação Musical. Acessado em: <<http://www.br-ie.org/pub/index.php/sbie/article/viewFile/1135/1038>>, 2009.
- PALMEIRA, J., HIROSE R., MESQUITA R. e COX, K. *ClassVision: Sistema Educativo para Invisuais*. Workshop Ensino de Informática da ERBASE (Escola Regional Bahia, Alagoas e Sergipe), 2005.
- PEREIRA, e outros. *Ateliê de Objetos de Aprendizagem: Uma Abordagem para o Ensino de Computação em Cursos Técnicos*. Revista Brasileira de Informática na Educação, V. 18, 2010.
- PERÚCIA, Alexandre. *Desenvolvimento de Jogos Eletrônicos Teoria e Prática*, São Paulo: Novatec, 2007.
- PORTO D. e COX, K. *Um Curso à Distância de Introdução à Programação Usando o LMS Moodle*. Trabalho de Conclusão de Curso, Departamento de Computação, Universidade Federal de Sergipe, 2011.
- SILVA, C. *Ambiente de Educação a Distância Sobre a Linguagem de Programação Java*. Universidade Corporativa Serpro: Disponível na *Internet*. www.tise.cl/archivos/tise2006/20.pdf, em set, acessado em 2010.
- SILVA, S. *A melhoria da Qualidade da Educação na Escola Pública: Desafio ao Uso das TIC*. Estudos IAT, Salvador. V. 1, N. 3, 2010
- SOUZA, P., COX, K., SILVA, L. e FAKHOURI, R. *Uma Proposta de Curso à Distância de Patologia*, Congresso Nacional de Informática Médica, Brasília – Brasil, 2004.
- TAROUCO, L. *Jogos educacionais*. Renote – Novas Tecnologias na Educação, Porto Alegre. V.2. 2011.

Metodologia PETIC – Planejamento Estratégico de TIC

Rogério Patrício Chagas do Nascimento

Resumo. O avanço da tecnologia e o mercado competitivo exigem que as organizações busquem inovações para encontrarem as melhores soluções aos seus problemas, garantia de qualidade e satisfação aos seus clientes. Para corresponder a estas expectativas é necessário que as organizações adotem modelos eficazes de gestão, menos convencionais e capazes de transformar as Tecnologias da Informação e Comunicação (TIC) em serviços de valor para os gestores de negócios, auxiliando as suas tomadas de decisão. Este capítulo faz um apanhado histórico sobre a criação e todo o processo de concepção da Metodologia PETIC pelo Grupo de Pesquisa em Engenharia de *Software* da UFS. O leitor é convidado a conhecer os diversos modelos, *frameworks* e metodologias de Governança de TIC. E a perceber a importância do Planejamento Estratégico (PE) para as organizações. A metodologia aqui apresentada representa um conjunto de normas e diretrizes para a concepção de um PE voltado para os processos de TIC dentro das organizações, abordando os cinco pilares dos Sistemas de Informação (SI), a saber: Dados, *Software*, *Hardware*, Telecomunicações e Pessoas. São também mostrados os principais componentes da PETIC e as diretrizes para a correta construção de cada um deles. Assim, são descritos o Artefato PETIC, o Catálogo de Processos de TIC, os Gráficos de Importância *versus* Custo e os Mapas de Gantt que modelam os processos de TIC em cada pilar dos SI. Além disso, são indicadas as diretrizes para se fazer a análise de Maturidade dos Processos de TIC da organização, o Alinhamento Estratégico com os objetivos organizacionais, a identificação dos Processos de TIC Críticos, bem como a visão geral do *framework* PETIC por meio do *Workflow* de Etapas da Metodologia PETIC.

Palavras-chave: Planejamento Estratégico de TIC (PETIC). Governança de TIC. Sistemas de Informação (SI). Tecnologias da Informação e Comunicação (TIC). Planejamento Estratégico (PE).

1 Introdução

Com as novas necessidades de aplicações e serviços, surgem novas áreas de pesquisa dos Sistemas de Informação (SI) e da Engenharia de *Software* (ES). Os avanços na área de Tecnologias

da Informação e Comunicação (TIC) ocorrem desde o final do século passado e seguem acelerados neste milênio, refletindo mudanças estruturais nas organizações (PILGER, 2007). Atualmente, as organizações que possuem as TIC incorporadas aos seus negócios também devem ter um Planejamento Estratégico (PE) voltado para estas novas áreas informacionais.

Hoje o foco da Tecnologia da Informação (TI) mudou, tanto que o termo TI passou a ser trocado pelo termo TIC, não somente pelo surgimento de novos meios, mas também pelas ferramentas comunicacionais (PILGER, 2007).

Dentro desse novo paradigma, novas ideias como colaboração e Gestão do Conhecimento poderão ser edificadas. É importante enfatizar que nenhuma infraestrutura por si só promoverá a colaboração entre as pessoas, essa atitude faz parte de uma cultura que deverá ser disseminada por toda a organização. É necessária uma grande mudança de paradigma.

Nesse contexto, é evidenciada a importância de uma estratégia que oriente a organização a alcançar seus objetivos. Com isso, cada organização tem que decidir quais estratégias deve seguir na hora de aplicar seu plano. Também deve escolher o que será melhor para ela, levando em conta a sua posição no mercado, os seus objetivos, as oportunidades e os seus recursos disponíveis.

No universo das organizações públicas ou privadas, a busca por novas tecnologias se faz necessário para atender essa nova demanda. Devido, principalmente, ao aumento dos dados e informações gerenciadas, ao número crescente de usuários, bem como à criação de novas unidades descentralizadas, localizadas em outras áreas geográficas.

O Grupo de Pesquisa em Engenharia de *Software* da UFS vem trabalhando desde 2008 num estudo que busca por novas estratégias de gestão que suportem as TIC de forma eficiente e eficaz. Nesse ínterim, percebeu a necessidade de se elaborar um PE para as organizações estudadas. O PE deve ser capaz de proporcionar um menor tempo de resposta às solicitações, presteza, flexibilidade, rapidez e confiabilidade nos processos de TIC da organização e no fornecimento de uma base de dados segura, consistente e volumosa.

Dos vários trabalhos realizados pelo Grupo, abstraíram-se três objetivos genéricos considerados primordiais para as atividades de informática corresponderem às expectativas das organizações: (i) buscar novas políticas e/ou estratégias para administrar as TIC com eficácia e eficiência (ii) viabilizar e integrar os serviços e sistemas que deverão apoiar os processos de negócio (institucionais), (iii) prover atendimento de qualidade aos seus usuários e demais *stakeholders*.

Diante desse cenário, o PE surge como uma valiosa “ferramenta” de auxílio à alta administração, pois permite nortear as ações gerenciais, através de um plano previamente determinado de metas e estratégias. Diminuindo, com isso, a possibilidade de tomada de decisões equivocadas, num mercado extremamente competitivo, sem margem para erro (BARBOSA e BRONDANI, 2005).

Os objetivos do PE devem estar alinhados com “o novo conceito de TIC mais abrangente do que os de processamento de dados, sistema de informação, engenharia de *software*, informática ou o conjunto de *hardware* e *software*, pois também envolve aspectos humanos, administrativos e organizacionais” (KEEN, 1993 apud LAURINDO *et al*, 2001). A finalidade da gestão estratégica, segundo Costa (2007, p. 35), visa assegurar o crescimento, a continuidade e a sobrevivência da instituição por meio da adaptação contínua de sua estratégia, de sua capacitação e de sua estrutura. Permite também que a instituição enfrente as mudanças observadas (ou previsíveis), antecipando-se a elas.

Mintzberg *et al.* (2010, p. 46) afirma “toda mudança estratégica envolve novas experiências, um passo no desconhecido, certa dose de risco”. Por isso, as organizações precisam funcionar não só com as estratégias, mas durante os períodos de formulação desta, até onde o prazo do plano seja estabelecido.

Para Rezende (2008a, p. 49), as organizações públicas ou privadas não atuam nem sobrevivem

sem informações. A necessidade de que as organizações públicas ou privadas sejam inteligentes, frente às mudanças constantes do meio ambiente interno e externo, faz com que as mesmas também se modifiquem e requeiram planejamento das suas informações, auxiliadas pelos sistemas informatizados e recursos de TIC. Segundo Costa (2007 p. 35), “todo plano estratégico precisa ter desafios, bem como os objetivos e metas a serem realizados ao longo do horizonte de tempo e planejamento. É a formalização desses elementos que fornece os alvos a serem alcançados pela organização”.

Com o objetivo de refletir esta nova forma de atuação, nasceu a metodologia denominada Planejamento Estratégico de TIC (PETIC). Este capítulo está dividido em 5 seções. A seção 2 aborda os principais aspectos teóricos e tecnológicos de suporte para a investigação realizada. Ela descreve uma parte da vasta revisão bibliográfica realizada para a fundamentação teórica da metodologia PETIC. A seção 3 descreve os principais componentes da PETIC, sua utilização e as principais diretrizes para a criação de um artefato PETIC e a utilização dos seus componentes. A seção 4 analisa os trabalhos desenvolvidos pelas várias equipes de trabalho e os estudos realizados em várias organizações públicas e privadas do Norte e Nordeste do país. A seção 5 conclui este capítulo tecendo considerações sobre as principais contribuições alcançadas e os possíveis trabalhos futuros.

2 Revisão Bibliográfica e Fundamentação Teórica

Para entender o PE é necessário examinar a estrutura organizacional da instituição que será o foco da questão, as barreiras ou obstáculos a serem enfrentados, quais recursos serão necessários para as devidas alterações, quem serão os agentes que irão tornar possíveis as mudanças e que benefícios o PE trará à organização.

Segundo Lima (2007), “o PE é um processo contínuo de formulação de estratégias organizacionais no qual se busca a inserção da organização e de sua missão no ambiente em que está atuando”.

O PE está relacionado com os objetivos estratégicos de médio e longo prazo que afetam a direção ou a viabilidade da organização. Mas, aplicado isoladamente, é insuficiente [...], é preciso que sejam elaborados de maneira integrada e articulada todos os planos táticos e operacionais, maximizando os resultados e minimizando as deficiências utilizando princípios de maior eficiência, eficácia e efetividade. Eles são os principais critérios de avaliação da gestão (LIMA, 2007).

A elaboração de um PE oferece uma visão de futuro. Ele deve conter a definição do negócio, a missão, a visão, a análise do ambiente interno e externo, a formulação das estratégias e a execução e controle do projeto. A missão deve ser simples e extremamente objetiva. Deve ser compartilhada por todos os dirigentes e funcionários.

A visão é um modelo mental de um estado ou situação altamente desejável, de uma realidade futura possível para a organização (COSTA, 2010, p. 36). Ela deve ser definida de maneira simples, objetiva, abrangente, compreensível para todos, tornando-se, assim, útil e funcional para os envolvidos com a organização.

Os valores são as características, virtudes, atributos, princípios e qualidades da organização que podem ser objeto de avaliação. Também é necessário haver um sentido claro sobre qual a razão da existência da empresa, pois é ela que define o foco, o objetivo e o propósito.

Segundo Mintzberg *et al* (2010 p. 137), “à medida que as organizações crescem [...] um novo papel emerge, então para o líder: o de desenvolver e articular a visão para a organização”. Esta visão é um alvo futuro, uma condição que nunca existiu e que agora é projetada do presente para o futuro da organização.

Rezende (2008b), diz que a organização preliminar do projeto também é chamada de

organização do PE, que deve ser constituída de atividades que devem ser elaboradas antes do efetivo desenvolvimento do projeto.

Ainda, segundo Rezende (2008b), é necessário definir os objetivos e adotar uma metodologia para esse projeto coletivo e participativo. Além dessas atividades, é recomendado entender, estudar, rever, elaborar ou ajustar os demais planejamentos públicos.

Todas as etapas do planejamento deverão ser amplamente discutidas coletivamente, pois se trata de um processo que leva ao estabelecimento de um conjunto coordenado de ação pública, visando um alvo predefinido.

O processo estratégico deve ser um processo de construção de consenso, pois vai atender aos interesses e necessidades de todas as partes envolvidas para que no futuro todos sejam beneficiados. Ele também é uma tentativa constante de aprendizado, competitivo e suscetível às mudanças.

2.1 Governança de TIC

De acordo com Weill e Ross (2006), Governança de TIC é o conjunto de práticas de gestão de TIC que busca auxiliar no balanceamento de riscos e oportunidades em um ambiente altamente dinâmico, otimizando tomadas de decisão, controlando custos, pessoas, contratos, possibilitando aos usuários a garantia da segurança dos serviços fornecidos pelas TIC, bem como especifica os direitos de decisão e responsabilidade na estrutura da organização, a fim de obter o uso desejável das TIC.

Segundo Fernandes e Abreu (2008), o principal objetivo da governança de TIC é “alinhar a TI aos requisitos do negócio. Esse alinhamento tem como base a continuidade do negócio, o atendimento às estratégias do negócio e o atendimento a marcos de regulação externos”.

A Governança de TIC vem sendo adotada com a utilização de alguns *frameworks* e modelos. Dentre eles merecem destaque o *Control Objectives for Information and Related Technology* (COBIT), a ISO/IEC 38500 e o VAL IT (ISACA, 2010). Para dar suporte ao uso de tais *frameworks* e a efetiva implantação de Governança de TIC, vários outros modelos são usados como mecanismos complementares.

Assim, os *frameworks* de governança (COBIT, ISO/IEC 38500 ou VAL IT) englobam outros modelos. Isso ocorre tendo em vista que o *framework* preocupa-se apenas com o que será implementado. A forma pela qual será realizada a gestão de serviços, o desenvolvimento de aplicações, a segurança da informação, dentre outros, será definida por modelos como o *Information Technology Infrastructure Library* (ITIL), ISO 20000, *Capability Maturity Model* (CMM) e ISO 27001. Estes modelos (ou mecanismos complementares) visam auxiliar o gestor na implementação efetiva da Governança de TIC.

O COBIT é um *framework* para gestão de TI focado principalmente no negócio da organização e nos processos de TIC que dão suporte a ela. O COBIT é uma opção para maximizar o Retorno Sobre Investimento (ROI) das TIC, uma vez que sua preocupação é fornecer uma métrica dos processos da organização.

O COBIT apresenta quatro domínios onde se encontram 34 processos (ISACA, 2010). Cada um desses processos pode ser analisado primordialmente de três maneiras: (i) Indicadores Chave de Desempenho, (ii) Indicadores Chave Objetivo ou de Sucesso e (iii) Fatores Críticos de Sucesso. Esses indicadores servem como métricas para definição de metas a serem atingidas. Cada um apresenta uma abordagem diferente para a situação.

O CMM pode ser descrito como um conjunto de práticas para a melhoria de processos existentes dentro de uma organização. A maior contribuição do CMM para este trabalho são os conceitos sobre os níveis de maturidade. No CMM são definidos cinco níveis de maturidade (SEI, 2009): (i) inicial, (ii) repetível, (iii) definido, (iv) gerenciado e (v) otimizado. O CMM

explica como definir o nível em que cada processo se encontra. Outra abordagem interessante é a definição de processos modelos. Estes servem como referência para e melhoria dos processos da organização.

A *Balanced Scorecard* (BSC) é uma metodologia de gerenciamento de desempenho organizacional. Visa dar suporte à estratégia e à visão organizacional. O principal objetivo da BSC é o alinhamento do planejamento estratégico com os processos existentes na organização. Esse objetivo é atingido a partir de 4 ações: (i) Esclarecer e traduzir a visão e a estratégia; (ii) Comunicar e associar objetivos e medidas estratégicos; (iii) Planejar, estabelecer metas e alinhar iniciativas estratégicas; e (iv) Melhorar o *feedback* e o aprendizado estratégico.

Segundo Kaplan e Norton (1997), a BSC é uma ferramenta de apoio usada para alinhar atividades de negócios à visão e estratégia da organização, melhorar comunicações internas e externas, e monitorar o desempenho da organização em relação às metas estratégicas definidas. É um sistema de gestão baseado em um conjunto de indicadores de desempenho que proporcionam aos gerentes uma visão rápida e abrangente de toda a empresa. Os componentes do BSC mais relevantes para esse trabalho são: (i) Objetivo estratégico, (ii) Metas e (iii) Planos de ação (BSCI, 2010).

A *Information Technology Balanced Scorecard* (IT BSC) é derivada do BSC, sendo projetada especificamente para tratar questões de TI (CRAM, 2007). A IT BSC pode ser especificamente customizada visando gerenciar necessidades da Governança de TIC. Segundo a metodologia IT BSC, além da definição das perspectivas, é necessário definir um conjunto de objetivos estratégicos que dará suporte a estas perspectivas, e, também, relacioná-los por meio do Mapa Estratégico. Além disso, métricas são identificadas, indicadores e metas são estabelecidos, e *benchmarks* e *baselines* são desenvolvidos (ISM, 2010).

O Guia do Conhecimento em Gerenciamento de Projetos (Guia PMBOK) tem como objetivo fornecer uma visão geral do que é amplamente reconhecido como boa prática dentro do conjunto de conhecimentos em gerenciamento de projetos (PMI, 2008). Os conhecimentos e práticas descritos nele são aplicáveis à maioria dos projetos na maior parte do tempo, existindo um consenso geral em relação aos valores e utilidades destes. A aplicação correta destas habilidades, ferramentas e técnicas podem aumentar as chances de sucesso em vários projetos diferentes. Este guia também fornece um vocabulário comum para as práticas de gerenciamento de projetos. O PMBOK define 9 áreas e 42 processos para serem analisadas.

2.2 Planejamento Estratégico de TI

O avanço da tecnologia e o mercado competitivo exigem que as organizações busquem inovações para encontrarem as melhores soluções aos seus problemas, garantia de qualidade e satisfação aos seus clientes, em um tempo menor de resposta e com custo inferior ao mercado. Essas exigências forçam as organizações a reverem seus valores de negócios, humanos e tecnológicos. Esses valores devem envolver toda a organização, principalmente se ela vai adotar as TIC para atingir seus objetivos e metas.

Para corresponder a estas expectativas será necessário adotar modelos eficazes de gestão, menos convencionais e capazes de transformar as TIC em serviços de valor para os gestores de negócios, auxiliando as suas tomadas de decisão.

Todas as TIC devem ser configuradas para atender às estratégias e às políticas da unidade de TI e, principalmente, aos SI e aos sistemas de conhecimento propostos para a organização (REZENDE, 2008a, p. 67).

Para O'Brien (2004), o planejamento de SI e de TI é o processo de identificação das aplicações baseadas em computadores para apoiar a organização na execução do seu plano de negócio e na realização dos seus objetivos organizacionais.

A análise organizacional, seus pontos fortes e fracos, suas ameaças e fraquezas são estudadas na organização para identificação de oportunidades, riscos existentes e potenciais, além dos recursos a serem aplicados.

Esses valores merecem atenção especial com análise estratégica e planejamento efetivo, envolvendo toda a organização, principalmente no que tange à adoção das TIC e respectivos recursos com o objetivo de alcançar as metas organizacionais.

A metodologia PETIC acrescenta novas diretrizes para a análise do cenário atual das organizações. E, ainda, adiciona mais uma grande área dos SI: a Telecomunicação, que representa o “C” de “Comunicação” na sigla PETIC.

2.2.1 Outras Iniciativas de Planejamento Estratégico

Segundo Rezende (2008b), uma metodologia de planejamento estratégico pode se constituir em uma abordagem organizada para alcançar o sucesso da organização por meio de passos preestabelecidos.

As principais metodologias de planejamento de TIC encontradas na revisão bibliográfica foram: *Business Systems Planning* (BSP), Planejamento apoiado no Conhecimento (PAC), Fatores Críticos de Sucesso (FCS), Método de Análise e Solução de Problema (MASP), Wetherbe, Porter/Millar, Planejamento de Cenários, Análise da Cadeia de Valor e *Business Process Redesign* (BPR).

A BSP foca nos recursos disponíveis para a implementação das TIC pela organização. Apresenta quatro elementos de base para o planejamento: processos, organização, sistemas e dados. O comportamento organizacional é interpretado como adesão da alta administração e não como mudanças de valores e comprometimento por parte dos usuários (BRODBECK, 2001).

O PAC é baseado em uma análise objetiva das necessidades e prioridades empresariais (BRODBECK, 2001). Já os FCS centram-se na definição daquilo que é imprescindível para a organização atingir: seus objetivos. O planejamento é mais focado na resolução de problemas prioritários (ROCKART, 1979). A Engenharia da Informação é semelhante ao BSP e FCS, mas enfatiza que o planejamento deve começar pelos dados e não pelos processos (BRODBECK, 2001).

O MASP propõe a divisão de problemas maiores em menores. Utiliza-se de técnicas de análise causal, onde os problemas são os efeitos e se buscam as causas (TERNER, 2008). A Wetherbe consiste de 4 etapas principais: planejamento estratégico de TI, análise de necessidades de informação, alocação de recursos e planejamento de projeto (WETHERBE, 2002).

A Porter/Millar consiste nas seguintes atividades: avaliar intensidade de informação, determinar o papel da TI, identificar e classificar possibilidades de criação de vantagem competitiva pela TI, pesquisar perspectivas de TI e desenvolver plano de TI (PORTER, 1989). O Planejamento de Cenários é indicado quando há muitas incertezas. Compreende identificar cenários e eventos futuros, probabilidades e impactos, cenários alternativos e definição de SI de apoio à alternativa escolhida (HEIJDEN, 1997).

A BPR refere-se a uma inovação total na estrutura organizacional e na forma de conduzir o negócio, baseada em mudanças radicais, absolutas, revolucionárias ou incrementais. Compreendem uma proliferação de metodologias, técnicas de gestão e implementação para atender o negócio como um todo – intrafuncional, interfuncional e interorganizacional (BRODBECK, 2001; HAMMER; CHAMPY, 1993).

Várias outras metodologias foram concebidas, tais como a *Tactical Systems Planning* (TSP), a *Account Planning Extended* (APX), a *Business Information Analyses and Integration Technique* (BIAIT) e a *Business Information Characterization Study* (BICS). Dentre as iniciativas mais recentes, destacam-se o PE com ênfase em conhecimento (BERMEJO, 2009), Práxis (AMARAL; VAREJÃO, 2007), Lutchen (2004), e *Business IT Strategic Planning* (MINGAIL,

2006).

É importante destacar que cada metodologia foca realidades ou problemas específicos. A aplicação da metodologia deve ocorrer de acordo com as necessidades e o contexto da organização. Todas elas apresentam vantagens e desvantagens. E, mais importante, em sua grande maioria, não focam as especificidades das TIC.

2.3 A Concepção da Metodologia PETIC

Após esta breve apresentação sobre os diversos *frameworks*, modelos e metodologias de Governança de TIC que pudessem suportar um PE nas áreas de TIC, a PETIC foi concebida como uma alternativa bastante adequada na implementação de um PE de TIC em organizações públicas ou privadas de qualquer porte.

Dentre os ensejos elencados que motivou a concepção da metodologia PETIC, os mais importantes foram:

- (i) A descoberta da necessidade de um guia simplificado que considerasse as limitações financeiras e de pessoal nas pequenas e médias organizações, sejam elas públicas ou privadas.
- (ii) A identificação da ausência de um artefato de planejamento estratégico em TIC nas pequenas e médias empresas, especialmente as públicas.
- (iii) A detecção de uma grande demanda na utilização de uma metodologia que fosse de fácil aplicação e não fosse tão complexa como as outras metodologias estudadas.
- (iv) O interesse em se criar uma metodologia que não necessitasse de formação prévia ou que os consultores não tivessem que fazer uma certificação específica para utilizá-la.
- (v) Iniciativas como a CNJ (2009) que obrigou todos os seus órgãos a estabelecerem um PE de suas ações.

3 Componentes da Metodologia PETIC

A metodologia PETIC representa um conjunto de normas e diretrizes para a concepção de um PE voltado para as TIC dentro das organizações. A PETIC aborda as cinco grandes áreas (ou os cinco pilares) dos SI, a saber: (i) Dados, (ii) *Software*, (iii) *Hardware*, (iv) Telecomunicações e (v) Pessoas.

Todos os componentes da metodologia PETIC foram concebidos de maneira modular e aberta, visando facilitar o seu uso incremental, por organizações de qualquer tamanho. Dessa forma, a metodologia PETIC pode produzir resultados robustos e sob medida para cada tipo de organização.

A denominação PETIC surgiu a partir de um trabalho acadêmico realizado por um grupo de alunos numa disciplina optativa do DCOMP da UFS, em meados de 2008. Naquele trabalho, foi criado um PE para ser aplicado no estudo de caso escolhido: o CPD da UFS. Posteriormente, este trabalho foi concluído na forma de um Trabalho de Conclusão de Curso (TCC) (SILVA, 2009). Este trabalho teve como propósito auxiliar o CPD da UFS a atingir suas metas através do planejamento estruturado de suas TIC, produzindo mais, com menos esforço.

Antes disso, no entanto, o autor já vinha pesquisando a respeito de Governança de TIC e ferramentas de e-Government (Governo Eletrônico), quando o autor ministrava a disciplina Sistemas de Informações Empresariais no Departamento de Engenharia Eletrônica e Informática (DEEI), da Universidade do Algarve em Portugal. Nessa época, um grupo de alunos do DEEI destacou em seus trabalhos acadêmicos o Planejamento Estratégico de Tecnologias da Informação (PETI) em Rezende (2008b). Desde então, o autor vem coordenando trabalhos acadêmicos nessa área.

Na primeira versão da metodologia PETIC, foram utilizados alguns conceitos e ferramentas extraídas do PMBOK e do IT BSC. Essa primeira versão, depois batizada de PETIC 1.0, foi elaborada e testada pelo período de dois anos no DCOMP da UFS.

Na seqüência, foi construída a versão PETIC 2.0, com o apoio de alunos do curso de Especialização em Gestão de Projetos de Tecnologia da Informação da UFS, em colaboração com alunos de PIBIC e PIBITI vinculados a projetos de pesquisa aprovados para os períodos 2009/2010 e 2010/2011.

A seguir, serão descritos os principais componentes da PETIC: o (i) Artefato PETIC, (ii) o Catálogo de Processos de TIC, (iii) os Gráficos de Importância *versus* Custos e os (iv) Mapas de Gantt dos pilares de SI. De maneira análoga, são discutidas as diretrizes para a correta construção de cada um desses componentes e, também, as diretrizes para se fazer (v) a análise de Maturidade dos Processos de TIC, (vi) o alinhamento estratégico com os objetivos organizacionais, (vii) a identificação dos Processos de TIC Críticos e, por fim, (viii) o *Workflow* das Etapas da Metodologia PETIC que mostra a seqüência e o relacionamento entre os principais elementos do *framework* PETIC.

3.1 O Artefato PETIC

Para a produção do artefato PETIC de uma organização sugere-se a criação de um grupo de trabalho composto por cinco pessoas (ou cinco equipes de trabalho). Eles podem ser divididos para analisar os processos de TIC dos cinco pilares principais de SI (dados, pessoas, telecomunicações, *hardware* e *software*) e propor melhorias aos mesmos. Cada membro (ou equipe de trabalho) do grupo deve ficar responsável por um dos pilares. Estes pilares são os responsáveis pela execução das atividades de entrada, processamento, saída, armazenamento e controle de dados e, em seguida, a conseqüente transformação desses dados em produtos de informação.

O primeiro passo importante na criação do artefato é descobrir a missão, a visão e os valores do setor/unidade de TIC da organização, assim como a análise do ambiente, a razão da sua existência, as atribuições de cada gerente, do diretor e até dos secretários. Pode-se fazer isso a partir da análise documental existente ou de entrevistas *in loco* aos responsáveis por cada pilar de SI dentro da organização.

A Figura 1 mostra o Artefato PETIC dividido em duas partes, sendo a primeira, a apresentação da organização, onde são mostradas:

- (i) A parte formal da organização (ramo, atividades, estrutura organizacional, missão e visão).
- (ii) O estado da arte em TIC (descrição das novas tecnologias que serão aplicadas no cenário desejado das cinco grandes áreas).
- (iii) Cenário desejado para os processos de TIC na Organização (resumo do alinhamento do estado da arte em TIC com os pilares de SI da organização).
- (iv) A metodologia de análise do PETIC (nomes e cargos das pessoas envolvidas e a metodologia de análise das informações que foram aplicadas).

A segunda parte é o desenvolvimento, onde são descritos o cenário atual, a descrição do cenário desejado e as prioridades, custos e execução de cada atividade planejada. As informações levantadas referem-se aos pilares de SI:

- (i) Dados (banco de dados, *backup*, redundância, *data warehouse* e segurança).
- (ii) *Hardware* (*desktops*, *notebooks*, *no-breaks*, *datashows*, *benchmark*, proteção contra sinistros, aquisição e suporte).
- (iii) Gestão de pessoas (contratação, treinamentos, salários, ergonomia, cargos, avaliação de desempenho e segurança no local de trabalho).

- (iv) *Software* (sistemas operacionais, desenvolvimento, aquisição e *frameworks* e métodos de acesso).
- (v) Telecomunicações (*VoIP*, *backbones*, pontos de acesso, arquitetura e segurança da rede, rádio, tv e *blogs*).

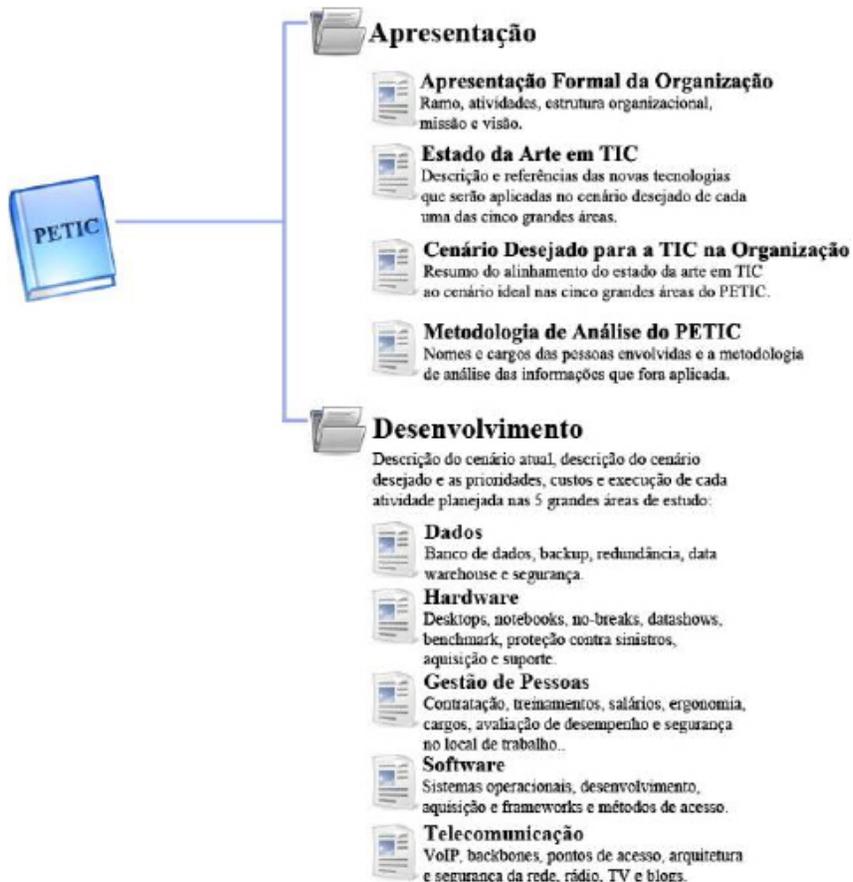


Figura 1: O Artefato PETIC.
Fonte: PINA & PALMEIRA (2010).

No segundo passo para a criação do artefato PETIC, a principal atividade do grupo é descobrir os objetivos estratégicos da organização dentro do escopo do período estabelecido para o artefato. Normalmente, os PE são realizados a cada 3, 4 ou 5 anos. Portanto, pode-se criar um artefato PETIC com duração de 3, 4 ou 5 anos. No entanto, anualmente todo o artefato deve ser revisado para se adequar as atividades de acordo com as mudanças, evoluções ou eliminação de ações previstas para aquele ano. Reestruturando, assim, todas as ações previstas para os anos subsequentes.

Para construir um artefato PETIC também é necessário saber a missão, a visão e os valores do setor/unidade de TIC da organização. Assim como, o motivo da existência da organização, o que ela faz, quem são os responsáveis, quem atende e de que forma atua em seu nicho de negócio.

A partir daí, é feito o levantamento de todos os processos de TIC da organização, identificado o nível de maturidade de cada um deles, os objetivos da empresa, as atribuições dos responsáveis pelos pilares dos SI e os pontos fortes e fracos.

3.2 Catálogo de Processos de TIC

A tabela 1 destaca a indexação dos processos de TIC. Por meio do “ID” de cada processo de TIC, pode-se facilmente identificar o sub-processo, seu sub-campo e a qual pilar de SI o processo pertence.

Tabela 1: Catálogo de Processos de TIC.

| ID | | | | |
|--------------|---------------|------------------------|------------|-----------|
| 1. | DADOS | Processo de TIC | Maturidade | Objetivos |
| 1.1 | Armazenamento | | | |
| 1.1.1 | | Banco de dados | | |

Fonte: MARCHI (2010c).

Com as informações do ID de cada processo de TIC bem indexadas, poder-se-á facilitar a identificação e a criação de novos processos. Inclusive, de forma automatizada como se pretende fazer nas próximas versões da PETIC. Assim, promove-se também a legibilidade das informações contidas no artefato PETIC.

O Catálogo de Processos de TIC vem sendo atualizado e incrementado à medida que estão sendo desenvolvidos os estudos de caso em Sergipe e no Amazonas. Dessa maneira, quando se inicia a concepção do artefato PETIC de uma organização, esta começa a analisar um conjunto de processos de TIC já amplamente conhecidos em outras organizações. Em se identificando um novo processo de TIC, este será adicionado ao Catálogo de Processos de TIC e, eventualmente, será utilizado no futuro por novas organizações que venham utilizar a metodologia PETIC.

3.3 Análise da Maturidade dos Processos de TIC

Na PETIC, a maturidade de cada um dos processos de TIC da organização é analisada tomando-se como base os níveis de maturidade semelhantes ao CMM (SEI, 2009), um dos modelos de Governança de TIC apresentados que contém práticas para a melhoria de processos. Dessa forma, a PETIC permite mostrar a atual situação dos processos de TIC da organização por meio de um valor atribuído a cada processo.

Tabela 2: Índices de Maturidade dos Processos de TIC da PETIC.

| Maturidade | Definição |
|------------|-----------------------|
| 0 | Processo Inexistente |
| 1 | Processo Mínimo |
| 2 | Processo Seguro |
| 3 | Processo Satisfatório |
| 4 | Processo Ideal |

Fonte: autor.

Convencionou-se que o valor do índice de maturidade compreenderia o intervalo de (0) zero a (4) quatro, com os respectivos significados informados na tabela 2.

Para se determinar em qual nível de maturidade o processo de TIC se encontra, é simples. Basta realizar as perguntas descritas na tabela 3 e verificar quais são as respostas acerca daquele processo.

Quando todas as respostas forem “sim”, o processo de TIC corresponderá àquele nível de maturidade. No caso de haver uma única resposta “não”, significa que o processo de TIC possui índice de maturidade com valor igual ao do nível anterior ao que foi feito a pergunta.

Por meio desse questionário, os consultores podem inferir a maturidade de cada processo de TIC identificado na organização, de acordo com os índices definidos na tabela 2.

Tabela 3: Questionário de Maturidade dos Processos de TIC.

| Nível 1 – Mínimo | Sim | Não |
|---|------------|------------|
| O processo existe? | | |
| Funciona mesmo que de maneira instável? | | |
| O processo é utilizado? | | |
| Existe interesse no processo? | | |
| Nível 2- Seguro | Sim | Não |
| Existe plano de contingência? | | |
| O processo pode ser considerado robusto? | | |
| O processo oferece riscos a outras áreas? | | |
| Em caso de falha o processo pode ser resgatado? | | |
| Existe documentação do processo? | | |
| Nível 3-Satisfatório | Sim | Não |
| Atende a necessidade da empresa? | | |
| Agrega valor à empresa? | | |
| É bem conhecido e utilizado por todos? | | |
| Cumpe o que promete realizar? | | |
| Está alinhado com os objetivos da empresa? | | |
| Nível 4- Estado ideal do processo. | Sim | Não |
| O processo é a melhor solução em custo/benefício? | | |
| É o mais atual? | | |
| É a melhor resposta para o problema? | | |

Fonte: Marchi (2010c).

Por exemplo, na tabela 4, os níveis de maturidade “4” mostrados nos dois primeiros processos são considerados ideais, enquanto que nos processos com ID “2.2.2” e “2.3.2” possuem maturidade satisfatória (“3”) e os demais processos são inexistentes.

Depois que todos os níveis de maturidade de todos os processos de TIC identificados forem aferidos é feito o alinhamento desses processos aos objetivos do setor/unidade de TIC da organização, como será visto na seção a seguir.

Tabela 4: Exemplo dos Índices de Maturidade.

| ID | Subárea | Processo de TIC | Maturidade |
|-------|-------------------------|---------------------------------------|------------|
| 2.1.1 | Compras | Política de aquisição de equipamentos | 4 |
| 2.2.1 | Voz | Voip | 4 |
| 2.2.2 | | Utilização de e-mail | 3 |
| 2.3.1 | Segurança e privacidade | Encriptação | 0 |
| 2.3.2 | | Integração de sistemas/informações | 3 |
| 2.4.1 | Bem estar | Acessibilidade | 0 |

Fonte: Lira (2011).

3.4 Alinhamento Estratégico com os Objetivos Organizacionais

O entendimento de alinhamento estratégico deve ser ampliado para além do aspecto conceitual, sendo utilizado como uma ferramenta de monitoramento e gestão das estratégias e objetivos da organização. Para agregar valor ao negócio é necessário que o alinhamento entre o planejamento de TIC e o PE da organização esteja de comum acordo, sempre buscando melhorias.

O alinhamento estratégico é um processo evolucionário e dinâmico, que requer forte apoio da alta gestão das organizações, associados às boas relações de trabalho, liderança forte, priorização adequada, confiança e efetiva comunicação, além do correto entendimento do ambiente de negócios, BENZ (2008 apud Pina & Palmeira, 2010).

O alinhamento entre as TIC e o negócio deve ser atingindo conjuntamente com o PE da instituição para que a missão, a visão, os objetivos e as metas sejam alcançados. No artefato PETIC, o alinhamento é explicitado por meio da associação do processo com um dos objetivos da organização, como pode ser visto na tabela 5.

Tabela 5: Exemplo de Alinhamento Organizacional na PETIC.

| Subárea | Processo de TIC | Maturidade | Objetivos |
|-------------------------|----------------------|------------|-----------|
| Voz | VoIP | 0 | 3 |
| | Utilização de e-mail | 3 | 1,2,3 |
| Segurança e Privacidade | Encriptação | 1 | 12 |
| Bem Estar | Acessibilidade | 0 | - |

Fonte: Lira (2011).

Na coluna “Objetivos”, os números “1”, “2”, “3” e “12” representam os números dos objetivos que foram previamente listados e numerados. E o símbolo “-” significa que nenhum objetivo da organização está associado diretamente àquele processo de TIC analisado.

3.5 Relevância dos Processos: identificação dos Processos de TIC Críticos e Prioritários

Definiu-se como Processos de TIC Prioritários os processos que dão suporte a um determinado processo de negócio considerado prioritário para o topo da organização. Normalmente, esses processos fazem parte das metas a alcançar naquele ano, ou dão suporte ao novo plano diretor da organização. São independentes de justificativas técnicas e normalmente estão atrelados a uma nova política adotada pela alta administração.

Por outro lado, Processos de TIC Críticos são definidos como processos problemáticos que necessitam de ações de melhorias com mais urgência, para que possam atender aos objetivos da organização.

Na PETIC convencionou-se que os Processos de TIC Críticos são os processos de TIC observados com índice de maturidade menor que 03 (três), segundo a tabela 2 mostrada anteriormente. Ou seja, os processos com índices 0, 1 e 2 (definidos respectivamente como inexistente, mínimo e seguro) são considerados críticos. Por exemplo, na mesma tabela 5 da sub-seção anterior, os processos “VoIP”, “Encriptação” e “Acessibilidade” seriam os processos críticos identificados.

Vários processos críticos podem ser identificados durante a criação dos artefatos PETIC. Nesses casos, o Gestor de TIC deve definir estratégias (ou ações) que possam ser executadas num determinado período de tempo para melhorar a maturidade do processo.

Na tabela 6, o ID “1.1.1.A” refere-se ao identificador da ação “Definição da política de compra”. Recebe a numeração que foi atribuída ao processo de TIC no Catálogo de Processos, seguido de

uma letra. Letras diferentes são acrescentadas para identificar quando mais de uma ação paralela se destine a um mesmo processo. Assim, como a ação “Documentação da política de compra” é uma segunda ação associada ao mesmo processo, esta é identificada por “1.1.1.B”. Caso uma ação necessite de outra ação de forma sequencial, convencionou-se que ao invés de se adotar uma letra diferente, deve-se enumerar a ordem da sequência das ações. Por exemplo, se os dois processos acima fossem sequencias seriam então definidos como “1.1.1.A1” e “1.1.1.A2”.

Tabela 6: Exemplos de ações de melhoria que podem ser definidas para dar suporte a um Processo de TIC Crítico. Neste caso, o processo cujo ID é “1.1.1”.

| ID | Ação | Descrição | Responsável (is) | Custo | Esforço | (...) |
|---------|---|---|--------------------|-------|---------|-------|
| 1.1.1.A | Definição da política de compra | Definir o processo de compra de equipamentos | Gerente de suporte | 0 | 8 h | |
| 1.1.1.B | Documentação da política de compra | Documentar o processo de compra de equipamentos | Estagiário Sênior | 0 | 4h | |

Fonte: Marchi (2010b).

A coluna “Ação” designa o título da ação. E a coluna “Descrição” descreve o que deve ser executado para a ação ser bem sucedida. O(s) “Responsável(is)” é(são) a(s) pessoa(s) ou órgão(s) responsável(is) pela execução da ação. O “Custo” é o valor da ação em termos monetários. E o “Esforço” é a quantidade de horas/pessoa que precisam ser designadas para a ação ser executada.

O símbolo “(…)” na última coluna da tabela 6 representa outros campos desejáveis para as ações de melhoria dos processos de TIC, a saber: Data de Início, Data Provável de Conclusão, e Importância do Processo de TIC.

3.6 Workflow das Etapas da Metodologia PETIC

A principal contribuição deste *workflow* é reconhecer o momento ideal de se discutir os resultados com os demais *stakeholders*. Seguindo corretamente este *workflow*, o fluxo de trabalho proposto pela Metodologia PETIC passa a ser interativo e engloba o que ocorre na prática durante a execução das atividades. Dessa maneira, verificou-se que se aumenta a taxa de aceitação por parte dos *stakeholders* garantindo que o Artefato PETIC final esteja alinhado com a realidade da empresa.

O *workflow* descrito na Figura 2 mostra, de forma modular, as principais diretrizes que devem ser seguidas para a criação do Artefato PETIC, a sua implantação numa organização e como deve ser feita a manutenção desse artefato. Como as etapas são modulares, caso o prazo seja emergencial, pode-se definir uma equipe maior para acelerar o trabalho em cada etapa.

Os primeiros elementos a serem observados são os objetivos do setor/unidade de TIC da organização. Depois é iniciada a construção do Artefato PETIC da organização tomando-se como base o Catálogo de Processos de TIC legado da PETIC. Para cada Processo de TIC da organização, listam-se os objetivos associados a ele. Em seguida, é realizada a Definição dos Níveis de Maturidade de todos os Processos de TIC da empresa. Por definição, convencionou-se que os Processos de TIC com maturidade < 3 serão considerados Processos de TIC Críticos para a organização.

Na etapa seguinte, identifica-se a Relevância dos Processos de TIC, ou seja, qual a importância de cada processo para se atingir os objetivos da organização. Na tentativa de se encontrar a

melhor solução, cria-se um Catálogo de Ações para cada Processo de TIC Crítico ou Prioritário encontrado. Na sequência, os dados do Catálogo de Ações criado são ilustrados em Gráficos de Importância *versus* Custo, de acordo com cada pilar de SI. Depois, são sobrepostos num único gráfico geral onde podem ser vistas todas as ações necessárias para a melhoria de todos os Processos de TIC Críticos ou Prioritários da organização.

Em posse dos Gráficos de Importância *versus* Custo, o Gestor de TIC discute com os demais *stakeholders* envolvidos com o PE da organização qual será a ordem de execução das ações. Caso seja aprovado o Catálogo de Ações delineado inicialmente, segue-se para a formulação dos Gráficos de Gantt. Caso seja reprovado, serão necessárias definições de novas ações para se alcançar melhores soluções. Esse ciclo se repete até a aprovação de todo o Catálogo de Ações.

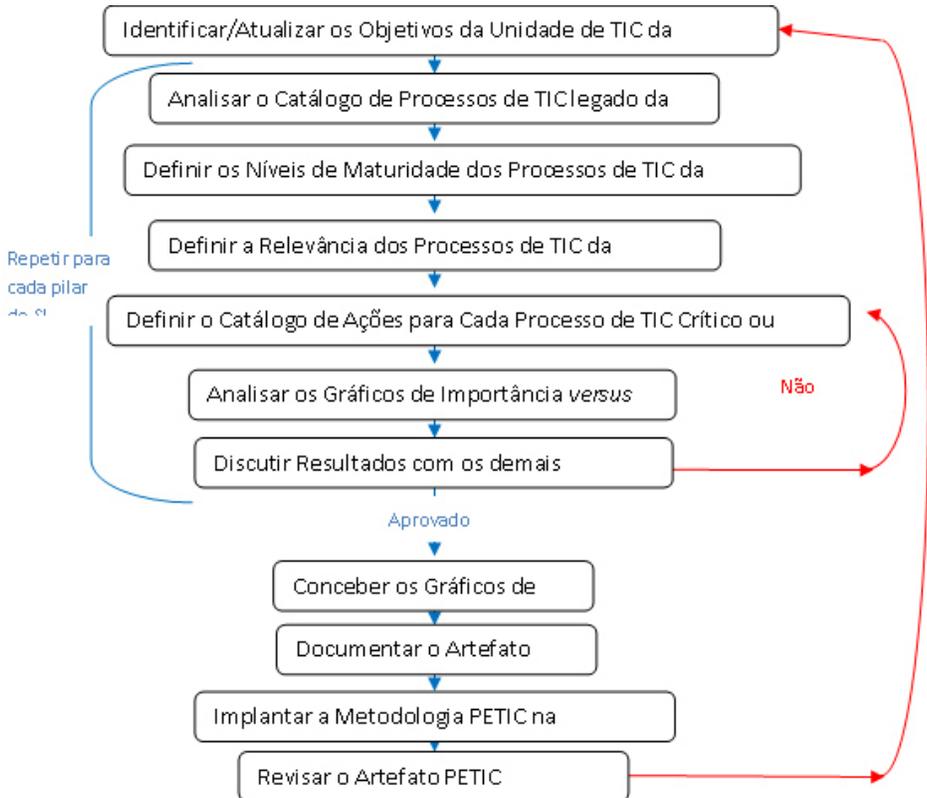


Figura 2: Workflow das Etapas da Metodologia PETIC.

Fonte: Marchi (2011).

Dessa forma, os Gráficos de Importância *versus* Custo auxiliam os gestores nas tomadas de decisão e na construção dos Mapas de Gantt pertinentes a cada pilar de SI.

Com a concepção dos Gráficos de Gantt, finaliza-se a Documentação de Artefato PETIC para aquela organização. Conforme mostrado anteriormente, o artefato será composto por todos os componentes criados durante o *workflow* citado na Figura 2.

Com o Artefato PETIC em mãos, o Gestor de TIC ganha uma ferramenta para monitoração e controle das ações de melhoria necessárias ao bom funcionamento das TIC em sua organização. Como todo documento de PE, o Artefato PETIC da organização deve ser atualizado anualmente, de acordo com as mudanças internas e externas à organização. O processo de revisão deve

verificar se os níveis de maturidade, processos críticos e prioritários continuam os mesmos. Também é verificado se as ações devem ser modificadas, ou, até mesmo, se novas ações devem ser criadas para dar suporte à nova realidade em cada ano. Em suma, todo o *workflow* deve ser retomado a partir da identificação/atualização dos objetivos organizacionais.

3.7 Gráficos de Importância *versus* Custos

Este gráfico permite fazer ilações sobre as prioridades de execução e avaliações sobre quais metas podem ser seguidas com base na importância e no custo. O Gráfico de Importância *versus* Custo ajuda o gestor a tomar a decisão sobre quais ações serão realizadas primeiro, pois ele consegue ter a uma visão holística de todas as ações, seus custos de implementação e sua importância para a implementação de melhorias nos processos de TIC críticos definidos anteriormente.

Tabela 7: Catálogo de Ações.

| ID | Processo | Ação | Responsável | Custo / esforço |
|-----------|-------------------------|--|---|-----------------|
| 2.1.2.8.A | Segurança e Privacidade | Pesquisar <i>Software</i> de Encriptação | Gerente de Infraestrutura de Dados (A) | 4 horas |
| 2.1.2.8.B | Segurança e Privacidade | Licitair <i>Software</i> de Encriptação | Gerente de Infraestrutura da Dados (A) | 720 horas |
| 2.3.2.2 | Voip | Configurar o sistema de telefonia via Voip para melhorar a comunicação | Gerente de Infraestrutura de Telecomunicações (B) | R\$1.134,00 |

Fonte: Lira (2011).

Tomando-se a tabela 7 e a Figura 3 como exemplo, a ação “2.1.2.8.A” precisa de 4 horas e a ação “2.1.2.8.B” necessita de 720 horas para serem executadas. Caberá ao Gestor de TIC, juntamente com os demais *stakeholders*, decidirem quais ações deverão ser executadas primeiramente, tendo em vista a importância e o custo da mesma.

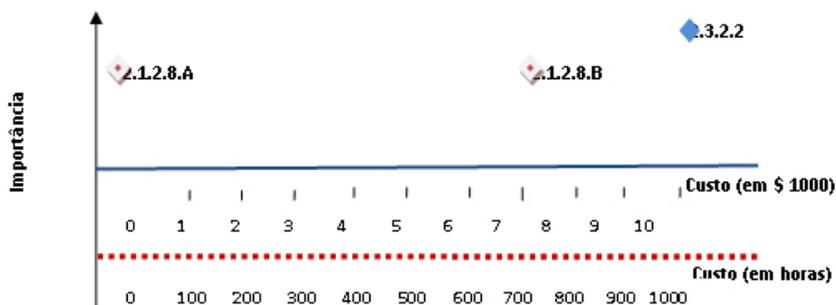


Figura 3: Gráfico de Importância *versus* Custos.

Fonte: Lira(2011).

Dessa forma, as ações e seus custos ficam claras, permitindo uma visão holística da organização e suas necessidades de TIC. A visão holística de uma empresa equivale a se ter uma “imagem única”, sintética de todos os elementos da empresa, que normalmente podem ser relacionados a visões parciais abrangendo suas estratégias, atividades, informações, recursos e organização (estrutura da empresa, cultura organizacional, qualificação do pessoal, assim como

suas interrelações). Uma vez definida a ordem das ações, são elaborados os Mapas de Gantt, como será visto a seguir.

3.8 Mapas de Gantt

O mapa de Gantt é um recurso que auxilia o Gestor de TIC a fazer o acompanhamento e o monitoramento das ações de melhorias dentro dos prazos estabelecidos.

Na tabela 8, é mostrado um exemplo de Mapa de Gantt gerado a partir de um artefato PETIC elaborado no estudo de caso de uma organização pública.

Tabela 8: Exemplo de Mapa de Gantt usado na PETIC.

| ID | Nome da Ação | Esforço ou Custo | Início | Término | Importância | Responsável |
|-----------|--|------------------|--------|---------|-------------|-------------|
| 2.1.2.8.A | Pesquisar Software de Encriptação | 4 horas | 03/12 | 03/12 | 3 | Gerente A |
| 2.1.2.8.B | Licitar Software de Encriptação | 720 horas | 04/12 | 05/12 | 3 | Gerente A |
| 2.3.2.2 | Configurar o sistema de telefonia via Voip para melhorar a comunicação | RS1134,00 | 02/11 | 12/11 | 2 | Gerente B |

Fonte: Lira (2011).

Para cada ação definida, são informados vários dados para a sua realização: o nome da ação, seu esforço ou custo, quando ela será iniciada, para quando está previsto o seu término, qual a sua importância e o responsável pela ação.

Aconselha-se a criação de cinco mapas, cada um referente a um pilar de SI. Quando todos os mapas estiverem prontos, deve-se juntá-los num único diagrama de Gantt de modo que aqui também se possa ter uma visão holística das ações planejadas em todos os pilares.

4 Estudos de Casos e Resultados Alcançados

Nas seguintes subseções são apresentados alguns dos resultados alcançados nos trabalhos desenvolvidos na concepção da Metodologia PETIC sob a supervisão do autor.

4.1 Trabalhos desenvolvidos por meio de Convênio de Cooperação Técnica

Estão sendo elaborados projetos discentes e de extensão que serão observados durante o decorrer do Projeto de Extensão e Colaboração Técnica entre a UFS e a UFAM chamado “*u-Caju: Arquitetura de Suporte a Sistemas Ubíquos que Propiciem Inclusão Digital no Estado de Sergipe*” (jan/2010 a dez/2012), sob a coordenação do autor deste capítulo (Nascimento, 2010).

Numa primeira fase, foram realizados estudos de casos em organizações locais através da elaboração de artefatos PETIC que fornecem uma visão objetiva quer do negócio, quer da própria arquitetura de informação da organização para que a organização possa impulsionar a mobilidade através de sistemas ubíquos, alcançando vantagens competitivas face aos seus concorrentes.

De forma sucinta, alguns objetivos específicos são descritos nas metas a seguir: (i) aplicação do estado da arte em soluções de sistemas ubíquos locais desenvolvidos com a cooperação UFS/UFAM; (ii) realizar definições de processos de organizações locais que sejam escolhidas como

estudos de caso; (iii) analisar como os sistemas ubíquos poderão apoiar estes processos; (iv) propor uma nova arquitetura de informação destas organizações tomadas como estudo de caso, e (v) fazer a análise de gestão da informação destas organizações (NASCIMENTO, 2010).

Dissertação de Mestrado Orientada no Programa de Pós-Graduação em Engenharia de Produção (PEP) da UFAM. Esta dissertação, concluída em dezembro de 2011, apresenta um estudo de caso para o CPD da UFAM por meio da utilização da Metodologia PETIC. Nela são descritos o cenário atual da organização, suas prioridades, custos e ações de melhorias, possibilitando ao gestor, após a concepção do artefato PETIC, acompanhar as atividades por meio dos Mapas de Gantt das grandes áreas de SI e de Gráficos de Importância *versus* Custos das Ações planejadas (LIRA, 2011).

Disciplinas obrigatórias e optativas ministradas na UFAM. Ainda no âmbito das atividades exercidas no Convênio de Cooperação Técnica com a UFAM, o autor orientou estudos de caso da Metodologia PETIC quando ministrou a disciplina obrigatória Planejamento Estratégico em Tecnologias e Comunicação para o Curso Sistemas de Informação e a disciplina optativa Gerência de Projetos para o Curso Ciência da Computação. Para acompanhar os projetos discentes e seus estudos de caso em organizações locais da região metropolitana de Manaus foi criado o blogue educacional “.: Edu-blog > PETIC | UFAM” que pode ser acessado pela URL <http://petic-ufam.blogspot.com/>.

4.2 Trabalhos desenvolvidos in loco na UFS

Todos os trabalhos apresentados a seguir contribuíram na concepção, contínua melhoria e atualização da Metodologia PETIC.

Dissertação de Mestrado Orientada no Programa de Pós-Graduação em Ciência da Computação (PROCC) da UFS. Esta dissertação de Mestrado, em andamento (Palmeira, 2010), mostrará a importância da Governança de TIC e do PETIC. Serão descritas as principais metodologias de Governança de TIC estudadas durante a revisão bibliográfica.

Dentre os trabalhos relacionados já pesquisados, merecem destaque: o Plano Estratégico para a Tecnologia de Informação da Universidade de São Paulo (USP), que utilizou a metodologia proposta em Hax e Majluf, citado em USP (2007); o Planejamento Estratégico de TI do Tribunal Superior Eleitoral (STI/TSE, 2009), que utilizou o IT BSC (CRAM, 2009); o Planejamento Estratégico em TI no Superior Tribunal de Justiça (MORAIS, 2009), que usou o BSC (KAPLAN; NORTON, 1997); e o Planejamento Estratégico de TIC do Poder Judiciário (CNJ, 2009), que usou ferramentas do IT BSC. O objetivo geral da dissertação é aperfeiçoar e formalizar a Metodologia PETIC, mantendo o seu foco na realidade de pequenas e médias organizações, alinhando-a aos principais modelos e *frameworks* de Governança de TIC. Pretende-se fazer uso de ferramentas automatizadas de apoio para suporte à elaboração, implantação, monitoramento e controle do artefato PETIC.

Os objetivos específicos a serem alcançados são: (i) descrever e analisar as principais metodologias de planejamento estratégico de TIC, além dos principais *frameworks* e modelos de Governança de TIC; (ii) analisar ferramentas de apoio à confecção, implantação, monitoramento e controle de planejamento estratégico; (iii) descrever as características e funcionalidades que a ferramenta de apoio deverá possuir para o alcance dos objetivos; (iv) desenvolver protótipo que demonstre as funcionalidades consideradas mais importantes; (v) aplicar a Metodologia PETIC em dois ou mais estudos de caso; e (vi) analisar resultados obtidos.

Monografias da Pós-Graduação em Gestão de Projetos do DCOMP. Mostraram a importância da Governança de TIC e do PE de TIC. Descreveram e analisaram algumas metodologias de planejamento de TIC, por meio de seu estudo e aplicação. Aplicaram o guia PETIC, concebido no DCOMP da UFS, juntamente com a ferramenta IT BSC em dois estudos de

casos: o CPD da UFS e o Tribunal Regional Eleitoral (TRE) de Sergipe. Identificaram fragilidades e melhorias para o guia PETIC. Foram elaborados um artefato baseado no PETIC para o CPD da UFS e outro artefato PETIC para o TRE de Sergipe (PINA e PALMEIRA, 2010).

Plano de Trabalho do Programa Institucional de Iniciação em Desenvolvimento Tecnológico e Inovação (PIBITI). Esse trabalho, em andamento, vem propondo inovações para a metodologia PETIC. Tem abordado as dificuldades encontradas nos estudos de caso e as soluções sugeridas para a modelagem de um novo *framework* de diretrizes.

Dos vários resultados já alcançados, os mais relevantes foram: (i) criação de níveis de maturidade para os processos de TIC; (ii) definição de um plano de trabalho ágil para o PE; (iii) escalabilidade para organizações de qualquer porte; (iv) criação de um guia para uso da metodologia; (v) apresentação do trabalho à comunidade acadêmica e empresarial do estado de Sergipe; (vi) estudos de caso orientados *in loco* em organizações de Sergipe; (vii) implementação de um modelo interativo; (viii) identificação dos processos e ações de maneira indexada, através do Catálogo de Processos de TIC; e a (viii) compilação dos processos de TIC encontrados nos estudos de caso, a qual servirá para alimentar uma base de dados com os possíveis processos de TIC a serem analisados em qualquer organização de maneira genérica. Também descreve o dia-a-dia do desenvolvimento do trabalho necessário para a construção de um artefato PETIC (MARCHI, 2010a; MARCHI, 2010b; MARCHI, 2010c; MARCHI, 2011).

Plano de Trabalho do Programa Institucional de Bolsas de Iniciação Científica (PIBIC). Estuda o alinhamento estratégico de um artefato PETIC com o PDI do DCOMP-UFS 2010/2012. O desenvolvimento de um artefato PETIC para o DCOMP foi o principal objetivo deste trabalho, além de propor melhorias para o desenvolvimento de tal planejamento. Para isto, primeiramente foi elaborado o planejamento para o DCOMP, com o intuito de compor o plano de desenvolvimento institucional (PDI) da UFS. Com base na identificação das necessidades do DCOMP, foi desenvolvido o PETIC que contém o planejamento para as áreas de TIC.

No desenvolvimento do PETIC para o DCOMP foi compilada uma lista de aplicações que se faz necessária para o suporte das atividades do departamento. Tais aplicações estão sendo estudadas em outro projeto chamado Portal de Serviços Acadêmicos (PSA) que está sendo desenvolvido por outros alunos participantes do Grupo de Pesquisa em Engenharia de *Software*. O documento PETIC do DCOMP já está contribuindo para a melhoria das TIC no DCOMP (CASSIMIRO, 2010; CASSIMIRO, 2011).

Trabalho de Conclusão de Curso de Graduação no DCOMP. Apresenta conceitos a respeito das TI e Comunicação. Mostra a importância do tratamento eficaz de informação e conhecimento por meio das TIC. Mostra a necessidade da concepção e execução do PETIC. Apresenta os fundamentos do PETIC e propõe um guia simplificado para a concepção do PETIC de forma a atender as limitações financeiras e de pessoal nas pequenas e médias empresas. Descreve todo o processo de documentação e implantação do guia proposto, o qual foi denominado posteriormente de Guia PETIC 1.0. Apresenta um estudo de caso no CPD da UFS (SILVA, 2009).

Disciplinas obrigatórias e optativas na UFS. Algumas disciplinas sob a regência do autor funcionaram como vetores do trabalho de pesquisa realizado na concepção da Metodologia PETIC. No âmbito da sua lotação no DCOMP como Professor Adjunto da UFS, o autor ministrou as disciplinas Gestão da Informação, Sistemas de Informação, Tópicos Especiais em Sistemas de Informação e Sistemas de Informações Empresariais.

Esta última disciplina foi ministrada na Especialização em Gestão de Projetos do DCOMP. Respectivamente, foram criados blogues educacionais para cada uma delas, a saber: “:: Edu-blog GI 2009 | DCOMP | UFS” (<http://gi-dcomp-2009.blogspot.com/>); “Edu-blog SI 2009 | DCOMP | UFS” (<http://si-dcomp-2009.blogspot.com/>); “:: edu-blog > PETIC 2009” (<http://petic-2009.com/>).

blogspot.com/); “:: edu-blog > Sistemas de Informações Empresariais” (<http://egp-sie.blogspot.com/>). Nestes blogs encontram-se documentos, ligações *Web*, referências e estudos de casos utilizados nas pesquisas relativas à Metodologia PETIC.

5 Conclusões e Trabalhos Futuros

As primeiras contribuições deste capítulo dizem respeito à compilação dos vários modelos, *frameworks* e metodologias usadas na Governança de TIC. Foram vistos conceitos e tecnologias que auxiliam as organizações de qualquer porte a produzirem seus próprios artefatos de planejamento.

Neste sentido, foram apresentadas em detalhes as principais diretrizes nas quais os consultores de TIC das organizações podem se basear para aplicar a Metodologia PETIC em sua própria organização, tornando menos complexa a elaboração de um planejamento de TIC.

Com a produção do artefato PETIC, em alguns estudos de caso constatou-se que a metodologia tem auxiliado na modernização da gestão local das organizações. Com isso, depreende-se que a metodologia PETIC pode servir como modelo para a administração moderna e pode aproximar mais as TIC com as áreas de negócio. Permitindo, também, que as TIC não sejam apenas operacionais, mas passem a ser vistas como um fator estratégico e de auxílio na transformação dos processos de negócio das organizações. Como a PETIC é baseada em processos de TIC, efetivamente, muitas vezes também ocorre uma reengenharia dos próprios processos de negócio da organização, pois permite a visualização de problemas e sugestões de ações de melhorias para os mesmos.

Os processos de TIC identificados empiricamente durante os estudos de caso foram incorporadas ao *framework* PETIC, criando-se assim o Catálogo de Processos de TIC que pode ser usado como referência na criação de outros artefatos PETIC. Deste modo, diminuiu-se o espaço para improvisações. E manteve-se um pensamento de desenvolvimento aberto, pois a PETIC continua a ser uma metodologia para qualquer tamanho de organização.

Neste capítulo também foram documentados todos os trabalhos acadêmicos realizados em Sergipe e no Amazonas acerca da metodologia PETIC. Pôde-se observar a descrição de vários incrementos à metodologia, bem como a sua melhoria contínua.

Atualmente, a metodologia PETIC está caminhando para a sua versão 3.0. Nela, continuam trabalhando alunos de PIBIC e PIBITI da UFS, agora em parceria com alunos de mestrado do PROCC da UFS e do PEP da UFAM.

Ainda em 2011, espera-se que a primeira dissertação de mestrado sobre a PETIC já esteja concluída no PEP da UFAM, por meio do convênio de cooperação técnica UFS/UFAM, sob a coordenação do autor.

Enquanto isso, também se iniciaram outros trabalhos de mestrado no PROCC da UFS que auxiliarão a automatização da concepção de um Artefato PETIC e de algumas tarefas do *Workflow* de Etapas da Metodologia PETIC. O primeiro destes trabalhos, descrito na subseção 4.2, deve ser concluído em 2012.

Talvez a implantação do PE em uma organização seja a fase mais difícil de realizar, dentre todas as etapas da metodologia adotada. Mudanças de hábito, instituição de novas atribuições e atividades, saídas das zonas de conforto e implantação de novos Processos de TIC são apenas algumas das novidades que poderão ser percebidas pelos colaboradores. Pensando nisso, está sendo concebido o segundo trabalho de mestrado no PROCC do DCOMP. Será observado o poder das novas mídias e os novos padrões de interatividade que nortearão a comunicação e o treinamento durante a implantação da Metodologia PETIC numa organização. Espera-se que esta dissertação de mestrado seja concluída em 2013.

Ainda em 2012 pretende-se divulgar a Metodologia PETIC em instituições europeias, por

meio de uma visita técnica à Universitat de València, na Espanha. Com isso, espera-se dar início a uma internacionalização do uso da PETIC. O que poderá motivar novos projetos de colaboração e pesquisa entre instituições estrangeiras e o DCOMP da UFS.

Ao final desses dois primeiros anos da sua criação, a Metodologia PETIC amadureceu consideravelmente, sobretudo no que diz respeito às suas funcionalidades. Espera-se que este amadurecimento a transforme em um guia ainda mais atrativo. Dessa vez, para além dos consultores, que seja principalmente atrativa para as organizações, auxiliando-as de maneira intuitiva nas suas tomadas de decisões.

Adicionalmente, espera-se também que a qualidade dos serviços das organizações melhore com a aplicação da PETIC, provendo redução de custos e de riscos estratégicos, provimento de investimentos e de despesas e proporcionando satisfação a todos os seus usuários ou clientes. Por fim, espera-se que a PETIC fique cada vez mais robusta, mais próxima da realidade das organizações e do efetivo desenvolvimento de um PE.

Por outro lado, almeja-se que o artefato criado nas organizações sirva como um guia para futuras tomadas de decisão e, ao mesmo tempo, que esteja alinhado ao PE da organização.

Neste momento encontra-se em fase de construção o Portal PETIC. Em breve o mesmo estará disponível na URL <http://petic.dcomp.ufs.br/>.

Agradecimentos

Estes resultados não seriam possíveis sem o apoio de algumas instituições e órgãos de fomento, seja através de subsídio financeiro ou apoio institucional. Agradecemos, assim, à EATIS.org Brasil – Associação Euro-Americana de Telemática e Sistemas de Informação, ao CNPq, à CAPES, à FAPITEC/SE e à própria UFS, através dos programas PIBIC e PIBITI de fomento à iniciação científica e à inovação tecnológica.

Referências Bibliográficas

- AMARAL, L.; VARAJÃO, J. *Planejamento de sistemas de informação*. 4 Edição. Lisboa: FCA, 2007.
- BERMEJO, Paulo H. S. *Planejamento Estratégico de Tecnologia da Informação com Ênfase em Conhecimento*. Florianópolis, 2009.
- BARBOSA, E. R; BRONDANI, G. Planejamento estratégico organizacional. *Revista Eletrônica de Contabilidade*. V. 1 dez.2004- [S.I], fev. 2005.
- BRODBECK, Ângela F. *Alinhamento Estratégico entre os Planos de Negócio e de Tecnologia da Informação: um Modelo Operacional para a Implementação*. Porto Alegre, 2001.
- BSCI – Balanced Score Card Institute, *Balanced Scorecard Basics*. Disponível em <<http://www.balancedscorecard.org/BSCRResources/AbouttheBalancedScorecard/tabid/55/Default.aspx>> acessado em 04/04/2010
- CASSIMIRO, Anderson *et al.* *Estudo de Caso do DCOMP/UFS através da Elaboração de um PETIC – Planejamento Estratégico de TIC*. EIC 2010 – Encontro de Iniciação Científica da UFS. São Cristovão, 2010.
- CASSIMIRO, Anderson. *PETIC DCOMP 2010-2014 - Planejamento Estratégico de TIC para o DCOMP*. Relatório Semestral PIBIC 2010-2011. UFS, São Cristovão, 2011.
- CNJ – Conselho Nacional de Justiça. *Resolução nº 99, de 24 de novembro de 2009*. Diário Oficial da União, Brasília, DF, 3 dez. 2009. Seção 1, p. 128-130, 2009.
- COSTA, Eliezer. *Gestão Estratégica da empresa que temos para a empresa que queremos*. 2 ed. São Paulo, 2007.
- CRAM, Alec. *The IT Balanced Scorecard Revisited*. Disponível em: <<http://www.isaca.org/>>

- AMTemplate.cfm?Section=20075&Template=/ContentManagement/ContentDisplay.cfm&ContentID=44791>. Acesso em: 19 out. 2009.
- HAMMER, M.; CHAMPY, J. *Reengenharia: revolucionando a empresa*. São Paulo: Campus, 1993.
- HEIJDEN, K. V. *Scenarios: the art of strategic conversation*. New York: John Wiley & Sons, 1998.
- ISACA. *Control Objectives for Information and related Technology (COBIT)*. Disponível em: <<http://www.isaca.org/Template.cfm?Section=COBIT6&Template=/TaggedPage/TaggedPageDisplay.cfm&TPLID=55&ContentID=31519>> acessado em 28/01/2010
- ISM. *Aligning IT to Organizational Strategy*. Disponível em <http://www.ism-journal.com/ITToday/AU2621_CH04.pdf> acessado em 14/03/2010
- KAPLAN, S R.; NORTON, P. D. *A estratégia em ação: balanced scorecard*. 4 ed. Rio de Janeiro: Campus, 1997.
- LIMA, Ricardo. *Planejamento estratégico alinhado com a tecnologia da da informação*. II Congresso de Pesquisa e Inovação da Rede Norte Nordeste de Educação Tecnológica. Anais. João Pessoa. 2007.
- LIRA, Lúcia M. B., NASCIMENTO, Rogério P. C. do, SAMPER, J J. *Planejamento Estratégico de TIC em um Centro de Processamento de Dado de uma Universidade Brasileira*. Revista Colombiana de Computación, v.11, n. 2, 2010.
- LAURINDO, F.J.B et al. *O papel da tecnologia da informação (TI) na estratégia das organizações*. Revista Gestão e Produção, v. 8, n. 2, p. 160-179, ago.2001.
- LUTCHEN, M. D. *Managing IT as a business: a survival guide for CEOs*. Hoboken: John Wiley & Sons, 2004.
- MARCHI, L. O; CASSIMIRO, A. P; NASCIMENTO, R. P. C. do. *Suggestion for PETIC 2.0: New Framework, Maturity Levels and Strategic Aligment*. In: EATIS 2010 – Euro-American Conference on Telematics and Information Systems, 2010, Cidade do Panamá. EATIS 2010 – Euro-American Conference on Telematics and Information Systems, 2010a.
- MARCHI, L. O. *Guia de Uso para o PETIC 2.0*. Relatório Técnico. DCOMP, UFS. São Cristovão, 2010b.
- MARCHI, L. O. et al. *Propostas para o PETIC 2.0: Novo Framework, Níveis de Maturidade e Alinhamento Estratégico*. EIC 2010 – Encontro de Iniciação Científica da UFS. São Cristovão, 2010c.
- MARCHI, L. O. *Atualização da Metodologia PETIC - Planejamento Estratégico de TIC*. Relatório Semestral PIBIC 2010-2011. UFS, São Cristovão, 2011.
- MINGAIL, H. B. *Business information technology strategic planning*. [S.I.: s.n.], 2006.
- MINTZBERG, H; AHLSTRAND B; LAMPEL J. *Safári de estratégia*. 2 ed. São Paulo, 2010.
- NASCIMENTO, Rogério P. C. do. *u-Caju – Arquitetura de Suporte a Sistemas Ubíquos que Propiciem Inclusão Digital no Estado de Sergipe*. Convênio de Cooperação Técnico-Científica, Cultural e Educacional UFS-UFAM, 2010.
- O'BRIEN, James. *Sistemas de informação e as decisões gerenciais na era da internet*. 2 ed. São Paulo, 2004
- PILGER, Marcio. *As TICs no cenário socioeconômico brasileiro*. In: Colóquio Mídia & Agenda Social. 1, 2007, Brasília. Anais. Brasília: Ed.Boulevard, 2007.
- PALMEIRA, J. C. *Um software de apoio ao Planejamento Estratégico de Tecnologia da Informação de Comunicação*. Pré-Projeto de Trabalho de Mestrado apresentado ao PROCC, UFS. 2010.
- PINA, E. C.; PALMEIRA, J.C. *Planejamento Estratégico de Tecnologia da Informação e Comunicação: Estudo de Caso CPD/UFS e TRE-SE*. Monografia apresentada no Curso de

- Especialização em Gestão de Projetos do DCOMP/UFS. 2010.
- PMI – Project Management Institute. *Um Guia do Conjunto de Conhecimentos em Gerenciamento de Projetos (Guia PMBOK®)*. Quarta edição, Four Campus Boulevard, Newtown Square, PA 19073-3299 EUA, 2008.
- PORTER, M. E. *Vantagem competitiva: criando e sustentando um desempenho superior*. Rio de Janeiro: Campus, 1989.
- REZENDE, Denis. *Tecnologia da informação e planejamento estratégico: alinhamento e Integração com a Inteligência Organizacional Privada ou Pública*. São Paulo: Brasport, 2008a.
- REZENDE, Denis. *Planejamento de sistemas de informação e informática: guia prático para planejar a tecnologia da informação integrada ao planejamento estratégico das organizações*. 3 edição. São Paulo: Atlas, 2008b.
- ROCKART, J. F. *Chief Executives Define Their Own Data Needs*. *Harward Business Review*. Março-Abril, 1979.
- SEI – Software Engineering Institute. *Capability Maturity Model (CMM)*. Disponível em <<http://www.sei.cmu.edu/cmmi/>> acessado em 10/12/2009
- SILVA, Wagner Amaral e. *PETIC: conceitos Fundamentais, Guia e sua Aplicação na Universidade Federal de Sergipe*. Trabalho de conclusão de curso apresentado ao Departamento de Computação (DCOMP) da UFS, 2009.
- SILVA, Ricardo Vidigal (Org.). *Gestão de Empresas na Era do Conhecimento*. Lisboa: Serinews Editora, 2003.
- TERNER, G. L. K. *Avaliação da aplicação dos métodos de análise e solução de problemas em uma empresa metal-mecânica*. Porto Alegre, 2008.
- USP. *Planejamento Estratégico para TI na USP*. São Paulo: Editora da Universidade de São Paulo, 2007.
- WEILL, P.; ROSS, J. *Governança de TI*. São Paulo: Makron Books, 2006.
- WETHERBE, J. C.; TURBAN, E.; MCLEAN, E. *Tecnologia da Informação para Gestão*. 3 edição. São Paulo: Bookman, 2002.

Casamento de Padrões Usando uma Máquina Virtual de Parsing para PEGs

Sérgio Queiroz de Medeiros

Resumo. Bibliotecas para o casamento de padrões (*pattern matching*) são uma facilidade presente em várias linguagens de programação, tais como Perl, Python, PHP, Ruby e Lua. Essas bibliotecas, comumente conhecidas como *regexes*, usam a sintaxe de expressões regulares e estendem expressões regulares com várias construções *ad-hoc*. Dado que essas extensões não possuem um modelo formal, o significado de alguns padrões não é claro, e é difícil prever a eficiência do casamento de um determinado padrão. Gramáticas de Expressões de Parsing (PEGs) são um formalismo que pode ser usado como base formal para o casamento de padrões, de modo a tornar preciso o significado de padrões. Além disso, PEGs possuem uma máquina virtual de parsing que pode ser usada para determinar a eficiência de um padrão. Neste capítulo, apresentamos uma visão geral de casamento de padrões e de PEGs, e discutimos a máquina virtual de parsing para PEGs. Discutimos também trabalhos que podem ser desenvolvidos nessa linha no DComp/UFS.

Palavras-chave: Casamento de padrões, Expressões regulares, PEGs, máquina de *parsing*.

1 Introdução

A Teoria da Computação é a área da Ciência da Computação que estuda modelos de computação. Nessa área, dada a descrição de um computador abstrato, deseja-se saber se um determinado problema pode ser resolvido nesse modelo de computação e com que eficiência essa solução pode ser alcançada (SIPSER, 1996).

Os autômatos, juntamente com a máquina de Turing, são um dos modelos de computação mais estudados na Teoria da Computação. Os autômatos finitos, que são autômatos com uma quantidade limitada de memória, são bastante usados em aplicações de casamento de padrões (*pattern matching*) (THOMPSON, 1968; COX, 2007).

Outro formalismo bastante usado em aplicações de casamento de padrões são as expressões regulares, que têm uma correspondência direta com autômatos finitos. As expressões regulares e os autômatos finitos possuem o mesmo poder expressivo, e descrevem a classe das linguagens regulares.

Em aplicações de casamento de padrões, a notação sucinta de expressões regulares é usada para descrever padrões. Um padrão é então convertido em um autômato finito que tenta encontrar esse padrão em um determinado texto de entrada.

O uso de expressões regulares e autômatos para o casamento de padrões teve início no fim dos anos 60 e começo dos 70, com o desenvolvimento das ferramentas Unix *grep* e *ed* (THOMPSON, 1968; COX, 2007). Atualmente, bibliotecas de casamento de padrões, popularmente conhecidas como *regexes*, estão entre as principais facilidades oferecidas por linguagens de programação tais como Perl, Python, PHP, Ruby e Lua (IERUSALIMSCHY, 2009). Embora *regexes* sejam baseadas em expressões regulares, elas possuem várias extensões sintáticas e semânticas que não podem ser descritas através de expressões regulares puras.

O uso dessas extensões foi motivado pelo fato de que expressões regulares puras não permitem descrever, ou não podem descrever facilmente, alguns padrões de texto. Por exemplo, como expressões regulares não possuem um operador de complemento, é difícil descrever um padrão para casar um identificador de um programa C, pois teríamos que excluir as palavras reservadas como *int* e *while*.

De modo geral, as extensões usadas por bibliotecas *regexes* não possuem uma base formal, sendo portanto um conjunto de construções *ad-hoc*. Devido à ausência de um modelo formal, o significado de padrões que usam essas extensões não é claro, e pode variar entre diferentes implementações de bibliotecas *regexes* (FRIEDL, 1997) ou até mesmo entre diferentes implementações de uma mesma biblioteca (FOWLER, 2003).

Como as extensões de *regexes* não podem ser descritas através de expressões regulares puras, elas também não podem ser convertidas para um autômato finito, de modo que *regexes* não possuem um modelo computacional bem definido. Em virtude disso, é difícil determinar a eficiência de um padrão e entender como podemos melhorá-la.

Gramáticas de Expressões de Parsing (*Parsing Expression Grammars – PEGs*) são um formalismo proposto por (FORD, 2004) e que segundo (IERUSALIMSCHY, 2009) são uma alternativa interessante como uma base formal para o casamento de padrões.

PEGs podem descrever todas as linguagens regulares e também as linguagens livres de contexto determinísticas. Em virtude disso, é possível descrever por meio de PEGs várias extensões *regexes*. O uso de PEGs para descrever extensões *regexes* nos permite dar um significado preciso a essas extensões.

Além disso, PEGs possuem uma máquina de parsing (MEDEIROS e IERUSALIMSCHY, 2008; IERUSALIMSCHY, 2009), onde uma dada PEG é compilada em um programa que é executado nessa máquina de parsing. Uma vez que PEGs podem expressar extensões *regexes*, a máquina de parsing para PEGs pode ser usada como um modelo de computação para *regexes*. Com base nesse modelo, podemos entender melhor como um determinado padrão é casado e como podemos modificá-lo de modo a tornar o seu casamento mais eficiente.

Esse capítulo apresenta uma visão geral de casamento de padrões e discute o uso de PEGs e de uma máquina virtual de parsing para PEGs em aplicações de casamento de padrões. Na próxima seção, discutimos autômatos e expressões regulares, e o uso desses formalismos no casamento de padrões. Em seguida, na seção 3, comentamos algumas extensões usadas por bibliotecas *regexes*. Na seção 4, vamos rever a definição de PEGs e discutir a transformação de *regexes* em PEGs. Na seção 5 discutimos a máquina virtual de parsing para PEGs e a compilação de *regexes* para programas dessa máquina. Por fim, na seção 6, apresentamos algumas conclusões e enumeramos alguns trabalhos futuros que podem ser desenvolvidos no DComp/UFS.

2 Autômatos Finitos e Expressões Regulares

Autômatos finitos são um modelo de computação simples que é bastante usado em aplicações

que envolvem o processamento de texto. Na Figura 1, podemos ver o exemplo de um autômato finito, que consiste de estados (representados por círculos) e de transições (representadas por setas) entre esses estados.

O autômato da Figura 1 é um exemplo de um Autômato Finito Determinístico (*Deterministic Finite Automata – DFA*). Esse autômato é finito pois possui uma quantidade limitada de memória (um número finito de estados), e é determinístico pois em cada estado há somente uma transição possível para um dado símbolo de entrada.

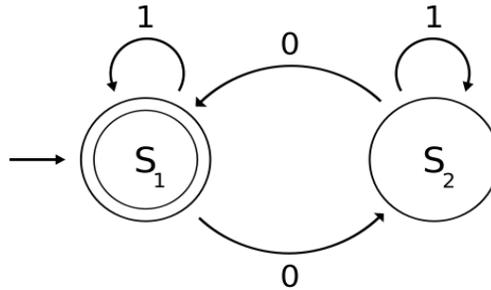


Figura 1 - Autômato finito que aceita cadeias de 0s e 1s que possuem um número par de 0s.

Quando o autômato está no estado S_1 há duas transições possíveis. Se a entrada começa com 0, o autômato vai para o estado S_2 e consome esse 0 da entrada, e se a entrada começa com 1 ele permanece no estado S_1 e consome esse 1 da entrada. As transições a partir do estado S_2 são similares: quando a entrada começa com 0 o autômato consome esse 0 e muda de estado, e quando a entrada começa com 1 o autômato consome esse 1 e permanece no estado S_2 .

Dada uma cadeia (*string*) de 0s e 1s, o autômato da Figura 1 termina no estado S_1 se a cadeia de entrada possui um número par de 0s, e termina no estado S_2 caso contrário.

O estado S_1 possui um círculo duplo, o que indica que ele é um estado de aceitação. Quando um autômato termina em um estado de aceitação após processar uma cadeia de entrada dizemos que essa cadeia pertence à linguagem do autômato. Assim, a linguagem do autômato da Figura 1 consiste nas cadeias que possuem um número par de 0s.

Os autômatos finitos descrevem as linguagens regulares, a mesma classe de linguagens que pode ser descrita através de expressões regulares. Expressões regulares usam uma notação algébrica para descrever linguagens. Na tabela 1, podemos ver a sintaxe de expressões regulares e a linguagem descrita por cada expressão:

Tabela 1: Expressões regulares e suas respectivas linguagens.

| Expressão Regular | Linguagem Correspondente |
|-------------------|--------------------------|
| \square | \square |
| ε | $\{ "" \}$ |
| a | $\{ "a" \}$ |
| $e_1 e_2$ | $L(e_1) \cup L(e_2)$ |
| $e_1 e_2$ | $L(e_1)L(e_2)$ |
| e^* | $L(e)^*$ |

A expressão regular \square representa a linguagem vazia, enquanto que a expressão regular ϵ representa a linguagem que possui apenas a cadeia vazia.

A expressão regular a representa uma linguagem que contém apenas uma cadeia com um símbolo. Por exemplo, a expressão regular 0 denota a linguagem $\{“0”\}$, enquanto que a expressão regular 1 denota a linguagem $\{“1”\}$,

Uma barra vertical é o operador de escolha de expressões regulares. A linguagem da expressão $e_1|e_2$ é a união das linguagens das expressões e_1 e e_2 .

A justaposição de duas expressões regulares representa a concatenação dessas expressões. A linguagem da expressão e_1e_2 é formada pela concatenação das cadeias em $L(e_1)$ com as cadeias em $L(e_2)$.

Por fim, a expressão e^* indica uma repetição. A linguagem dessa expressão consiste nas cadeias que podem ser formadas concatenando as cadeias em $L(e)$ zero ou mais vezes. A linguagem de e^* sempre contém a cadeia vazia.

No fim dos anos 60, Thompson propôs o uso de expressões regulares para o casamento de padrões (THOMPSON, 1968). De acordo com a abordagem de Thompson, os padrões de texto seriam descritos através de expressões regulares e essas expressões seriam então convertidas em autômatos finitos que buscariam esse padrão em um texto de entrada.

Por exemplo, em uma ferramenta de casamento de padrões o usuário digitaria a expressão regular $(1|0^*0)^*$, que descreve as cadeias de 0 s e 1 s que possuem um número para de 0 s, e essa expressão seria convertida no autômato da Figura 1, que aceita essa linguagem.

3 Regexes

Bibliotecas de casamento de padrões, comumente conhecidas como *regexes*, são uma das principais funcionalidades oferecidas por linguagens de programação tais como Perl, Python, PHP, Ruby e Lua.

Embora regexes sejam baseadas em expressões regulares, elas apresentam várias extensões sintáticas e semânticas em relação a expressões regulares puras. A seguir, iremos ver algumas extensões de bibliotecas regexes. Uma discussão mais completa sobre essas extensões pode ser encontrada em (OIKAWA *et al*, 2010) e em (IERUSALIMSKY, 2009).

Uma extensão bastante comum são as classes de caracteres. Classes de caracteres são delimitadas por colchetes e casam qualquer um dos caracteres que pertencem à classe. Por exemplo, a classe de caracteres $[012]$ casa os caracteres 0 , 1 e 2 , e pode ser descrita através de expressões regulares puras como $[012]$.

Bibliotecas regexes também possuem o padrão $.$, que casa qualquer caractere. Uma expressão regular equivalente a essa extensão é uma escolha com todos os caracteres que podem aparecer no texto de entrada.

Outra extensão comum em bibliotecas regexes é a repetição e^+ , que casa uma expressão e uma ou mais vezes. Um padrão da forma a^+ , que casa um ou mais as , pode ser reescrito como a expressão regular aa^+ .

Por sua vez, o operador de repetição $?$ indica o casamento opcional de uma expressão. Um padrão da forma $a?$, que casa o caractere a uma vez ou nenhuma, pode ser reescrito como ea .

Em bibliotecas regexes, os operadores de repetição são gulosos. Isso quer dizer que uma repetição casa a maior porção possível da entrada, desde que isso leve a um casamento bem sucedido do padrão como um todo.

Por exemplo, dado o padrão a^*a e a entrada aaa , a repetição aa^* casa os dois primeiros as da entrada. Se essa repetição casasse todos os as da entrada, então o casamento do padrão a não seria bem sucedido, e portanto o casamento da expressão a^*a também não seria bem sucedido.

Para implementar essa semântica de repetição e o operador de escolha, as bibliotecas regexes usam retrocesso global, um assunto que discutiremos na próxima seção. Até o momento, vimos extensões de regexes que podem ser facilmente convertidas para expressões regulares puras. Contudo, há extensões regexes como *backreferences* e predicados que não podem ser convertidas para expressões regulares puras, pois elas permitem casar linguagens que não são regulares, ou são difíceis de descrever através de expressões regulares puras.

Dado que algumas extensões de regexes não possuem expressões regulares equivalentes, também não é possível convertê-las para autômatos finitos. Assim, autômatos finitos não podem ser usados como um modelo computacional para regexes.

Uma consequência da ausência de um modelo formal é que o significado de algumas extensões não é claro, e pode variar entre diferentes implementações de bibliotecas regexes (FRIEDL, 1997). Já a ausência de um modelo computacional faz com que seja difícil determinar a eficiência de um dado padrão. Além disso, a falta de um modelo computacional leva a implementações complexas de bibliotecas regexes.

Na próxima seção discutimos Gramáticas de Expressões de Parsing (PEGs) e como podemos transformar expressões regulares e extensões regexes através de PEGs.

4 Gramáticas de Expressões de Parsing (PEGs)

Gramáticas de Expressões de Parsing (*Parsing Expression Grammars – PEGs*) são um formalismo proposto por (FORD, 2004) como uma alternativa para descrever linguagens. PEGs podem descrever todas as linguagens livres de contexto determinísticas, e também linguagens que não livres de contexto. É um problema em aberto se PEGs podem descrever ou não a classe de linguagens livres de contexto.

A descrição de uma PEG é semelhante à descrição de uma Gramática Livre de Contexto (*Context-Free Grammar – CFG*) e consiste de um conjunto de produções da forma $A \rightarrow p$, onde A é um não-terminal (ou variável), e p é uma expressão de parsing. Na tabela 2, temos a descrição dos operadores usados para construir expressões de parsing, que possuem uma sintaxe similar a de padrões de bibliotecas regexes. Os operadores na tabela 2 aparecem em ordem não crescente de prioridade, assim um dado operador possui prioridade maior ou igual do que os operadores que aparecem embaixo dele nessa tabela.

Tabela 2: Operadores para construir expressões de parsing.

| Expressão de Parsing | Descrição |
|----------------------|-------------------------|
| " | Cadeia de literais |
| [] | Classe de caracteres |
| . | Qualquer caractere |
| $p?$ | Casamento opcional |
| p^* | Zero ou mais repetições |
| $p+$ | Uma ou mais repetições |
| p | Predicado de afirmação |
| $!p$ | Predicado de negação |
| $p_1 p_2$ | Concatenação |
| $p_1 p_2$ | Escolha ordenada |

Aspas simples são usadas para construir expressões de parsing que casam uma única cadeia de caracteres. Por exemplo, a expressão de parsing *'abc'* casa a cadeia *abc*, e a expressão de parsing

" casa a cadeia vazia.

Assim como em bibliotecas regexes, colchetes são usados para construir classes de caracteres, e a expressão \bullet casa qualquer caractere.

PEGs possuem os operadores de repetição $?$, $*$ e $+$ que são usados por bibliotecas regexes. Mais adiante, veremos que em PEGs esses operadores possuem uma semântica possessiva.

Os operadores $\&$ e $!$ são os operadores de predicado em PEGs. O operador $\&$ representa um predicado de afirmação. O casamento de uma expressão de parsing da forma $\& p$ é bem sucedido se o casamento de p é bem sucedido, e falha caso contrário. Já o operador $!$ representa um predicado de negação. O casamento da expressão de parsing $!p$ é bem sucedido quando o casamento de p falha. A expressão de parsing $\& p$ é equivalente a $!!p$. Predicados não consomem caracteres da entrada.

A expressão de parsing $p_1 p_2$ representa uma concatenação. O casamento dessa expressão é bem sucedido se o casamento de p_1 é bem sucedido para uma determinada entrada, e o casamento de p_2 é bem sucedido para a entrada que restou do casamento de p_1 .

Por fim, a expressão p_1/p_2 representa uma escolha ordenada, que oferece uma forma limitada de retrocesso (*backtracking*) e é uma característica distintiva de PEGs. Ao tentar casar essa expressão, tentamos primeiro casar p_1 . Se esse casamento é bem sucedido, então o resultado da escolha é o resultado do casamento de p_1 , e o casamento de p_2 não é realizado. Caso contrário, se o casamento de p_1 falha, então o resultado da escolha ordenada é o resultado do casamento de p_2 .

A escolha ordenada de PEGs possui uma semântica diferente da escolha de bibliotecas regexes, onde temos um retrocesso global. Em bibliotecas regexes que implementam um casamento mais à esquerda (*leftmost matching*), dado o padrão $e_1 e_2$, tentamos primeiro o casamento de e_1 , e se o casamento de todo o padrão (incluindo a expressão que segue essa escolha) é bem sucedido, então o resultado da escolha é o resultado do casamento de e_1 . Caso contrário, se e_1 ou alguma expressão seguinte falhar, então o resultado da escolha é o resultado do casamento de e_2 .

Para ilustrar essa diferença entre PEGs e regexes, vamos considerar a expressão de parsing $(a'/ab')c'$, o padrão regex $(a[ab])c$, e a entrada abc .

No caso do padrão regex $(a[ab])c$, temos que o casamento da primeira alternativa da escolha é bem sucedido e consome o caractere a da entrada. Contudo, o casamento da expressão c que segue a escolha não é bem sucedido para a entrada bc , de modo que o casamento do padrão como um todo falha. Devido ao retrocesso global de regexes, tentamos casar a segunda alternativa da escolha, o que leva a um casamento bem sucedido do padrão como um todo.

No caso da expressão de parsing $(a'/ab')c'$, o casamento da alternativa a' da escolha é bem sucedido, de modo que a alternativa ab' é descartada, e o casamento da expressão que segue a escolha falha. Nesse caso, como em PEGs não há um retrocesso global, o casamento da expressão $(a'/ab')c'$ falha.

De modo geral, para que uma expressão de parsing da forma $(p_1/p_2)p_3$ case a mesma entrada de um padrão regex da forma $(e_1/e_2)e_3$, temos que distribuir a expressão p_3 entre as alternativas da escolha. Assim, a expressão $(p_1/p_2)p_3$ seria reescrita como $p_1 p_3 / p_2 p_3$.

Com base nessa abordagem, reescrevemos $(a'/ab')c'$ como $'ac'/abc'$. Agora, dada a entrada abc , o casamento da primeira alternativa de $'ac'/abc'$ falha, e o casamento da segunda alternativa é bem sucedido e casa toda a entrada.

Devido ao retrocesso local de PEGs, os operadores de repetição $?$, $*$ e $+$ são possessivos, isso quer dizer que após casar uma entrada, uma repetição não devolve caracteres dessa entrada se o casamento da expressão que segue a repetição falha. Por exemplo, dada a expressão de parsing $'a^*a'$, temos que o seu casamento nunca é bem sucedido, pois dada uma entrada como aaa , a expressão $'a^*a'$ casa todos os as da entrada, e portanto o casamento da expressão $'a'$ falha.

No caso de regexes, dado o padrão a^*a e a entrada aaa , a repetição a^* também casa toda a entrada e o padrão como um todo falha. Porém, devido ao retrocesso global essa repetição devolve o último a para a entrada de modo que o casamento do padrão a seja bem sucedido.

Para simular o retrocesso global de um padrão da forma $e_1 * e_2$ em PEGs, vamos criar uma PEG que é equivalente à expressão regular $e_1 e_1^* e_2$, que descreve a mesma linguagem do padrão anterior. Note que ao reescrevermos $e_1 * e_2$, obtivemos um novo padrão onde essa expressão aparece novamente. Em PEGs, podemos descrever uma expressão recursiva por meio de não terminais. Assim, a seguinte PEG é equivalente à expressão $e_1 e_1^* e_2$, onde p_1 é equivalente a e_1 e p_2 é equivalente a e_2 :

$$A \rightarrow p_1 A \mid p_2$$

Nessa PEG, a expressão p_1 pode casar zero ou mais vezes. Se o casamento de p_1 falha, tentamos casar p_2 . Se o casamento desta expressão falha, então o casamento do não terminal A falha, o que faz com que o último casamento de p_1 seja desfeito, e o casamento de p_2 seja tentado novamente. Dessa forma, simulamos o retrocesso global de regexes em PEGs.

Além das transformações anteriores, que trataram de expressões regulares puras, podemos transformar extensões usadas por regexes em PEGs equivalentes. Por exemplo, é possível expressar a semântica de predicados, repetições possessivas, repetições preguiçosas e subpadrões independentes através de PEGs. Uma discussão da transformação de regexes em PEGs equivalentes pode ser vista em (OIKAWA *et al.*, 2010) e em (MEDEIROS *et al.*, 2011).

A seguir discutimos a máquina de parsing para PEGs e a compilação de PEGs para programas dessa máquina.

5 Máquina de Parsing para PEGs

Nesta seção, vamos descrever a máquina virtual de parsing para PEGs. Além disso, vamos mostrar a tradução de algumas expressões de parsing para programas da máquina virtual.

A máquina de parsing para PEGs é similar às máquinas virtuais de linguagens imperativas, ela possui um conjunto de instruções que mudam o seu estado.

O estado da máquina é descrito pelos seguintes elementos: um contador de programa, que

possui o endereço da próxima instrução a ser executada; um registrador que armazena a posição corrente da entrada; e uma pilha que armazena endereços de retorno e entradas de retrocesso.

Um endereço de retorno consiste apenas de um novo valor para o contador de programa, enquanto uma entrada de retrocesso armazena tanto um novo valor para o contador de programa quanto uma posição da entrada.

A seguir, temos a descrição textual das instruções da máquina de parsing para PEGs:

- **Char a**: tenta casar o caractere *a*. Se o próximo caractere da entrada é *a*, essa instrução incrementa em uma unidade a posição corrente da entrada. Caso contrário, ela falha.
- **Any**: incrementa em uma unidade a posição corrente da entrada caso o fim da entrada não tenha sido alcançado, e falha caso contrário.
- **Choice I**: empilha uma entrada de retrocesso, onde *l* é o endereço relativo da instrução alternativa.
- **Jump l**: faz um desvio para a instrução no endereço *l*.
- **Call l**: empilha o endereço da próxima instrução e faz um desvio para a instrução no endereço *l*.
- **Return**: desempilha um endereço e faz um desvio para esse endereço.
- **Commit l**: finaliza o casamento de uma escolha. Essa instrução desempilha a entrada de retrocesso no topo da pilha, ignorando-a, e então faz um desvio para a instrução no endereço *l*.
- **Fail**: força uma falha. No caso de uma falha, a máquina desempilha entradas da pilha até encontrar uma entrada de retrocesso. Após isso, o novo estado da máquina passa a ser o endereço armazenado pela entrada de retrocesso juntamente com o resto da pilha.

De modo formal, podemos descrever o estado da máquina ou como uma tripla $NxNxPilha$, que corresponde ao contador de programa, à posição corrente da entrada e à pilha, ou como **Fail** $\langle Pilha \rangle$, que corresponde ao estado da máquina quando uma falha ocorre.

Na Figura 2, apresentamos a semântica operacional da máquina. Essa semântica descreve como o estado da máquina muda de acordo com cada instrução. Daqui em diante, vamos usar *pc* para nos referir ao contador de programa, *i* para nos referir à posição corrente da entrada, e *h* para nos referirmos à pilha. O programa que está sendo executado, representado por *P*, e a entrada corrente, representada por *E*, que aparecem na Figura 2, são valores globais. A seguir, mostraremos exemplos de compilações de PEGs para programas da máquina virtual de parsing.

| | | | |
|---|-------------------|--|----------------|
| $\langle pc, i, h \rangle$ | → Char a | $\langle pc+1, i+1, h \rangle$ | $E[i]=a$ |
| $\langle pc, i, h \rangle$ | → Char a | Fail $\langle h \rangle$ | $E[i] \neq a$ |
| $\langle pc, i, h \rangle$ | → Any | $\langle pc+1, i+1, h \rangle$ | $i+1 \leq (E)$ |
| $\langle pc, i, h \rangle$ | → Any | Fail $\langle h \rangle$ | $i+1 > (E)$ |
| $\langle pc, i, h \rangle$ | → Choice l | $\langle pc+1, i+1, (pc+l, i):h \rangle$ | |
| $\langle pc, i, h \rangle$ | → Jump l | $\langle pc+l, i, h \rangle$ | |
| $\langle pc, i, h \rangle$ | → Call l | $\langle pc+l, i, (pc+1):h \rangle$ | |
| $\langle pc', i, pc:h \rangle$ | → Return | $\langle pc, i, h \rangle$ | |
| $\langle pc, i, h1:h \rangle$ | → Commit l | $\langle pc, i, h \rangle$ | |
| Fail $\langle pc:h \rangle$ | → <i>any</i> | Fail $\langle h \rangle$ | |
| Fail $\langle (pc, i):h \rangle$ | → <i>any</i> | $\langle pc, i, h \rangle$ | |

Figura 2: Semântica operacional da máquina de parsing.

5.1 Compilação de PEGs

Para começar, vamos ver a compilação da expressão de parsing **'abc'**. O programa gerado por essa expressão de parsing pode ser visto a seguir:

```
Char 'a'
Char 'b'
Char 'c'
```

A compilação da expressão acima gera três instruções, onde cada uma delas casa um caractere. Como na máquina de parsing as instruções são executadas de maneira sequencial, para executar dois programas em sequência precisamos apenas concatenar as suas instruções.

Sejam p_1 , p_2 e p_3 expressões de parsing, a compilação da concatenação $(p_1/p_2)p_3$ geraria o seguinte programa:

```
Choice L1
  <p1>
Commit L2
L1: <p2>
L2: <p3>
```

O programa anterior começa empilhando uma entrada de retrocesso com o endereço de onde começa o programa correspondente a p_2 . A entrada de retrocesso também possui a posição corrente da entrada, pois p_1 e p_2 tentam casar a entrada a partir da mesma posição. Em seguida, o programa correspondente à p_1 , indicado por $\langle p1 \rangle$, é executado. Caso o casamento de p_1 seja bem sucedido, descartamos a entrada de retrocesso que foi empilhada anteriormente e fazemos um desvio para o endereço $L2$, onde começa o programa relacionado à expressão p_3 , de modo a não executar o programa associado a p_2 . Caso contrário, a entrada de retrocesso será desempilhada e usada para estabelecer o novo estado da máquina, que irá executar o programa correspondente a p_2 , que começa no endereço $L1$.

Como vimos anteriormente, ao tentar simular a semântica de regexes através de PEGs, uma expressão da forma $(p_1/p_2)p_3$ seria transformada em $p_1 p_3 / p_2 p_3$. Nesse caso, o seguinte programa seria gerado:

```
Choice L1
  <p1>
  <p3>
Commit L2
L1: <p2>
L2: <p3>
```

Agora o programa $\langle p3 \rangle$ é executado logo após o programa $\langle p1 \rangle$, e quando a execução desses programas é bem sucedida é feito um desvio para a instrução no endereço $L2$. Caso contrário, se ocorre uma falha durante a execução de $\langle p1 \rangle$ ou de $\langle p3 \rangle$, é feito um desvio para a instrução no endereço $L1$.

Para encerrar essa visão geral do processo de compilação, vamos ver a compilação de uma

repetição da forma $p_1 * p_2$, que é mostrada a seguir:

L1: Choice L2

<p1>

Commit L1

L2: <p2>

Esse programa começa empilhando uma entrada de retrocesso com a posição corrente da entrada e o endereço $L2$, onde inicia o programa correspondente a p_2 . Em seguida, tentamos casar p_1 .

Em caso de um casamento bem sucedido, descartamos a entrada de retrocesso empilhada anteriormente e fazemos um desvio para o endereço $L1$. Nesse ponto, empilhamos uma nova entrada de retrocesso. Note que essa nova entrada de retrocesso possui o mesmo endereço da entrada empilhada anteriormente, mas contém uma posição da entrada diferente, uma vez que p_1 casou um prefixo da entrada.

Quando há uma falha ao executar o program <p1>, é feito um desvio a entrada de retrocesso é desempilhada e usada para estabelecer o novo estado da máquina, que irá executar as instruções a partir do endereço $L2$. Nesse endereço começa o programa <p2>.

Como discutimos anteriormente, se o casamento de p_2 falha, o padrão como um todo falha. Para que um novo casamento seja tentado, assim como é feito em bibliotecas regexes, devemos reescrever a expressão de parsing anterior como $A \rightarrow p_1 A / p_2$, onde A é um novo não terminal. O programa correspondente à expressão $p_1 A / p_2$ é mostrado a seguir:

A: Choice L1

<p1>

Call A

Commit L2

L1: <p2>

L2: Return

Esse programa começa empilhando uma entrada de retrocesso, e então executa o programa correspondente à primeira alternativa da escolha $p_1 A / p_2$. Quando o programa <p1> é executado com sucesso, executamos novamente o programa associado ao não-terminal A . Quando <p1> falha, é feito um desvio para a instrução no endereço $L1$. No fim desse programa, há uma instrução *Return*. Essa instrução faz um desvio para a próxima instrução depois da instrução que fez uma chamada ao programa do não-terminal A .

Com isso concluímos a apresentação do conjunto básico de instruções da máquina de parsing, que é suficiente para descrever todas as PEGs. Além dessas instruções básicas, há outras instruções que são usadas para gerar programas mais eficientes ou mais sucintos. Por exemplo, a máquina possui uma instrução *Charset*, que é usada para casar uma classe de caracteres. Uma descrição mais completa das instruções da máquina de parsing pode ser encontrada em (IERUSALIMSCHY, 2009) e em (MEDEIROS e IERUSALIMSCHY, 2008).

6 Conclusão

Bibliotecas de casamento de padrões são uma das principais facilidades oferecidas por linguagens de programação como Perl, Python, PHP, Ruby e Lua. Essas bibliotecas, comumente

chamadas de bibliotecas *regexes*, são baseadas na sintaxe de expressões regulares e possuem um conjunto de extensões em relação a expressões regulares puras.

Essas extensões geralmente não possuem uma base formal, nem podem ser expressas através de expressões regulares puras. Com isso, o significado de algumas extensões não é claro, e pode variar entre as diferentes bibliotecas *regexes*, ou até mesmo entre diferentes implementações de uma mesma bibliotecas *regex*.

PEGs são um formalismo no qual podemos expressar várias das extensões presentes em bibliotecas *regexes*. Em (MEDEIROS *et al*, 2011) e (OIKAWA *et al*, 2010) é apresentada uma transformação para converter expressões regulares e extensões *regexes* em PEGs.

Uma vantagem do uso de PEGs como um modelo formal para o casamento de padrões é que com isso podemos dar um significado preciso aos padrões.

Além disso, PEGs podem ser compilados em programas de uma máquina virtual de parsing, e essa máquina pode ser usada como um modelo de computação para raciocinarmos sobre o casamento de um determinado padrão e como podemos tornar o casamento desse padrão mais eficiente.

6.1 Pesquisa sobre PEGs e *regexes* na UFS

Uma linha de pesquisa que pode ser desenvolvida na UFS é o uso de PEGs para o casamento de padrões. Nessa linha, podemos destacar dois temas principais:

- Transformação de padrões *regexes* em PEGs: esse tema envolve o estudo de bibliotecas *regexes* e a transformação de *regexes* em PEGs. Nesse tema o objetivo é dar uma semântica clara a padrões *regexes* e gerar PEGs que casem a entrada de maneira eficiente.
- Máquina virtual de PEGs: nesse tema estudaremos a compilação de PEGs para programas da máquina virtual, com o objetivo de adaptar essa compilação para gerar programas que tenham um desempenho melhor, o que pode envolver a extensão do conjunto de instruções da máquina virtual.

Na pesquisa sobre PEGs e *regexes*, usaremos a biblioteca LPeg (IERUSALIMSCHY, 2007; IERUSALIMSCHY, 2009) como modelo de implementação de uma máquina virtual para PEGs.

A partir das funcionalidades oferecidas por LPeg, podemos implementar uma nova biblioteca para o casamento de padrões que é baseada em PEGs. Essa biblioteca implementaria a semântica de bibliotecas *regexes* e seria uma ferramenta onde aplicaríamos os resultados dessa pesquisa.

Ao fim do trabalho, essa biblioteca deve ser lançada como um *software* livre e disponibilizada para a comunidade.

Uma outra linha de pesquisa que pode ser desenvolvida na UFS é o estudo da correspondência entre classes de Gramáticas Livres de Contexto e PEGs. Esse estudo seria uma continuação de (MEDEIROS, 2010).

Referências Bibliográficas

- COX, R. Regular Expression Matching Can Be Simple And Fast . Disponível em <http://swtch.com/~rsc/regex/regex1.html>, 2007.
- FORD, B. Parsing Expression Grammars: A Recognition-Based Syntactic Foundation. Principles of Programming Languages: ACM, 2004.
- FOWLER, G. An interpretation of the POSIX regex standard. Disponível em <http://www2.research.att.com/~gsf/testregex/re-interpretation.htm>, 2003.
- FRIEDL, J. Mastering Regular Expressions. O'Reilly Media, Inc., 1997.
- IERUSALIMSCHY, R. A Text Pattern-Matching Tool based on Parsing Expression Grammars. *Software - Practice & Experience*: vol. 39 (3), pp. 221—258, Jonh Wiley & Sons, 2009.

- IERUSALIMSCHY, R. LPeg: Parsing Expression Grammars for Lua. Disponível em <http://www.inf.puc-rio.br/~roberto/lpeg>, 2007.
- MEDEIROS, S., IIERUSALIMSCHY, R. A Parsing Machine for PEGs. Dynamic Languages Symposium: ACM, 2008.
- MEDEIROS, S. Correspondência entre PEGs e Classes de Gramáticas Livres de Contexto. Tese de doutorado: PUC-Rio, 2010.
- MEDEIROS, S., MASCARENHAS, F., IERUSALIMSCHY, R. From Regular Expressions to PEGs. Simpósio Brasileiro de Linguagens de Programação, 2011 (a ser publicado).
- OIKAWA, M., IERUSALIMSCHY, R., de Moura, A. L. Converting regexes to Parsing Expression Grammars. Simpósio Brasileiro de Linguagens de Programação: SBC, 2010.
- SIPSER, M. Introdução à Teoria da Computação. Cengage Learning, 1996.
- THOMPSON, K. Programming Techniques: Regular expression search algorithm. Communications of the ACM: vol. 11 (6), pp. 419—422, ACM, 1968.

Seção 2

Computação Inteligente

Métodos computacionais para a detecção, caracterização e extração de padrões de atividade a partir de dados biomédicos

Carlos Alberto Estombelo-Montesco

Resumo. Na área de processamento de sinais biomédicos a extração de padrões de informação baseada em um conjunto de medidas adquiridas no tempo é considerada de suma importância. A qualidade desta informação extraída permite avaliar o funcionamento dos diversos órgãos. Neste trabalho se descreverá o método de análise de componentes dependentes para auxiliar a extração de componentes de interesse, a partir de medidas multivariadas. Observa-se que os sinais de interesse raramente são registrados de forma isolada, e sim misturados com outros sinais superpostos, ruído e artefatos fisiológicos ou ambientais, onde a relação sinal-ruído é geralmente baixa. Nesse contexto, a estratégia a ser utilizada baseia-se na ACD (Análise de Componentes Dependentes) que permitirá extrair um pequeno número de fontes, de potencial interesse, com informações úteis. A estratégia ACD para extração de informação é aplicada em três importantes problemas, na área de processamento de sinais biomédicos: (1) detecção do sinal do feto em magnetocardiografia fetal (MCGf); (2) detecção da atividade de resposta elétrica do estômago em magnetogastrografia (MGG); e, (3) detecção de regiões ativas do cérebro em experimentos de imagens por ressonância magnética funcional (*Functional Magnetic Resonance Imaging*, fMRI). Os resultados, nos três casos estudados, mostram que o método utilizado como estratégia, é efetivo e computacionalmente eficiente para extração de sinais de interesse. A partir desse ponto e dando continuidade aos trabalhos desenvolvidos e relatados neste artigo mostra-se série de trabalhos futuros a serem desenvolvidos em nível de Graduação no Departamento de Computação (DCOMP) e em nível de Pós-Graduação por meio do Programa de Pós-Graduação em Ciência da Computação (PROCC) dentro da área de pesquisa de Computação Inteligente da Universidade Federal de Sergipe (UFS).
Palavras chave: Reconhecimento de Padrões. Análise de Componentes Independentes. Componentes Ortogonais. Sinais biomédicos.

1 Introdução

Um objetivo importante na área de processamento de sinais biomédicos é a extração de informação baseada em um conjunto de medidas feitas ao longo do tempo (JAMES; HESSE,

2005). Em grande parte dos casos, sinais biomédicos são medidas eletromagnéticas, embora outras quantidades físicas ou químicas também possam ser avaliadas. Essas medidas, geralmente realizadas utilizando aparelhos multicanais, são registradas usando sensores distribuídos espacialmente no corpo humano, dando origem a um conjunto de dados correlacionados no tempo e/ou no espaço.

Com relação ao sinal adquirido, deve-se conhecer que a informação, inerente às medidas, depende geralmente do domínio de aplicação, sendo necessário considerar as posições e o número de sensores. Os sinais de interesse raramente são registrados de forma isolada, mas observados misturados com sinais espúrios, ruídos e artefatos fisiológicos ou ambientais, em consequência, a relação sinal-ruído é geralmente baixa. Às vezes, ainda, os dados, contêm medidas superpostas, as que são geradas por fontes localizadas em diferentes partes do corpo humano.

Assim, ruído para uma imagem ou um sinal é, de forma geral, qualquer variação que represente o desvio a partir do verdadeiro sinal ou imagem (KAUFMAN *et al.* 1989). A fonte dos ruídos pode ser de origem sistêmica ou aleatória, este último pode ser denominado não repetitivo e, pode ter algum tipo de distribuição estatística. A estrutura deste ruído e sua caracterização são consideradas, aqui, como todo o que modifica o sinal, ou seja, a fonte original, sendo indesejável durante a avaliação da componente.

Por outro lado, médicos e/ou especialistas procuram nos sinais biomédicos, prévio treinamento e experiência, diferentes padrões de atividade com alguma distribuição espacial em particular. Assim, tentam descobrir e/ou desfazer a mistura visualmente observada. Portanto, é de interesse para o diagnóstico de doenças, se através da automatização e análise de sinais biomédicos, fosse possível conseguir:

- Reverter a mistura e isolar um conjunto de sinais biomédicos em suas componentes constituintes, chamadas também de fontes;
- Fornecer a informação sobre o número de fontes distintas que estão sendo observadas;
- Rastrear as mudanças no número de fontes identificadas, suas distribuições espaciais e sua morfologia, ao longo do tempo.

Nesse sentido, a separação cega de fontes (*Blind Source Separation*, BSS) pode ser utilizada para a Análise de Componentes Independentes – ACI (ou em inglês *Independent Component Analysis*, ICA), sendo uma ferramenta computacional que oferece soluções perante os problemas citados.

A ICA é uma técnica que, essencialmente, extrai um conjunto de fontes subjacentes ou componentes independentes a partir de um conjunto de variáveis aleatórias, medidas ou sinais. Esta técnica, basicamente, utiliza conjuntos de dados multivariados para definir um modelo, baseado em dados observados. Essa técnica considera que as componentes são desconhecidas, misturadas linearmente, tentando reverter o processo de mistura através da separação de fontes e da exploração da independência estatística.

Assim, encontram-se disponíveis um quantitativo de sensores para inúmeras aplicações, porém, na prática um número limitado de fontes são de interesse, sendo o restante considerado como ruído. Para esse tipo de aplicações é necessário desenvolver algoritmos de aprendizagem confiáveis, robustos e efetivos, que permitam extrair apenas uma pequena quantidade de fontes, que sejam de potencial interesse, e que ofereçam informações úteis (BARROS *et al.* 2000).

Neste trabalho é mostrado o método que foi o resultado de trabalhos de pesquisas do autor, recebendo o nome de análise de componentes dependentes (*Dependent Component Analysis*, DCA), e aplicado para separação de fontes em três importantes problemas, na área de processamento de sinais biomédicos: (1) detecção do sinal do feto em magnetocardiografia fetal (MCGf) (DE ARAUJO *et al.* 2005); (2) detecção da atividade de resposta elétrica do estômago em magnetogastrografia (MGG) (ESTOMBELO-MONTESCO *et al.* 2007a); e, (3) detecção de

regiões ativas do cérebro em experimentos de imagens por ressonância magnética funcional (*Functional Magnetic Resonance Imaging*, fMRI) (ESTOMBELO-MONTESCO *et al.* 2007b). O trabalho foi estruturado seguindo esta mesma sequência incluindo os fundamentos necessários para sua melhor compreensão.

2 Contexto teórico dos problemas a serem abordados

De forma geral os sinais biomagnéticos surgem na área de Biomagnetismo onde procuram medir e analisar os campos magnéticos gerados por órgãos humanos (CARNEIRO *et al.* 2000). Dessa forma buscar um maior entendimento sobre os processos fisiológicos dos diversos órgãos, além de desenvolver e aprimorar técnicas de diagnóstico não-invasivas e eficientes.

Os campos magnéticos produzidos pelos humanos são muito fracos, sendo sua potência de 10^{-9} até 10^{-14} T, quando comparado com o campo magnético da Terra, de 10^{-4} T respectivamente. (CARNEIRO *et al.* 2000). Assim, campos de maior intensidade, gerados pelo organismo podem ser medidos por um dispositivo chamado de magnetômetro de fluxo saturado, ou *fluxgate*. No entanto, a maioria dos campos, por serem de pouca intensidade, são medidos por meio de um dispositivo criogênico, de grande sensibilidade, denominado superconductor de interferência quântica, ou com a sigla SQUID (CARNEIRO *et al.* 2000). Este dispositivo foi utilizado para coletar dados no contexto de Magnetocardiografia fetal e Magnetogastrografia mostrados a seguir.

2.1 Magnetocardiografia Fetal (MCGf)

Dentre as diversas aplicações do biomagnetismo, a magnetocardiografia (MCG) – medida restrita a campos magnéticos gerados pelo coração – é uma das mais ressaltantes em virtude da grande incidência de doenças cardiovasculares na população (COMANI *et al.* 2004). Os sinais do coração são medidos por sensores SQUID posicionados sobre a pele que fornecem informação das fontes de atividade elétrica cardíaca sem necessidade de fixação de eletrodos no paciente, diferente das medidas eletrofisiológicas convencionais realizadas (CARNEIRO *et al.* 2000).

A magnetocardiografia fetal (MCGf) é a subárea da MCG voltada para a medida de campos magnéticos gerados pelo coração do feto. Neste caso, os sensores SQUID são posicionados sobre o abdômen da mãe. A MCGf apresenta grandes vantagens quando comparada com a eletrocardiografia fetal (ECGf). Nesta última, os sinais obtidos por eletrodos apresentam muita interferência pelo coração da mãe, como consequência de sinais elétricos com baixa relação sinal/ruído; e, pela presença da *vernix caseosa* na pele do feto, a partir da vigésima quinta semana de gestação, sendo esta considerada uma substância de baixa condutividade e que atua como isolante elétrico (DE ARAUJO *et al.* 2005).

O coração do feto produz campos magnéticos mais fracos, quando comparados com os da mãe. Ainda, apresenta interferências tais como o sinal cardíaco e a respiração da mãe e, inclusive o próprio movimento fetal. Outra dificuldade observada é o pequeno volume do coração e a distância com relação aos sensores, considerando que estes sinais magnéticos fetais exigem uma ótima relação sinal-ruído. Desta forma, a MCGf apresenta uma boa relação sinal-ruído durante todo o período de gestação e permite uma ótima definição na forma de onda, sendo uma excelente técnica para acompanhar a atividade cardíaca fetal (BAFFA *et al.* 1995).

Da mesma forma, a MCGf pode ser usada a partir da décima quinta semana de gestação, muito antes da ECGf (DE ARAUJO *et al.* 2005), o que permite a detecção precoce de anomalias congênitas tornando possível realizar uma intervenção antes do parto (CARNEIRO *et al.* 2000).

2.2 Magnetogastrografia (MGG)

As contrações dos músculos da parede do estômago durante a digestão, assim como no coração, produzem campos elétricos factíveis de ser medidos, utilizando para tal finalidade eletrodos posicionados sobre a pele, técnica denominada de eletrogastrografia (EGG). Assim a atividade elétrica estomacal é medida por sensores SQUID, utilizando a técnica conhecida como magnetogastrografia (MGG) (CARNEIRO *et al.* 2000).

A atividade elétrica gástrica (Figura 1) é constituída por duas componentes: ritmo elétrico de base (REB), não necessariamente associado aos movimentos de contração estomacal, observada como onda elétrica lenta (com frequência de três ciclos por minuto em humanos); e, uma atividade de resposta elétrica (ARE), que ocorre em conjunção com os movimentos de contração estomacal, que é caracterizada por potenciais de ação durante a fase de platô do ritmo elétrico de base (IRIMIA *et al.* 2006; MINTCHEV 2007).

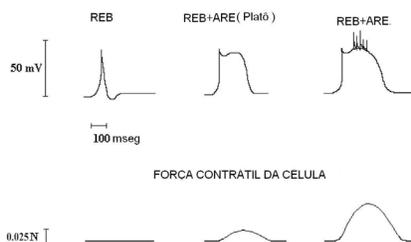


Figura 1: Registro da atividade elétrica gástrica “*in vitro*” e sua relação com a atividade contrátil do estômago.

Fonte: (MINTCHEV, 2007).

A detecção e a caracterização da atividade de resposta elétrica são consideradas de grande importância em gastroenterologia. Da mesma forma, a MGG tem potencial para oferecer uma importante contribuição nesta área, por ser uma técnica não invasiva. Porém, os principais problemas para a detecção da atividade de resposta elétrica por meio de MGG decorrem do fato que, o sinal da atividade de resposta elétrica possui amplitude relativamente baixa e está sujeito a interferências de outros sinais corporais, como os cardíacos e respiratórios.

2.3 Imagem funcional por Ressonância Magnética (fMRI)

A área de neuroimagem funcional teve um grande progresso nos últimos anos. Imagens com resolução anatômica milimétrica e resolução temporal da ordem de milissegundos possibilitam, de forma não-invasiva, a caracterização espaço-temporal de uma gama de processos cognitivos, bem como a sua utilização em algumas rotinas clínicas (DE ARAUJO, 2002).

O princípio da imagem funcional por ressonância magnética (fMRI)¹ para o mapeamento de regiões cerebrais ativas é relativamente simples. As informações de interesse, desde o ponto de vista clínico, obtidas através da fMRI são, geralmente, alcançadas através de diferença entre imagens que refletem dois estados de atividade cerebral distintos - em atividade e em repouso.

Os neurônios quando ativos consomem oxigênio presente nas moléculas de hemoglobina no sangue. Assim, quando incrementada a atividade em uma determinada região do cérebro é provocado um aumento do fluxo sanguíneo e do consumo de oxigênio. Reação que provoca alterações posteriores nas concentrações de oxi-hemoglobina e deoxi-hemoglobina (formas oxigenada e a não-oxigenada da molécula de hemoglobina, respectivamente).

Por sua vez, a oxi-hemoglobina é considerada diamagnética e a deoxi-hemoglobina é

considerada paramagnética, o que ocasiona diferenças na susceptibilidade magnética. Assim, quando alterada esta susceptibilidade no cérebro, também é alterado o tempo de relaxamento, como resposta característica do tecido frente uma sequência apropriada de pulsos de radiofrequência, o qual altera o contraste das imagens, sendo denominado de efeito BOLD (*blood oxygen level dependent*)². Técnicas para a aquisição rápida de imagens permitem explorar o efeito BOLD e obter uma visualização de diversos padrões na atividade cerebral (DE ARAUJO, 2002).

3 Análise de Componentes

A análise de componentes são métodos computacionais multidimensionais baseados em álgebra matricial e estatística que nos permite decompor um conjunto de variáveis e determinar/ descobrir padrões latentes (ocultos) nos próprios dados. Esses métodos podem considerar soluções quando a hipótese indica que há algum tipo de associação ou considerar outro tipo de soluções quando é considerado algum tipo de independência entre as variáveis para formular uma solução.

3.1 Análise de Componentes Independentes (ACI)

A Análise de Componentes Independentes (ACI) refere-se à separação dos sinais registrados (de fato é considerada uma mistura linear de sinais) em suas fontes originais, isto é, se procura saber como eram esses sinais antes da mistura. Essa aproximação é feita por meio de transformadas aplicadas aos dados coletados.

Esse problema pode ser facilmente exemplificado com o problema denominado *cocktail-party*, para tal o enunciado é: “Em um salão encontram-se pessoas conversando, simultaneamente, e com música de fundo. O som da sala será captado por dois microfones em locais diferentes. A finalidade neste contexto é separar os sinais da fala e da música, obtidas misturadas através dos microfones” (COMON, 1994). A Figura 2 mostra a transformação direta \mathbf{A} e a transformação inversa \mathbf{W} , ou seja, a aplicação do sistema de mistura e vice-versa. As componentes independentes ou fontes (fala de pessoas e música) são combinadas linearmente produzindo as observações $\mathbf{X} = [X_1, \dots, X_M]^T$ (sinal captado pelos microfones), a essas observações é aplicada a transformação linear \mathbf{W} , obtendo-se as componentes independentes estimadas, designadas por $\hat{\mathbf{S}} = [\hat{S}_1, \dots, \hat{S}_M]^T$.

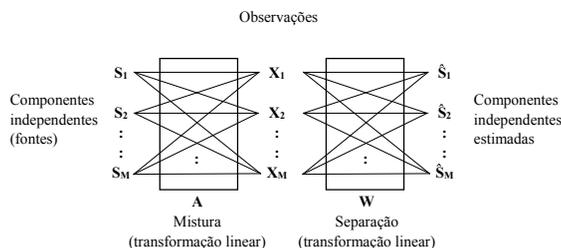


Figura 2: Aplicação do sistema de mistura \mathbf{A} e vice-versa \mathbf{W} .

Fonte: (FERREIRA, 2002).

1 Por ser de uso já consagrado em português, será adotada a sigla em inglês fMRI para representar o termo “imagem funcional por ressonância magnética”.

2 BOLD é uma sigla em inglês, conhecida em português como contraste “dependente no nível de oxigênio no sangue” (DNOS).

3.2 Análise de Componentes Dependentes

Nos trabalhos de pesquisa relatados neste trabalho buscaram utilizar um único algoritmo, para a separação de fontes, que incorpore estratégias na solução de problemas de permutação e de ambiguidade, existentes para a ACI. Uma das características desejadas para o novo algoritmo é que ele possa utilizar sinais não ortogonais (BARROS; CICHOCKI, 2002), ou seja, que possa relaxar a suposição de independência estatística entre as fontes, diminuindo a redundância, onde os sinais de saída devem ter uma medida mínima de correlação cruzada.

Desta forma, para diferenciar o novo método do método ACI, ele é denominado de análise de componentes dependentes (ACD) que pode ser encontrado com maiores detalhes nessas referências: (BARROS; CICHOCKI, 2002) (DE ARAUJO *et al.* 2005) (ESTOMBELO-MONTESCO *et al.* 2007b) (ESTOMBELO-MONTESCO *et al.* 2007a).

Alguns algoritmos propostos na literatura usam a chamada aprendizagem de tipo ponto-fixo, onde é extraído um sinal por vez. Assim, o método de aprendizagem utilizado com esta técnica ACD alicerça-se neste tipo de aprendizagem, sendo a diferença dada pela informação a priori requerida e, pela extração da componente (função base) de forma direta durante a varredura de dados. Esta é uma função realizada por meio da entrada de uma referência interna, no caso a hipótese, que assume ser da própria unidade processadora, ou neurônio, como mostrado na Figura 3. Neste caso, se presume que a entrada dessa referência pode ser um padrão que pertenceu previamente ao neurônio, considerando sua adaptação ao ambiente e suas características genéticas.

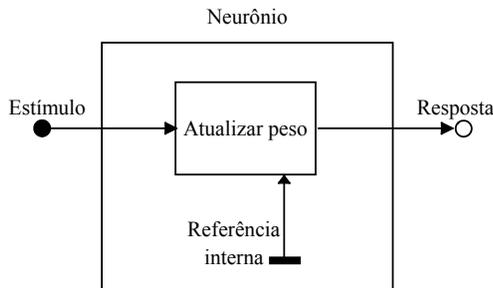


Figura 3: Modelo para o neurônio que inclui uma referência interna (ou padrão).

Fonte: (BARROS; CICHOCKI, 2002).

Inúmeras aplicações encontram-se disponíveis para um grande número de sensores. Porém, somente alguns sinais fonte são de interesse, sendo o restante considerado ruído de interferência. Para o caso do problema *cocktail-party*, onde é essencial extrair a voz específica de um único indivíduo, em vez de separar todos os sinais fonte disponíveis a partir do vetor de sinais, é necessário desenvolver algoritmos de aprendizagem confiáveis, robustos e efetivos que permitam extrair apenas sinais fonte de interesse (BARROS *et al.* 2000).

4 Componente fetal a partir de magnetocardiogramas

Na continuação se descreve uma aplicação do algoritmo de ACD mostrado anteriormente para MCGf, onde o sinal de interesse foi extraído usando um atraso temporal τ_i obtido da modelagem do sistema por autoregressão e, identificando a componente de interesse no sinal. Os resultados mostram que o método é efetivo na remoção do sinal materno, sendo ressaltada sua eficiência computacional. Finalmente, são comparados estes resultados com os obtidos por meio de outro método, conhecido como FastICA (HYVARINEN; OJA, 1997).

Os dados de MCGf foram coletados de dois fetos assintomáticos de 29 a 32 semanas de gestação. Os registros foram realizados com um biomagnetômetro de 37 canais (Magnes II, 4D Neuroimaging) que coletava os sinais do feto e da mãe, de forma misturada (DE ARAUJO *et al.* 2005).

Para ter uma ideia dos dados, a Figura 4 mostra 5 dos 37 canais registrados pelo biomagnetômetro. A figura mostra 10 segundos de dados típicos, onde é observado que o MCG da mãe e o MCG do feto se sobrepõem no tempo.

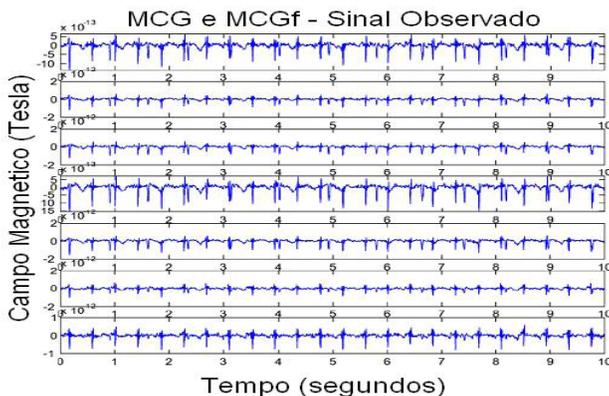


Figura 4: Dez segundos de dados observados, para 5 dos 37 canais de MCGf registrados. Os registros mostram os sinais da mãe e do feto sobrepostos.

Fonte: (DE ARAUJO *et al.* 2005).

O algoritmo ACD foi aplicado para extrair o MCGf a partir dos sinais observados. A complexidade dos dados de MCGf é devida a que os sinais cardíacos, materno e fetal, são quase-periódicos e são similares.

No entanto, é possível demonstrar com o modelo ACD, que os sinais cardíacos, materno e fetal, são caracterizados por valores diferentes quanto a frequência e amplitude. Na Figura 5 observa-se que os batimentos cardíacos do feto têm uma frequência maior que os batimentos da mãe, ainda, a amplitude do MCG da mãe é maior que o MCG fetal.

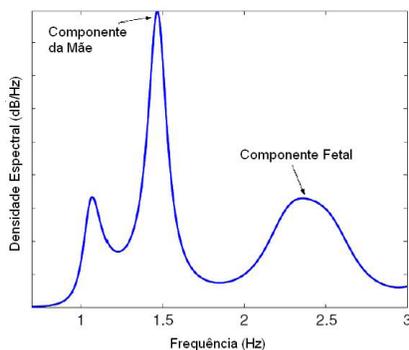


Figura 5: Espectro de potência de um registro simples. Dois picos correspondentes à componente fetal e à componente da mãe são observados.

Fonte: (DE ARAUJO *et al.* 2005)

Após encontrar um pólo em 1,5 Hz, sinal consistente com uma taxa de batimento cardíaco no

adulto, e outro em 2,4 Hz, consistente com uma taxa de batimento cardíaco fetal, foi aplicado o algoritmo de ACD conforme descrito anteriormente. A Figura 6(c) mostra o sinal cardíaco fetal extraído pelo método de ACD, onde o MCGf é claramente preservado e a interferência materna é removida.

Para comparar o resultado do algoritmo ACD com o resultado obtido pelo método de FastICA, o método FastICA foi aplicado ao mesmo conjunto de dados, resultando na Figura 6(a) e Figura 6(b). De forma geral, o método FastICA utiliza varias etapas durante o processo, os quais são evitados quando utilizado o algoritmo proposto. Assim, enquanto no FastICA é necessário encontrar a melhor combinação de sinais de entrada, para formar agrupamentos específicos e obter um resultado estável, no algoritmo proposto é feito numa etapa só.

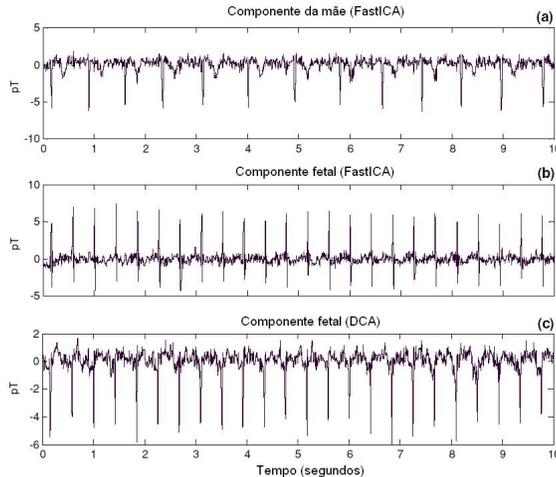


Figura 6: Componentes separadas do MCG da mãe (a) e do MCG fetal (b), obtidas por fastICA, e MCG fetal (c) extraído por ACD baseado na análise espectral auto-regressiva. Esta última é considerada um sinal mais consistente com os batimentos cardíacos fetais. Basicamente, a diferença entre (b) e (c) é o método utilizado, tendo cada um seu próprio forma para aproximar a componente fetal.

Fonte: (DE ARAUJO *et al.* 2005).

5 Componentes gástricas a partir de magnetogastrogramas (MGG)

Na continuação se descreve uma aplicação do método de ACD para a detecção das duas componentes da atividade elétrica gástrica após uma refeição, do ritmo elétrico de base (REB) e da atividade de resposta elétrica (ARE). Conforme mencionado anteriormente, os sinais de MGG são altamente contaminados pela respiração, movimento e artefatos cardíacos, além de considerar a possibilidade de interferência pela atividade mioelétrica originada por órgãos, o que pode confundir a interpretação e a análise do MGG (LIANG *et al.* 2000). Em particular, a detecção da ARE é difícil devido a apresentar um baixo nível de amplitude, possuindo sobreposição espectral com os sinais cardíacos, impossibilitando o uso dos métodos de análise existentes, considerados como padrão.

O método de ACD é aplicado para separar os sinais REB e ARE de outras fontes de interferência, mesmo em casos com baixas taxas da relação sinal-ruído. Da mesma forma, é utilizada uma representação tempo-frequência (RTF) das componentes extraídas, baseada em *wavelets* para examinar as características temporais da REB e da ARE.

Os registros foram realizados em um sistema de gradiômetros de primeira ordem (*Magnes, Biomagnetic Technologies, Inc*), sendo colocados no interior de uma sala blindada contra campos

magnéticos de alta intensidade. O sistema conformado por duas unidades de magnetômetros, A e B, cada uma contendo 37 canais.

Três registros de sinais com 10 minutos de duração foram adquiridos. O primeiro registro pré-prandial, antes da ingestão do alimento teste. Seguidamente, um alimento teste (um sanduíche de pão com queijo de 250 kcal) foi oferecido aos sujeitos antes da segunda medida (primeiro registro pós-prandial). Finalmente, o segundo registro pós-prandial foi adquirido durante o registro de sinais com mais de 10 minutos de duração.

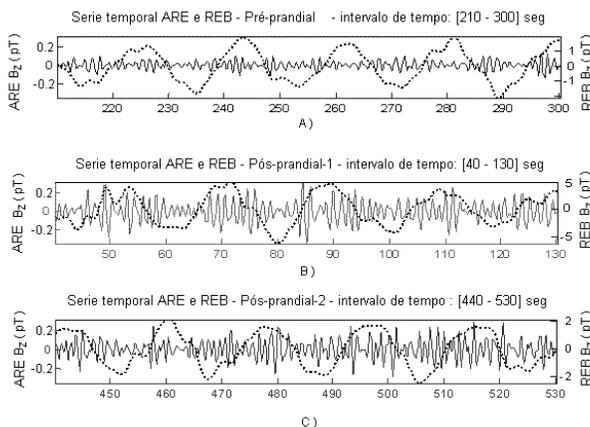


Figura 7: Sinais do REB (linha pontilhada) e da ARE (linha sólida) extraídos usando a ACD. (a) componentes extraídas no registro pre-prandial, (b) e (c) componentes extraídas para o primeiro e segundo registro pós-prandial.

Fonte: (DE ARAUJO *et al.* 2005).

A Figura 7 mostra os três registros, um em cada painel, com as componentes do sinal gástrico associadas ao REB e à ARE extraídas pela ACD. Usualmente para o domínio temporal, a componente do REB possui maior amplitude que a componente da ARE. Assim, para melhor visualização é conveniente mostrar as duas componentes, REB e ARE, cada uma superposta em sua própria escala. Do lado direito mostra-se a escala usada para o sinal do REB e, do lado esquerdo, mostra-se a escala usada para o sinal da ARE. O sinal do REB é *um upstroke* seguido por um platô e, pela fase de despolarização lenta com uma frequência aproximada de 3 cpm. Note a diferença de amplitude entre o sinal do REB no registro pré-prandial e o sinal da ARE nos dois registros pós-prandiais. Sendo a escala do eixo x em segundos.

A Figura 7(b) e (c) mostram que a componente da ARE é sempre encontrada no estômago. Pode-se notar que o sinal da ARE pós-prandial tem maior amplitude que o sinal pré-prandial (Figura 7(a)), especialmente durante a fase do platô do REB, aproximadamente em 50 seg., 70 seg., 90 seg. e 110 seg. da Figura 7(b), respectivamente.

6 Componente auditivo a partir de dados de fMRI

Trabalhos recentes sugerem que o potencial dos métodos de separação cega de fontes, como a ACI, para a análise de dados de fMRI vão em aumento, especialmente quanto informação *a priori*, quanto a disponibilidade, é quando incorporada à estimativa das componentes (SUZUKI *et al.* 2002).

Esses trabalhos encorajaram a proposta do uso da ACD para analisar dados em fMRI. Nas aplicações supracitadas, baseadas na ACI, foram utilizadas informações sobre o protocolo

experimental adotado, embora, no caso da ACD, esta informação esteja contida no próprio conjunto de dados. Da mesma forma, a abordagem de ACD utilizada para fMRI mantém a ideia de extrair apenas a componente de interesse, baseada na estrutura dos próprios dados. Característica considerada um avanço em relação a trabalhos anteriores (CALHOUN *et al.* 2005), pois deixa-se de precisar de informação *a priori* sobre o paradigma experimental e, passa-se a precisar apenas de informação contida no próprio conjunto de dados.

O maior problema encontrado refere-se à determinação de fonte de estímulo auditivo a partir de medidas de fMRI. O paradigma experimental foi baseado na escuta passiva de uma história complexa, com estrutura narrativa padrão. Cada experimento composto por seis blocos de controle de 27,5 s, onde o indivíduo permanecia em repouso, sendo alternados com cinco blocos durante os quais o indivíduo recebia o estímulo.

O protocolo utilizado pode ser observado na Figura 8. O eixo horizontal, graduado em termos de número de pontos, mostra a duração de cada bloco, tanto em repouso como em estímulo. Deve-se observar que o primeiro bloco (repouso) tem uma duração menor, pois seus pontos fazem parte do período no qual o equipamento foi calibrado. Os 64 pontos do eixo horizontal são equivalentes a 289,8 segundos. Os primeiros 12,7 s foram utilizados para calibração do aparelho.

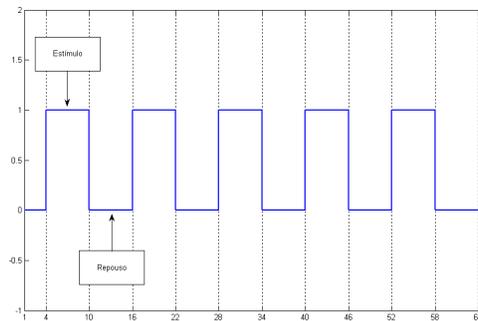


Figura 8: Esquema do protocolo em blocos usado durante o experimento.
Fonte: (ESTOMBELO-MONTESCO *et al.* 2007b).

Na Figura 9 se observa uma amostra dos dados observados, sem a aplicação de qualquer processo de extração. De forma específica, a imagem tem 128x128 pixels, e pertence à fatia 12 do total de 16 fatias, correspondente ao instante do ponto 32 de 64 pontos.

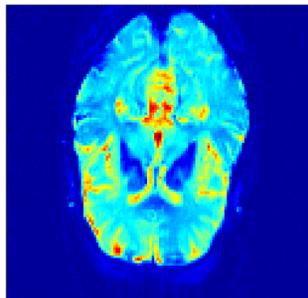


Figura 9: Exemplo de medida de fMRI sem processamento de extração.
Fonte: (ESTOMBELO-MONTESCO *et al.* 2007b).

Considerando que o mecanismo de contraste BOLD aliado às técnicas de aquisição rápida permite a visualização direta de um grande número de processos cerebrais, procurando séries

temporais, resultantes de etapas de pré-processamento, que favoreçam a eliminação de alguns artefatos conhecidos, antes de ser analisadas com ferramentas estatísticas ou algoritmos de extração. Os seguintes tipos de pré-processamento foram realizados baseados em dados: pré-processamento para a correção temporal entre fatias; pré-processamento para a correção de movimento; e pré-processamento para a redução na dimensão dos dados.

A avaliação do desempenho é importante para conhecer a eficácia do método, sendo a maioria das vezes realizada por comparação com outros métodos, considerada padrões. Entretanto, diante da dificuldade de obter um teste padrão para a fMRI, é comum estudar seu desempenho por meio de dados simulados (STURZBECHER, 2006).

Assim, foram gerados, artificialmente, conjuntos de dados em 3-D, de 64x64 voxels com séries temporais da função de resposta hemodinâmica (FRH). Utilizando para tal finalidade uma equação de modelação analítica (GLOVER, 1999):

$$h(t) = \left(\frac{t}{d}\right)^a \exp\left(-\frac{t-d}{b}\right) - c \left(\frac{t}{d'}\right)^{a'} \exp\left(-\frac{t-d'}{b'}\right)$$

Neste modelo, $d=ab$ é o momento do pico da FRH e $d' = a'b'$ é o momento do *undershoot* da FRH. Sendo os parâmetros utilizados: $a = 6$, $a' = 12$, $b = b' = 0,9$ e $c = 0,35$. Assim, na Figura 10 se observa o comportamento da função para estes parâmetros.

Nas colunas de 1 a 32 encontram-se os voxels com padrão de ativação, ou seja, contendo um sinal BOLD, enquanto nas colunas de 32 a 64 contém o ruído, modelado por um padrão gaussiano uniforme.

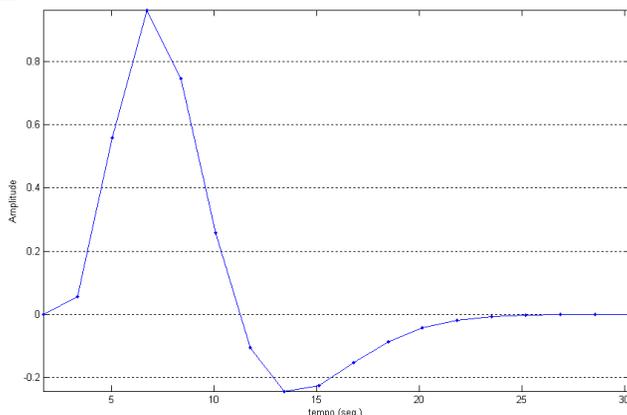


Figura 10: Modelo da função hemodinâmica utilizada nos dados simulados.

Fonte: (ESTOMBELO-MONTESCO *et al.* 2007b).

Na Figura 11 se observa uma série temporal baseada no modelo da FRH anteriormente mostrada.

Antes de processar os dados de fMRI com as 16 fatias, é interessante realizar a análise da fatia que contém a maior quantidade de atividade neural. Nesse sentido, não conhecendo a fatia com maior quantidade de ativação neural, devido ao experimento auditivo, deverá ser, empiricamente, escolhida a melhor fatia. Uma das formas mais simples é realizar uma correlação cruzada, artificial, de cada fatia com um sinal variável no tempo, formada por impulsos de tipo ativado-desativado, como se dá no protocolo experimental.

Este teste empírico permitiu inferir que a fatia 12 poderia conter a maior intensidade do sinal

procurado. A partir deste ponto foi possível aplicar o método de ACD, pois ele pode ter um melhor desempenho quando utilizados os dados da fatia 12.

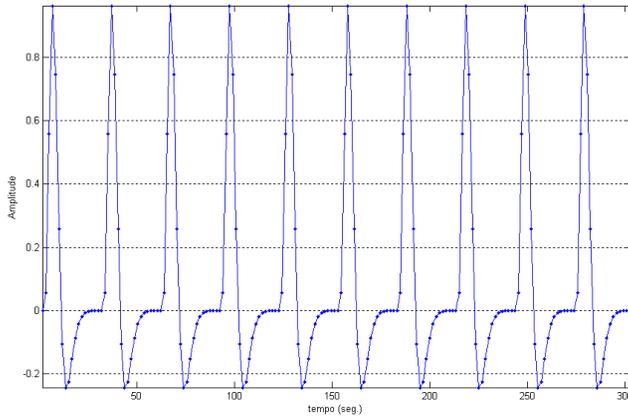


Figura 11: Série temporal de funções de resposta hemodinâmicas baseadas no modelo da Figura 10.
Fonte: (ESTOMBELO-MONTESCO *et al.* 2007b).

Outra questão foi a quantidade de séries temporais processadas. Considerando que cada fatia é composta por 128x128 pixels e, que cada pixel gera uma série temporal, assim tem-se que a quantidade de dados é enorme. No caso deste experimento a resolução espacial foi boa, 128x128 pixels respectivamente; em contraste com a resolução temporal que foi menor (64 pontos de referência amostrados), porém suficiente para que a função de resposta hemodinâmica seja reconhecida.

Por outro lado, precisava-se trabalhar apenas com os pixels que sejam relevantes ou, pelo menos, com aqueles que estejam na área do cérebro. Para tal fim, foi aplicada uma máscara sobre a fatia selecionada, a qual detectou a região do cérebro que contém os pixels. Assim, somente as séries temporais desses pixels foram utilizadas no processo de extração.

Após os diferentes tipos de pré-processamento, descritos acima, foi necessário identificar, com base na informação intrínseca dos dados, a informação *a priori* necessária para a determinação do atraso temporal, a ser usado na modelagem AR.

Uma vez identificada a informação *a priori*, o próximo passo é realizar uma redução na dimensão (branqueamento) dos dados, para trabalhar com os dados mais significativos. O procedimento de extração propriamente dito é feito para cada fatia, ou seja, uma a uma. A matriz original de 3 dimensões, <pixel X, pixel Y, ponto amostrado no tempo>, foi transformada em uma matriz bidimensional, com número de pixels por número de pontos amostrados: <16384, 64>. Esta última matriz é a entrada para o método de ACD, deve observar-se que existe uma relação- quando mapeada cada posição do pixel (vetor linha) da matriz, em sua posição na matriz original- que é usada para posterior localização e criação do mapa de ativação.

A Figura 12 mostra os dados de fMRI, após ser processados com a ACD, a saída é a componente da fonte auditiva, existindo uma forte correlação com o protocolo e, o paradigma usado no domínio do tempo, da Figura 8. Para um sinal quase-periodico, a ACD identifica a componente baseada no atraso temporal determinado a partir das características temporais da resposta hemodinâmica auditiva.

Em consequência, a componente da Figura 12 foi correlacionada com as séries temporais das imagens, obtendo-se como resultado os mapas de ativação, os quais foram sobrepostos às imagens, utilizando-se o *software* de visualização MRIcro. As ativações detectadas foram

localizadas na área do córtex, correlacionada com o experimento auditivo descrito anteriormente, observado na Figura 13. Nessa mesma Figura se observa o mapa de localização da atividade obtido por correlação cruzada entre a componente extraída (representando a função de resposta hemodinâmica auditiva) e o conjunto de dados originais (observações), obtidos por fMRI.

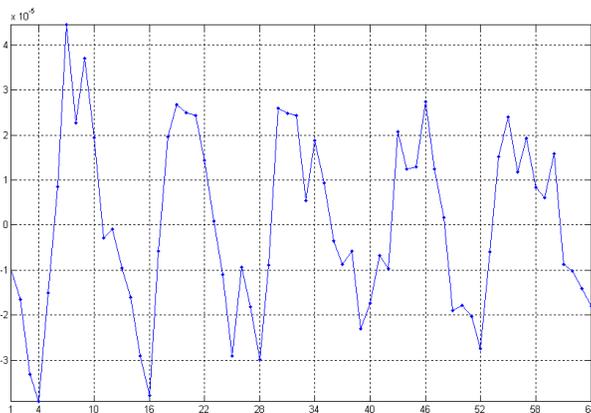


Figura 12 : Componente auditiva extraída.

Fonte: (ESTOMBELO-MONTESCO *et al.* 2007b).

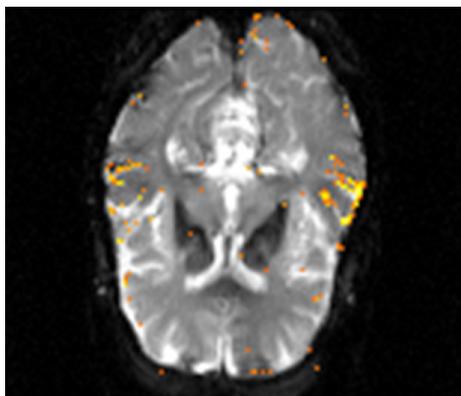


Figura 13 : Área de atividade detectada no experimento de fMRI auditivo (fatia 12 do total de 16 fatias).

Fonte: (ESTOMBELO-MONTESCO *et al.* 2007b).

7 Discussão e conclusões

Os resultados obtidos mostram: a) a extração da componente fetal a partir de magnetocardiogramas, resultados que além do baixo custo, são alentadores quando comparados com o método FasICA da literatura. As etapas de pré-processamento são simples, sendo de importância o cálculo do atraso do sinal a ser extraído, para a posterior extração da componente por meio da ACD.

Outros resultados estão diretamente relacionados com os registros de Magnetogastrografia, especificamente, a detecção, a caracterização da atividade de resposta elétrica (ARE) e o ritmo elétrico de base (REB). Este artigo mostra resultados não reportados antes na literatura de extração de componentes tendo como interesse a componente de interesse (DE ARAUJO *et al.* 2005). Medidas realizadas, com outros voluntários, foram analisadas, confirmando o aumento da

atividade elétrica após a refeição com o alimento teste.

Os resultados da terceira aplicação, com fMRI, foram corroborados pela literatura. Nesta aplicação o número de sinais a serem analisados aumentou significativamente, sendo que cada pixel representava uma serie temporal com uma imagem que variava ao longo do tempo. Assim, cada componente desejada estava relacionada com o experimento de tipo auditivo e, sua extração indicou a existência de atividade cerebral nas áreas laterais do cérebro. Esta abordagem utiliza a ideia de estrutura de correlação, entre as componentes, assim é considerada uma abordagem de maior interesse que aquela por correlação entre a fonte e o protocolo utilizado.

Na maioria dos casos, a extração de todas as fontes foi desnecessária. A formulação deste método designado por ACD levou a uma simples regra de aprendizagem que permite a obtenção de resultados comparáveis com as melhores técnicas de uso na área, onde as vezes, a formulação pode ser bastante complexa. De fato, este método não é afetado (não é sensível) pela dimensão dos dados, como no caso de dados fMRI.

A ACD obteve um bom desempenho a) na detecção e extração dos sinais desejados para as diferentes aplicações, MCG fetal e Magnetogastrografia, onde foram usados sinais biomagnéticos (37 canais); e, b) no mapeamento de atividades cerebrais utilizando fMRI, através de protocolos bem definidos. Resultados que abrem possibilidades para outros tipos de aplicações no processamento de sinais biomédicos, com vários sensores de aquisição no tempo.

O atraso temporal foi uma variável muito utilizada com a componente de interesse, a qual tem um determinado período ou quase-periódico de repetição. Embora, esta condição pode ser relaxada, se encontrada uma referência temporal.

A partir desse ponto e dando continuidade aos trabalhos já desenvolvidos e relatados neste trabalho percebe-se uma serie de trabalhos futuros a serem desenvolvidos em nível de Graduação no Departamento de Computação (DCOMP) e em nível de Pós-Graduação por meio do Programa de Pós-Graduação em Ciência da Computação (PROCC) da Universidade Federal de Sergipe (UFS) e dentro da área de pesquisa de Computação Inteligente, eles são: a) utilizar exaustivamente a ACD para futuras aplicações clínicas validando o resultado com o auxilio dos especialistas na área para dessa forma tentar contribuir efetivamente com a nossa sociedade; b) Atualmente o método de ACD trabalha sobre a dimensão do tempo, sendo de grande interesse que a ACD seja estendido para extrair componentes com características espaço-temporais; c) O estudo das imagens de ressonância magnética funcional requer uma utilização eficiente do *hardware* e *software* disponível, nesse sentido existe um planejamento pelo grupo de pesquisa para a utilização de *hardware* de alto desempenho com o intuito de aproveitar o processamento paralelo das unidades de processamento gráfico (GPU - *Graphics Processing Unit*), sendo esta linha de pesquisa um dos interesses que serão alinhados aos trabalhos já desenvolvidos e relatados neste artigo; d) A exploração de dados multivariados se torna um desafio em qualquer área do conhecimento desde que o objetivo seja detectar e/ou reconhecer estes padrões que estão escondidos ou misturados com outras variáveis; e) Outro desafio a ser considerado é a construção de novos modelos de algoritmos que permitam extrair informações de interesse sem a necessidade de utilizar informação a priori (externa ao modelo) e considerando somente a estrutura dos próprios dados coletados, isto é, descobrir o que os dados querem dizer ou revelar diante da grande quantidade variáveis; f) Como a disponibilidade de um teste padrão para fMRI esta longe de ser obtido dado que os estudos invasivos são restritos ou estes são considerados não viáveis frente a um comitê de ética, se faz necessário avaliar e/ou comparar os métodos por meio de dados simulados, gerados artificialmente para assim posteriormente fazer uso das ferramentas estatísticas e validar os mesmos; g) Na exploração dos dados multivariados geralmente é utilizada a hipótese de que as variáveis (em análise) são linearmente independentes ou que estas devem ser extraídas considerando a existência de uma correlação igual a zero; no

entanto uma hipótese que pode trazer resultados mais robustos é aquela que considera que as variáveis são independentes estatisticamente sendo esta última uma restrição mais forte que a correlação indicada anteriormente, abrindo um leque de novos modelos e mais robustos para o grupo de pesquisa.

Referências Bibliográficas

- AKAIKE, H. 1974, "New Look at Statistical-Model Identification 2", *Ieee Transactions on Automatic Control*, vol. AC19, no. 6, pp. 716-723.
- ALDROUBI, A.; UNSER, M. 1996, *Wavelets in medicine and biology* Boca Raton.
- ANDRĂ, W.; NOWAK, H. 2006, *Magnetism in Medicine : A Handbook* Wiley-vch.
- AURANEN, T. 2002, Nonparametric statistical analysis of time-frequency representations of magnetoencephalographic data, HELSINKI UNIVERSITY OF TECHNOLOGY.
- BAFFA, O., WAKAI, R. T., SOUSA, P. L., VERZOLA, R. M. M. 1995, "Fetal Heart-Rate Monitoring by Magnetocardiograms 1", *Brazilian Journal of Medical and Biological Research*, vol. 28, no. 11-12, pp. 1333-1337.
- BARROS, A. K.; CICHOCKI, A. 2001, "Extraction of specific signals with temporal structure", *Neural Computation*, vol. 13, no. 9, pp. 1995-2003.
- BARROS, A. K., CICHOCKI, A. "Neural Coding by Redundancy Reduction and Correlation", in the VII Brazilian Symposium on Neural Networks (SBRN'02), pp. 223-226.
- BARROS, A. K., CICHOCKI, A., OHNISHI, N. "Extraction of Statistically Dependent Sources with Temporal Structure", in VI Brazilian Symposium on Neural Networks (SBRN'2000), pp. 61-65.
- BOARDMAN, A., SCHLINDWEIN, F. S., ROCHA, A. P., LEITE, A. 2002, "A study on the optimum order of autoregressive models for heart rate variability", *Physiological Measurement*, vol. 23, no. 2, pp. 325-336.
- CALHOUN, V. D., ADALI, T., STEVENS, M. C., KIEHL, K. A., PEKAR, J. J. 2005, "Semi-blind ICA of fMRI: a method for utilizing hypothesis-derived time courses in a spatial ICA analysis", *Neuroimage*, vol. 25, no. 2, pp. 527-538.
- CARNEIRO, A. A. O., FERREIRA, A., MORAES, E. R., DE ARAUJO, D. B., SOSA, M., BAFFA, O. 2000, "Biomagnetismo: Aspectos Instrumentais e Aplicações", *Revista Brasileira de Ensino de Física*, vol. 22, pp. 324-338.
- CICHOCKI, A., RUTKOWSKI, T., SIWEK, K. "Blind signal extraction of signals with specified frequency band", in 12th IEEE Workshop on Neural Networks for Signal Processing, 2002, pp. 515-524.
- COMANI, S., MANTINI, D., PENNESI, P., LAGATTA, A., CANCELLIERI, G. 2004, "Independent component analysis: fetal signal reconstruction from magnetocardiographic recordings", *Computer Methods and Programs in Biomedicine*, vol. 75, no. 2, pp. 163-177.
- COMON, P. 1994, "Independent Component Analysis, A New Concept", *Signal Processing*, vol. 36, no. 3, pp. 287-314.
- DE ARAUJO, D. B. 2002, SOBRE NEUROIMAGENS FUNCIONAIS POR MAGNETOENCEFALOGRAFIA E RESSONÂNCIA MAGNÉTICA: NOVOS MÉTODOS E APLICAÇÕES, DFM.
- DE ARAUJO, D. B., BARROS, A. K., ESTOMBELO-MONTESCO, C., ZHAO, H., DA SILVA, A. C. R., BAFFA, O., WAKAI, R., OHNISHI, N. 2005, "Fetal source extraction from magnetocardiographic recordings by dependent component analysis", *Physics in Medicine and Biology*, vol. 50, no. 19, pp. 4457-4464.
- ESTOMBELO-MONTESCO, C. A., DE ARAUJO, D. B., SILVA FILHO, A. C. R., MORAES, E. R., BARROS, A. K., WAKAI, R. T., BAFFA, O. Dependent component analysis for the

- magnetogastrographic detection of human electrical response activity. *Physiological Measurement* 28, 1-16. 2007a. Ref Type: Journal (Full)
- ESTOMBELO-MONTESCO, C. A., STURZBECHER, M., BARROS, A. K., BAFFA, O., DE ARAUJO, D. B. "A new data-driven analysis for functional MRI", in *Workshop on Cerebral Perfusion and Brain Function: Novel Techniques and Applications*, Salvador, Bahia.
- FERREIRA, A. J. 2002, *Aplicação da análise em componentes independentes na compressão de imagem*, Engenharia Electrotécnica e de Computadores, Univ. Técnica de Lisboa.
- GLOVER, G. H. 1999, "Deconvolution of impulse response in event-related BOLD fMRI", *Neuroimage*, vol. 9, no. 4, pp. 416-429.
- HYVARINEN, A. OJA, E. 1997, "A fast fixed-point algorithm for independent component analysis", *Neural Computation*, vol. 9, no. 7, pp. 1483-1492.
- HYVARINEN, A. OJA, E. 2000, "Independent component analysis: algorithms and applications", *Neural Networks*, vol. 13, no. 4-5, pp. 411-430.
- IRIMIA, A., RICHARDS, W. O., BRADSHAW, L. A. 2006, "Magnetogastrographic detection of gastric electrical response activity in humans", *Physics in Medicine and Biology*, vol. 51, no. 5, pp. 1347-1360.
- JAMES, C. J. HESSE, C. W. 2005, "Independent component analysis for biomedical signals", *Physiological Measurement*, vol. 26, no. 1, p. R15-R39.
- KAUFMAN, L., KRAMER, D. M., CROOKS, L. E., ORTENDAHL, D. A. 1989, "Measuring Signal-To-Noise Ratios in Mr Imaging", *Radiology*, vol. 173, no. 1, pp. 265-267.
- LIANG, H., LIN, Z., MCCALLUM, R. W. 2000, "Artifact reduction in electrogastrogram based on empirical mode decomposition method", *Medical & Biological Engineering & Computing*, vol. 38, no. 1, pp. 35-41.
- MADISETTI, V. K. WILLIAMS, D. B. 1998, *The digital signal processing handbook*.
- MARPLE, L. S. 2005, *Digital Spectral Analysis with Applications in C, FORTRAN, and MATLAB* Pearson Higher Education.
- MINTCHEV, M. P. ELECTRICAL PHENOMENA IN THE HUMAN STOMACH. Department of Electrical and Computer Engineering, University of Calgary, (Canada) . 2007. 2007. Ref Type: Electronic Citation
- MORAES, E. R., TRONCON, L. E. A., BAFFA, O., OBA-KUNYIOSHI, A. S., WAKAI, R., LEUTHOLD, A. 2003, "Adaptive, autoregressive spectral estimation for analysis of electrical signals of gastric origin", *Physiological Measurement*, vol. 24, no. 1, pp. 91-106.
- RIOUL, O. VETTERLI, M. Wavelet and signal processing. *IEEE Signal Processing Magazine* 8[4], 14-38. 1991. Ref Type: Magazine Article
- SADOWSKY, J. 1996, "Investigation of signal characteristics using the continuous wavelet transform", *Johns Hopkins Apl Technical Digest*, vol. 17, no. 3, pp. 258-269.
- SPYERS-ASHBY, J. M., BAIN, P. G., ROBERTS, S. J. 1998, "A comparison of fast Fourier transform (FFT) and autoregressive (AR) spectral estimation techniques for the analysis of tremor data", *Journal of Neuroscience Methods*, vol. 83, no. 1, pp. 35-43.
- STURZBECHER, M. 2006, *Detecção e caracterização da resposta hemodinâmica pelo desenvolvimento de novos métodos de processamento de Imagens Funcionais por Ressonância Magnética*, Faculdade de Filosofia Ciências e Letras de Ribeirão Preto da Universidade de São Paulo – Área de Concentração: Física Aplicada a Medicina e Biologia.
- SUZUKI, K., KIRYU, T., NAKADA, T. 2002, "Fast and precise independent component analysis for high field fMRI time series tailored using prior information on spatiotemporal structure", *Human Brain Mapping*, vol. 15, no. 1, pp. 54-66.
- TALLON-BAUDRY, C., BERTRAND, O., DELPUECH, C., PERNIER, J. Oscillatory gamma-Band (30-70 Hz) Activity Induced by a Visual Search Task in Humans. *The Journal of Neuroscience*

17[2], 722-734. 1997. Ref Type: Journal (Full)

WAKAI, R. T., STRASBURGER, J. F., LI, Z., DEAL, B. J., GOTTEINER, N. L. 2003, "Magnetocardiographic rhythm patterns at initiation and termination of fetal supraventricular tachycardia", *Circulation*, vol. 107, no. 2, pp. 307-312.

Computação Afetiva personalizando interfaces, interações e recomendações de produtos, serviços e pessoas em ambientes computacionais

Maria Augusta Silveira Netto Nunes

Resumo. Este capítulo descreve o trabalho que vem sendo realizado no Departamento de Ciência da Computação da Universidade Federal de Sergipe no que tange o uso de Computação Afetiva e Sistemas de Recomendação. A gama de trabalhos aqui apresentados representam o estado da arte e o estado da técnica das pesquisas científicas e tecnológicas orientadas e/ou co-orientadas pela prof. Maria Augusta S. N. Nunes. O capítulo também, abre perspectivas considerando a prospecção tecnológica internacional, incentivando que a produção acadêmica e tecnológica de qualidade consiga de forma efetiva atingir o empresariado produtor de software, especialmente no que tange o *e-commerce* e *e-services*, no Brasil.

Palavras-chave: Computação Afetiva. Sistemas de Recomendação. Personalidade. Emoção. *PersonalityML*.

1 Introdução

Estudos recentes de psicólogos, neurologistas, antropólogos e cientistas computacionais (Damásio, 1994), (Simon, 1983), (Picard, 1997), (Trappl et al, 2003), (Thagard, 2006) e (Nunes, 2009) têm provado o quão importante os aspectos psicológicos humanos, tais como emoção e personalidade, são no processo de tomada de decisão humana.

Os mesmos estudos provam que esses aspectos sutis e inerentes a personalidade humana influenciam de maneira efetiva e particular suas interações interpessoais potencializando a personalização na interação humano-humano podendo substancialmente favorecer aspectos de processos comerciais convencionais na oferta de produtos e serviços no mundo real. Alguns estudos (Reeves and Nass, 1996) têm sido conduzidos indicando que os humanos respondem psicologicamente a computadores e outras mídias como se esses fossem, também, humanos.

Considerando esse aspecto, não importa que recurso computacional o computador estará usando, entretanto, em todos os casos, o computador estará, potencialmente, tomando decisões e trabalhando com as pessoas e para as pessoas. Assim, para o computador, o entendimento da natureza psicológica humana é extremamente relevante e necessária para que o mesmo

possa melhorar sua compreensão do ser com quem interage, melhorando, assim, o nível de personalização e otimização dos seus processos de tomada de decisão visando potencializar a interação humano-computador e conseqüentemente a personalização de ambientes comerciais também em ambientes *web* por meio de Sistemas de Recomendação em *e-commerce* e *e-services*, por exemplo.

Considerando a sobrecarga de informação disponibilizadas na *web* dificilmente a personalização de informações, produtos e serviços tem se dado de forma efetiva no Brasil. Na Europa e Estados Unidos esse problema tem sido contornado pelo uso efetivo de Sistemas de Recomendação que manipulam a grande massa de informação disponível na *web* filtrando o que realmente interessa ao usuário de *e-commerce* e *e-services*.

Dessa forma, a *web* brasileira vem perdendo um grande potencial mercadológico porque o empresariado vem negligenciando esse aspecto. Há uma estranha contradição nessa questão, pois a Academia brasileira produz ciência e tecnologia suficiente para inovar as técnicas utilizadas em *e-commerce* e em *e-services* no Brasil, entretanto o conhecimento produzido é subutilizado pelo empresariado brasileiro, o que acaba acarretando o déficit de nossa tecnologia comercial se comparado a Europa e Estados Unidos.

Assim, esse capítulo propõe diminuir esse déficit disponibilizando um portfólio dos trabalhos no que tange tanto o estado da arte como o da técnica dos trabalhos em andamento que direcionam a área de Computação Afetiva e Sistemas de Recomendação, principalmente na linha de *e-commerce* e *e-services*.

1.1 Como ler o capítulo

Como descrito anteriormente esse capítulo é um conjunto de trabalhos de IC, IT e TCC desenvolvidos no DCOMP, todos eles orientados e/ou co-orientados pela autora. Os trabalhos de IC, IT e TCC que deram origem ao capítulo em apresentação são oriundos de propostas da autora já descritas como perspectivas futuras da sua Tese de Doutorado (Nunes, 2008) e também contemplados em seu projeto de Estágio Probatório¹. A metodologia de apresentação dos trabalhos seguirá por seções autocontidas indicando a origem da autoria (trabalho de IC, IT ou TCC desenvolvidos sob a guarda do projeto guarda-chuva de Estágio Probatório da autora).

1.2 Organização do texto

Na seção 2, a Computação Afetiva é descrita, disserta-se sobre personalidade, suas abordagens e alguns instrumentos de mensuração, seguindo-se pela descrição de emoção e alguns instrumentos de mensuração.

Logo, disserta-se sobre identidade e como ela pode ser modelada, padronizada e implementada em computadores. A seguir, introduz-se Sistemas de Recomendação e Sistemas de Combinação Social, incluindo perspectivas do uso de personalidade.

Finalmente, apresenta-se os trabalhos em desenvolvimento com parceria de outras instituições, seguido pelos potenciais cenários de aplicação futuros, apresentando-se por fim, na seção 3, a conclusão e trabalhos futuros.

Note que o objetivo desse capítulo não é esgotar a teoria sobre os assuntos tratados, a perspectiva do texto é apresentar como a equipe da autora tem contribuído nesses assuntos, dissertando-se brevemente sobre os mesmos e apresentando as contribuições da equipe. Caso o leitor deseje aprofundar-se sobre os assuntos por favor dirija-se às referências citadas em cada seção.

1 “E-Psico Um Sistema de Recomendação baseado em Traços de personalidade: influenciando a tomada de decisão computacional no contexto de E-services <

2 Áreas de pesquisa relacionadas

2.1 Computação Afetiva

Desde a década de 70, cientistas computacionais, como citado na seção anterior, principalmente da área de Computação Afetiva buscam modelar e implementar aspectos psicológicos humanos em ambientes computacionais. Na Computação Afetiva estuda-se como os computadores podem reconhecer, modelar e responder às emoções humanas (dentre outros aspectos) e, dessa forma, como podem expressá-las através de uma interface/interação computacional (Picard, 1997). Acredita-se que permitindo que agentes artificiais expressem/compreendam fisiológica e verbalmente uma emoção e/ ou personalidade, em uma interação humano-computador, é possível induzir e despertar emoções em humanos.

O principal objetivo de se promover esse interfaceamento afetivo é contribuir para o aumento da coerência, consistência, predicabilidade e credibilidade das reações e respostas computacionais providas durante a interação humana via interface humano-computador.

2.1.1 Personalidade

2.1.1.1 Abordagens (SOUZA and NUNES, 2011)

Não existe na literatura em psicologia uma definição única para a personalidade. De fato, talvez uma conceituação que se aproxime daquela em que o termo é utilizado corriqueiramente seja trazida por Schultz (1990), que o define como um conjunto permanente e único de características que produzem respostas que não se modificam ainda que em diferentes situações. No entanto, embora essa operacionalização possa ser utilizada de forma razoavelmente genérica, as definições de personalidade são muito melhor compreendidas se encaradas a partir da teoria (ou abordagem) da Personalidade à qual pertencem.

As Teorias da Personalidade foram criadas para facilitar a compreensão acerca de si mesmo e dos outros (Cartwright, 1979). Mas existe uma pluralidade de conceitos com pontos de vistas epistemológicos diferentes, onde cada abordagem teórica possui suas próprias definições, origem e desenvolvimento acerca da personalidade. Afinal, pensar em personalidade é mesmo pensar em um construto complexo. Hall *et al* (1998) separam as diversas teorias existentes acerca da personalidade em quatro grandes grupos de abordagens: (1) ênfase na psicodinâmica; (2) ênfase na realidade percebida; (3) ênfase na aprendizagem; e (4) ênfase na estrutura. Os trabalhos nessa seção referem-se a esta última, pois as teorias com ênfase na estrutura foram as que mais influenciaram o desenvolvimento de testes para mensuração da personalidade (maiores informações sobre as outras ênfases em (Santos e Nunes, 2011).

As teorias com ênfase na estrutura possuem, como sua própria categorização deixa claro, uma preocupação central com a forma como se estrutura a personalidade. Isto é, ainda que a dinâmica e o desenvolvimento da personalidade também recebam alguma atenção, a característica definidora das teorias nesta abordagem é a busca por uma taxonomia, por um conjunto sistemático de características que possa ser usado para resumir a personalidade de um indivíduo. Três dos grandes estudiosos da personalidade que desenvolveram teorias que se encaixam nessa abordagem foram Henry Murray, Gordon Allport e Raymond Cattell (Hall *et al*, 1998).

Murray desenvolveu uma teoria da personalidade conhecida como Personologia. O autor enfatizava que a personalidade é uma abstração formulada pelos teóricos e não simplesmente uma descrição do comportamento do indivíduo. Para Murray, a personalidade de um indivíduo se refere a uma série de eventos que idealmente abrangem toda a sua vida. A definição da personalidade

deveria, portanto, refletir os elementos duradouros e recorrentes do comportamento, bem como os elementos novos e únicos.

A teoria da personalidade de Murray parte de alguns postulados. O teórico afirmava que a personalidade funcionaria como agente organizador cujas funções seriam integrar os conflitos e as limitações aos quais o indivíduo está exposto, satisfazer suas necessidades e fazer planos para a conquista de metas futuras. Assim, os comportamentos humanos estariam relacionados uns aos outros de forma temporal através de ações que desembocam na tentativa de satisfazer dadas necessidades, que são acompanhadas de um determinado sentimento ou emoção (Hall *et al*, 1998).

Murray *et al* (1938) definem uma necessidade como um potencial ou prontidão para responder de determinada forma sob dadas circunstâncias. Assim, a necessidade representaria o fato de que certa tendência deve ser recorrente no comportamento do indivíduo. As necessidades que guiam o comportamento e formam a personalidade foram categorizadas pelos autores em primárias (de base biológica), como fome, sede, sexo e evitar a dor; e secundárias (derivadas daquelas de base biológica ou inerentes à natureza psicológica), como agressão, autonomia, reconhecimento, entre outras.

Necessidades mais fortes em um indivíduo se expressariam mais frequentemente e levariam a determinados comportamentos recorrentes – dos quais, por fim, poder-se-ia abstrair a sua personalidade. Allport foi o primeiro teórico a trabalhar o conceito de traços psicológicos para embasar uma teoria da personalidade, conhecida como Teoria do Traço (Allport & Allport, 1921; Allport, 1927). Para o autor, cada ser humano possui traços de personalidade comuns e individuais, e a intensidade com que esses traços se expressam é diferente para cada indivíduo. Por exemplo, duas pessoas podem ter um traço de personalidade “calmo”, e cada uma delas ter um nível de “calma” diferente. Essas diferenças seriam fruto da história de vida de cada um e das influências externas e ambientais sobre a pessoa. No entanto, o traço latente seria o mesmo, e a personalidade dos indivíduos poderia, portanto, ser descrita com base nos traços.

Com base nos estudos da linguagem natural, seguindo uma hipótese de que analisar a linguagem ajudaria a entender a personalidade, Allport e Odbert (1936, como citado por Hutz *et al*, 1998; Loehlin, 1992) selecionaram um amplo número de palavras – 17.953 palavras, entre traços comuns e individuais –, em sua maioria adjetivos, que fossem capazes de descrever traços de personalidade como, por exemplo, os adjetivos *agressivo* e *sociável*.

O uso de descritores de traços da linguagem natural para identificar fatores que permitam entender características de personalidade decorre da idéia léxica que afirma que as diferenças individuais mais significativas nas interações diárias das pessoas são codificadas na linguagem (Goldberg, 1982, como citado por Hutz *et al*, 1998). Isto é, se um traço de personalidade é saliente, capaz de gerar diferenças individuais socialmente relevantes, as pessoas notarão essa característica e, já que ela é importante, vão querer falar sobre ela. Como consequência, uma palavra ou expressão terminará sendo inventada para descrevê-la.

Cattell desenvolveu a Teoria de Traço Fatorial-Analítica, uma tentativa de elaborar uma “tabela periódica” de elementos da personalidade. De início, o autor reduziu a lista criada por Allport a 4.500 adjetivos que descreveriam os traços de personalidade. Com o avanço de suas pesquisas, Cattell destilou essa lista e novamente reduziu-a drasticamente, dessa vez para 171 adjetivos, os quais foram agrupados empiricamente e conceitualmente em 35 pares bipolares (Goldberg, 1990; Hutz *et al*, 1998; Loehlin, 1992). Os trabalhos de Cattell e de outros autores serviram de base para várias análises fatoriais, que em grande parte convergiram em uma solução similar de cinco fatores. Essa solução fatorial ficou popularmente conhecida como *o Big Five*, expressão traduzida para o português como o modelo dos Cinco Grandes Fatores (CGF).

O modelo dos Cinco Grandes Fatores (CGF), ou *Big Five*, da personalidade é uma versão

moderna da Teoria do Traço. Este modelo representa um avanço conceitual e empírico no campo da personalidade, descrevendo dimensões humanas básicas de forma consistente e replicável (Hutz *et al*, 1998). Norman (1963, como citado por Loehlin, 1992) denominou os Cinco Fatores de *Surgency*, *Agreeableness*, *Conscientiousness*, *Emotional Stability* e *Culture*. Atualmente, diversas outras denominações já foram sugeridas. Em língua portuguesa, os Cinco Grandes Fatores podem ser definidos como Extroversão, Amabilidade (ou Socialização), Conscientização (ou Realização), Neuroticismo (ou Instabilidade Emocional) e Abertura (ou Abertura à mudança) (Hutz *et al*, 1998; Berger, 2003).

Cada um dos fatores resume aspectos distintos do comportamento que tendem a se apresentar juntos empiricamente (Loehlin, 1992). O fator Extroversão caracteriza pessoas sensíveis, assertivas, ativas e impulsivas. O fator Socialização, pessoas gentis, úteis e despreocupadas, com comportamento pró-social. O fator Realização representa pessoas organizadas e deliberadas, com comportamento de responsabilidade social. O Neuroticismo caracteriza pessoas ansiosas, mal-humoradas e autopunitivas. Por fim, o fator Abertura reflete pessoas criativas, curiosas, abertas a novas experiências e com traços de facilidade intelectual (Berger, 2003; Loehlin, 1992).

A tabela 1, adaptada de Loehlin (1992), apresenta alguns adjetivos que caracterizam os cinco fatores. McAdams (1992 como citado por Hutz *et al*, 1998) observa que os Cinco Fatores se referem a informações fundamentais que geralmente se quer ter sobre pessoas com quem se interage. As pessoas costumam querer saber se o estranho, o visitante ou o aprendiz com quem vão interagir é: 1) ativo e dominante ou passivo e submissivo; 2) socialmente agradável ou desagradável, amigável ou frio, distante; 3) responsável ou negligente; 4) louco, imprevisível ou “normal”, estável; 5) esperto ou tolo, aberto a novas experiências ou desinteressado pelo que foge ao cotidiano.

Segundo as teorias estruturais, são os traços de personalidade que levam os indivíduos a procurar, interpretar e depois reagir aos eventos da vida de maneira própria. Durante a idade adulta, por exemplo, a principal fonte de continuidade do desenvolvimento é a estabilidade da personalidade. E, em geral, os Cinco Grandes Fatores permanecem estáveis nessa fase da vida. A classificação alta ou baixa de um indivíduo em cada um dos Cinco Fatores é determinada pelas influências interativas dos genes, da cultura, da forma inicial de criação e das experiências e escolhas feitas no final da adolescência e no início da idade adulta.

Tabela 1: Adjetivos característicos dos Cinco Grandes Fatores de Personalidade.

| | Extroversão | Socialização | Realização | Neuroticismo | Abertura |
|-----------------------|--------------------|---------------------|-------------------|---------------------|-----------------|
| Pólo do rótulo | Ativo | Altruísta | Confiável | Ansioso | Artístico |
| | Aventureiro | Amigável | Consciente | Apreensivo | Curioso |
| | Barulhento | Carinhoso | Eficiente | Emotivo | Engenhoso |
| | Energético | Confiante | Minucioso | Instável | Experto |
| | Entusiástico | Cooperativo | Organizado | Neuroso | Imaginativo |
| | Exibido | Gentil | Prático | Preocupado | Inteligente |
| | Sociável | Sensível | Preciso | Temeroso | Original |
| Tagarela | Simpático | Responsável | Tenso | Sofisticado | |
| Pólo oposto | Acanhado | Antipático | Desatento | Calmo | Comum |
| | Introvertido | Brigão | Descuidado | Contido | Simple |
| | Quieto | Bruto | Desorganizado | Estável | Superficial |
| | Reservado | Crítico | Distraído | Indiferente | Tolo |
| | Silencioso | Frio | Imprudente | Sereno | Trivial |
| | Tímido | Insensível | Irresponsável | Tranquilo | Vulgar |

Fonte: (Souza and Nunes, 2011).

A força desses Fatores pode flutuar antes dos 30 anos, porque o contexto social e as escolhas pessoais da adolescência e do início da idade adulta despertam novos padrões de personalidade

que não eram manifestos na primeira infância. Entretanto, por volta dos 30 anos, os Cinco Fatores geralmente se estabilizam por completo, e assim permanecem por toda a vida (Berger, 2003).

Cabe aqui, no entanto, a ressalva de que a estabilidade da personalidade não é regra imutável. Mudanças significativas no ambiente do qual faz parte o adulto, especialmente quando as circunstâncias da vida são drasticamente alteradas, são de fato capazes de produzir modificações na personalidade. O divórcio ou morte do cônjuge, um novo casamento ou uma nova carreira, o abandono de um antigo vício, a transferência para uma cultura estrangeira ou uma conversão religiosa são alguns exemplos de fatores sócio-ambientais capazes de exercer forte influência sobre a personalidade, mesmo na idade adulta (Berger, 2003).

2.1.1.1.1 Abordagem de Traços

Como descrito anteriormente na Psicologia não existe um consenso para a definição de Personalidade sendo que a mesma pode ser definida segundo muitas abordagens. Uma abordagem bastante interessante é a abordagem de Traços de Personalidade que permite diferenciar psicologicamente pessoas usando traços mensuráveis e conceituáveis factíveis de modelagem e implementação em computadores (Nunes, 2009).

Os Traços de Personalidade, como já descrito, foram historicamente definidos por Allport (1927) que criou 17.953 Traços (Traços “comuns” e “individuais”) para descrever a personalidade de um indivíduo. Considerando que a maioria das diferenças individuais (representadas pelos Traços individuais) eram insignificantes nas interações diárias humanas, objetivando limitar exponencialmente o número de definições de Traços, os pesquisadores assumiram que todos os homens eram identificáveis “como algum outro homem”.

Considerando isso, pesquisadores reduziram mais de 99% dos Traços, pois eles consideraram que somente cinco fatores eram replicáveis. Como resultado, o modelo *Big Five* (John and Srivastava, 1999) foi criado. Porém, mesmo que o *Big Five* representasse grande eficiência na representação da estrutura de Personalidade, ele não garantia exaustivamente todas as dimensões de personalidade. Dessa forma, *facetats* também foram criadas e usadas pelos psicólogos para dotar o *Big Five* de características mais detalhadas (Goldberg *et al*, 2006). Para extrair Traços humanos (baseado nas dimensões do *Big Five* e suas respectivas *facetats*) psicólogos geralmente usam diferentes instrumentos de mensuração.

2.1.1.2 Instrumentos de Mensuração

2.1.1.2.1 Testes de Personalidade

Um teste psicológico pode ser definido como “um conjunto de tarefas pré-definidas, que o sujeito precisa executar numa situação geralmente artificializada ou sistematizada, em que o seu comportamento na situação vai ser observado, descrito e julgado, e essa descrição é geralmente feita utilizando-se números” (Pasquali, 2001, p. 18).

Os testes, baseados em pesquisas empíricas, são capazes de descrever diferenças psicológicas entre os indivíduos. Como já referido, a teoria da personalidade proveniente da abordagem com ênfase na estrutura foi a que mais influenciou o desenvolvimento dos testes para mensuração da personalidade. Um teste de personalidade com base nessas teorias geralmente revela um conjunto de traços do sujeito que o diferencia dos outros. Existem inúmeros testes e inventários que se propõem a investigar e avaliar traços de personalidade e, em geral, eles tomam por base uma das muitas teorias que se sustentam na abordagem com ênfase na estrutura. Um exemplo é o Inventário Fatorial de Personalidade (IFP). O IFP é um instrumento baseado no *Edwards*

Personal Personality Schedule (EPPS), desenvolvido por Allen L. Edwoods, baseado na teoria das necessidades básicas formuladas por Murray (Murray *et al*, 1938).

Este inventário traz diversas afirmativas inseridas em uma escala tipo *Likert*, composta de sete pontos. Os pontos da escala definem o item progressivamente como “nada característico” ao sujeito do teste até “totalmente característico”. O inventário visa avaliar o indivíduo em 15 necessidades ou motivos psicológicos, expostos na tabela 2. Essas necessidades indicariam taxonomicamente a personalidade do indivíduo.

Tabela 2: Necessidades da Teoria da Personalidade de Murray avaliadas pelo IFP.

| Necessidades | Sujeito com Alto Escore |
|--------------------|---|
| Assistência | Tem grandes desejos e sentimentos de piedade, compaixão e ternura. |
| Intracepção | É intraceptivo, se deixa conduzir por sentimentos e inclinações difusas. |
| Afago | Busca apoio e proteção, desejo de satisfação, de afago, apoio e proteção. |
| Deferência | Demonstra respeito, admiração e reverência. |
| Afiliação | Procura dar e receber afeto dos amigos. É confiante e de boa vontade. |
| Dominância | Possui autoconfiança, desejo de controlar, influenciar ou dirigir. |
| Denegação | Tende a submeter-se passivamente à força externa, aceitar desafios. |
| Desempenho | Mostra ambição e empenho, dominação, manipulação e organização. |
| Exibição | É vaidoso, deseja impressionar, ser ouvido e visto. |
| Agressão | Tem tendência à raiva, irritação e ódio. |
| Ordem | Costuma manter a ordem, limpeza, organização, equilíbrio e precisão. |
| Persistência | Leva a cabo qualquer trabalho iniciado, por mais difícil que pareça. |
| Mudança | Tem desejo por novidade, aventura. Não constrói ligação permanente com lugares, objetos ou pessoas. |
| Autonomia | Possui alta resistência à coerção e à oposição |
| Heterossexualidade | Demonstra interesse em manter relações com indivíduos do sexo oposto. |

Fonte: (Souza and Nunes, 2011).

Nunes (2008) cita outros exemplos de testes como o 16PF (Cattell's 16 Personality Factors Questionnaire) e o 6FPQ (Six Factor Personality), ambos baseados nos trabalhos de Cattell, e diversos outros baseados no modelo do *Big Five* ou Cinco Grandes Fatores (CGF). Para o presente projeto, foram elencados testes baseados no *Big Five*, pois as pesquisas mais recentes na área da testagem em personalidade parecem ter estabelecido um consenso entre os teóricos quanto à solidez dos Cinco Grandes Fatores, que têm sido considerados como o melhor modelo estrutural disponível na atualidade para a descrição da personalidade (Hutz *et al*, 1998).

Pesquisas longitudinais extensivas e transversais entre homens e mulheres em diversos países e etnias têm evidenciado esses cinco grupos básicos de traços de personalidade como sempre presentes (Hutz *et al.*, 1998) e, por isso, optou-se por testes com base nesta teoria para o uso em computadores. Foram escolhidos o *Neo-International Personality Item Pool* (NEO-IPIP) e o *Ten-Item Personality Inventory* (TIPI).

O IPIP é um consórcio de domínio público, definido como uma colaboração científica para o desenvolvimento de medidas avançadas de personalidade e outras diferenças individuais (Goldberg, 1999; Goldberg *et al*, 2006). Este inventário utiliza a estrutura fatorial do *Big Five* para avaliar a personalidade. O Inventário NEO-IPIP é um inventário de 300 itens criado por meio do consórcio IPIP, e é uma versão gratuita do inventário *Neo Personality Inventory* (NEO-PI-R), o qual é descrito como um dos inventários comerciais mais robustos, conhecidos e

validados no mundo, e utiliza o modelo dos Big Five (Johnson, 2000a). O Inventário NEO-IPIP foi utilizado e validado por Johnson (2000b, 2005), As 300 questões do NEO-IPIP são divididas igualmente entre os Cinco Grandes Fatores, e cada um dos Fatores se subdivide em seis facetas. Às afirmativas nos itens do inventário, o sujeito respondente atribui um valor de concordância do quanto elas o representam em uma escala tipo *Likert* de cinco pontos.

O TIPI é uma inventário bastante resumido, composto por dez itens, também de domínio público. Gosling *et al* (2003) aplicaram este inventário a estudantes universitários americanos e encontraram uma correlação forte com as escalas no NEO-PI-R. No entanto, os autores alertam que o TIPI é um instrumento de mensuração bastante curto, que deve ser utilizado apenas se a personalidade não for o tópico primário de interesse na pesquisa, já que uma mensuração breve é mais frágil quanto a suas propriedades psicométricas.

Ambos os inventários (NEO-IPIP e TIPI), embora de domínio público, são apresentados em língua inglesa. A equipe da autora desenvolveu uma versão brasileira (língua portuguesa do Brasil) dos testes TIPI e NEO-IPIP. A tradução foi realizada pela equipe composta por uma tradutora, Danilo Reinert e Diogo Araújo Sousa. (A versão traduzida dos inventários na íntegra encontra-se em (Sousa e Nunes, 2011) ou no *software* desenvolvido por Daniel Moraes apresentado a seguir (Nunes *et al*, 2010b).

2.1.1.2.1.1 Personality Inventory PV1.0 (Portuguese Version) (NUNES *et al*, 2010b)

O *Personality Inventory* PV1.0 (Nunes *et al*, 2010b) torna-se então uma opção interessante a ser utilizada como ferramenta de entrada de dados explícita para obtenção dos traços de personalidade dos usuários. Este teste possui um *feedback* positivo em mais de 99% dos casos, como descrito por Johnson (2005). Dessa forma, a equipe da autora propôs uma inovação em termos de interface ao NEO-IPIP e TIPI, oferecendo mais usabilidade ao ambiente proposto originalmente por Johnson e Gosling.

(i) *Detalhes da modelagem:*

As medições de Personalidade extraídas seguem os padrões propostos por Johnson (2000a). Para que os resultados pudessem ser mantidos em um local seguro e persistente, optou-se por uma aplicação *web*, com interface *web* e um banco de dados virtual, localizado em nossos servidores. A interface foi desenvolvida de forma simplificada e objetiva para que os resultados fossem obtidos com eficácia, como apresentado na Figura 1.



Figura 1: Interface inicial do ambiente.

Fonte: Nunes *et al* (2010a).

Conforme a Figura 1, para que a medição de Personalidade seja feita, o usuário precisa criar

uma conta e responder pelo menos um dos questionários disponíveis, como apresentado na Figura 2.



Figura 2: Medidas Disponíveis.
Fonte: Nunes *et al* (2010).

Após a seleção do inventário o usuário deve responder o questionário como apresentado na Figura 3.

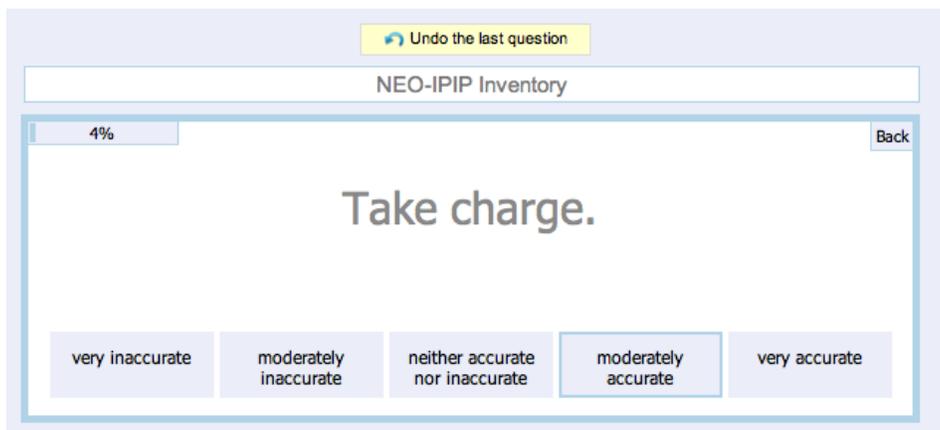


Figura 3: Questão 1 do Teste NEO-IPIP.
Fonte: Nunes *et al* (2010).

Uma vez respondido o questionário do NEO-IPIP, por exemplo, os valores atribuídos a cada uma das questões respondidas são utilizados para calcular o resultado. No cálculo, o resultado é normalizado e é atribuído um valor entre 1-100 para cada um dos itens do Big Five, bem como para as suas facetas. Note que somente então o relatório descritivo dos traços de personalidade do usuário é gerado e disponibilizado.

Na Figura 4, parte do relatório descritivo dos traços de personalidade do usuário, extraído do inventário de Personalidade NEO-IPIP é apresentado. Uma vez visualizado seu prognóstico, o usuário poderá re-visualizar o resultado através da própria aplicação ou exportá-la via *PersonalityML* (descrito nas próximas seções desse capítulo).

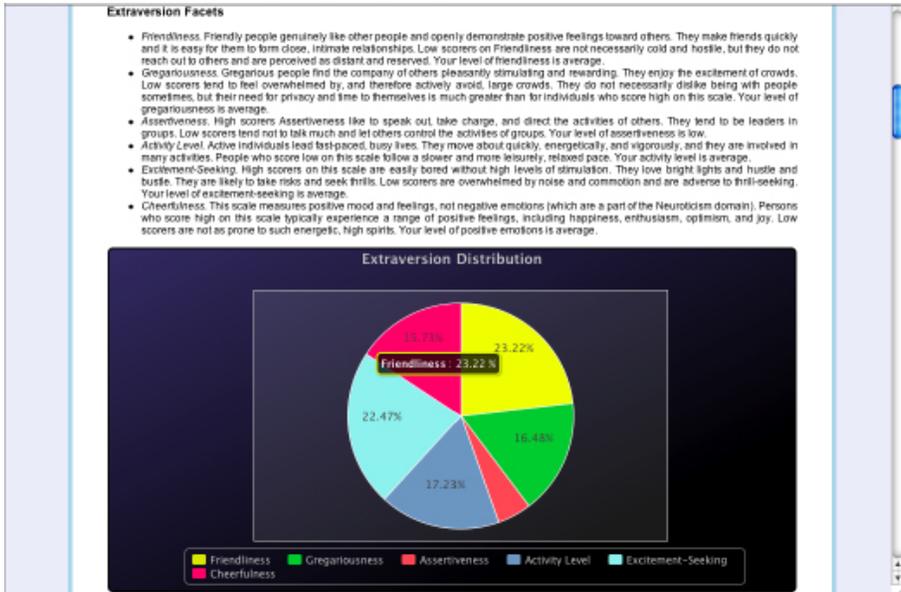


Figura 4: Relatório descritivo parcial do NEO-IPIP.
Fonte: Nunes *et al* (2010).

Detalhes da implementação:

A solução para o problema da medição de personalidade envolve basicamente três tarefas: (1) Cadastro do Usuário, (2) Aplicação de Teste de Personalidade, (3) Cálculo dos Resultados com base no Big Five e normalização precisa. As tarefas são apresentadas na Figura 5.

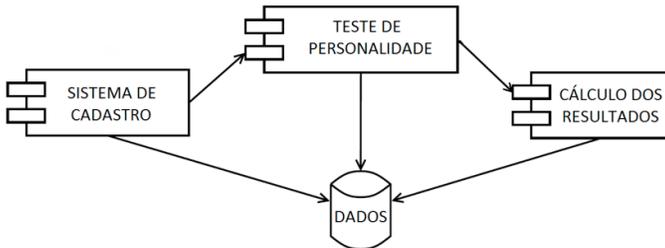


Figura 5: Diagrama dos Módulos do Personality Inventory PV1.0
Fonte: (Nunes *et al*, 2010)

A aplicação foi desenvolvida com o uso de tecnologias bastante utilizadas no desenvolvimento de aplicações *web*, entre elas, HTML, CSS, Javascript, Ajax, PHP e Base de Dados MySQL.

- A aplicação foi segmentada basicamente em três camadas: Camada de Interface: foi desenvolvida com o uso de HTML, CSS e JavaScript. Esta camada é responsável pela interação homem-máquina, ou seja, é ela que apresenta as janelas da aplicação, as imagens, os textos, os botões e tudo mais que esteja relacionado à interface com o usuário.
- Camada do meio – *Middleware*: é a camada que gerencia a comunicação entre a Camada de Interface e a Camada de Dados. A Camada de Interface através de requisições HTTP, com o uso de Ajax, solicita uma determinada informação da base de dados e a *Middleware* retorna os dados requeridos em forma de JSON. Esta camada foi desenvolvida em PHP.
- A Camada de Dados: foi desenvolvida com o uso do MySQL, que é um Sistema de

Gerenciamento de Banco de Dados (SGBD) livre (com base na GPL²) e que utiliza a linguagem SQL como interface. É nesta camada que todos os dados dos usuários ficarão armazenados para futura utilização, tais como as informações de autenticação dos usuários, as suas questões respondidas, os seus resultados e tudo mais que precisar ser mantido de forma persistente na aplicação.

Note que o grande diferencial do *Personality Inventory PVI.0* é que o *software* permitirá exportar os valores referentes à personalidade de cada usuário cadastrado. A exportação é realizada através de um padrão de arquivo pré-definido em forma de XML (*PersonalityML* (Nunes *et al* 2010a; Bezerra *et al*, 2011)). Desta forma, estes dados poderão ser usados em quaisquer sistemas computacionais.

2.1.1.2.2 Teclado e Mouse (Porto *et al*, 2011)

Segundo Gosling (2008) “deixamos pistas sobre nossa Personalidade em tudo o que fazemos, em nossos objetos, onde interagimos, onde vivemos”. Sendo assim, através dos padrões de navegação, digitação, entre outros, de cada usuário, seria possível adquirir características psicológicas, através de uma abordagem implícita e transparente ao usuário. Um pensamento natural é que esta seria a melhor forma de obtenção dos traços de personalidade dos usuários, uma vez que esta abordagem exigiria menor esforço cognitivo se comparado aos tradicionais testes de Personalidade aplicados, como discutido anteriormente.

Segundo Porto *et al* (2011), existem muitos trabalhos que tentam extrair a personalidade dos usuários a partir de técnicas não intrusivas, tais como trabalhos que avaliam a escolha dos usuários para customização de algumas aplicações, a saber, Windows Media Player, ICQPlus e o Winamp, através da tecnologia de *skins*, e dependendo da escolha, pode-se prever a personalidade do usuário, fazendo comparações com a escolha de outros usuários. O objetivo inicial desse IC/TCC era usar tanto o ritmo de digitação como o uso do mouse para tentar extrair a personalidade do usuário, mas ao decorrer do trabalho, acabou-se por focar apenas no uso do teclado. A extração de traços de personalidade é realizada pelo teclado, através da plataforma desenvolvida apresentada na Figura 6 e 7.

The image shows a web form titled "PersonalKey" with the instruction "Informe os dados abaixo:". The form contains four input fields:

- E-mail:** A text input field containing "gutanunes@dcomp.ufs.br". Below the field is the label "Insira seu e-mail".
- Já fez algum curso de Digitação?:** A radio button selection field with the option "Sim" selected.
- Com que frequência você usa o computador?:** A radio button selection field with the option "Todo Dia" selected.
- Qual a sua experiência com computadores?:** A radio button selection field with the option "Menos de 2 anos" selected.

Figura 6: Tela inicial do PersonalKey.

Fonte: Porto *et al* (2011).

² “GNU General Public License (Licença Pública Geral), GNU GPL ou simplesmente GPL, é a designação da licença para *software livre* idealizada por Richard Stallman no final da década de 1980, no âmbito do projeto GNU da Free Software Foundation (FSF)” (Wikipedia 2010).

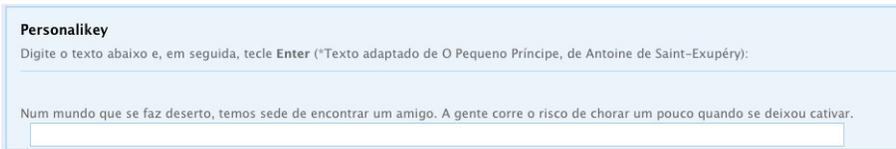


Figura 7: Interface de extração de padrões de digitação.

Fonte: Porto *et al* (2011).

O *PersonalKey* inicia o processo de extração da personalidade do usuário obtendo informações de como o usuário digita determinado texto, essa informação pode ser obtida através da latência entre teclas consecutivas (o tempo entre o apertar de uma tecla e da subsequente). Esse tempo é obtido através de um evento de teclado, o *keyPress*. O evento aciona um *timer* que é interrompido quando o evento é acionado novamente, ou seja, o *timer* conta o tempo decorrido entre uma tecla pressionada e outra, o tempo de latência.

Segundo Porto *et al* (2011), o processo de extração de traços de personalidade através do ritmo de digitação pode ser dividido em duas fases. A primeira fase é a coleta de dados, e a segunda fase é extração da personalidade através de técnicas de agrupamento de dados (*clustering*).

Na primeira fase, existe uma frase padrão que um usuário deve digitar duas vezes, como apresentado na Figura 7, enquanto o usuário está digitando o *timer* no evento *keyPress* está recolhendo os tempos de intervalos e colocando-os em um vetor, quando por fim o usuário tecla “Enter” indicando o término da digitação, os valores são armazenados num vetor chamado D. Como o usuário digita a mesma frase duas vezes, então ao final são formados dois vetores, D1 e D2.

Na segunda fase do algoritmo, intitulado *PersonalKey*, é feita a extração da personalidade utilizando técnicas de *clustering*. Quando um usuário forma os vetores D1 e D2 este é armazenado num banco de dados. Este banco de dados já foi alimentado previamente com os valores de D1 e D2, ou seja, ritmo de digitação, de outros usuários, assim como os valores dos traços de personalidade desses mesmos usuários no modelo Big Five (extraídos através do questionário NEO-IPIP implementado em português no *Personality Inventory PV1.0*, já apresentado anteriormente).

Com esses dados o algoritmo *PersonalKey* faz uma clusterização com todos os usuários, inclusive o usuário que acabou de digitar levando em conta somente os valores de ritmo de digitação. Ao fim do processo, o usuário que acabou de digitar é alocado com outros usuários que tem um ritmo de digitação similar.

A partir desse grupo, o algoritmo monta uma matriz com os valores da personalidade dos usuários, menos o do usuário que acabou de digitar, porque este valor ainda não foi inferido, obviamente. Mas tendo as personalidades dos usuários que se assemelham em ritmo de digitação, nossos testes demonstraram que é possível inferir alguns traços no usuário que acabou de digitar, são eles, Extroversão, Interesses artísticos e Autodisciplina (ainda em análise no TCC de Porto e Costa).

Ou seja, tendo a informação de como um usuário digita, este pode ser comparado com um banco de dados que contem informações de outros usuários, informações estas que são o ritmo de digitação e a personalidade do usuário, e através de técnicas de agrupamento de dados (*clustering*), o usuário é isolado em um grupo que tem o mesmo padrão de digitação, e então os traços de sua personalidade são inferidas a partir dos usuários que fazem parte do mesmo grupo.

Os testes realizados para essa abordagem tendem a confirmar que quando os usuários são agrupados levando em consideração o ritmo de digitação, os traços Extroversão (um dos cinco grandes fatores do BIG FIVE, como já apresentado anteriormente), Interesses Artísticos (que é

uma das facetas do fator Abertura a Experiências) e Autodisciplina (faceta do fator Realização) (ver sobre facetas no trabalho de (Nunes, 2008)) são correlacionados. Usuários que são agrupados num mesmo cluster tendem a ter o mesmo valor para estes traços, e, portanto usuários que não soubessem o valor desses traços, tendo sido agrupados em um cluster, poderiam ter seus valores inferidos pelos outros elementos do mesmo cluster.

2.1.2 Emoção

Considerando que os aspectos psicológicos, tais como emoção e personalidade, são importantes e influenciam o processo de tomada de decisão dos humanos; considerando ainda que a intensidade e/ou tipo de emoção despertado em um humano sofre grande influência da personalidade (mesmo que a personalidade humana não aparece explicitamente ela influencia as emoções diretamente), agora é o momento, também de introduzir emoções.

Pesquisadores em Computação Afetiva têm extraído e implementado as emoções explicitamente. Isso porque as emoções são mais facilmente mensuráveis e interpretáveis e podem influenciar diretamente na ação-interação dos usuários e tomada de decisão computacional. Na realidade, as emoções são instantâneas, têm uma vida curta e mudam constantemente, diferentemente da personalidade que é muito mais estável e, normalmente, em circunstâncias normais, imutável. Segundo trabalhos (Bianchi- Berthouze *et al*, 2002; Lisetti, 2002) é possível perceber que a personalidade é mais abrangente e implica na emoção.

Lisetti (2002) descreve um modelo complexo para representar esses aspectos psicológicos em agentes inteligentes (virtual/real) que interagem socialmente. Este modelo é composto de uma taxonomia de afeição, estado de ânimo, emoção e personalidade, como apresentado na Figura 8.

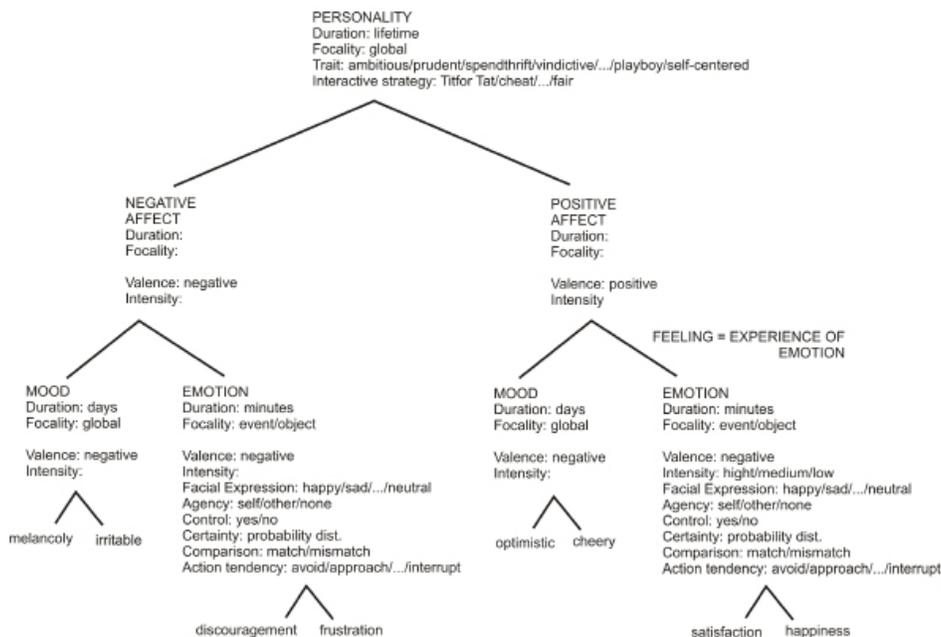


Figura 8: Modelo hierárquico de personalidade, afeição, estado de ânimo e emoção.

Fonte: Adaptado de Lisetti (2002).

Na Figura 8, apresenta-se a personalidade no topo do modelo hierárquico (o que demonstra

seu maior poder). Note que em sistemas de agentes inteligentes, os mesmos, dotados de diferentes personalidades, podem experimentar diferentes tipos e intensidades de emoções. Emoções, essas, que podem prever e influenciar as ações desse agente (real/virtual) durante sua interação em um ambiente computacional. Note, também, que a emoção está na base hierárquica do modelo e, conseqüentemente, ela não influencia os outros aspectos. Ao contrário, ela tende a ser influenciada por eles. Assim a personalidade, no topo do modelo, influencia as emoções na base.

A emoção é volátil e dinâmica, muda constantemente e afeta diretamente as ações do usuário/agente gerando atitudes positivas e/ou negativas em um ambiente. Em contrapartida, a personalidade é muito mais estável, porém muito mais difícil de ser extraída durante a interação humano-computador podendo prever as necessidades e comportamento do usuário.

Na versão do *Affective Knowledge Representation* (AKR - Representação do Conhecimento Afetivo) descrito por Lisetti (2002), as emoções são modeladas em 16 diferentes dimensões chamadas de componentes emocionais, sendo que cada emoção difere das demais quanto aos valores atribuídos a cada componente. Note que existem várias linhas divergentes que descrevem a teoria da emoção. Essas teorias são baseadas geralmente numa das seguintes abordagens: (i) Biológicas; (ii) Comportamentais; (iii) Relacionadas a fenômenos; (iv) Expressões faciais; (v) Cognitivas.

No entanto, existe um ponto importante entre todas estas diferentes teorias: há uma aceitação universal de que existe um conjunto básico de emoções (geralmente menor que dez). Segundo Lisetti (2002) a partir desse conjunto de 9 emoções básicas existem a formação de outras emoções derivadas. As emoções derivadas são formadas a partir de variações entre os atributos e/ou combinação das emoções básicas. O modelo de Lisetti (2002) é uma derivação da *OCC model* (1988) que é, ainda uma variação das emoções básicas (que são 6) definidas por Ekman (1994) e representáveis através de expressões faciais.

2.1.2.1 Instrumentos de Mensuração

Considerando que emoções são voláteis e mais facilmente extraídas em uma curta interação com o usuário alguns pesquisadores vem estudando em como extrair emoções de uma forma menos intrusiva do que os tradicionais sensores fisiológicos. Leão em seu TCC sob orientação de Leonardo N. Matos e co-orientação da autora (Leão, 2010) inicia um trabalho nesse sentido, usando expressões faciais como forma de extração da emoção do usuário, ele utiliza como base as 6 emoções básicas de Ekman.

2.1.1.2.1 Detecção de Expressões Faciais (Leão, 2010; Leão et al 2011)

Segundo Leão (2010), Ekman em um de seus trabalhos sobre as expressões faciais humanas, demonstra evidências que suportam a universalidade das expressões faciais. Essas “expressões faciais universais” são as que representam felicidade, tristeza, raiva, medo, surpresa e aversão (Ekman 1994). Também foram estudadas as expressões em diversas culturas e tipos faciais (ocidentais, africanos, chineses, japoneses, por exemplo). Ekman desenvolveu um sistema de codificação para as expressões faciais onde os movimentos faciais são descritos como um conjunto de action units (AU).

Cada AU tem como base estudos relacionados à anatomia dos músculos faciais. Muitos pesquisadores se inspiraram nessas AU fazendo o uso de processamento de imagem e vídeo para automaticamente rastrear características faciais e então utilizá-las para categorizar as diferentes expressões. Ainda segundo Leão (2010) vários pesquisadores tal como Pantic e Rothkrantz fornecem um apanhado de abordagens diferentes utilizadas na tentativa de fazer o reconhecimento automático de expressões faciais.

Assim, Leão (2010) e Leão *et al* (2011) descreve um sistema de detecção automática, em tempo real, que utiliza como entrada um vídeo ou as imagens capturadas através de uma *webcam*. Inicialmente ocorre a detecção da face do indivíduo na tomada de vídeo e a marcação de pontos, denominados pontos de leitura, em regiões de interesse (olhos e boca, por exemplo), como mostrado na Figura 9.

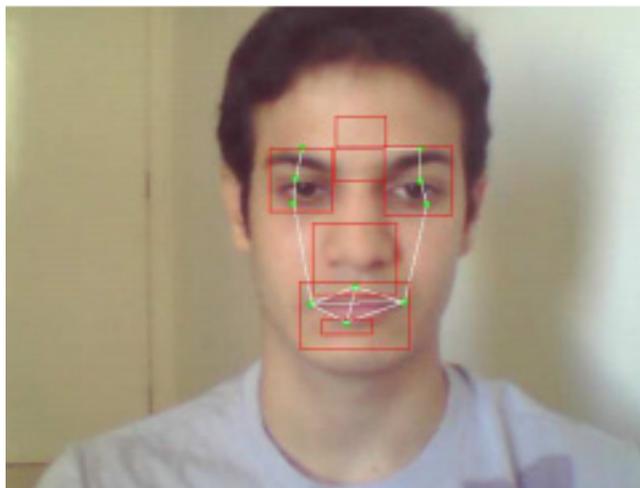


Figura 9: Regiões de interesse apontadas pelo sistema implementado na face inicialmente detectada.
Fonte: Leão *et al* (2011).

E é baseado nos movimentos faciais que ocorre a classificação da expressão facial, como apresentado na Figura 10.

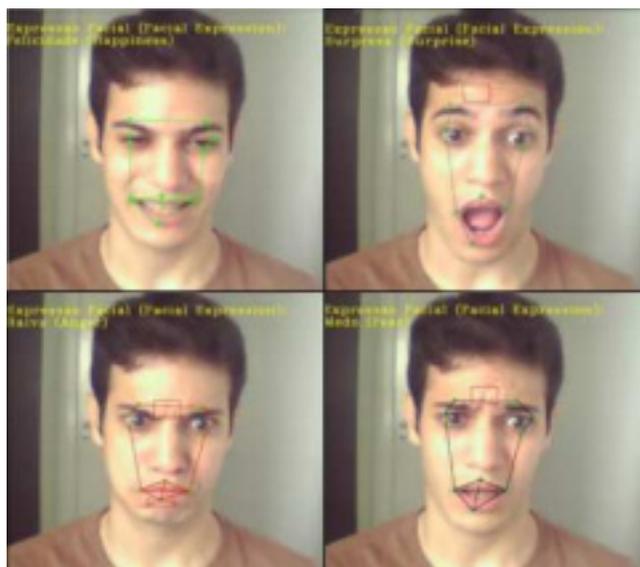


Figura 10: Movimentos faciais e classificação da expressão facial.
Fonte : Leão *et al* (2011).

Para a utilização da implementação basta que o usuário execute a aplicação. A captura

das imagens é feita através da *webcam* ou do vídeo e, assim, é iniciado o algoritmo detector automático de expressões faciais que é utilizado até que a face do usuário seja detectada. Após isso, o mesmo algoritmo é executado na região da face encontrada para segmentar as regiões dos olhos esquerdo e direito, boca e nariz.

No término dessa etapa esse algoritmo deixa de ser executado. Na próxima etapa o algoritmo de fluxo óptico entra em execução para realizar a leitura contínua da movimentação dos pontos colocados nas diferentes regiões da face. As mudanças na movimentação dos pontos em relação à distância inicial entre esses pontos na face inicial, que é interpretada como neutra, caracteriza a presença de algumas AU, além da presença ou ausência de rugas, a Figura 10 mostra o sistema sendo utilizado. Para mais informações por favor consultar (Leão, 2010; Leão *et al*, 2011).

2.2 Identidade (Jesus e Nunes, 2011)

Segundo Jesus e Nunes (2011) para falar de identidade é necessário inicialmente falar do processo de socialização que se inicia no ingresso do indivíduo ao mundo. Ao nascerem todos se deparam com um cenário social, resultante da produção humana, que já está pronto e organizado – realidade objetiva.

A socialização é um processo pelo qual a realidade objetiva se apresenta como realidade subjetiva, ou seja, ao nascer, a criança entra e se apropria de uma realidade cuja construção não participou. Sobretudo não quer dizer que o indivíduo seja apenas um reflexo puro e simples dessa realidade objetiva. Assim, a socialização, que tem como primeiro grupo socializador a família (a não ser em casos excepcionais), se refere à maneira como o indivíduo se insere em uma determinada cultura. A diferença e a igualdade são a primeira noção de identidade, uma vez que vamos nos diferenciando e nos igualando conforme os vários grupos sociais de que fazemos parte.

Quando alguém fala sobre sua identidade, de fato descreve características da sua personalidade: sua história de vida, suas atividades, fala-se de algo que é igual a si mesmo e traz coerência aos diferentes elementos da vida do indivíduo, unindo três tempos em um relato – presente, passado e futuro. Para Brandão (1986), a identidade explica o sentimento pessoal e a consciência da posse de um eu, de uma realidade individual que a cada um se torna, diante dos outros “eus”, um sujeito que é único e é reconhecido pela sua individualidade. O termo identidade se aplica à delimitação que permite a distinção de uma unidade gerada pela socialização e garantida pela individualização.

A formação da identidade depende da diferenciação do eu em relação ao outro. Os psicanalistas e psicólogos sociais relatam que a diferenciação do eu acontece no momento em que nos diferenciamos do outro, ou seja, ao descobrir o outro é possível se tornar alguém. Segundo a visão da psicologia clássica, Identidade é definida pela autoconsciência/visão que cada pessoa possui de si mesma, enquanto que na Psicologia Social e Sociologia, Identidade pode ser definida como a forma que cada pessoa é vista sob os olhos da sociedade.

O modo como se define a identidade é influenciado em grande parte pelo contexto, considerando que a identidade é o fundamento da personalidade, já que por meio daquela podemos ter indicações destas, ambientes diferentes podem moldar a personalidade.

Ciampa (1993) usa a concepção psicossocial e define a identidade como uma metamorfose, ou seja, está em constante mudança, mas apresenta-se de forma estática por meio da aparência da identidade. Isto é, a ação é que constrói a identidade, porém essa é fragmentada e *coisificada* sob a forma de personagem: “Eu sou o que faço em determinado momento”. E não é possível, a todo instante, mostrar todas as facetas que o indivíduo possui. Sendo assim, a dinâmica da construção da identidade é congelada tornando-se uma não metamorfose.

No seu processo de construção, que perdura a vida do indivíduo, a identidade, para Jacques

(1998), se configura como determinada e determinante ao mesmo tempo. Essa configuração decorre em um contexto histórico social em que o sujeito constrói, se insere e se apropria do seu contexto de forma ativa. Gosling (2008) diz que um aspecto importante que pode revelar algo sobre como vemos e como queremos ser vistos no mundo, são as afirmações de identidade, que é o modo que tomamos posse de nossos espaços pessoais. Essas afirmações podem dizer muito sobre nossa história de vida. Uma maneira comum e simples dessa apropriação do ambiente é adorná-lo com objetos que trazem lembranças e simbolismos deliberados. Dessa forma, essas afirmações estão presentes em nossas vidas em tudo o que fazemos.

Boyd (2002) descreve dois aspectos diferentes da Identidade: a noção internalizada do “eu” (Identidade Interna) e a versão projetada da internalização do “eu” (Identidade Social). Nessa mesma linha, Erikson (1980), por exemplo, acredita que Identidade (EGO) tem uma representação pessoal interna (Identidade Interna) bem como uma representação social (Identidade Social). Giddens (1991) concorda que sem experiências sociais o “eu” não pode internalizar evolução. Giddens ainda afirma que a identidade de um indivíduo não é estática, ela pode ser representada em constante evolução, principalmente porque o componente social é dinâmico e está sempre sendo modificado.

Mead (1934), ainda define “eu” e “mim”, onde “mim” representa o aspecto socializado da pessoa (Identidade Social), enquanto que o “eu” representa como a pessoa se define em relação aos outras pessoas da sociedade (Identidade Individual). Ainda, para Gosling (2008), essas afirmações de identidade são dirigidas a si mesmo (identidade interna) e aos outros (identidade social). As duas maneiras possuem função psicológica específica. As afirmações direcionadas aos outros servem para sinalizar como as pessoas desejam ser vistas e por esse motivo, utilizam-se de significados comuns a todos – como exemplo, no Brasil, onde o voto é secreto, é comum em período de eleição, notar que eleitores usam afirmações de suas escolhas, colando adesivos, vestindo camisas (...) do partido eleitoral escolhido em locais visíveis a todos.

São afirmações direcionadas ao outro que transmitem em seu significado não só a escolha eleitoral, mas os valores, crenças e a visão que o indivíduo tem do mundo e que por meio da manifestação, nesse caso, da escolha partidária, comunicam como deseja ser visto e informa como vê o mundo. Já as afirmações direcionadas a si mesmo podem não dar o significado certo para quem observa – como exemplo, a foto de um líder político muito influente, colada na parte interior da porta de um guarda-roupa. Essa é uma afirmação de identidade direcionada a si mesmo, já que está em um lugar visível apenas pelo usuário do ambiente, a nível individual, e individual também será o significado.

A identidade tratada como um dado (informação) capta um aspecto representacional deixando de lado aspectos constitutivos de sua produção. É introduzida nessa perspectiva uma complexidade – uma vez que a identidade pressuposta é resposta, ela é vista como dada anulando o contínuo processo de identificação (Ciampa, 1993). Constitui-se, assim, uma expectativa generalizada onde o indivíduo agirá de acordo com o que é.

Gosling (2008), afirma que traços da personalidade podem ser encontrados nas pistas que as pessoas deixam em seus ambientes pessoais. Impressões que são percebidas através de minuciosas decodificações de decorações de ambientes pessoais, como quarto, escritório e banheiro pessoal. Nessa perspectiva, as generalidades transmitidas pelos estereótipos, apesar de desconsiderarem a variedade das características dentro de um grupo social, podem ser utilizados como hipóteses de trabalho, no sentido de ter uma “dica” que direciona as observações.

Gosling (2008) traz como instrumento para agrupar traços da personalidade, a estrutura Big Five, o Modelo dos Cinco Grandes Traços, são eles: abertura, conscienciosidade, extroversão, sociabilidade e neurose (como já visto anteriormente). Nesse sentido as impressões percebidas pelas pistas deixadas pelos indivíduos em seus ambientes podem e vão dizer algo sobre a

personalidade. É importante ressaltar que entender a personalidade não se resume apenas em somar traços de caráter, mas assemelha-se a resolução de um enigma.

No mundo virtual, que é também considerado um ambiente em que relações sociais acontecem, a perspectiva da identidade é percebida pelas posições tomadas diante de determinadas situações. Características entendidas como próprias do indivíduo são consideradas parte de sua identidade e nas relações sociais, tanto virtuais quanto “reais”, para que o outro possa ter contato e identificar-se com o eu, faz-se necessário manifestações de sua identidade.

O adorno do ambiente virtual, participação de grupos virtuais (comunidades do Orkut) são exemplos de manifestações de sentimento de pertença a determinados grupos, uma vez que o grupo é formado a partir de algum significado comum a todos do grupo. Essa é uma maneira de observar uma característica da identidade. Porém no ambiente virtual existem os “*fakes*”, considerados como uma falsificação da identidade, porém o uso de *fake* pode acontecer como a maneira de ser outro, já que no mundo “real” não é permitido a essa pessoa demonstrar e viver determinadas características que permanecem no anonimato no âmbito real, mas virtualmente, encontra o espaço para se manifestar. Donath (1999; 2000) afirma que conhecer a identidade da pessoa é vital para uma adequada personalização de um ambiente no mundo virtual. Goffman (1959) afirma, ainda, que as pessoas se esforçam para se apresentarem como pessoas “aceitáveis” aos olhos da sociedade (em comunidades virtuais, por exemplo).

Considerando que o usuário é um agente em um ambiente computacional (*web*, por exemplo) onde a interface, ou a adaptação da mesma, é fundamental para que se crie uma personalização da interação com o usuário contextualizado-o em sua zona de conforto e necessidades emergentes. Muitas características da identidade pessoal (aspectos psicológicos e habilidades sociais) do usuário devem ser percebidas e podem ser captadas por meio de símbolos disponíveis na interface dos ambientes computacionais.

Porém cada símbolo pode ser interpretado diferentemente por cada sujeito/usuário, considerando como o mesmo os projeta durante suas interações com o mundo real. Da mesma forma, os símbolos projetados via interface são, também, interpretados diferentemente por cada usuário, emergindo, dessa forma, diferentes aspectos psicológicos dos mesmo durante sua interação no ambiente podendo influenciar definitivamente (positiva ou negativamente) na interação do usuário nos ambientes computacionais atuais.

Considerando a identidade como um canal importante onde as características objetivas e subjetivas das pessoas emergem, denomina-se de fundamental importância seu uso em computadores. Note que tecnicamente, em Ciência da Computação, a tecnologia usada para formalizar a identidade em um dado ambiente computacional é pelo uso de Perfil do Usuário (Identidade Interna) e Reputação do Usuário (Identidade Social). A próxima seção descreve, principalmente, a identidade interna (Para mais informações sobre reputação ver em (Cazella *et al*, 2010)).

2.2.1 Identidade Interna: Perfil do usuário /Modelo do Aluno

Donath (1999) afirma que para a formação eficiente de uma identidade virtual é crucial que o usuário tenha definida sua identidade interna. No mundo virtual a identidade interna do usuário é definida por ele próprio similar ao mundo real (algumas vezes também é descoberta através de técnicas de *Machine Learning*). A identidade interna é geralmente armazenada no Perfil do Usuário.

Perfis de Usuários são conceitos aproximados, eles refletem o interesse do usuário com relação a vários assuntos em um momento particular. Cada termo que um perfil de usuário expressa é, num certo grau, características de um usuário particular (Poo *et al*, 2003) incluindo todas informações diretamente solicitadas a ele e aprendidas implicitamente durante sua interação na

web (Carreira *et al*, 2004). Fisicamente, o perfil do usuário pode ser visto como uma base de dados onde a informação sobre o usuário, incluindo seus interesses e preferências, é armazenada e pode ser dinamicamente mantida (Rousseau *et al*, 2004), (Poo *et al*, 2003).

Na *web* encontram-se muitos tipos de perfis de usuário com diferentes graus de complexidade, eles são desenvolvidos no contexto de *e-commerce* e *e-services*, por exemplo. Kobsa (2007) cria uma *Modelagem Genérica de Usuário* para ser usada como um *shell* para a criação de categorias de informação sobre o usuário objetivando personalizar as aplicações *web*. O modelo proposto por Kobsa é um dos mais reputados. Paiva (Paiva *and* Self, 1995) também desenvolveu um *Shell* de *Modelo de Aluno* chamado TAGUS, criado para melhor modelar os alunos para atividades de aprendizado. Outros Modelos de Aluno foram propostos por Self (1974), Boulay (2003) e são historicamente referenciados.

Considerando ainda definições de Modelo/perfil de Usuário, Heckmann (2005), (Heckmann *and* Kruguer, 2003) propôs uma Ontologia de um *Modelo Geral de Usuário (GUMO)*. O *GUMO* é a ontologia mais completa existente hoje, abrange desde as informações mais básicas do usuário, como dados demográficos e informações de contato, incluindo características fisiológicas e características psicológicas. Essa ontologia foi criada inicialmente para ser utilizada na Computação Ubíqua, servindo para armazenar e compartilhar os dados do usuário provenientes de diferentes recursos tecnológicos interligados entre si.

A Ontologia de Heckmann é extremamente detalhada e pode ser implementada de acordo com o interesse do projetista de um *shell* de perfil de usuário. Na Figura 11 apresenta-se as dimensões básicas propostas por Heckmann no *GUMO*.

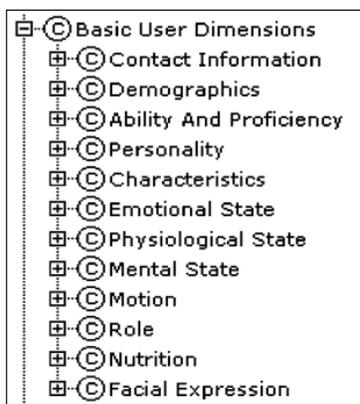


Figura 11: Dimensões básicas do usuário segundo modelo GUMO.

Fonte: Heckmann (2005).

Como o interesse dessa seção é a representação de componentes psico-afetivos na identidade interna do usuário (Perfil/Modelo de Usuário), abaixo destaca-se a ontologia de Heckmann (2005) no tocante a estes aspectos:

- **Emoção:** essa dimensão da ontologia *GUMO* permite representar o humor e o estado emocional do usuário. O estado emocional geralmente é ativado por uma emoção relacionada a algum evento que o usuário está envolvido em determinado momento, podendo ter uma duração curta, de poucos minutos a uma hora; enquanto, o humor é mais difuso e de longa duração, considerando de poucas horas a poucos dias.

A representação das características do humor é apresentado na Figura 12, seguido pela representação do estado emocional através do uso de emoções, apresentado na Figura 13.



Figura 12: lista de humores segundo modelo GUMO .
Fonte: Heckmann (2005).



Figura 13: lista de emoções segundo modelo GUMO.
Fonte: Heckmann (2005).

- **Personalidade:** essa dimensão da ontologia *GUMO* permite representar as características de personalidade de um usuário. Apesar de existirem diversas teorias de personalidade, Heckmann optou por utilizar algumas delas, como por exemplo, a teoria de abordagem de tipos de Jung, a abordagem de Traços de Allport, a abordagem de Eysenck, entre outras. Heckmann (2005) também propõe o uso das características propostas pela teoria Interpessoal.

A Personalidade, diferentemente da emoção, possui traços/características estáveis, ou seja, após ser extraída do usuário, dificilmente irá modificar-se. A representação da personalidade e das características é apresentada na Figura 14.



Figura 14: Personalidade e características representadas no modelo GUMO.

Fonte: Heckmann (2005).

Em se tratando de Modelos de Usuário dotados exclusivamente de características psicológicas, essa seção ilustra dois momentos:

- o primeiro, permite a representação de emoções do usuário;

- o segundo permite a representação da personalidade do usuário.

(i) *Emoções*

Lisetti (2002) propõe o MOUE objetivando modelar as Emoções utilizando o *Affective Knowledge Representation*. O AKR utiliza da abordagem cognitiva baseada em tendências de ação ligadas as emoções. Ele apresenta uma taxonomia de personalidade, afeição, humor e emoções bem como modela as emoções em 16 diferentes dimensões que são chamadas de componentes emocionais, sendo que cada emoção difere das demais nos valores que são atribuídos a um ou mais componentes. Os 16 componentes emocionais utilizados por (Lisetti, 2002) são Expressão Facial, Valência, Intensidade, Duração, Foco, Agência, Novidade, Intencionalidade, Controlabilidade, Modificabilidade, Segurança, Legitimidade, Norma Externa, Norma Interna, Tendência de Ação e Cadeia de Eventos, como visto na tabela 3.

Tabela 3: Componentes emocionais do AKR.

| Componente | Descrição |
|--------------------------|---|
| Facial Expression | Expressão facial: guarda a expressão facial associada à emoção. |
| Valence | Valência: Mensura a agradabilidade de um estado afetivo, ou seja, se ele é positivo ou negativo para o usuário. |
| Intensity | Intensidade: Mede a "força" de uma emoção (por exemplo, a emoção medo pode receber diferentes nomes dependendo apenas da sua intensidade: assustado, apreensivo, aterrorizado, etc.). |
| Duration | Duração: Tempo pelo qual persiste um estado afetivo. |
| Focality | Foco: Indica se o estado emocional é global (se está relacionado a vários eventos ocorridos) ou local (quando é focado em um evento ou um objeto isolado). |
| Agency | Agente: Refere a quem pratica a ação que desencadeia essa emoção. |
| Novelty | Novidade: Usada para indicar se o novo estímulo apresenta-se compatível com as expectativas do sujeito. |
| Intentionality | Intencionalidade: É usada para se referir ao agente que causou a intenção: a própria pessoa ou um agente externo. |
| Controlability | Controlabilidade: Indica o quanto o sujeito acredita que pode controlar a situação corrente, aquilo que ocorre à sua volta (devido a emoção que está sentindo no momento). |
| Modifiability | Modificabilidade: É usada para se referir a perspectiva de tempo ou a julgamento de que o curso dos acontecimentos é capaz de mudar. |
| Certainty | Segurança: É usado para se referir a antecipação dos acontecimentos que virão e de quão convicto está o sujeito sobre as conseqüências desta situação. |
| Legitimacy | Legitimidade: Indica se a emoção é experimentada como um estado legítimo. |
| External Norm | Norma Externa: Indica se o evento obedece às normas e/ou convenções culturais ou ainda se corresponde às expectativas de terceiros. |
| Internal Norm | Norma Interna: Indica se o evento está de acordo com os padrões, concepções e idéias pessoais do próprio sujeito. |
| Action Tendency | Tendência de Ação: Indica a ação (ou conjunto de ações) mais provável consistente com um estado emocional, em outras palavras, como a pessoa reagiria. |
| Casual Chain | Cadeia de eventos que causaram o estímulo. |

Fonte: Lisetti (2002).

No MOUE o *User Model* é implementado sobre a ótica do AKR, e armazena as informações relativas às emoções do usuário obtidas através de multisensores (câmeras, mouse, teclado, microfone e computadores portáteis) e modeladas nas 16 diferentes dimensões chamadas de componentes emocionais. Nesse modelo cada emoção difere das outras pelos valores que são atribuídos a cada componente (informações adicionais em (Bianchi-Berthouze e Lisetti, 2002)).

Considerando ainda emoções, Bezerra *et al* (2011) citam o *XEP-0107 - User Mood*. O *XEP-*

0107 - *User Mood* (Saint and Meijer, 2008) é uma extensão feita em xml para o protocolo XMPP (Extensible Messaging and Presence Protocol), projeto ainda em andamento, criada para guardar e transportar informações referentes ao humor do usuário. Esse modelo foi baseado, em sua maior parte, no *AKR*, mas também levou em consideração os estudos de Ortony *et al* (1988) e Frijda (1986), bem como uma lista de humores e estados físicos retirados de estudos de frequências de palavras na língua inglesa para definir quantas e quais são as emoções básicas.

Ainda em emoções, existe a *EmotionML* (*Emotion Markup Language*) criada pelo W3C (*WWW Consortium*) com o objetivo de padronizar a representação das emoções utilizando XML (W3C, 2009). A *EmotionML* foi projetada para servir em três áreas distintas:

- Anotação manual de dados, para ser utilizada quando a informação advém da análise de vídeos (*web-cam*), fala, textos e análise de expressão facial.
- Reconhecimento automático dos estados emocionais relacionados com o comportamento do usuário, utilizada quando a informação advém de sensores específicos, tais como medidor de frequência cardíaca, temperatura corporal, etc;
- Geração do comportamento do sistema.

A *EmotionML* é uma linguagem dinâmica, apesar do fato de ainda ser muito jovem. Seus principais elementos são: (<emotion>; <category>; <dimensions>; <appraisals>; <action-tendencies>; <intensity>; <metadata>. Onde <emotion> é a tag que representa uma única emoção, e seus filhos (opcionais): <category> que serve para identificar em que grupo a emoção pertence; <dimensions> serve para definir as dimensões que serão trabalhadas na emoção; <appraisals> descreve variáveis de avaliação como, por exemplo, agradabilidade; <action-tendencies> representa a tendência de ação, ou conjunto de tendências de ação; <intensity> representa a intensidade da emoção; <metadata> representa os dados sobre os próprios dados, essa tag possui alguns filhos, por exemplo <confidence> que é utilizada para saber se os dados estão descritos de forma segura.

(ii) *Personalidade*

Considerando que a personalidade implica em Emoções e cada pessoa ou agente que tem emoções tem uma personalidade; e, geralmente, a personalidade não aparece explicitamente mesmo que influencie as emoções diretamente. Pesquisadores em Computação Afetiva têm modelado, padronizado e implementado as emoções explicitamente. Isso acontece porque as emoções são mais facilmente mensuráveis e interpretáveis e podem influenciar diretamente na ação-interação dos usuários e tomada de decisão computacional. Na realidade, como já discutido, as emoções são instantâneas, têm uma vida curta e mudam constantemente, diferentemente da personalidade que é muito mais estável e, normalmente, imutável. Dessa forma, é possível perceber que a personalidade é mais abrangente e implica na emoção. Buscando aumentar o poder representativo da *EmotionML*, Nunes e equipe tem trabalhado na tentativa de modelar e padronizar a representação de personalidade. Em termos de modelagem de personalidade Nunes (2008; 2009) e Nunes *et al* (2008) propõe um Modelo Psicológico de Usuário (*UPP Model*) capaz de armazenar a personalidade de um usuário considerando a abordagem de traços. Esse perfil foi utilizado pela autora em um Sistema de Recomendação, provando que a implementação dos traços de personalidade e seu uso em um perfil de usuário efetivamente melhoram a tomada de decisão computacional personalizando satisfatoriamente o *feedback* ao usuário.

O *UPP* é utilizado pela autora e sua equipe em outras aplicações, tais como, no *Software Personality Inventory PV 1.0* utilizado para extrair os traços psicológicos do usuário através do inventário NEO-IPIP (Johnson, 2000a) e TIPI (Gosling, 2008) traduzidos para a língua portuguesa (Nunes *et al* 2011b). Ainda em personalidade, a equipe tem trabalhado na criação da *PersonalityML* com o objetivo de padronizar a representação da personalidade utilizando XML.

2.2.1.1 Estendendo a EmotionML (Nunes *et al*, 2010c)

A primeira tentativa da equipe em padronizar a representação da personalidade foi pensando em uma extensão da *EmotionML*. Nunes *et al* (2010c) propõem estudos iniciais para uma extensão da EmotionML 1.0 permitindo a inclusão de outros elementos psico-afetivos que aumentam o poder representativo da linguagem. Esses elementos consideraram o modelo hierárquico AKR (Lisetti 2002), estendendo a linguagem com o objetivo de aumentar a capacidade representativa da mesma. O aumento da representatividade permite aos sistemas computacionais identificar com maior precisão a Emoção em foco. A proposta de extensão inclui os componentes psico-afetivos: Afeição, estados de Ânimo e Personalidade. Um fragmento da estrutura proposta para EmotionML é:

```
<personality>
  <affection>
    <mood>
      <emotion>
        <dimensions set = "emotionComponents">
          <facialExpression>
          <valence>
          <intensity>
          <duration>
          <focality>
          <agency>
          <novelty>
          <controlability>
          <intentionality>
          <modifiability>
          <certainty>
          <legitimacy>
          <externalNorm>
          <internalStandard>
          <actionTendency>
          <casualChain>
        </dimensions>
        <category> </category>
        <appraisals> </appraisals>
        <action-tendencies> </action-tendencies>
        <intensity> </intensity>
      </emotion>
    </mood>
  </affection>
</personality>
```

Onde <personality> indica um tipo de personalidade, <affection> afeição, <mood> o estado de ânimo, <emotion> uma emoção, <dimensions set = "emotionComponents"> as dimensões daquela emoção. Os elementos filhos e os atributos definidos na versão original da *EmotionML* continuam válidos na extensão, já que complementam a descrição da emoção.

2.2.1.1 PersonalityML(Bezerra *et al*, 2011)

Considerando que a Personalidade é o ponto chave da representação da cadeia afetiva e que sua representação está limitada pela atual versão da *EmotionML* 1.0. A primeira versão da extensão continuou um pouco limitada no sentido da representação da personalidade, dessa forma a equipe de Nunes decidiu criar primeiramente a *PersonalityML* e depois futuramente incorporá-la a *EmotionML*. A partir do estudo da Personalidade buscou-se a criação de um novo XML

Schema para definir a estrutura de um documento representativo específico da personalidade. A seguir, na Figura 15 apresenta-se um exemplo dos dados modelados na *PersonalityML* e gerados pelo *Personality Inventory PVI.0*, já apresentado anteriormente.

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <personalityml xmlns="http://www.w3.org/2011/personalityml" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
3  <!--Exemplo de representação da Personalidade seguindo a abordagem de traços
4  e a teoria de S. Gosling-->
5  <personality>
6  <emphasis name="Structure">
7  <approach name="Traits">
8  <model name="Big Five Model">
9  <theory author="Sam Gosling">
10 <inventory test="TIPI">
11 <factors set="Factors TIPI checks">
12 <factor name="extraversion" score="100"/>
13 <factor name="agreeableness" score="50"/>
14 <factor name="conscientiousness" score="42"/>
15 <factor name="emotional-stability" score="50"/>
16 <factor name="openness-to-experience" score="100"/>
17 </factors>
18 </inventory>
19 </theory>
20 </model>
21 </approach>
22 </emphasis>
23 </personality>
24 </personalityml>

```

Figura 15: Exemplo do PersonalityML para o TIPI

A *PersonalityML* esta ainda em desenvolvimento.

2.3 Sistemas de Recomendação

Recomendação é um processo social deliberativo executado por pessoas quando as mesmas desejam demonstrar seu grau de apreciação sobre alguém ou alguma coisa. No mundo digital, Sistemas de Recomendação (SR) surgiram a partir da década de 90 como aplicações que forneciam sugestões personalizadas aos usuários sobre produtos e serviços que eles poderiam se interessar, diminuindo consideravelmente a sobrecarga de ofertas (Resnick, 1997).

Atualmente, as recomendações tendem a ser bastante solicitadas. Note que no processo humano de recomendação de produtos ou serviços, as pessoas que tomam a decisão e geram a recomendação tendem a utilizar não só fatores técnicos para essa tomada de decisão, utilizando também fatores psicológicos. Os Sistemas de Recomendação são considerados como uma rica área de pesquisa que têm várias aplicações práticas, inclusive como sistemas que promovem recomendação de pessoas tão bem como promove a recomendação de produtos e serviços. Em 2005, Terveen e McDonald (2005) redefiniram esses Sistemas de Recomendação específicos, chamando-os de Sistemas de Combinação Social.

Entretanto tem se notado que Sistemas de Recomendação e de Combinação Social está engatinhando no uso de aspectos psicológicos humanos em suas recomendações. Muitas vezes devido a dificuldade de extração intencional desse aspecto psicológicos humanos, outras vezes pelo não (re) conhecimento de sua relevância. Porém as pesquisas emergentes na área (Gonzalez, 2007), (Nunes, 2008) bem como outras descritas em (Nunes e Cazella, 2011) vem demonstrando a crescente importância desses aspectos considerados como altamente significante em processos de tomada de decisão humanos e por conseguinte também na tomada de decisão computacional, principalmente, em Sistemas de Recomendação, por exemplo.

Nesse contexto Gonzalez (2007) propõe um primeiro modelo baseado em aspectos

psicológicos. Ele usa Inteligência Emocional para melhorar a recomendação de cursos em um site de e-training. Em 2008, Nunes (2008) publica o primeiro trabalho que usa traços de personalidade para recomendar produtos, serviços ou pessoas em Sistemas de Recomendação. Este trabalho revelou indícios de que o uso de traços de personalidade efetivamente proporciona uma melhora na recomendação, podendo efetivamente ser um aliado à criação de eficientes grupos/equipes de trabalho, como descrito a seguir.

2.3.1 Sistemas de Combinação Social

Segundo Forsyth *apud* in (Wikipedia 2010) “um grupo são duas ou mais pessoas que estão mutuamente conectadas por relacionamentos sociais”. Sociologicamente, um grupo é definido como “um conjunto de seres humanos que compartilha certas características, interagem uns com os outros, aceitam expectativas e obrigações como membros do grupo, e compartilham uma identidade em comum”. Para Bartle (2007): “a principal razão para a formação de um grupo é a interação física baseada em uma necessidade ou em um problema comum. Quanto mais indivíduos compartilharem suas atividades, mais eles interagem entre si e mais alta será a probabilidade de eles virem a formar um grupo”. Segundo Shaw (1976) a composição do grupo pode influenciar em seu desempenho contemplando, assim, duas dimensões significativas para a composição do mesmo: homogeneidade/heterogeneidade de competências e homogeneidade/heterogeneidade de traços sócio-emocionais e de personalidade.

A teoria de atração interpessoal pode predir a maneira como personalidades humanas interagem entre si. Nass *et al* (1995) têm demonstrado um extenso estudo na literatura psicológica que indica forte relacionamento entre similaridade/homogeneidade e atração, isto significa que pessoas preferem interagir com outros (estranhos ou não) que tenham personalidade similar/homogênea a si, a interagir com outros que tenham personalidades diferentes/heterogêneas. Normalmente, as pessoas preferem interagir com outras que tenham a personalidade similar a sua (Nass e Lee, 2000), (Reeves e Nass, 1996).

Grupos que são mais homogêneos em termos sócio-emocionais e de personalidade despendem menos tempo em interações sócio-emocionais necessárias para manter a coesão do grupo e, portanto, concentram-se mais nas interações que são relevantes para a resolução de tarefas. Por outro lado, grupos incompatíveis em termos de características sociais e emocionais têm mais problemas para atingir a resolução de tarefas e despendem muito mais tempo com problemas sócio-emocionais. Além de que, membros em grupos compatíveis são, na maioria dos casos, mais satisfeitos com as interações do grupo.

De acordo com Moerk (1972), “um grupo é formado por indivíduos com uma ampla variedade de experiências e expectativas prévias, que freqüentemente se apresentam como traços de personalidade, e que influenciam o comportamento dos indivíduos no grupo”. Como descrito anteriormente, o ingresso de um indivíduo em um determinado grupo, bem como sua permanência neste, são decisões tomadas pelos sujeitos e sobre as quais as características psicológicas dos envolvidos possuem grande peso. Nos últimos anos pesquisadores, como Damásio (1994) entre outros, vêm percebendo o quão fundamental as características psicológicas são no processo de tomada de decisão humana e por isso vêm ampliando as tentativas em aliá-los também ao processo de tomada de decisão computacional, como discutido anteriormente.

Note que cientistas da Computação começaram a considerar o uso de personalidades (e outros aspectos sócio-emocionais) em computadores para, entre outros objetivos, seja possível a criação de dinâmicas de grupo mais poderosas e eficientes que possam ser usadas das mais diferentes formas, desde recomendação de pessoas em redes sociais, em cursos EaD, até a criação de agentes virtuais mais credíveis (Prada *et al*, 2010), (Prada *et al*, 2009), (Nunes *et al*, 2010a).

2.3.1. 2 Group Recommender 1.0 (Nunes *et al* 2010)

Atualmente existem alguns estudos que demonstram como a utilização de Sistemas de Recomendação/Combinação Social para a formação de grupos de trabalho pode auxiliar no processo do *e-learning*, como em (Romero *et al*, 2009), onde o banco de dados de alunos de *e-learning* é utilizado como base para a construção de um sistema de descoberta de grupos, baseados em algoritmos de lógica difusa. Este processo se mostrou eficaz, quanto ao rendimento das turmas.

Os aspectos psicológicos humanos são fatores de grande peso no processo de tomada de decisão humano, como já descrito anteriormente. No entanto, os Sistemas de Recomendação não tem utilizado efetivamente esses fatores psicológicos. Essa seção mostra uma ferramenta, desenvolvida por Danilo Reinert, onde o foco principal é a possibilidade de criação de equipes de trabalho eficientes coordenado por tutores (considere a esfera de ação de um curso de EAD na UAB criado para atender um publico de 5000 alunos por turma) onde alunos deverão ser distribuídos em subgrupos. O papel dessa ferramenta é recomendar equipes de trabalho considerando as características do tutor e a similaridade de Traços de Personalidade de seus alunos. A interface do Sistema de Recomendação intitulado *Group Recommender 1.0* é apresentado na Figura 16.

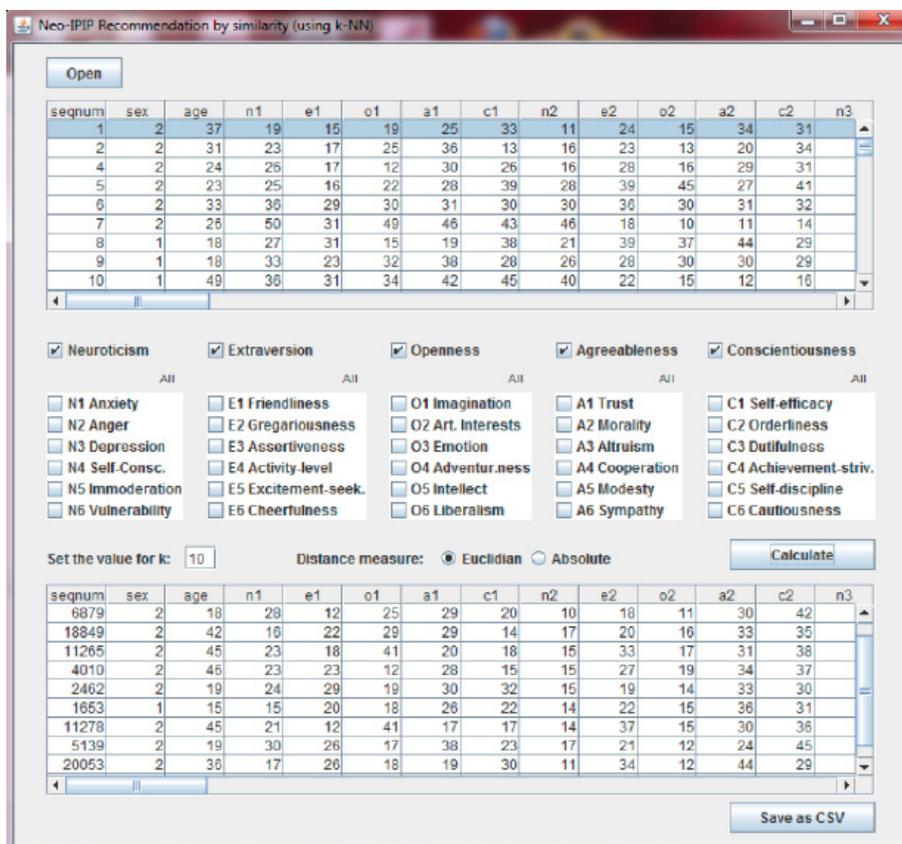


Figura 16: Group Recommender V1.0.

Fonte: Nunes *et al* (2010).

O *software* experimental foi desenvolvido na plataforma JAVA a fim de proporcionar maior mobilidade de execução. O propósito desse Sistema de Recomendação é recomendar pessoas

baseado exclusivamente na personalidade. A personalidade das pessoas foi medida utilizando o *Personality Inventory PVI.0*.

Esse inventário classifica as pessoas em trinta características derivadas do Big-Five. Cada um dos cinco domínios do Big Five possui seis facetas que o compõe. Para fins de validação do *software*, foi utilizada uma database de dados já existente com mais de 20.000 pessoas classificadas de acordo com o NEO-IPIP extraída e fornecida por Johnson (2005). A recomendação busca levantar as pessoas mais similares a alguma selecionada (na UAB, será o Tutor).

O algoritmo utilizado na recomendação foi o k-NN (Nunes 2009). Ele mostrou-se o mais adequado para o propósito do sistema. O princípio do k-NN (Schafer *et al* 2001) é extrair de um determinado repositório de dados, os k exemplos mais próximos a um que se deseja comparar. Com o k-NN, tem-se a opção de determinar uma quantidade k de pessoas, a qual se deseja que o sistema recomende. O Sistema então faz o cálculo levando em consideração sua base de dados, e retorna, assim, as k pessoas mais similares à que foi previamente selecionada.

A fórmula utilizada para calcular a proximidade dos perfis nesse sistema foi a distância Euclidiana. Outra fórmula que seria interessante de ser utilizada para esse fim - similaridade - seria a distância dos cossenos, mas essa não foi implementada nessa primeira versão experimental. Outro ponto positivo em se utilizar o k-NN é a versatilidade que ele dispõe de se comparar um ou mais atributos que se deseje, não necessariamente todos. Vê-se, também, na figura 9, que o usuário tem a opção de selecionar quais atributos ele deseja comparar com o intuito de trazer os k mais similares de acordo somente com esses atributos.

2.3.1.3 Group Recommender 2.0 (Telles *et al* 2010)

O sistema implementado, chamado *Group Recommender 2.0*, foi uma atualização da versão anterior. A primeira versão do *Group Recommender* foi uma versão desktop proporcionando somente a geração de equipes através da similaridade.

O *Group Recommender 2.0* foi desenvolvido para arquitetura *web*, facilitando seu acesso e uso. Além de inovar no quesito similaridade & complementaridade na formação de grupos/equipes. Para utilização do sistema é necessário que os usuários/alunos respondam ao *Personality Inventory PVI.0* (Nunes *et al* 2011b). Através do modelo de personalidade extraído via inventário são gerados os dados necessários para a geração de grupos/equipes.

Os dados utilizados para o cálculo da recomendação de grupos/equipes são os *scores* (pontuação) de cada característica de personalidade atribuída ao usuário que finalizou o inventário. Para tanto, o método define aleatoriamente um usuário chamado “perfil referência” (perfil é o conjunto de informações de um usuário: *scores*, nome e e-mail), que será utilizado de base para o cálculo da distância entre os demais perfis. A forma de calcular é através da fórmula da distância euclidiana n-dimensional, pois o cálculo da distância poderá ser composto por n características psicológicas entre os usuários.

Depois de obtido o resultado entre o perfil referência e cada um dos outros perfis, é listado num vetor ordenado de forma crescente todas as pessoas para retirar K (número representando a quantidade de pessoas por grupo) usuários do começo da lista (pois tiveram a maior similaridade, ou seja, menor distância) e formar o primeiro grupo. Estes perfis serão excluídos da lista e o processo será repetido com os demais até não restar mais perfis sem grupos formados.

O *Group Recommender 2.0* utiliza a base de dados real do *Personality Inventory PVI.0* para realizar sua recomendação com veracidade. Com o uso dos e-mails dos alunos o sistema buscará seus resultados e fará os cálculos necessários para criar uma lista de outros alunos com maior similaridade/compatibilidade psicológica. Na versão dois, além de informar as características similares desejáveis no grupo, é possível também informar os pesos em percentual para cada uma das características marcadas (aplica-se o percentual em cima da característica antes de aplicar

a fórmula da distância euclidiana). Na Figura 17 temos um exemplo do *Group Recommender 2.0*, note que os e-mails são informados, as características e pesos são fornecidos, bem como a quantidade de integrantes de cada grupo.

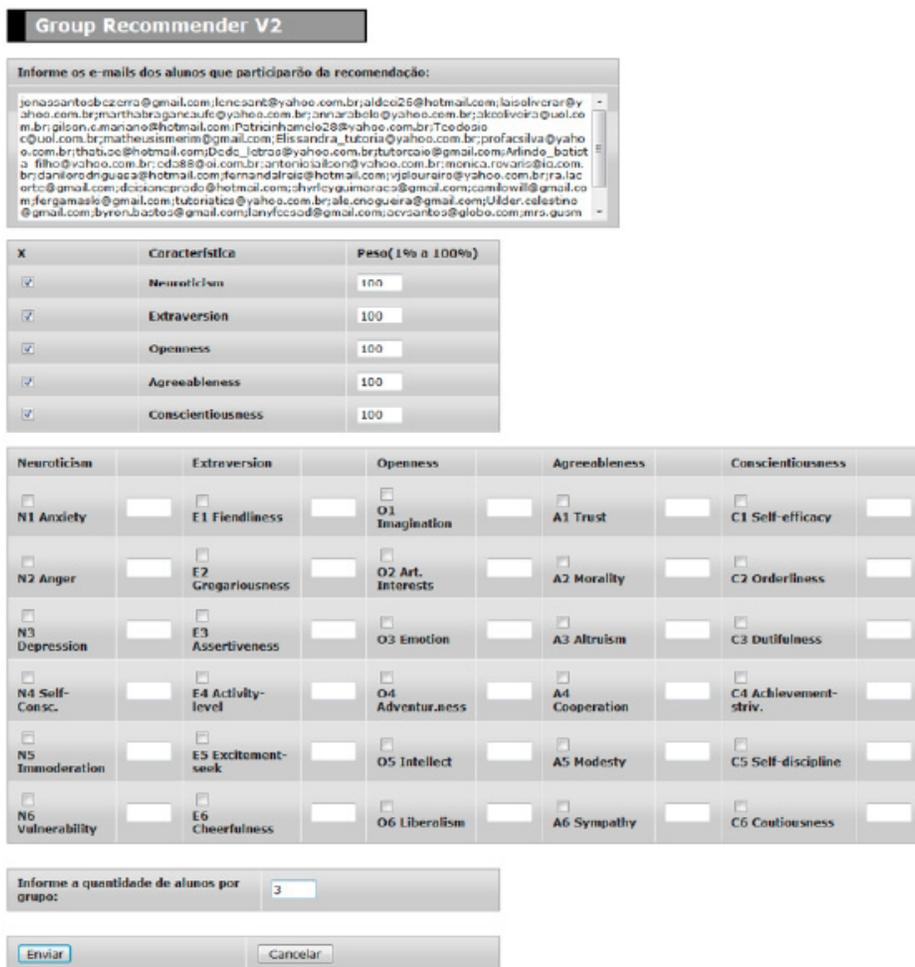


Figura 17: Group Recommender V2.0.

Fonte: Nunes *et al* (2010).

Após o envio dos dados para realização dos cálculos necessários, são exibidas na mesma tela várias tabelas representando cada grupo recomendado pelo *software*, informando o nome do usuário e a distância entre o perfil referência e o perfil corrente.

2.4 Cenários de aplicação de Computação Afetiva em parceria com outras IES

2.4.1 TV Digital (Trevisan 2011)

O trabalho aqui apresentado é resultado de uma parceria do orientador Silvio César Cazella prof. Adjunto da UFCSPA e UNISINOS no Rio Grande do Sul. O trabalho ora apresentado

concorreu com mais de 60 trabalhos oriundos da UFRGS, PUC e UNISINOS sendo agraciado através de uma premiação: Primeiro Lugar na categoria de Inovação – Prêmio ASSESPRO-RS 2011.

O protótipo em TVD implementado por Trevisan (2011) acessa um servidor HTTP e busca as recomendações de programação disponíveis para o usuário. O telespectador pode identificar o quanto o programa recomendado atende às suas necessidades ou desejos. Esta informação alimenta o sistema como uma nova avaliação, aumentando a precisão de futuras sugestões. A base de dados de programas de televisão foi disponibilizada pela empresa Revista Eletrônica para o presente trabalho e pôde ser obtida via FTP, no formato XML.

A grade de programação utilizada no protótipo foi extraída entre 21/06/2010 a 05/07/2010 (quinze dias). A fim de viabilizar o experimento e devido à quantidade de usuários e o tempo de experimentação, foi decidido pela importação de somente programas dos canais abertos da televisão brasileira: o SBT, a RBS, a Bandeirantes, a TV COM, a TV Futura e a Rede TV. Desta forma, foi obtida uma base de dados com uma espacialidade não tão alta e com programas mais conhecidos.

O contexto cujo usuário está inserido no momento da recomendação é informado pelo mesmo de forma explícita a partir de uma lista pré-definida, porém configurável. Estes contextos foram elencados empiricamente a partir da análise dos gêneros, subgêneros e classificação etária dos programas disponíveis na grade de programação. Como os contextos são configuráveis, é possível a aplicação de um algoritmo de mineração de dados no momento em que há a posse de uma base de dados com uma quantidade maior de avaliações e extrair novos contextos, talvez mais significativos.

Os contextos elencados para a execução do protótipo foram por exemplo, 1) Nome do contexto: Sozinho; Descrição: Adulto assistindo sozinho; Aparelho: Televisor; 2) Nome do contexto: Com Crianças; Descrição: Acompanhado de crianças; Aparelho: Televisor; 3) Nome do contexto: Recebendo Amigos; Descrição: Recebendo amigos para um jantar; Aparelho: Televisor; 4) Nome do contexto: Com Namorado(a); Descrição: Acompanhado do namorado(a)/esposo(a); Aparelho: Televisor; 5) Nome do contexto: Locomovendo-se; Descrição: Viajando de ônibus para o trabalho; Aparelho: Celular.

Os testes de personalidade foram realizados utilizando a aplicação *Personality Inventory PVI.0* já descritos anteriormente nesse capítulo. A seleção do contexto em que o usuário se encontra é realizada de forma explícita pelo mesmo antes de receber uma recomendação. Pode ser alterada a qualquer momento, pressionando a tecla verde do controle remoto. Quando a lista é exibida, basta clicar com a tecla de comando “para cima” e “para baixo” do controle remoto até posicionar sobre o contexto desejado e, então, pressionar o botão “ok” do mesmo, conforme a Figura 18.

Uma vez definido o contexto do usuário, o sistema acessa o servidor e busca uma lista de recomendação para o usuário. É possível que nenhum programa seja recomendado para o usuário, neste momento, pelo algoritmo descrito no modelo do sistema anteriormente. Para que o usuário não fique sem receber nenhuma recomendação, o sistema busca por programas que se enquadrem no contexto selecionado e que possua os gêneros, subgêneros e classificações etárias melhor avaliadas por este usuário no passado.

Neste ponto, o sistema acaba realizando uma filtragem baseada em conteúdo e, desta forma, o usuário não fica frustrado por não receber nenhum item recomendado e acaba com o problema do *cold start*. Outro ponto positivo de recomendar itens mesmo que o algoritmo de filtragem colaborativa não retorne nenhuma recomendação é a alimentação do sistema com o *feedback* do usuário que é enviado para cada recomendação gerada, o que de certa forma contribui para o sucesso das suas próximas utilizações.

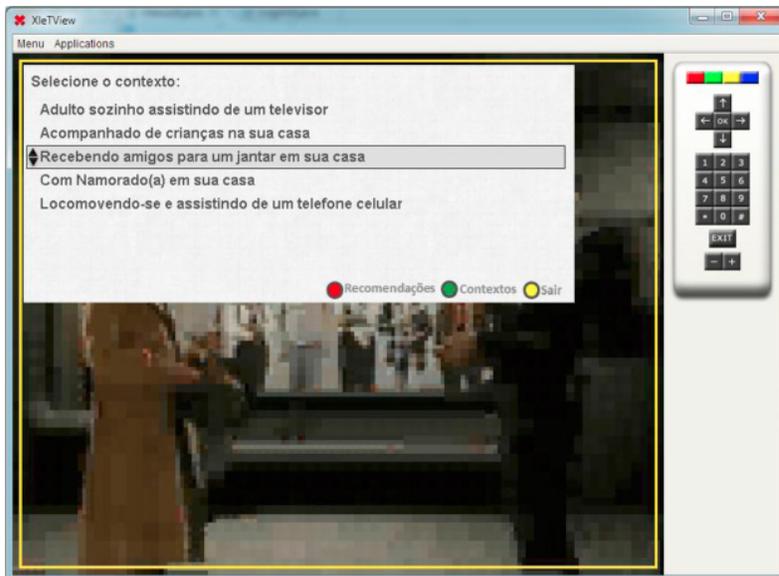


Figura 18: Seleção explícita de contexto.

Fonte: Trevisan (2011).

A lista de itens recomendados apresenta o título do programa, o ícone do canal, a classificação etária e a avaliação dada pelo usuário representada pelas estrelas ao lado de cada programa (Figura 19). Para navegar pelos itens e avaliá-los, basta que o usuário utilize as teclas de navegação do controle remoto “para cima” e “para baixo” a fim de posicionar-se no item desejado, e a tecla “para esquerda” e “para direita” para aumentar ou diminuir a nota dada para o programa recomendado.

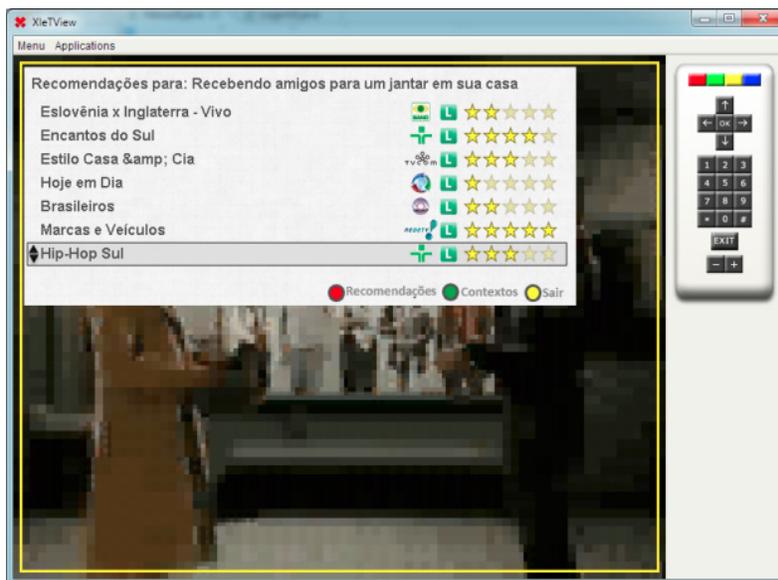


Figura 19: Lista de recomendações fornecidas pelo TvPlus.

Fonte: Trevisan (2011).

Cada avaliação realizada pelo usuário pela TV é enviada para o servidor e armazenada na tabela de avaliações. Quando o algoritmo de recomendação for executado novamente por qualquer usuário do sistema, estas informações transmitidas no *feedback* realizado serão processadas e poderão gerar uma recomendação diferente das anteriormente criadas.

2.5 Outros cenários possíveis

Recomendação de produtos baseados nas características subjetivas dos mesmos

Este cenário é original e, futuramente, será usado largamente em Sistemas de Recomendação comerciais (*e-commerce*). Atualmente os Sistemas de Recomendação convencionais usados em *e-commerce* oferecem produtos e serviços para os seus clientes, geralmente, baseados em informações convencionais do usuário, tais como, informação demográfica, preferências, entre outras. Como exemplos: livros, na *Amazon.com*, *Submarino.com.br*, músicas no *MyStrands* e filmes no *MovieLens*. Na intenção de fornecer uma melhor otimização e personalização nas recomendações prestadas pelos Sistemas de Recomendação os sites comerciais deveriam mudar drasticamente a maneira que tem representado as informações de usuários e de produtos:

Dados do usuário : devem ser enriquecidos com aspectos psicológicos, tais como, Traços de Personalidade, Inteligência Emocional e Habilidades Sociais (Nunes, 2008);

Informações de produtos: uma descrição mais subjetiva deve ser adicionada a descrição tradicional de dados do produto. Características subjetivas em dados convencionais podem ser descritas como meta-dados dos dados, baseados na perspectiva dos aspectos psicológicos já medidos em humanos.

Na tabela 4, apresenta-se uma hipótese de como os dados convencionais podem ser representados subjetivamente usando características psicológicas.

Tabela 4: Representação das informações convencionais e subjetivas de produtos e usuários.

| Nowadays | | Future | |
|---------------------|---|---------------------|---|
| <i>Books</i> | -number of pages -language -category -textual description | <i>Books</i> | -the author writing style -desired Emotions after reading -desired psychological aspects as pre-condition to read a book the designing of the book Personality of the book and the characters |
| <i>Reader/ User</i> | -subject interests -favorite artists, writers - demographic information | <i>Reader/ User</i> | -Personality Traits of each user emotional Intelligence of user Soft Skills of user |

Fonte: Nunes (2008).

3 Conclusões e Trabalhos futuros

Segundo Nunes e Aranha (Nunes e Aranha 2009), do ponto de vista mercadológico, a identificação do comportamento social assim como a interação entre as pessoas sempre foi algo valioso a ser explorado pela área de marketing das empresas. Saber identificar o público alvo e a disseminação viral de uma informação pode significar altas margens de lucro.

Com a *Internet*, essas informações começaram a ser mais facilmente mapeadas digitalmente. Não demorou para aparecer ferramentas como *IRC*, *ICQ*, *MSN* que centralizam a comunicação

das pessoas em um único mapa. Logo depois *ParPerfeito* começa a relacionar as pessoas segundo afinidades conjugais. O *MySpace* encontrou a música como um grande fio condutor de relacionamentos afetivos e popularizou o conceito. O *Orkut* e o *Facebook* deram continuidade e aproximaram pessoas com afinidades perdidas no tempo. O último chegou a ser avaliado em 50 bilhões de reais. O fenômeno chamou tanta atenção para as redes sociais que iniciativas simples como o sistema *Geni* que faz a disseminação viral traçando um mapa genealógico global foi logo em seguida avaliado em 100 milhões de reais.

Encontrar afinidades psicológicas continuam entre os itens de maior valor. Os novos nichos são *LinkedIn* -afinidade profissional, *CineDica*-afinidades por filmes e *GoodReads*-afinidades por livros. Como as iniciativas começaram a se proliferar, as novas tendências são novamente de centralização. O forte crescimento do *Twitter* que abrange qualquer informação e o lançamento do *Google Talk* como unificador de *Gmail* e *MSN*. A grande vantagem do *Twitter* é que ele é partidário de serviços abertos, sendo assim, todos terão acesso às informações para construir sua própria aplicação de afinidades psicológicas sem se preocupar na divulgação do canal.

Dessa forma, conclui-se que existe uma tendência e necessidade de que os programas *web* se utilizem na essência de seus processos internos de tomada de decisão de tecnologias que permitam incorporar a extração, modelagem e uso de características psicológicas do usuário, produtos ou serviços em questão nos processos recomendatórios. Sem dúvida, essas novas representações criarão uma nova perspectivas de como essas informações serão tratadas, representadas e manipuladas na *web* em um futuro muito próximo. Nesse contexto serão criadas novas técnicas de mineração que possam contextualizar conhecimento afetivo, seja através de *web-mining* na análise de *clickstreams* associados a símbolos de conotação afetiva, ou seja através de *text-mining* analisando o sentimento do autor expresso em um texto.

Essas tecnologias inteligentes, juntamente com novas *markup languages* criadas para representar características psicológicas, seguindo-se o exemplo da W3C que padroniza Emoções, podem ajudar a enriquecer os textos na *web* com informações relevantes para a tomada de decisão. Finalmente, tudo isso, aparecerá como uma febre viral que potencializará e, muito, os lucros empresariais

Esse capítulo apresentou um portfólio dos trabalhos desenvolvidos e em desenvolvimento pela autora e sua equipe. Muitos deles possuem ainda um vasto caminho de contribuições, o que aponta linhas de trabalhos futuras. Os principais interesses atuais nas linhas de Computação Afetiva, é o desenvolvimento de Sistemas de Recomendação que usem aspectos afetivos visando melhorar a performance da personalização em dispositivos móveis tanto em termos de interfaces, interação ou mesmo técnica de personalização que efetivamente busque a satisfação dos clientes quando da busca por produtos serviços ou pessoas em ambientes do mundo real.

Agradecimentos

Esse capítulo foi desenvolvido baseado numa coletânea de textos já publicados em Livros, Journals, Congressos e Relatórios técnicos/pesquisa referente a projetos da autora.

Referências Bibliográficas

- ALLPORT, F. H., & ALLPORT, G. W. Personality Traits: Their Classification And Measurement. *Journal Of Abnormal And Social Psychology*, 16, 6–40, 1921.
- ALLPORT, G. W. Concepts Of Trait And Personality. *Psychological Bulletin*, 24, 284–93.
- BARTLE, Phil. Formação e Desenvolvimento de Grupos, 1927. Disponível em <http://www.scn.org/mpfc/modules/bld-grpp.htm> . Acesso em 20/08/2010.

- BERGER, K. S. *The Developing Person Through The Life Span*. 6ª Ed. Worth Publishers, 2003.
- BEZERRA, J. S. ; NUNES, M. A. S. N. . Desenvolvimento De Metodologias De Extração De Perfil Psicológico De Usuário Para Aplicação Em Sistemas De Recomendação Objetivando Personalização De Produtos E Serviços Em E-Commerce- Interface- Plano 1. 2011. (Relatório De Pesquisa).
- BIANCHI-BERTHOUE, N., LISETTI, C. Modeling Multimodal Expression Of User's Affective Subjective Experience. *User Modeling And User-Adapted Interaction* 12: 49-84, 2002.
- BOULAY, B. du. Caring for the Learner in honour of John Self. *Int. J. Artif. Intell. Ed.* 13, 1, 2003.
- BOYD, D. Faceted id/entity: Managing representation in a digital world. Master's thesis, Cambridge, MA, 2002.
- BRANDÃO, Carlos R. Identidade E Etnia, Construção Da Pessoa E Resistência. São Paulo, Brasiliense, 1986.
- CARTWRIGHT, D. *Theories And Models Of Personality*. Wcb, 1979.
- CARREIRA, R., CRATO, J.M., GONÇALVES, D. and JORGE, J. A. Evaluating adaptive user profiles for news classification. In IUI '04: Proceedings of the 9th international conference on Intelligent user interfaces, pages 206–212, New York, NY, USA. ACM Press, 2004.
- CAZELLA S. C., NUNES, M. A. S. N., REATEGUI, E. A . A ciência da opinião: estado da arte em sistemas de recomendação. In: andré ponce de leon f. De carvalho; tomasz kowaltowski. (org.). (org.). Jai: jornada de atualização em Informática da SBC. Rio de Janeiro: Editora da PUC Rio, 2010, v. , p. 161-216.
- CIAMPA, Antônio Da Costa. A Estória Do Severino E A História Da Severina – Um Ensaio De Psicologia Social. São Paulo, Brasiliense, 1993.
- DAMASIO, Antonio R. Descartes' Error: Emotion, Reason, and the Human Brain. Quill, New York, 1994.
- DONATH, J. S. Being Real: Questions of Tele-Identity. In: Ken Goldberg, editor, *The Robot in the Garden: Telerobotics and Telepistemology in the Age of the Internet*, chapter 16, pages 296–311. The MIT Press, first edition, 2000.
- DONATH, J.S. Identity and Deception in the Virtual Community. In M. A. Smith and P. Kollock, editors, 1999.
- EKMAN, P. Strong Evidence For Universals In Facial Expressions. *Psychol. Bull.*, 1994.
- ERIKSON, Erik H. *Identity and the Life Cycle*. Norton, 1980.
- FIORI, A. R. T. Romantic regressions: An analysis of behavior in online dating systems. Master's thesis, MIT Media Lab, 2004.
- FIORI, A. R. T.; DONATH, J. S Online personals: an overview. In CHI'04: CHI '04 extended abstracts on Human factors in computing systems, pages 1395–1398, New York, NY, USA. ACM Press, 2004.
- FRIJDA, N. *The Emotions*. New York: Cambridge University Press. 1986.
- GIDDENS, A. *Modernity and Self-Identity. Self and Society in the Late Modern Age*. Stanford university Press, Stanford, California, 1991.
- GOFFMAN, E. *The Presentation of Self in Everyday Life*. Anchor Book, 1959.
- GOLDBERG, L. R. An alternative “Description of personality”: The Big-Five factor structure. *Journal of Personality and Social Psychology*, 59, 1216-1229, 1990.
- GOLDBERG, L. R. A Broad-Bandwidth, Public-Domain, Personality Inventory Measuring The Lower-Level Facets Of Several Five-Factor Models. *Personality Psychology In Europe*, 7, 7–28, 1999.
- GOLDBERG, L. R., & COLS. The International Personality Item Pool And The Future Of Public-Domain Personality Measures. *Journal Of Research In Personality*, 40, 84–96, 2006.
- GONZALEZ, G., DE LA ROSA, J.L., And MONTANER, M. Embedding Emotional Context Inrecommender Systems. In The 20th International Florida Artificial Intelligence Research Society Conference-Flairs, Key West, Florida, 2007.
- GOSLING, S. D., RENTFROW, P. J., SWANN Jr., W. B. A Very Brief Measure Of The Big-Five Persoanlity Domains. *Journal Of Research In Personality*, 37, 504–528, 2003.
- GOSLING, Sam. *PsIU, Dê Uma Espiadinha: O Que As Suas Coisas Dizem Sobre Você*; Tradução Marcio De Paula S. Hack. – Rio De Janeiro: Elsevier, 2008.
- HALL, Calvin S.; LINDZEY, Gardner; CAMPBELL, John, B. *Theories Of Personality*. John Wiley & Sons, Inc., 1998.
- HECKMANN, D. Ubiquitous User Modeling. Phd thesis, Technischen Fakultäten der Universität des Saarlandes, Saarbrücken-Germany, 2005.
- HECKMANN, D. and KRUGER, A. A user modeling markup language (UserML) for ubiquitous computing.

- In 8th International Conference on User Modeling, LNAI 2702, page 393-397, Johnstown, PA, USA. Springer, Berlin Heidelberg, 2003.
- HUTZ, C. S., Nunes, C. H., SILVEIRA, A. D., SERRA, J., ANTON, M., & WIECZOREK, L. S. O. Desenvolvimento De Marcadores Para A Avaliação Da Personalidade No Modelo Dos Cinco Grandes Fatores. *Psicologia: Reflexão E Crítica*, 11(2), 1998.
- JACQUES, M. G. et al Psicologia Social Contemporânea: Livro-Texto. Petrópolis, Rj. Vozes, 1998.
- JOHNSON, J. A. Predicting Observers Ratings Of The Big Five From The Cpi, Hpi, And Neo-Pi-R: A Comparative Validity Study. *European Journal Of Personality*, 14, 1–19, 2000a.
- JOHNSON, J. A. *Web-Based Personality Assessment*. In 71st Annual Meeting Of The Eastern Psychological Association, Baltimore, Usa, 2000b. Retirado Em [Http://Www.Personal.Psu.Edu/ J5j/Vita.Html](http://www.personal.psu.edu/~j5j/vita.html).
- JESUS, S. G. ; NUNES, M. A. S. N. . Percepção De Aspectos Da Identidade De Usuários Em Um Ambiente Computacional. 2011. (Relatório De Pesquisa).
- JOHNSON, J. A. Ascertaining The Validity Of Individual Protocols From Webbased Personality Inventories. *Journal Of Research In Personality*, 39(1), 103–129, 2005.
- KOBSA, A. Generic user modeling systems. In P. Brusilovsky, A. Kobsa, and W. Nejdl, editors, The Adaptive Web, volume 4321 of Lecture Notes in Computer Science, chapter 4, pages 136–154. Springer Verlag, 2007.
- LEAO, L. P. Detecção De Expressões Faciais Para Plataforma Moodle. 2010. Trabalho De Conclusão De Curso. (Graduação Em Ciência Da Computação) - Universidade Federal De Sergipe.
- LEAO, L. P. ; MATOS, L. N. ; NUNES, M. A. S. N. . Detecção De Expressões Faciais: Uma Abordagem Baseada Em Análise Do Fluxo Optico. In: Wticg, 2011, Salvador. Wticg 2011, 2011.
- LISETTI, C. Personality, Affect And Emotion Taxonomy For Socially Intelligent Agents. In Proceedings Of The Fifteenth International Florida Artificial Intelligence Research Society Conference, Pages 397-401. Aaai Press, 2002.
- LOEHLIN, J. C. *Genes And Environment In Personality Development*. United States Of America: Sage Publications, 1992.
- LORENZ, K. Z. *Os Fundamentos Da Etologia*. (P. M. Cruz & C. C. Alberts, Trad.) São Paulo: Unesp (Obra Original Publicada Em 1981), 1995.
- MEAD, G. H. *Mind, Self, and Society*, volume 1. Univeristy of Chicago, Chicago, charles w. morris edition, 1934.
- MOERK, E. L. Effects of personality structure on individual activities in a group and on a group process, *Human Relations*, v.25 n.6 pp. 505-513, 1972.
- MURRAY, H. A., et al. *Explorations In Personality*. New York: Oxford University Press, 1938.
- NASS, C et al. Can computer personalities be human personalities? *International Journal Human-Computer Studies*, 43(2):223–239, 1995.
- NASS, C. and LEE, K. M. Does computer-generated speech manifest personality? an experimental test of similarity-attraction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (The Hague, The Netherlands, April 01 - 06, 2000). CHI '00. ACM, New York, NY, 329-336, 2000.
- NUNES, M. A. S. N. ; CERRI, Stefano A. ; BLANC, N. Improving Recommendations By Using Personality Traits In: In: International Conference On Knowledge Management-In: International Conference On Knowledge Management-I.Know08, 2008, Graz-Austria. International Conference On Knowledge Management-I.Know08. V. 1. P. 92-100, 2008.
- NUNES, M. S. N. *Recommender System Based On Personality Traits*. (Tese De Doutorado). Université Montpellier 2-Lirimm- França, 2008.
- NUNES, M. A. S. N. ; ARANHA, C. N. . Tendências à Tomada de Decisão computacional. In: W3C, 2009, São Paulo. W3C, 2009.
- NUNES, M. A. S. N. Psychological Aspects In Lifelike Synthetic Agents: Towards To The Personality Markup Language (A Brief Survey). Renote. *Revista Novas Tecnologias Na Educação*, V. 7, 2009a.
- NUNES, M. A. S. N. Recommender Systems Based On Personality Traits: Could Human Psychological Aspects Influence The Computer Decision-Making Process?. 1. Ed. Berlin: Vdm Verlag Dr. Müller. V. 1. 2009.
- NUNES, M. A. S. N. ; BEZERRA, J. S. ; REINERT, D. ; MORAES, D. ; PEREIRA, E. ; PEREIRA, A. J. S. . Computação Afetiva E Sua Influência Na Personalização De Ambientes Educacionais: Gerando

- Equipes Compatíveis Para Uso Em Ava Na Ead. In: Glaucio José Couri Machado. (Org.). Educação E Ciberespaço: Estudos, Propostas E Desafios. Aracaju: Virtus Editora, 2010, V. 1, P. 308-347
- NUNES, M. A. S. N. ; MORAES, D. ; REINERT, D. . Personality Inventory - Pv 1.0 (Portuguese Version). 2010b. (Registrado No Inpi) (Disponível em <http://www.personality-research.com/>)
- NUNES, M. A. S. N., BEZERRA, J. S., OLIVEIRA, A. Estendendo O Conhecimento Afetivo Da Emotionml In: Ihc. 2010, Belo Horizonte, Ihc2010. Porto Alegre: Sbc., P.197 -200.,2010c
- NUNES, M.A.S.N. e CAZELLA, S. C. O que sua personalidade revela ? Fidelizando clientes *web* através de Sistemas de Recomendação e traços de personalidade. Minicursos do *Webmedia*. SBC.Capítulo 5. 2011
- ORTONY, A., CLORE, G. L., And COLLINS, A. The Cognitive Structure Of Emotions. Cambridge University Press, Usa, 1988.
- PAIVA, A. and SELF, J.A. Tagus - a user and leamer modelling workbench. User Model. User-Adapt. Interact., 4(3):197–226, 1995.
- PASQUALI, L. (Org.). Técnicas de exame psicológico – TEP. São Paulo: Casa do Psicólogo, 2001.
- PICARD, R.W., «Emotion research by the people, for the people». *Emotion Review*. July 2010 vol. 2 no. 3 p.250-254. 2010.
- PICARD, R. W. *Affective Computing*. MIT Press, Cambridge, MA, USA, 1997.
- POO, D., CHNG, B. and GOH, J.M. A hybrid approach for user profiling. In HICSS '03: Proceedings of the 36th Annual Hawaii International Conference on System Sciences (HICSS'03) Washington, DC, USA. IEEE Computer Society, 2003.
- PORTO, S. M.; COSTA, W.S. ; NUNES, M. A. S. N; MATOS, L. N. Desenvolvimento De Metodologias De Extração De Perfil Psicológico De Usuário Para Aplicação Em Sistemas De Recomendação Objetivando Personalização De Produtos E Serviços Em E-Commerce. Relatório Técnico De Pesquisa. Universidade Federal De Sergipe, 2011.
- PRADA, R., MA, S., NUNES, M. A. S. N. Personality in Social Group Dynamics In: International Conference on Computational Science and Engineering- CSE '09, 2009, Vancouver. International Conference on Computational Science and Engineering- CSE '09. v.4. p.607 – 612, 2009.
- PRADA, R. ; CAMILO, J. ; NUNES, M. A. S. N. . Introducing Personality Into Team Dynamics. In: Ecai - European Conference On Artificial Intelligence-Frontiers In Artificial Intelligence And Applications, 2010, Lisbon. Ecai. Ios Press, 2010. V. 215. P. 667-672, 2010.
- REEVES, B. and NASS, C. The media equation: how people treat computers, television, and new media like real people and places. Cambridge University Press, New York, NY, USA, 1996.
- RESNICK, P., & VARIAN, H. R. Recommender Systems. *Communications Of The Acm*, 40(3), 56-8, 1997.
- ROMERO, C., & Cols. *Evolutionary Algorithms For Subgroup Discovery In E-learning: A Practical Application Using Moodle Data*. University Of Córdoba, 2009.
- ROUSSEAU, B., BROWNE, P. , MALONE, P. FOSTER, P. and MENDIS, V. Personalised resource discovery searching over multiple repository types: Using user and information provider profiling. In ICEIS (5), pages 35–43, 2004.
- SAINT, P., MEIJER, A. R. (2008). Xep-0107: User Mood. Xmpp. In Standards Foundation. Disponível Em <Http://Xmpp.Org/Extensions/Xep-0107.Html>. Acesso Em Agosto De 2010.
- SANTOS, A. B. NUNES, M. A. S. N. . Desenvolvimento De Metodologias De Extração De Perfil Psicológico De Usuário Para Aplicação Em Sistemas De Recomendação Objetivando Personalização De Produtos E Serviços Em E-Commerce Reconhecimento De Traços Psicológicos Via Conversação. 2011. (Relatório De Pesquisa).
- SCHAFER, J. B. KONSTAN, J. and RIEDL, J. E-commerce recommendation applications. *Data Mining Knowledge Discovering*, 5(1-2):115-153, 2001.
- SCHULTZ, D. *Theories Of Personality*. 4ª Ed. Brooks/Cole, 1990.
- Self, J.: Student Models in Computer-Aided Instruction. *International Journal of Man-Machine Studies* 6(2): 261-276 , 1974.
- SIMON, H.A. Reason in Human Affairs. Stanford University Press, California, 1983.
- SOUSA, D. A. NUNES, M. A. S. N. . Aspectos Teóricos E Mensuração Do Construto Psicológico Personalidade Desenvolvimento De *Plugin Moodle* Para A Formação De Grupos De Trabalho Para Uso Na Ead Ufs. 2011. (Relatório De Pesquisa).
- SHAW, M. E. (1976) *Group Dynamics*. McGraw-Hill, 1976.
- TEERVEN, L. and MCDONALD, D.W. Social matching: A *framework* and research agenda. *Acm Trans.*

- Comput.-Hum. Interact., 12(3):401–434, 2005.
- TELES, I. P ; NUNES, M. A. S. N. . Desenvolvimento De *Plugin* Moodle Para Recomendação De Objetos De Aprendizagem Para Uso Na Ead Ufs, 2011. (Relatório De Pesquisa).
- THAGARD, Paul. Hot Thought: Machanisms and Applications of Emotional Cognition.A Bradford Book-MIT Press, Cambridge, MA, USA, 2006.
- TRAPPL, Robert; Payr, Sabine and Petta, Paolo editors. Emotions in Humans and Artifacts. MIT Press, Cambridge, MA, USA, 2003.
- TREVISAN, Luiz. Tv Plus: Um Modelo De Sistema De Recomendação De Programas Para Tv Digital. Trabalho De Conclusão De Curso (Graduação Em Sistemas De Informação) - Universidade Do Vale Do Rio Dos Sinos, São Leopoldo, RS, 2011.
- W3C 2009. Emotion Markup Language (EmotionML) 1.0 W3C - Working Draft 29 October 2009. Disponível em: <http://www.w3.org/TR/2009/WD-emotionml-20091029/>.
- WIKIPÉDIA 2010. Disponível em http://pt.wikipedia.org/wiki/Ambiente_virtual_de_aprendizagem. Acesso em 20/08/2010.

Seção 3

Sistemas Distribuídos e Arquitetura de Computadores



Novas Direções em Engenharia Web

Admilson de Ribamar Lima Ribeiro

Resumo. A engenharia *web* possibilita a construção de tecnologias para *World Wide Web* (WWW), sendo parte de um tópico maior chamado de ciência *web* que abrange a *Internet* em um contexto de ciência completo. Os trabalhos apresentados na literatura até aqui se limitam a essa abordagem, sem considerar que as tecnologias *web* surgem em decorrência de novas aplicações emergentes na WWW. Este capítulo aborda as principais tecnologias *web* do ponto de vista dos requisitos das aplicações, isto é, como as aplicações emergentes na WWW estão levando ao desenvolvimento de novas tecnologias para satisfazer os requisitos dessas aplicações. Novos requisitos como ataques de segurança inesperados, falhas de *hardware*, e ambientes de rede dinâmicos precisam ser agora considerados. As principais tecnologias para satisfazer esses requisitos e apresentados neste capítulo são: computação pervasiva, *web services*, sistemas *peer-to-peer*, computação em grade e aplicações *multimídia*. Algumas questões de pesquisa sobre o assunto são apresentadas no final do texto e também o que está sendo desenvolvido, nessa área, no departamento de computação (DCOMP) da Universidade Federal de Sergipe (UFS).

Palavras-chave: *Web Services*. Computação Pervasiva. Sistemas *Peer-to-peer*. Computação em Grade. Aplicações *Multimídia*.

1 Introdução

A *World Wide Web* (WWW) consiste em um espaço de informações onde os recursos são identificados por URIs (*Uniform Resource Identifier*) com protocolos para suportar a interação entre os agentes de *software* e formatos para representar os recursos de informação. De um ponto de vista analítico podemos estudar a WWW para encontrar as leis que geram ou explicam determinados fenômenos. Por outro ponto de vista, sintético, formalismos e linguagens são criados para suportar determinado comportamento desejado. Neste capítulo, abordaremos esse último ponto de vista, ou seja, uma abordagem chamada de engenharia *web*.

Também faremos uma abordagem do ponto de vista dos requisitos das aplicações. Dessa maneira não consideraremos aquelas tecnologias que se preocupam basicamente com o formato da representação das informações tais como *web* semântica, processamento de linguagem natural e personalização (BERNERS-LEE *et al.*, 2006). Assim, apenas as tecnologias que se concentram predominantemente em agentes de *software* (aplicações) e protocolos serão consideradas, por

exemplo, sistemas *peer-to-peer*, *web services*, computação em grade entre outras (COULORIS *et al.*, 2001).

Essas novas tecnologias surgiram da necessidade de satisfazer os requisitos de aplicações emergentes na WWW atual. Essas novas aplicações se utilizam de comunicação *multimídia*, mobilidade, computação embarcada, comunicação de grupo e alta disponibilidade e levam ao aparecimento de sistemas que precisam considerar seus requisitos. Requisitos como ataques de segurança inesperados, falhas de *hardware*, e ambientes de rede dinâmicos precisam ser agora considerados.

Outros trabalhos abordam o estudo da engenharia *web*, mas o consideram no âmbito da ciência *web* e não partem da perspectiva dos requisitos das aplicações (BERNERS-LEE *et al.*, 2006). Portanto, este trabalho tem como perspectiva a descrição do campo da engenharia *web* considerando os requisitos que as novas aplicações *web* precisam satisfazer. O principal objetivo é descrever as novas direções que a engenharia *web* vem tomando, considerando apenas as novas tecnologias que se concentram predominantemente em agentes de *software* (aplicações) e protocolos.

Além desta introdução, o capítulo está organizado em mais sete seções. Nas seções de número 2 a 6, as seguintes tecnologias *web* serão consideradas: computação pervasiva, *web services*, sistemas *peer-to-peer*, computação em grade e aplicações *multimídia*. Na seção 7, discutiremos algumas questões de pesquisas relevantes em engenharia *web*. Finalmente, na seção 8 faremos as considerações finais do capítulo, onde destacaremos os principais projetos na área de engenharia *web* que estão sendo desenvolvidos no Departamento de Computação (DCOMP) da Universidade Federal de Sergipe (UFS).

2 Computação Pervasiva

A computação pervasiva está mudando completamente a maneira como as pessoas interagem com o computador. As aplicações pervasivas possibilitam o acesso à informação e serviços a qualquer hora e em qualquer lugar, moldando uma nova sociedade de informação, com os usuários acessando a informação na *Internet* através de diversos dispositivos móveis, tais como telefones celulares, *personal digital assistants* (PDAs) e *laptops*. Assim, em vez de se preocupar com as questões de gerenciamento do ambiente computacional, os usuários simplesmente acessam suas aplicações e dados sempre que for desejado, independente de hora e lugar.

Software para computação pervasiva executa em pequenos dispositivos sem fio, tais como PDAs, telefones celulares, etc., que estão conectados em redes sem fio. Esse ambiente apresenta características próprias, a saber: limitação de recursos dos dispositivos móveis (pouca memória, bateria de vida curta, CPU (*Central Processing Unit*) de baixa velocidade); limitações apresentadas pela rede sem fio e mobilidade dos usuários.

Essas características introduzem novos requisitos que os sistemas de *software* devem satisfazer: capacidade de perceber mudanças no ambiente de execução, capacidade de localizar e endereçar elementos móveis, capacidade de se autoconfigurar, uso eficiente dos recursos no elemento móvel e do canal de comunicação sem fio, capacidade de migrar funcionalidade, alto grau de tolerância a falhas e mecanismos para autenticação e cifragem de dados (ENDLER & SILVA, 2002).

Devido às variações na disponibilidade dos recursos locais dos dispositivos sem fio e também a variações nas características do canal de comunicação sem fio, faz-se necessário o monitoramento contínuo do contexto de execução, o qual se refere a tudo que pode afetar o comportamento de uma aplicação. Podem-se identificar dois níveis de ciência de contexto (CAPRA *et al.*, 2001): ciência de dispositivo e ciência de ambiente. Ciência de dispositivo refere-se a tudo que reside no dispositivo físico onde a aplicação está executando, por exemplo, memória, tamanho da tela, energia da bateria, poder de processamento, etc. Ciência de ambiente refere-se a tudo que está

fora do dispositivo físico, por exemplo, largura de banda, conectividade de rede, localização, outros computadores, etc.

Ciência do contexto de execução requer que o programador de aplicação saiba qualquer informação coletada do sistema operacional de rede como, por exemplo, a localização do dispositivo. A capacidade de localização e endereçamento é necessária devido à mobilidade dos usuários e pode ser feita em diversos níveis do sistema. A própria infra-estrutura de uma rede sem fio pode fornecer esse serviço, como ocorre no padrão IS-41 dos sistemas celulares de segunda geração (PETERSON & DAVIE, 2007). Alguns protocolos de rede já fornecem essa facilidade como IP (*Internet Protocol*) móvel (PETERSON & DAVIE, 2007). As aplicações móveis precisam satisfazer esse requisito, caso não seja suportado em outro nível do sistema.

Os mecanismos de monitoramento podem ser feitos através de *polling*, notificação de eventos ou de uma combinação das duas e pode ocorrer em qualquer nível do sistema (*hardware, firmware, software* de sistema ou aplicação). O monitoramento do ambiente de execução fornece informações necessárias para adaptação dinâmica do sistema para um novo ambiente de execução. Assim, o sistema deve ser projetado para suportar mecanismos de configuração dinâmica.

A limitação dos recursos dos dispositivos móveis (pouca memória, bateria de vida curta, CPU de baixa velocidade) obriga que o sistema de *software* que executa nesses dispositivos tenha baixa carga computacional em detrimento de novos requisitos não funcionais que poderiam ser acrescentados. Há um compromisso entre a carga computacional que o dispositivo suporta e os requisitos não funcionais para possibilitar um mínimo de sobrecarga possível.

A possibilidade de ocorrência de desconexões entre o elemento móvel e a rede fixa, e também a necessidade de poupar recursos do dispositivo móvel, tornam necessária a existência de mecanismos que possibilitem a migração de tarefas de processamento dos elementos móveis para a rede fixa e vice-versa.

Em um ambiente de rede sem fio, os componentes distribuídos no dispositivo móvel conectam-se à infra-estrutura de rede para acessar dados e solicitar serviços. É comum que quando o cliente está conectado, o servidor que forneceria determinado serviço não está, isto é, o cliente e o servidor normalmente não executam ao mesmo tempo. Uma forma de comunicação assíncrona é necessária para fornecer interações entre componentes cliente e servidor que não executam ao mesmo tempo.

Em uma rede móvel, os dispositivos podem falhar devido principalmente à falta de energia. Nos casos em que o reabastecimento não seja possível, o sistema deve prover mecanismos de tolerâncias a falhas que possam garantir o funcionamento da rede.

Os requisitos de segurança específicos desse ambiente são devidos ao fato de que o meio de comunicação sem fio é um meio acessível a todos, e também à portabilidade dos dispositivos que podem ser perdidos ou furtados. Utiliza-se de criptografia para evitar que os dados sejam acessados por intrusos que monitoram o meio sem fio, ao passo que é exigida a autenticação do dono do dispositivo para evitar que intrusos acessem os dados locais ou remotos. Essa autenticação pode ser feita por *smart cards* que são inseridos no dispositivo móvel.

3 Web Services

Na WWW, a maioria das aplicações cliente-servidor necessita da interação de uma pessoa (normalmente, executando um *software* chamado de navegador *web*) com uma máquina (um computador executando um *software* chamado de servidor *web*). Por exemplo, a utilização por um usuário de um navegador *web* para acessar determinados documentos de um servidor *web*.

Entretanto, existe uma grande demanda para que uma aplicação possa interagir com outra sem a intervenção de um ser humano. Esse interesse apareceu da necessidade de interação das

aplicações comerciais das empresas. Por exemplo, fazer um preenchimento de um formulário de pedido e ao mesmo tempo obter a informação se determinado item ainda está no estoque. Quando essas aplicações estão na mesma empresa, a integração é chamada de EAI (*Enterprise Application Integration*). Denomina-se de integração B2B (*Bussiness-to-Bussiness*) quando a interação ocorre com aplicações de empresas diferentes (BERNERS-LEE *et al.*, 2006).

Essa interação entre aplicações sem a necessidade de um usuário levou ao conceito de *web services*. Assim, *web services* é uma arquitetura *web* que permite que ocorra a interação entre aplicações *web* automaticamente, liberando o usuário de manipulações extras.

A implementação de uma arquitetura *web services* é feita por meio da oferta de serviços *web* para serem consumidos por usuários, daí a origem do nome *web services*. Nesse sentido, *web services* são pedaços de códigos distribuídos que se comunicam usando trocas de mensagens com objetivo de resolver determinada tarefa. Atualmente, existem duas arquiteturas de *web services* (PETERSON & DAVIE, 2007): SOAP (*Simple Object Access Protocol*) e REST (*REpresentational State Transfer*).

3.1 Arquitetura SOAP

A arquitetura SOAP aborda o problema da interação de aplicações por meio da geração de protocolos de aplicação e transporte que são customizados para uma aplicação específica. Os principais componentes dessa arquitetura são (PETERSON & DAVIE, 2007): um *framework* para especificação de protocolo, ferramentas de *software* para automaticamente gerar implementações de protocolos dado as especificações, e especificações parciais que podem ser reutilizadas para geração de outros protocolos.

A arquitetura SOAP compreende dois *frameworks* que foram padronizados pelo *World Wide Web Consortium* (W3C) denominados de WSDL (*Web Services Definition Language*) e SOAP (PETERSON & DAVIE, 2007). WSDL é o *framework* para especificação de protocolos de aplicação enquanto SOAP é o *framework* responsável pela especificação de protocolos de transporte. Ambos os *frameworks* são especificados usando a linguagem XML (*eXtended Markup Language*) para facilitar a utilização por intermédio das ferramentas de *software* como compiladores e serviços de diretórios.

A WSDL segue um modelo de operação procedural para especificação do protocolo de aplicação. A interface *web service* a ser especificada consiste de duas partes (PETERSON & DAVIE, 2007): uma parte abstrata e outra concreta. A parte abstrata especifica as operações necessárias para interação entre o cliente e um *web service* enquanto a parte abstrata especifica a ligação (*binding*) necessária das operações com o protocolo de transporte a ser utilizado. A WSDL possui ligações predefinidas para HTTP (*HyperText Transfer Protocol*) e SOAP.

O SOAP é um *framework* usado para definir protocolos de transporte com características exatas para suportar um determinado protocolo de aplicação. Essas características podem incluir: confiabilidade, segurança, roteamento e padrões de troca de mensagens. O SOAP objetiva também que essas características possam ser reutilizadas em outras definições de protocolos de transporte, que define essas características como componentes reutilizáveis chamados de blocos de cabeçalho.

3.2 Arquitetura REST

A arquitetura REST se baseia na arquitetura *web* com algumas modificações para suportar *web services* (PETERSON & DAVIE, 2007). A arquitetura REST aborda o problema de integração de aplicações considerando um *web service* como mais um recurso da WWW que pode ser identificado por um URI e acessado via HTTP. Como o protocolo HTTP dá suporte a um

pequeno conjunto de métodos que são freqüentemente bloqueados por *firewalls*, o modelo REST resolve essa limitação deslocando as informações do protocolo para o *payload*. O *payload* é uma representação do estado abstrato de um recurso. Por exemplo, um método GET pode retornar uma representação do corrente estado do recurso e um método POST pode enviar uma representação do estado desejado do recurso.

Muitos servidores *web* (*Amazon, Yahoo, Google e eBay*) oferecem interfaces de serviço que permitem aos clientes manipular seus recursos *web* (PETERSON & DAVIE, 2007). Isso permite que aplicações de outros fornecedores construam serviços com valor agregado sobre aqueles fornecidos por essas empresas. Por exemplo, controle de inventário e aquisição sobre o serviço da Amazon.com. Mais de 50.000 desenvolvedores se registraram para uso desses serviços *web* nos dois anos após eles serem introduzidos (eles foram introduzidos em 2002). A Amazon.com deixou seu sistema acessível por ambas arquiteturas de *web services*, e cerca de 80% do consumo dos serviços foram feitos via interface REST por vários anos (PETERSON & DAVIE, 2007).

4 Sistemas *Peer-to-peer*

Um sistema *peer-to-peer* consiste em explorar os recursos de um grande número de computadores de uma maneira não centralizada. Consideramos a distribuição horizontal que divide os componentes do sistema em clientes e servidores. Nessa distribuição, um cliente ou servidor opera em sua própria porção do conjunto completo de dados. Os processos que constituem um sistema *peer-to-peer* são todos iguais. Cada processo agirá como um cliente e um servidor ao mesmo tempo. Esses processos são chamados de processos *serventes* (COULORIS *et al.*, 2001).

Um usuário desse sistema deve ser capaz de localizar um recurso de interesse (arquivo de dados ou de música) sem auxílio de uma entidade centralizada e também que os recursos estejam disponíveis em larga escala. Para conseguir esses dois objetivos, o sistema deve ser projetado por meio de uma rede de sobreposição onde os nós contêm os recursos e são interligados para que se possa percorrer em busca de determinado recurso.

4.1 Redes de Sobreposição

Desde seu início, a *Internet* adotou um modelo de comunicação em que os roteadores dentro da rede são responsáveis por encaminhar pacotes da origem ao destino, e os programas aplicativos são executados nos *hosts* conectados às extremidades da rede. Nos últimos anos, porém, novas aplicações estão sendo distribuídas em toda a *Internet*, e em muitos casos, esses aplicativos fazem suas próprias decisões de encaminhamento. Tendo em vista esse objetivo, redes de sobreposição são construídas com o intuito de migrar parte da complexidade de roteamento para a camada de aplicação, facilitando, por exemplo, buscas determinísticas na rede (PETERSON & DAVIE, 2007).

Uma rede de sobreposição (PETERSON & DAVIE, 2007) pode ser definida como uma rede lógica construída sobre uma rede física já existente, onde cada nó da rede lógica também existe na rede original, porém a rede lógica processa e encaminha pacotes de uma forma específica definida pela aplicação. Os enlaces que conectam os nodos de uma rede de sobreposição são construídos como túneis através da rede subjacente. Múltiplas redes de sobreposição podem existir sobre a mesma rede subjacente, cada uma implementando seu próprio comportamento específico da aplicação. Redes de sobreposição podem ficar sobrepostas umas sobre outras. A maioria das redes de sobreposição trata a *Internet* como a rede subjacente.

As redes de sobreposição oferecem uma maneira de se acrescentar e testar novas funcionalidades sem utilizar a rede subjacente, bastando acrescentar essas funcionalidades para os nodos

sobrepostos. Caso essa nova funcionalidade se mostre mais adequada que aquela implantada na rede subjacente, pode-se adotar essa nova funcionalidade permanentemente. Por exemplo, pode-se testar um novo protocolo de roteamento para a *Internet* sem precisar de um roteador que suporte esse novo protocolo. Nesse contexto, as redes de sobreposição estão rapidamente emergindo como o mecanismo de escolha para a introdução de novas funcionalidades na *Internet*.

5 Computação em Grade

Os sistemas de computação em grade são utilizados para tarefas de computação de alto desempenho. Esse tipo de sistema costuma ser montado como federação de computadores onde cada computador fica em um domínio administrativo diferente. Portanto, são diferentes dos sistemas de computação em *cluster*, que apesar de serem projetados para computação de alto desempenho, os seus computadores ficam em um mesmo domínio.

Diferentemente da homogeneidade dos sistemas de computação em *cluster*, os sistemas de computação em grade apresentam alto grau de heterogeneidade. Eles podem conter diferentes tipos de *hardware*, sistemas operacionais, redes, domínios administrativos, políticas de segurança, assim por diante.

Os recursos de diferentes organizações são reunidos para permitir a colaboração de um grupo de pessoas ou instituições. A colaboração é realizada na forma de uma organização virtual. Pessoas que pertencem à mesma organização virtual possuem direitos de acessos a diferentes recursos, tais como servidores de computação, facilidades de armazenamento e bancos de dados.

Foster *et al.* (2001) propôs uma arquitetura de quatro camadas, como mostra a Figura 1. A camada base provê interfaces para recursos locais em um *site* específico. A camada de conectividade consiste de protocolos de comunicação para suporte de transações acessando múltiplos recursos. A camada de recurso é responsável pelo gerenciamento de um simples recurso. A camada coletiva manuseia acesso a múltiplos recursos. Por fim, a camada de aplicação consiste das aplicações que operam dentro de uma organização virtual.

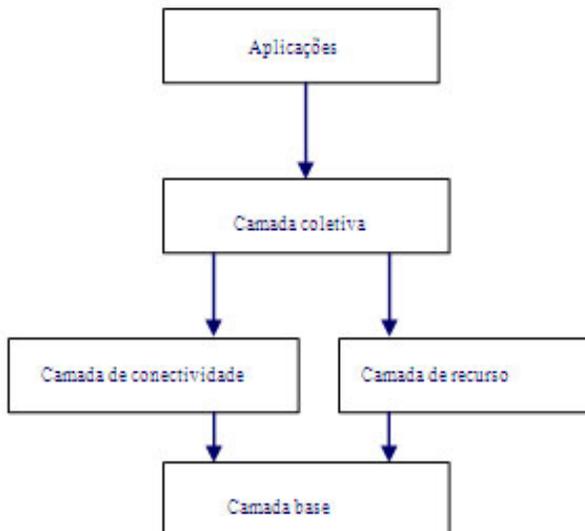


Figura 1: Arquitetura em camadas para sistemas de computação em grade.

Fonte: (FOSTER *et al.*, 2001)

6 Aplicações *Multimídia*

Nos primeiros dias da *Internet*, as aplicações manuseavam apenas dados que não variavam com o tempo, tais como transferência de arquivos, envio de e-mails, e assim por diante. Hoje em dia, há necessidade de se trabalhar com dados em tempo real, como amostras de dados digitalizados que contêm informações de voz ou vídeo. O processamento desses tipos de dados deu origem a um novo tipo de aplicação chamada de aplicações *multimídia*.

Assim, aplicações *multimídia* são aquelas que envolvem voz, vídeo e texto. Elas são divididas em dois tipos (PETERSON & DAVIE, 2007): interativas e *streaming*. Nas aplicações interativas, todo o arquivo *multimídia* (vídeo ou voz) é transferido de um servidor para um cliente antes de ser apresentado, semelhante ao processo de *download* de uma página *web* só com texto. Nas aplicações *streaming*, há um computador que gera a informação de vídeo ou áudio e depois a transmite através da *Internet* em mensagens; um cliente captura e apresenta a informação à medida que ela chega. Algumas aplicações *multimídias* são: vídeo sob demanda, videoconferência, entre outras.

O mais importante requisito que essas aplicações devem satisfazer é a capacidade de poderem se comunicar umas com as outras. Isso acarreta que as aplicações devem possuir o mesmo método de codificação e compressão de voz e vídeo.

Os requisitos das aplicações *multimídias* são atendidos através do projeto de protocolos de propósitos específicos para essas aplicações. Os protocolos de transporte para essas aplicações devem atentar para (PETERSON & DAVIE, 2007): diversidade de aplicações, interação entre as aplicações e a sincronização dos fluxos de áudio e vídeo. Os protocolos de reserva de recursos devem ser usados para requisição de recursos na rede de modo que determinada garantia de qualidade de serviço seja fornecida para a aplicação. Finalmente, precisamos de um protocolo de controle de sessão caso queiramos fazer alguma aplicação síncrona como telefonia baseada em IP.

7 Questões Proeminentes em Engenharia Web

O crescimento contínuo da WWW é um fato consumado e desejado. O armazenamento de uma grande quantidade de informação necessitará de novas tecnologias para sua facilidade de recuperação. Novas aplicações *multimídias* e novos serviços devem requerer esforços de pesquisas no mundo acadêmico e de negócios. Novos ambientes *web*, como redes *peer-to-peer*, foram criados para impulsionar a WWW como um todo e devem ser investigados. Serviços é uma área chave na WWW onde novos modelos de desenvolvimento devem ser pensados.

O *software* age como uma abstração em cima do *hardware* que nos permite especificar máquinas computacionais em termos de funções lógicas. Essas funções facilitam a especificação e solução de problemas de uma maneira mais intuitiva. A inclusão de serviços na WWW abre mais uma possibilidade de uma nova abstração: podemos agora desenvolver abstrações em cima do *software*. Isso leva ao seguinte questionamento: Que métodos de descrição de serviços nós permitirá não nos preocuparmos como eles serão executados?

Vários métodos de especificação formal de processos foram desenvolvidos e aplicados no domínio de serviços. Tanto especificação algébrica como especificação baseada em modelo estão sendo utilizadas em *web services*. Em geral, a especificação algébrica de serviços em um nível abstrato tem sido a maior área de pesquisa em serviços (DEUTSCH *et al.*, 2007) (FU *et al.*, 2004) (NARAYANAN & McILRAITH, 2003).

WS-Net (BERNERS-LEE *et al.*, 2006) é uma linguagem de descrição arquitetural baseada na teoria das redes de Petri colorida, que descreve um componente *web service* em termos dos serviços que ele fornece para outros componentes; os serviços que ele requer para funcionar e

suas operações internas. O resultado é um modelo que engloba os aspectos locais e globais do sistema de serviço, facilita a integração *web service* para obter novos objetivos, enquanto provê um formalismo para avaliação da integração (ZHANG *et al.*, 2004).

Álgebras de processo também têm sido aplicadas em serviços. O uso de uma álgebra formal permite a ocorrência do desenvolvimento e da avaliação. Ferrara (2004) descreve o mapeamento entre uma álgebra de processo e BPEL4WS (uma notação baseada em XML padronizada para descrever processos de negócios executáveis). Esse mapeamento permite a criação de serviços em BPEL4WS seguido de sua avaliação e verificação usando álgebra de processo. É possível também a geração de código BPEL4WS automaticamente a partir do uso da álgebra para especificar os serviços desejados.

Existem muitos dispositivos de *hardware* que ainda não podem se conectar a WWW. O ambiente formado por esses dispositivos possui características diferentes do ambiente formado por máquinas tradicionais interligadas por uma rede cabeada. Isso leva a mudanças significativas nas tecnologias da engenharia *web*. São exemplos desse ambiente: computação móvel, computação pervasiva, sistemas *peer-to-peer* e computação em grade.

A computação móvel apresenta novas demandas em termos de engenharia *web*. Nesse ambiente, o poder computacional disponível é pequeno e os usuários estão em movimento com acesso e largura de banda variáveis. A apresentação de informação para o usuário varia de um PC (*Personal Computer*) com bastante informação até pequenos dispositivos com pequenas telas com restrições de apresentação com um navegador *web* (BALUJA, 2006) (MAEKAWA *et al.*, 2006). O acesso móvel na WWW já é dominante em muitas nações devido aos baixos preços dos dispositivos; confiabilidade das conexões sem fio e o aumento da autonomia das pequenas baterias (PARIKH & LAZOWSKA, 2006). Pesquisa nessa área é importante para a distribuição equitativa dos recursos *web*.

No ambiente de computação pervasiva algumas questões precisam ser enfrentadas. Por exemplo, descoberta de serviço deve ocorrer sem a interferência humana. Serviços devem ser capazes de alertar eles mesmo para facilitar a descoberta. Padrões para publicações de serviços são requeridos para (LASSILA & ADLER, 2003):

- Garantir segurança e privacidade;
- Confiança na confiabilidade do serviço;
- Compensação para o provedor do serviço;
- Exatidão na composição com outros serviços invocados para obter algum objetivo ou resolver o problema em questão.

Existem muitas aplicações para redes *peer-to-peer* no contexto *web*. A questão que se apresenta é que funções essenciais para a experiência *web* podem ser preservadas em sistemas autônomos fracamente acoplados. Devido ao grande número e à heterogeneidade dos nós *peer-to-peer*, métodos tradicionais de engenharia tais como experimentação *online* ou simulação em larga escala são inapropriadas. A escala licenciada pela WWW, que continuará a ocorrer em redes *peer-to-peer*, torna a teoria de modelamento de rede essencial (SAROIU *et al.*, 2002) (LIANG *et al.*, 2004), mas também se deve esperar experimentação, inovação e empreendedorismo para conduzir o caminho neste campo.

O problema com a computação em grade é a coordenação do comportamento de um grande número de computadores, que exploram recursos não usados de maneira oportuna. Como nesse ambiente há neutralidade administrativa ou de plataforma, padrões abertos são, portanto necessários e a grade requer descrições abstratas dos recursos computacionais.

8 Considerações Finais

Descrevemos cinco tecnologias atualmente em uso na WWW: *web services* usada para facilitar

a interação entre as aplicações; aplicações *multimídia* que estão precisando de novos protocolos de controle de sessão e congestionamento; computação pervasiva, computação em grade e sistemas *peer-to-peer* que possibilitam o processamento de aplicações e o encaminhamento de pacotes de uma maneira inovadora.

Algumas tecnologias de engenharia *web*, por exemplo, *web* semântica não foi abordada devido à limitação do escopo do trabalho que se limitou a apresentar apenas as tecnologias relacionadas com agentes de *software* e protocolos.

Não é fácil apontar para um ponto específico aberto em engenharia *web*. Todo o campo está aberto para novas tecnologias devido ao aparecimento de novas aplicações que são inventadas diariamente. Mas um ponto é necessário observar: reconhecer que aplicações emergentes necessitam de substancial mudança nos protocolos de rede. Os protocolos de rede precisam de alguma maneira incorporar os requisitos das aplicações.

Caminhamos para uma arquitetura de rede altamente adaptativa. Essa nova arquitetura deverá prover várias funcionalidades. Desde prover a transmissão de dados e código em um mesmo fluxo até considerar questões de autogerenciamento. Assim, a preocupação que agora se apresenta é a organização de sistemas distribuídos como sistemas de realimentação de controle de alto nível que permitam adaptação automática a mudanças. Esse novo paradigma é conhecido como computação autônoma (KEPHART & CHESS, 2003).

No Departamento de Computação (DCOMP) da Universidade Federal de Sergipe (UFS) alguns projetos na área de engenharia *web* já foram iniciados. Descreveremos apenas os principais projetos aqui.

Estamos desenvolvendo um laboratório de engenharia *web* chamado de LEW que executa na *Internet* por meio de um portal *web* que suporta as atividades de pesquisa, ensino e extensão. Esse portal já está em operação desde 2010 e conta com vários alunos cadastrados.

O ERLab é um outro projeto de pesquisa e ensino cuja finalidade é a criação de um laboratório de acesso remoto para a realização de pesquisas *on-line*. Através desses laboratórios, obtêm-se procedimentos que garantem as atividades de experimentos de pesquisa de forma autônoma, aumentam a disponibilidade efetiva dos laboratórios, e propiciam atendimento a um grande número de pesquisadores. Dessa forma, há uma racionalização dos custos associados à pesquisa.

Ainda neste contexto é possível utilizar a *Internet* como um meio de ensino de computação. Sendo assim, estamos desenvolvendo um juiz *online* que será disponibilizado na *Internet*, através de um portal *web* que está sendo desenvolvido utilizando *software* de código aberto.

Todos os projetos seguem a temática da área de engenharia *web* que consiste na aplicação de abordagens sistemáticas, disciplinadas e quantificáveis para o desenvolvimento, operação e manutenção de aplicações *web*.

Referências Bibliográficas

- BALUJA, S.: Browsing on small screens: Recasting *Web*-page segmentation into an efficient machine learning framework. In: *Proceedings of WWW 2006*, <http://www2006.org/programme/files/pdf/2502.pdf>, 2006.
- BERNERS-LEE T, HALL W, HENDLER J A, O'HARA K, SHADBOLT N e WEITZNER D A: A Framework for *Web Science*. In: *Foundations and Trends in Web Science* **1**, **1**: 1-130, 2006.
- CAPRA, L., EMMERICH, W., MASCOLO, C.: *Middleware for Mobile Computing*. Technical report, University College London, Dept. of Computer Science, 2001.
- COULORIS, G., DOLLIMORE, J., e KINDBERG, T. *Distributed Systems: Concepts and Design*. Third edition. Addison-Wesley, 2001.
- DEUTSCH, A., SUI L., e VIANU, V.: Specification and verification of data-driven *Web* services. In: *Proceedings of PODS '04*, 2004.
- ENDLER, M. & SILVA, F. J. S.: Requisitos e Arquiteturas de *software* para computação móvel. In: III Workshop de Comunicação sem Fio e Computação Móvel, Recife, PE, 2001.

- FERRARA, A.: *Web services: A process algebra approach*. In: *Proceedings of the 2nd International Conference on Service-Oriented Computing (ICSOC 2004)*, (M. Aiello, M. Aoyama, F. Curbera, and M. P. Papazoglou, eds.), pp. 242–251, New York: ACM Press, 2004.
- FOSTER, I., KESSELMAN, C., e TUECKE, S.: The Anatomy of the Grid, Enabling Scalable Virtual Organizations. In: *Journal of Supercomputer Applications*, (15)3:200-222, 2001.
- FU, X., BULTAN, T., e SU, J.: Analysis of interacting BPEL *Web services*. In: *Proceedings of the World Wide Web Conference 2004*, 2004.
- KEPHART, J., O. & CHESS, D., M.: The Vision of Autonomic Computing, *Computer*, pp. 41-50, 2003.
- LASSILA, O. & ADLER, M.: Semantic gadgets: Ubiquitous computing meets the Semantic Web. In: *Spinning the Semantic Web: Bringing the World Wide Web to its Full Potential*, (D. Fensel, J. Hendler, H. Lieberman, and W. Wahlster, eds.), pp. 363–376, Cambridge MA: MIT Press, 2003.
- LIANG, J., R., KUMAR, R., ROSS, K., W.: Understanding KaZaA, Working paper, <http://cis.poly.edu/~ross/papers/UnderstandingKaZaA.pdf>, 2004.
- MAEKAWA, T., HARA, T., NISHIO, S.: Image classification for mobile *Web* browsing. In: *Proceedings of WWW 2006*, <http://www2006.org/programme/files/pdf/2506.pdf>, 2006.
- NARAYANAN, S. & McILRAITH, S.: Analysis and simulation of *Web services*. *Computer Networks*, vol. 42, no. 5, pp. 675–693, 2003.
- PARIKH, T., S. & LAZOWSKA, E., D.: "Designing an architecture for delivering mobile information services to the rural developing world," *Proceedings of WWW 2006*, <http://www2006.org/programme/files/pdf/5507.pdf>, 2006.
- PETERSON, L. L. & DAVIE, B. S.: *Computer Networks: A systems approach*. 4th edition, Morgan Kaufmann, San Francisco, 2007.
- SAROIU, S., GUMMADI, K., P., DUNN, R., GRIBBLE, S., D., LEVY H., M.: An analysis of internet content delivery systems," *Proceedings of the 5th Symposium on Operating Systems Design and Implementation*, 2002.
- ZHANG, J., CHUNG, J.-Y., CHANG, C. K., KIM, S., W.: WS-Net: A Petri-net based specification model for *Web services*. In: *Proceedings of IEEE International Conference on Web Services (ICWS '04)*, 2004.

P&D em Arquitetura de Sistemas Computacionais e Sistemas Embarcados

Edward David Moreno Ordonez

Resumo. Este capítulo apresenta algumas iniciativas de pesquisa que estão em andamento no DCOMP/UFS e destacando os desafios e questões relevantes para realizar atividades de P&D (Pesquisa e Desenvolvimento) em arquitetura de sistemas computacionais, focando em aspectos de projeto e otimização de plataformas embarcadas para soluções de segurança e algoritmos que requeiram de aceleradores em *hardware*, visando além de um bom desempenho, otimização de código, o menor consumo de energia. Outro aspecto é o projeto específico de processadores dedicados e o projeto e uso de sistemas multicore e GPUs para soluções de segurança e aplicações específicas que requeiram alto desempenho, observando a possibilidade de ter o projeto em FPGAs ou em plataformas embarcadas com capacidade de multicore on-chip.

Palavras-Chave: Arquiteturas Avançadas de Computador. Sistemas Embarcados. Segurança em *Hardware*. Avaliação de Desempenho.

1 Introdução

Os avanços nas tecnologias de integração de circuitos integrados têm tornado viáveis os sistemas integráveis em um único *chip*. Esses tipos de sistemas integram processadores programáveis e componentes de *hardware* dedicados permitindo que sejam colocados em um único chip de forma a atender requisitos de desempenho necessários às aplicações que possuem natureza multifuncional.

Projetos com as características descritas acima são extremamente complexos porque as decisões tomadas no início tendem a ter impacto no seu desempenho final e também porque o projeto de circuitos integrados possui uma rápida inserção no mercado. À medida que novos padrões são definidos surgem então novas aplicações e, em especial para atender essas novas aplicações, torna-se vantajoso proporcionar **programabilidade**, **reconfigurabilidade** e **escalabilidade** nesse tipo de dispositivo, no momento em que grandes velocidades de processamento podem ser alcançadas apenas com um pequeno ajuste nos parâmetros da arquitetura no intuito de adaptá-la a aplicações de natureza específica (Paar 1999)(ECC 2006)(Rasheed 2007).

Com essas alternativas a disposição, torna-se mais fácil para os projetistas basearem seus chips e sistemas em um tipo de arquitetura que forneça soluções eficientes e otimizadas para um determinado domínio de aplicações, somente sintonizando e ajustando os elementos reconfiguráveis presentes na arquitetura e no sistema para comportar a aplicação em questão.

Dessa maneira, a área de computação reconfigurável (via utilização de circuitos programáveis – FPGAs) tem surgido como uma alternativa tecnológica que pode adaptar-se com muita facilidade ao uso de um processador de propósito geral ao mesmo tempo em que mantém as vantagens relativas a desempenho do *hardware* dedicado (Yang 2003)(Huffmire 2008).

Com essa velocidade e adaptabilidade inerente, a computação reconfigurável tem um grande potencial para ser explorado especialmente em aplicações que necessitam de um bom desempenho com prototipagem rápida (diminuindo o tempo e o custo do projeto em qualquer uma das fases pertencentes a um projeto de circuitos e sistemas digitais: desenvolvimento, implementação e teste) adequado aos parâmetros reais de funcionamento das mais diversas aplicações modernas existentes hoje.

A maioria dessas aplicações alvo é de natureza industrial e própria do setor eletro-eletrônico em países industrializados e em fase de modernização tecnológica, tendo um forte impacto sendo uma excelente alternativa na pesquisa e desenvolvimento quando associada ao setor das pequenas e médias empresas que precisam, utilizam e disponibilizam de alguma maneira circuitos e sistemas digitais e de computação.

Por essa última razão, seria importante salientar que a área de computação reconfigurável poderá trazer muitos benefícios acadêmicos, industriais e comerciais; permitindo e facilitando assim uma maior interação entre pesquisadores, professores, alunos, profissionais e empresários dos mais diversos setores da economia. Além disso, com as novas facilidades de projeto e desenvolvimento de circuitos integrados facilitadas com o uso de FPGAs e SoC torna-se plausível a utilização e desenvolvimento dessa tecnologia em aplicações industriais próprias em regiões que estejam em fase de desenvolvimento.

Geralmente o alvo dos sistemas reconfiguráveis são aplicações que inerentemente possuam paralelismo de dados, alta regularidade e grandes requisitos de vazão (*throughput*). Alguns exemplos dessas aplicações são: compressão de vídeo (transformadas discretas tais como DCT – *Discrete Cosine Transform*, e estimação de movimento), processamento gráfico e de imagens, multimídia, encriptação de dados e transformadas usadas em DSPs (*Digital Signal Processing*) (Paar 2011)(Koc 2011)(Fan 2010)(Kastner, 2010).

Nesse sentido, a utilização de FPGAs para essas aplicações justifica-se por varias razões: (i) Alta velocidade: é possível otimizar velocidade adequando e sintonizando os parâmetros críticos do sistema. (ii) Nível Flexível de Segurança: É relativamente fácil adequar as novas arquiteturas e sistemas a novos *insights* e otimizações. (iii) relação Desempenho/Custo: Pode-se selecionar e projetar unidades funcionais, levando em consideração o custo e o desempenho. Por exemplo, projetar poucas e lentas unidades aritméticas, ou rápidas e variadas unidades. (iv) Moderação dos custos: o projeto com FPGAs é relativamente baixo em custo.

A área de segurança em *hardware* e a respectiva integração de diferentes serviços de segurança em sistemas embarcados ainda são carentes de métodos, técnicas, e principalmente, protótipos eficientes em desempenho e consumo de energia.

Por esse motivo, ainda há a possibilidade de criar, capacitar, ensinar, promover e desenvolver novas metodologias e expandir a utilização dessas inovações tecnológicas em aplicações tanto educacionais quanto industriais, que usam circuitos programáveis e outras plataformas de sistemas embarcados. Nesse intuito, as atividades de P&D alvo de interesse da comunidade focalizarão no projeto e desenvolvimento de soluções de segurança (estudo, projeto, otimização de consumo de energia e desempenho de algoritmos de criptografia e serviços de segurança)

em *hardware* e plataformas embarcadas, com forte ênfase nos aspectos de projeto e definição de arquiteturas e projeto de processadores para aplicações específicas de segurança.

O capítulo está organizado em 5 seções, e cada uma delas oferece informações da área e finaliza com questões que ainda precisam ser melhor investigadas. A seção 2 apresenta informações sobre segurança em sistemas embarcados e *hardware* específico aplicado a aspectos de segurança da informação. A seção 3 aborda detalhes do uso de sistemas multicore e GPUS. A seção 4 enfatiza em técnicas e questões de pesquisa sobre como diminuir e/ou otimizar o consumo de energia de aplicações computacionais. A seção 5 finaliza com algumas conclusões.

2 Hardware Específico para Soluções de Segurança

O conceito de segurança da informação é muito mais amplo do que a simples proteção dos dados em nível lógico. Para proporcionar uma segurança real é preciso estar atento a vários fatores internos e externos. Em primeiro lugar há a necessidade de se caracterizar o sistema que armazenará as informações para poder identificar as ameaças, e assim definir as possíveis soluções.

Segundo Stallings (2008) e Paar (2011) existe a seguinte subdivisão: (i) **Sistemas Isolados** - São aqueles que não estão conectados a nenhum tipo de rede e, (ii) **Sistemas Interconectados** - Atualmente, a maioria dos computadores existentes pertence a uma rede de computador, enviando e recebendo informações do exterior quase que constantemente. Neste tipo de sistema sempre há riscos potenciais.

Segundo Menezes (2011) e Stallings (2008), o objetivo de segurança da informação, pode ser classificado em:

- (i) **Segurança Física** - Esta categoria engloba todos os assuntos relacionados com a salvaguarda e o suporte físico da informação. Neste nível estão as medidas contra incêndio, sobrecargas elétricas, políticas de *backup*, etc;
- (ii) **Segurança da Informação** - Nesta categoria apresenta-se a preservação da informação frente a pessoas não autorizadas. É neste nível que entra a criptografia (simétrica e assimétrica);
- (iii) **Segurança do Canal de Comunicação** - Raramente os canais de comunicação são seguros. Devido ao fato da maior parte das redes existentes pertencerem a terceiros, se torna muito improvável assegurar totalmente a segurança nela;
- (iv) **Problemas de Autenticação** - Devido aos problemas nos canais de comunicação, se torna necessário assegurar que a informação que se recebe em um computador vem realmente de quem se pensa que a enviou. Neste nível é conveniente a utilização da Assinatura Digital;
- (v) **Problemas de Suplantação** - Nas redes existe o problema crônico de que qualquer usuário autorizado possa acessar as informações, mesmo de fora. Então, tem-se que garantir que usuários não estão sendo suplantados por intrusos. Normalmente se utilizam mecanismos baseados em *password* para solucionar este problema;
- (vi) **“Descarte de Informação” ou Não Repúdio** - É necessário haver uma política de não descarte de informação, pois esta, se armazenada, pode revelar o autor de uma determinada ação. Ex: um usuário ter realizado um ato de comércio e posteriormente tentar negar esse mesmo ato.

Uma definição um pouco mais técnica, é que a Criptografia é o estudo de técnicas matemáticas que relatam os aspectos da informação, assim como a confidencialidade, integridade da informação, autenticação de entrada e autenticação de dados na origem (Menezes 2006) (Menezes 2010)(Paar 2011).

A tabela 1 mostra a expansão dos objetivos principais dos Sistemas Criptográficos.

Tabela 1: Objetivos dos Sistemas Criptográficos.

| Características | Descrição |
|---|--|
| Privacidade ou confidencialidade | Manter informações secretas disponíveis somente para quem está autorizado |
| Integridade dos dados | Garantir que a informação não seja alterada por meios desconhecidos ou não autorizados. |
| Autenticação ou identificação da entidade | Validação da identidade de uma entidade |
| Autenticação de mensagens | Confirmar a origem da informação |
| Assinatura | Maneira de ligar a informação à entidade |
| Autorização | Comunicar, à outra entidade, de permissão oficial para fazer algo ou ser alguém. |
| Validação | Maneira de fornecer tempo limite de autorização para usar ou manipular informações e recursos |
| Controle de Acesso | Restringir o acesso aos recursos à entidades privilegiadas |
| Certificação | Confirmação de uma informação por uma entidade confiável |
| Selo de criação | Gravação da hora de criação ou tempo de existência da informação |
| Testemunhar | Verificação da criação ou existência da informação por uma entidade diferente da criadora |
| Recibo | Reconhecimento de que a informação foi recebida |
| Confirmação | Reconhecimento de que o serviço foi providenciado |
| Propriedade | Maneira de prover uma entidade com os direitos legais para usar ou transferir um recurso para outras entidades |
| Anonimato | Ocultar a identidade de uma entidade envolvida no processo |
| Não-repúdio | Impedir a recusa de compromissos ou ações anteriores |
| Anulação | Revogação de certificação ou autorização |

Fonte: Menezes (2006).

As primitivas criptográficas são estruturas elementares utilizadas para a construção dos criptosistemas e dos protocolos criptográficos. A Figura 1 apresenta o modelo de classificação das diversas primitivas criptográficas, proposto por Menezes (2011), que também sugere os seguintes critérios para avaliação de tais primitivas:

- (i) Nível de segurança** – Normalmente é calculada com base na quantidade de operações necessárias para alcançar o seu objetivo;
- (ii) Funcionalidade** – As primitivas precisam ser combinadas para atender aos objetivos de segurança da informação. Quais delas atendem melhor estas características são determinadas pelas propriedades básicas desta primitiva;
- (iii) Método de Operação** – A utilização das primitivas depende dos modos de operação possíveis;
- (iv) Desempenho** – Refere-se a eficiência com que a primitiva executa sua função;

(v) **Facilidade de Implementação** – Refere-se a complexidade exigida pela primitiva para sua implementação tanto em *hardware* quanto em *software*.

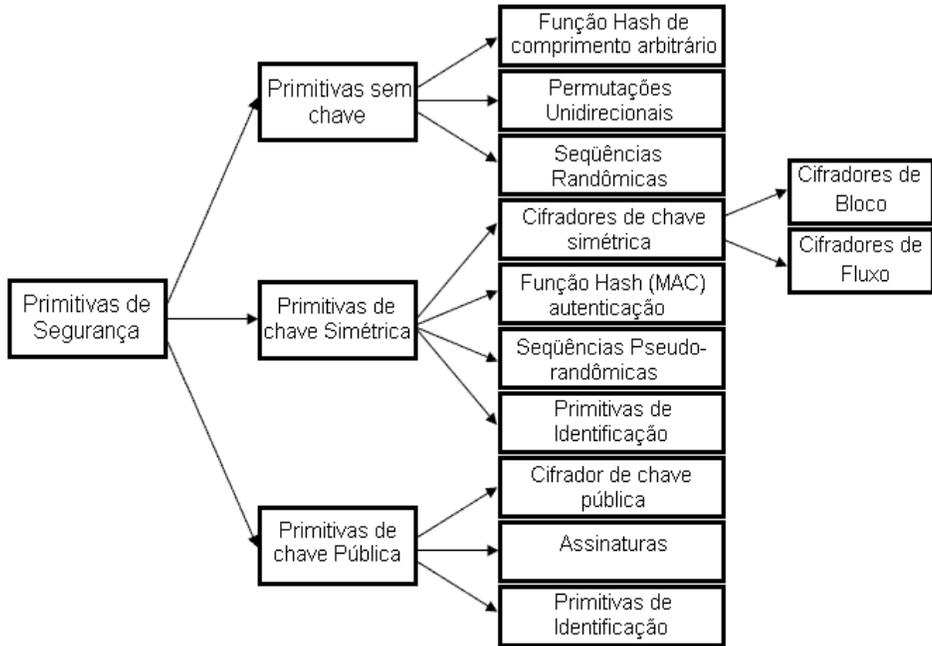


Figura 1: Classificação das Primitivas Criptográficas.
Fonte: Menezes (2011).

Segundo Srivaths (2005) a funcionalidade da segurança pode ser vista em 3 níveis: primitivas criptográficas, protocolos de segurança e aplicações de segurança. Na Figura 2 é possível observar diferentes exemplos de solução, em cada uma desses níveis.

Existem várias implementações na literatura e grupos pesquisando na área de “*hardware security* – segurança em *hardware*”, apesar disso, esta área ainda é carente de soluções eficientes que levem em consideração aspectos de desempenho e funcionalidade, fortalecendo também o critério de facilidade de implementação.

Por isso, se deseja focalizar principalmente o nível 1, primitivas criptográficas, e em alguns módulos do nível 2, visando o uso de primitivas criptográficas modernas na integração de serviços de segurança, sempre sob a visão de projeto de *hardware* específico e eficiente para que essas soluções sejam implementadas e usadas em sistemas embarcados, similar ao realizado em Huffmire (2010) e Hao (2006).

Por outro lado, é inevitável que indivíduos mal intencionados se organizem para criar ataques mais eficientes e inteligentes. Da mesma forma, devem ser criados sistemas integrados de segurança, principalmente conhecendo a importância da integração de serviços de segurança para a melhoria da prevenção, detecção e atuação em situações anômalas (Kastner 2003).

Os modelos de integração de serviços de segurança atuais determinam qual a relação entre um conjunto específico de serviços de segurança para que esses possam integrar informações para prevenir ou tratar anomalias do sistema. Contudo, os modelos atuais propõem soluções sob um conjunto limitado de serviços, ignorando ou desprezando a existência de outros (Wu 2001) (Kastner 2003). Assim, verifica-se que existe dificuldade em definir uma estratégia para a criação de um modelo de integração de serviços de segurança.

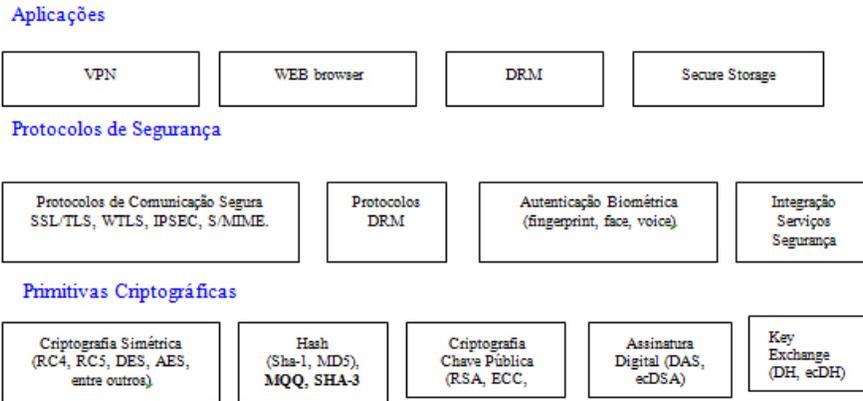


Figura 2: Classificação Moderna de Soluções de Segurança.
Fonte: Srivaths (2005).

2.1 Questões em Aberto em Sistemas Embarcados

Apesar dos inúmeros trabalhos na área de sistemas embarcados, ainda há questões em aberto que permitem focar pesquisas em arquiteturas específicas para aplicações de segurança em *hardware* e criando soluções criptográficas e integração de serviços de segurança, avaliando seu desempenho e consumo de energia. As soluções em *hardware* devem ser eficientes, com especial ênfase em processadores para aplicações específicas e plataformas embarcadas na área de segurança. Nesse processo, também é importante o uso de técnicas de simulação que ajudarão a descobrir características arquiteturais das aplicações em estudo, pois poderão se detectar gargalos e/ou pontos críticos de desempenho e consumo de energia.

A curto e médio prazo, pode se pensar nos seguintes assuntos:

- Estudar e Implementar algoritmos (ex. criptografia, de compressão, de imagens, entre outros) em *hardware*, especificamente FPGAs e SoC, e plataformas embarcadas como microcontroladores, DSP, redes de sensores, plataformas com pouco poder computacional, quando comparados aos tradicionais sistemas computacionais;
- Projetar processadores para aplicações específicas de segurança em *hardware* em FPGAs e SoC, eficientes em desempenho e consumo de energia;
- Projetar processadores para aplicações específicas de processamento de sinais e imagens, ou aceleradores em *hardware*, em FPGAs e SoC, eficientes em desempenho e consumo de energia;
- Conhecer, por meio de técnicas de simulação, as características de execução de algumas dessas técnicas em plataformas embarcadas; de modo a descobrir pontos críticos de desempenho e consumo de energia, que auxiliarão na proposta de otimizações e melhoramentos na descrição de arquitetura específicas e eficientes;
- Propor arquiteturas específicas para essas aplicações, visando principalmente o projeto de processadores de aplicações específicas, com respectiva prototipação em FPGAs, da área de segurança, processamento de imagens, e aceleradores e aplicações críticas de performance;
- Caracterizar o uso de memória e consumo de energia de algumas dessas soluções quando executando em sistemas embarcados (processadores embarcados, microcontroladores, FPGAs, SoC, DSP, e plataformas embarcadas);
- Analisar a possibilidade de criar soluções que tenham reconfiguração parcial e/ou total,

- aproveitando as tecnologias reconfiguráveis;
- Projetar sistemas de segurança (algoritmos criptográficos modernos, tais como algoritmo MQQ, ECC – Criptografia com curvas elípticas e HECC – Criptografia com Curvas Hiper Elípticas, hashing do padrão SHA-3, integração de serviços de segurança) em *hardware* (FPGAs, microcontroladores, DSPs, SoC) eficientes em processamento, velocidade, uso de memória e consumo de energia.

3 Hardware Específico, Arquiteturas Multi-core e GPUs

Um processador multicore consiste em múltiplos processadores dispostos em um mesmo encapsulamento físico e sua respectiva interface com uma placa mãe. Processadores multicore foram introduzidos em vários segmentos da academia e do mercado. Para Domeika (2008), a motivação básica é a performance, pois o uso de processadores multicore pode resultar em execução mais rápida, aumento do *throughput*, e baixo consumo necessário para aplicações embarcadas.

Um sistema multiprocessador consiste de múltiplos processadores inseridos em um único sistema. Os processadores que compõem um sistema multiprocessador podem ser core simples ou multicore. A Figura 3 mostra três diferentes layouts de sistemas: um core simples/processador simples; um sistema multiprocessador e; um multiprocessador/sistema multicore.

Um processador multicore possui dois ou mais cores em um único chip. Segundo Savage (2008) um Multicore difere de multiprocessador nos seguintes aspectos: (i) latência de comunicação - multicore tipicamente possui latência menor que a latência de um multiprocessador; (ii) largura de banda - tipicamente mais alta que a largura de banda de um multiprocessador, pois os processadores estão fisicamente mais próximos e; (iii) número de processadores - multicore apresenta menos processadores que um sistema multiprocessador.

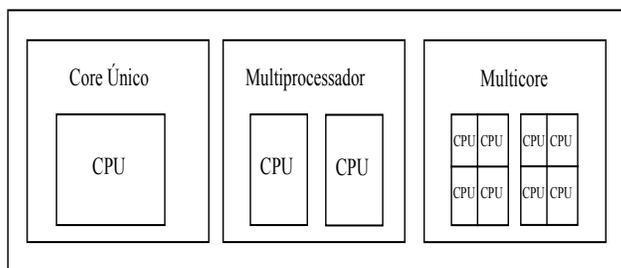


Figura 3: Três configurações de sistema.

Fonte: Domeika (2008).

Uma classificação de processadores multicores é feita com base no tipo do processador presente no mesmo chip (Savage 2008). Um multicore homogêneo é formado por dois ou mais processadores de mesmo conjunto de instruções. Um multicore heterogêneo é composto por processadores com conjuntos de instruções diferentes, possuindo a vantagem de empregar um determinado processador na execução de uma determinada tarefa.

Seguindo a sugestão de Inoue (2009), existe outra classificação de processadores multicore é relativa a como os processadores dispõem-se em relação ao sistema embarcado como um todo. Existem duas classificações chamadas por simétricas e assimétricas, as quais diferem em termos do acesso a memória e a comunicação entre processos.

Conforme explica Meakin (2009), no multiprocessamento simétrico (SMP) os processadores do sistema possuem a mesma visão do sistema e compartilham a memória principal. Isto significa

que valores armazenados na memória principal podem ser acessados e modificados por cada um dos processadores. Em multiprocessamento assimétrico (AMP) a visão do sistema é diferente entre os processadores do sistema e regiões de memória podem estar separadas implicando que dois ou mais processadores não acessam diretamente e não podem modificar uma mesma região da memória.

A comunicação entre os processadores em um sistema SMP é realizada por compartilhamento de memória e o sistema operacional que provê funcionalidades de sincronização e exclusividade de acesso a regiões da memória.

Em sistemas AMP com regiões de memória separadas a comunicação é realizada por meio de passagem de mensagens facilitando que servidores empacotem dados e transfira-os de um S. O. para outro S. O. utilizá-los. Um padrão de comunicação que suporta comunicação entre processadores em um sistema AMP é o *Multicore Communication Application Programming Interface* (MCAPPI) (Meakin, 2009). Uma outra técnica para compartilhar valores em um sistema AMP é prover um mapeamento de memória que usa *hardware* especializado para permitir que dois cores compartilhem espaços de endereçamento distinto entre cores processadores.

Segundo Domeika (2008), dentre os principais segmentos de mercado para processadores multicore e suas possíveis aplicações, pode-se citar:

- (i) Infraestrutura de telecomunicação sem fio, na qual cada telefone celular pertence a um conjunto de processamento que recebe e transmite informações de outro telefones celulares, traça rotas para voz e dados através da rede de comunicação e servir de ligação a outros telefones celulares fora da cobertura de torres de sinal. O benefício primário de processadores multicore neste cenário é vazão de dados, a habilidade de controlar mais ligações. O desafio principal em tomar vantagem de processadores multicore para muitas dessas aplicações são a próprias aplicações em si: elas tendem a ser grandes o que torna difícil conversão para *multi-thread*. Muitas destas aplicações executam sobre sistemas operacionais proprietários que não suportam SMP.
- (ii) Controle Industrial permite que os computadores auxiliem no monitoramento e direcionamento de fábricas que manufaturam uma vasta gama de produtos utilizados no mundo industrializado. Monitoramento eficiente, aquisição de dados e distribuição de informações no chão de fábrica levam a uma produção mais eficiente. Neste contexto, dentre as aplicações industriais alvo para processadores multicore pode-se citar aplicações para instrumentação teste e medições, interfaces homem máquina, PCs industriais e sistemas de controle para automação.
- (iii) Militar, aéreo e governamental são segmentos que possuem dois requisitos acima e além da necessidade funcional específica da aplicação: primeiro, o sistema deve se integrar bem com os sistemas já desenvolvidos e oferecer flexibilidade para futuros desenvolvimentos - Este requisito é desejável para utilizar produtos comerciais de prateleira e economizar; segundo, estes sistemas necessitam operar em ambientes hostis e, ainda assim, por longo tempo.
- (iv) Infraestrutura de segurança de dados de empresas e grandes corporações requerem uma ampla gama de produtos de rede que podem ser classificados como um segmento de mercado para processadores multicore. Alguns destes produtos de rede como o firewall, rede virtual privada e *Secure Sockets Layer* (SSL) podem exigir alto poder computacional em caso de ataque. Detectar e se defender de ataques em massa requer análise de pacotes que exige alto poder computacional, sendo possível extrair vantagens com o uso de processadores multicore.
- (v) O armazenamento de dados tem crescente demanda e dispositivos capazes de armazenar terabytes de dados são cada vez mais comuns e necessários para o armazenamento dos

mais variados arquivos. Os consumidores desejam acessar seus dados a partir de qualquer lugar e exigem segurança, além de que é necessário executar backup e restauração correta dos dados em caso de falha.

- (vi) O segmento de aplicações para área médica e de saúde proporciona que sistemas embarcados administrem dados de pacientes, sejam empregados em dispositivos de mão para acessarem dados armazenados, auxiliar em diagnósticos, medições complexas e sistemas para tratamento. Processadores multicore podem oferecer alta performance e baixo consumo para aplicações como, por exemplo, sistemas de geração de imagens 3D a partir de scanners que requerem máquinas grandes, fixas e complexas podendo torná-las portáteis e mais rápidas.
- (vii) Vigilância digital emprega processamento de vídeo e câmeras de segurança para permitir processamento, análise e armazenamento de grande volume de dados de informações de vídeo para auxiliar nas investigações criminais. Nestes sistemas, processos automáticos podem identificar e traçar pontos de rostos, bem como classificar movimentos suspeitos. O uso de processadores multicore provê a capacidade de processar um grande volume de dados com os mais sofisticados algoritmos de processamento disponíveis.

3.1 Multicores em FPGAs

Nesta seção apresentam-se alguns trabalhos acadêmicos direcionados especificamente para prototipação e uso em dispositivos FPGA. Um FPGA provê a flexibilidade necessária para a implementação e a verificação funcional de vários projetos de circuitos digitais (Gonzalez, 2006) (Gericota, 2003)(Zemva, 1998).

Um multicore 4x4 em *mesh network* direcionado para uso em FPGAs da empresa Altera é apresentado por Minhas (2009). Um sistema de interconexão é utilizado para unir 16 processadores Nios II dispostos em 4 chips FPGA Altera Stratix II. Em cada FPGA, são mapeados 4 processadores Nios II conectados por meio de um *Address-Mapped Resource Network Interface* (RNI). Foi descrita uma *hardware abstraction layer* (HAL) a qual tem base no padrão *message passing interface* (MPI) e os aplicativos utilizam a camada HAL para se comunicarem com o RNI. O RNI transfere mensagens com um tamanho máximo de 512 bytes com 32 bits de dados e 20 bits de cabeçalho. Os autores do trabalho afirmam que o MPI é o gargalo do sistema. A Figura 4 mostra o diagrama do sistema e a configuração da memória usada para comunicar com o RNI.

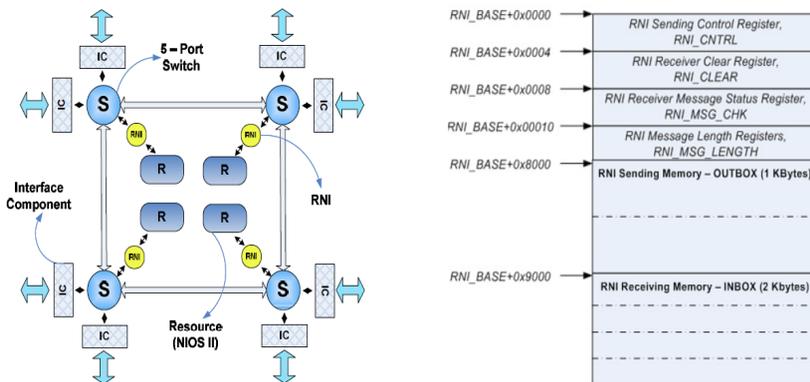


Figura 4: Diagrama do *Quad-Core Noc* e configuração RNI.

Fonte: Autor (2009).

Os *switches* são conectados entre si e à interface. Esta interface prove a lógica de ligação entre a ponte e entre os nós. O RNI se comunica com os processadores Nios II por meio do barramento Avalon da empresa Altera. Isto é conectado no sistema como um escravo Avalon o qual permite ao RNI se comportar como uma memória para o processador Nios II. Utilizando espaços de memória (*memory offsets*), pacotes de dados e informações de controle podem ser lidos ou escritos no RNI.

Para verificar a funcionalidade do sistema, os autores desenvolveram uma aplicação com base em lâmpadas, a qual comprovou a comunicação entre os nós do sistema. A aplicação permite definir o nó destinatário da instrução, a qual liga ou desliga a lâmpada presente no destino.

Em outro trabalho, os autores Kavadias *et Al.* (2010) propuseram um sistema composto por quatro processadores Xilinx Microblaze. O uso de memórias locais por core permite comunicação direta entre cores, com menor atraso e menor consumo de energia em relação ao uso de comunicação com base em coerência de *cache*, especialmente com as arquiteturas CMP tornando-se mais difundidas.

Neste trabalho os autores projetaram uma interface para *caches* integrados em network, a qual combina o uso da flexibilidade dos *caches* e a eficiência de memórias locais. Este trabalho apresenta uma arquitetura que provê acesso local e remoto às memórias, tanto para *words* de dados individuais ou blocos de múltiplos *words*.

O sistema foi implementado em um FPGA Xilinx Virtex5 e quatro Xilinx Microblaze foram integrados no network desenvolvido. A Figura 5 mostra a arquitetura do sistema desenvolvido.

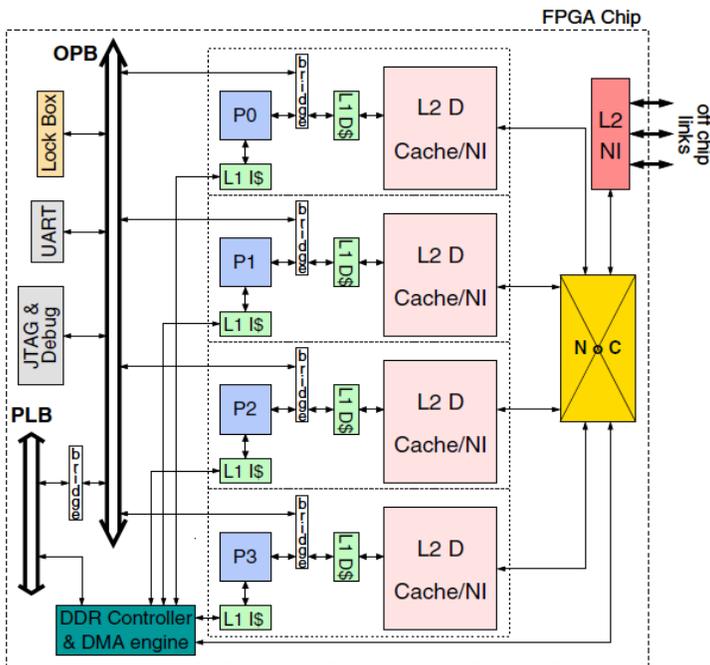


Figura 5: Protótipo do sistema com 4 Xilinx Microblaze em rede.

Fonte: Kavadias (2010).

McCalpin (2011) utilizaram uma adaptação do benchmark STREAM para avaliar a largura de banda do sistema, o qual é destinado a verificar a largura de banda em diferentes níveis de hierarquia de memória. O benchmark STREAM copia três *arrays* de dados de uma memória

remota para uma memória local, executa cálculos simples sobre os elementos deste array e envia de volta os resultados destas computações. Outros *benchmarks* padronizados também foram aplicados para se conhecer a escalabilidade do sistema e a performance.

Outro trabalho foi realizado por Bobda et. Al. (2007). Os autores apresentam uma técnica de projeto para um multiprocessador adaptativo on-chip, especificamente para dispositivos FPGAs. A técnica consiste em dois passos nos quais a infraestrutura de *hardware* é gerada e então, os processadores, os periféricos e os componentes de *hardware* customizáveis são configurados. Para automatizar o desenvolvimento do projeto, ter o máximo de liberdade a partir das ferramentas de projeto e esconder sua complexidade ao projetista, uma plataforma independente foi desenvolvida. Como estudo de caso os autores implementaram um problema chamado singular value decomposition (SVD) em uma placa Xilinx ML310 com até oito processadores.

O sistema consiste de um processador PowerPc e dois processadores MicroBlaze em um FPGA Virtex II Pro 30. A largura de banda medida para o sistema foi 36 Mbytes/s e inclui a transação completa necessária para codificar, enviar e decodificar uma mensagem. A arquitetura conceitual deste sistema é visualizada na Figura 6.

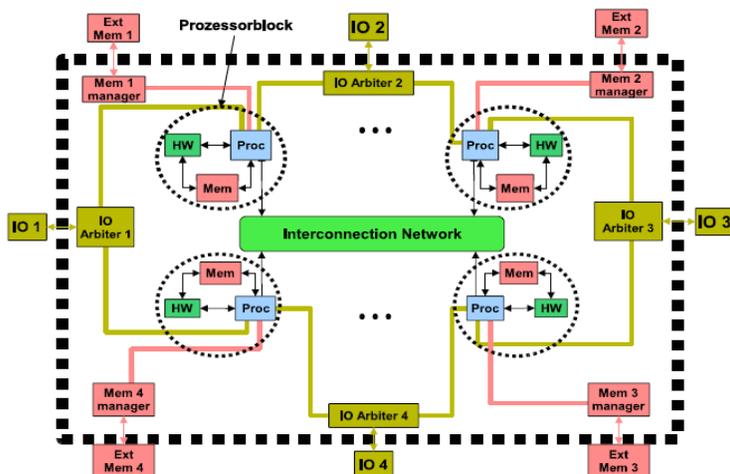


Figura 6: Sistema proposto com processadores Xilinx Microblaze.

Fonte: Bobda (2007).

Ihrig *et al.* (2010) apresentam uma ferramenta para automação de fluxo de projeto chamada ACME a qual facilita a emulação do *hardware* em novos networks de interconexão para amplos projetos de multicores. A ferramenta é voltada a projetistas de computador e network, os quais tem conhecimento de projeto digital mas podem não se sentir confortáveis com linguagens de descrição de *hardware* e fluxos de síntese. A ferramenta ACME utiliza interface gráfica que permite uma mistura de componentes de *hardware* com algoritmos de *software* escritos em C, sendo que cada um contém definições de usuário em termo de atrasos e vazão em termos de ciclos de sistema. A ferramenta ACME automaticamente gera um emulador de *hardware* de ciclos precisos, o qual integra blocos de *hardware* sintetizados com processadores soft-cores que executam o código C.

A Figura 7 mostra o modelo conceitual do ambiente de emulação no qual cada switch é emulado utilizando uma combinação de lógica em FPGA e processadores soft-cores tais como o MicroBlaze, PicoBlaze ou Nios. A lógica básica, como por exemplo os multiplexers, os pipelines,

A empresa Aeroflex Gaisler desenvolveu o Leon3 (Gaisler 2011). O Leon3 é um processador 32 bit com base na arquitetura SPARC V8 e suporte a configurações de multiprocessamento.

O processador é totalmente sintetizável e até 16 processadores podem ser implementados em multiprocessamento assimétrico (AMP) ou multiprocessamento síncrono (SMP). Uma configuração típica com quatro processadores é capaz de alcançar 1400 DMIPS de performance. O multiprocessador Leon3 está disponível totalmente em código fonte, sob a licença GNU GPL para avaliação, pesquisa e propósitos educacionais. Para uso comercial, uma licença de baixo custo é disponibilizada. A Figura 9 mostra a arquitetura interna do sistema Leon3, que segue a arquitetura padrão sugerida por Aeroflex Gaisler.

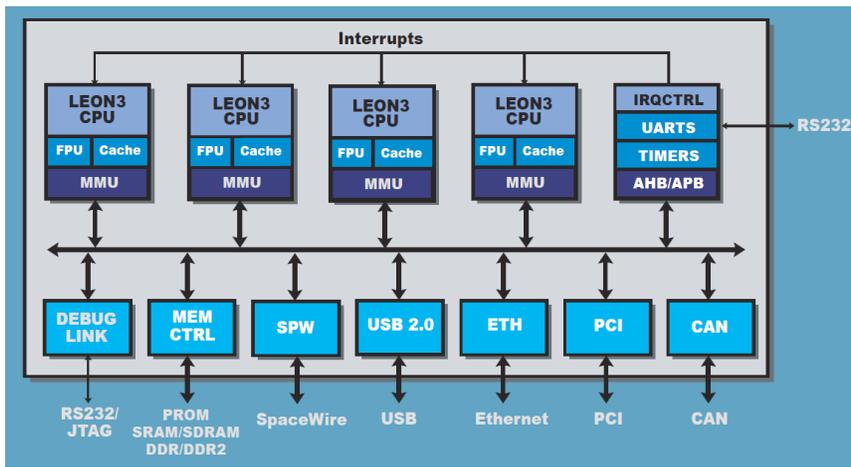


Figura 9: Arquitetura interna padrão do Leon3.

Fonte: Gaisler (2011).

Segundo a empresa Aeroflex Gaisler, o multiprocessador Leon3 provê alta performance com baixa frequência que soluções com processadores únicos. Isto resulta em economia significativa de custos e potencia, enquanto mantém total compatibilidade com ferramentas EDA existentes e fluxo de desenvolvimento.

O Leon3 pode ser utilizado em ambas configurações SMP ou AMP. O processador provê suporte para coerência de cache, enumeração de processadores e conjunto de interrupções. Uma única interface de depuração permite depuração de *hardware* não invasiva para ambos sistemas com um ou múltiplos processadores e provê para todos registradores internos e memória. Um arbitro *round-robin* para o barramento AMBA provê o uso honesto para todos os processadores.

O Leon3 é altamente configurável. A configuração de cada processador em termos de tamanho de cache, FPU e uso do MMU pode ser definida individualmente. Configurações assimétricas tal como dois processadores principais com Unidade de Ponto Flutuante, MMUe dois processadores de E/S são suportadas.

Para configurações SMP, Leon3 suporta os sistemas operacionais VxWorks, Linux 2.6 SMP e eCos. Para configurações AMP, Leon3 suporta os sistemas operacionais RTEMS e μ CLinux.

As principais características do sistema Leon3 são: (i) altamente configurável; (ii) implementação flexível utilizando de 1 a 16 processadores; (iii) tamanho de dados e cache de instruções entre 0k ate 2 Mbytes em cada CPU; (iv) arquitetura SPARC V8 com conjunto de instruções compatível para multiprocessador; (v) 400 MHz em um processo ASIC de 0.13 μ m, atingindo 1400 Dhrystone MIPS de performance com quatro processadores; (vi) cache snooping

embutido para coerência de dados; (vii) modo de baixo consumo onde processadores podem ser desligados individualmente e; (viii) código fonte em VHDL ou netlist.

3.2 Questões em Aberto sobre Projeto de Processadores Específicos

Antes de oferecer assuntos sobre multicore é interessante ter soluções dedicadas para aplicações específicas, conforme a lista seguinte. Depois se pode pensar em ter as mesmas soluções baseadas em um único processador ou em uma única plataforma processada, mas com opções de múltiplos elementos de processamento (multicore).

- Projetar sistemas novos e modernos de segurança em *hardware* e em sistemas embarcados, por exemplo, soluções para redes de sensores, soluções para aplicações automotivas, soluções para aplicações médicas, entre outros.
- Propor novas arquiteturas eficientes e específicas para sistemas de segurança e processamento de imagens em *hardware* e plataformas embarcadas.
- Projetar *hardware* específico para algoritmos criptográficos modernos, como AES, SHA-3, ECC, HECC, e outros.
- Projetar *hardware* específico para processamento de sinais biométricos, como voz, reconhecimento de Iris, reconhecimento de fingerprint, entre outros.
- Projetar aceleradores em *hardware* para aplicações/algoritmos de processamento de imagens, aplicações biomédicas, multimídia, entre outros.
- Projetar processadores específicos para soluções e integração de serviços de segurança e para tratamento de informações contidas em imagens.
- Gerar IP core de algumas soluções de segurança e de processamento de imagens, que requeiram de otimizações em *hardware*.
- Projetar sistemas de segurança visando a integração de serviços.

3.3 Questões em Aberto sobre Multicore e GPUs

- Projetar sistemas multicore eficientes em consumo de energia.
- Paralelizar aplicações que requeiram alto desempenho e adequá-las a plataformas multicore de forma eficiente e escalável.
- Criar ferramentas que paralelizem automaticamente aplicações e as executem de forma eficiente em plataformas multicore.
- Projetar e avaliar sistemas Multi-core e uso de GPUs em soluções de segurança e processamento de imagens que requeiram alto desempenho.
- Considerando a nova geração da arquitetura unificada dos *hardwares* gráficos (GPUs), lançada pela nVidia, que veio tornar capaz a execução de programas genéricos, e que existem aplicações com alto grau de paralelismo, e que transformam as GPUs em verdadeiros clusters; existe ainda necessidade de estudar e aplicar as GPU em várias aplicações que requeiram o uso de alto processamento para alto volume de dados.
- Construir ferramentas que auxiliem no processo automático do uso eficiente de GPUs em aplicações de alto volume de informações.
- Verificar o uso de GPUs em soluções modernas de segurança: IDS e IPS de alto desempenho, em algoritmos criptográficos de desempenho crítico, em algoritmos biométricos e etc.

4 Otimização do Consumo de Energia

Nesta seção são explicados os métodos mais comuns empregados para a estimativa e medição

do consumo de energia em diferentes plataformas computacionais: notebook, PC desktop e simulação arquitetural de um microprocessador específico.

As técnicas mais usadas na literatura são: (i) medição do nível de descarga da bateria para um notebook usando comandos do próprio Linux; (ii) estimativa teórica com base nas características do processador, em um PC tipo desktop; (iii) simulação de uma arquitetura usando alguma ferramenta de simulação e (iv) medição do consumo real usando um osciloscópio.

4.1 Medição de Descarga de Baterias

Esta técnica de medição é semelhante a aquela usada em (Costa, 2007). Quando o tempo de execução de uma aplicação é muito rápido não é possível medir a descarga da bateria usando os comandos próprios do Linux e por esse motivo recomenda-se usar um *Shell script* para que o mesmo seja executado um determinado número de vezes e verificar o nível de carga da bateria após esse número de execuções. Assim, o consumo de energia de cada ciclo de execução do algoritmo é calculado usando o consumo de energia total dividido pela quantidade de ciclos que o algoritmo foi executado. Também é calculado o tempo de execução de cada algoritmo dividindo o tempo de processo total pelo número de execuções indicada no script.

Após a aquisição dos dados são feitos os seguintes cálculos para a obtenção do consumo do dispositivo ao executar o algoritmo:

- (a) Tempo de execução = tempo de execução total / quantidade de execuções;
- (b) Consumo de bateria por execução = consumo total da bateria (Ah)/quantidade de execuções;
- (c) Corrente média = Energia consumida (Ah)/Tempo de execução (hora);
- (d) Potência (W) = corrente média x Voltagem da bateria;
- (e) Consumo (J) Potência média x tempo de execução (s).

A aplicação dessa medição é feita em alguma plataforma específica executando em algum sistema operacional específico. Nestes casos, os testes devem ser realizados várias vezes (realizados em grupos de 50, 100, 200, 500 e 1000 execuções) para cada arquivo em cada algoritmo, e a medição do descarregamento da bateria é feita utilizando a diferença do nível da carga da bateria, apresentada no arquivo `/proc/acpi/battery/BAT1/state` do sistema operacional Linux, ao iniciar e ao terminar a execução de cada algoritmo.

Para a medição do tempo de processo utilizado pela bateria costuma-se utilizar o comando **“time”**, presente no Linux, ao processar o conjunto de execuções de cada algoritmo. Para a execução desses conjuntos recomendasse escrever um Shell Script para cada um dos algoritmos em que são passados parâmetros como a quantidade de execuções e o arquivo de entrada, e os parâmetros da aplicação.

Visando obter uma estimativa um pouco mais precisa do consumo energético do algoritmo que se deseja analisar, recomenda-se medir o consumo do dispositivo nas mesmas condições de funcionamento com o processador ocioso, para tal processo pode-se escrever um *Shell script* que faz com que o processador fique ocioso por determinado tempo.

Esse script escrito deve possuir como parâmetros a quantidade de testes desejados, e os parâmetros do algoritmo. Por exemplo, caso o processo é de encriptação ou decriptação, então o arquivo a ser criptografado e as características da chave e etc. No script é definida a função **getInfo** que captura as informações da bateria. A função básica do script é capturar as informações da bateria, em seguida executar o algoritmo.

4.2 Estimativa usando as Características do Processador

Para a aplicação desta técnica baseada em uma estimativa do consumo de energia requer-se apenas a quantidade de ciclos de *clock* necessários para a execução do algoritmo e as características do processador (como voltagem, velocidade máxima e corrente máxima) (Costa, 2007)(Lin, 2006). Após a aquisição destes dados são feitos os seguintes cálculos para a obtenção do consumo do processador quando executa um algoritmo:

- Tempo por execução (s) = Número de ciclos necessários/velocidade do processador (ciclos/s);
- Potência do processador = voltagem x corrente (A);
- Corrente média – Energia consumida (Ah)/Tempo de execução (hora);
- Consumo (Joules) = Potência do processador x tempo de execução.

$$\text{Tempo por execução (s)} = \frac{\text{Número de ciclos necessários}}{\text{Velocidade do processador } \left(\frac{\text{ciclos}}{\text{s}}\right)}$$

4.3 Análise via técnicas de Simulação

Há vários simuladores, que analisam o comportamento de um algoritmo ou algoritmo seguindo as características e modelos de consumo de alguma plataforma computacional específica. Como exemplo desses simuladores, podemos citar o Sim-Panalyzer (Austin 2010), que é um simulador de consumo de energia baseado no simulador de processadores SimpleScalar (Austin 2002), e simula uma arquitetura computacional completa (com CPU, cache e hierarquia de memória). Com base nesse modelo é possível simular programas reais executando sobre tais plataformas.

Como o resultado é tido com base em uma simulação de arquitetura detalhada, o Sim-Panalyzer consegue modelar de forma detalhada tanto a potência dinâmica quanto a de fuga total do processador alvo, além de fornecer detalhadamente o consumo de cada componente simulado (exemplo: cachê de instruções, cachê de dados, unidades de ponto flutuante, barramentos, unidades de I/O, entre outros), conforme se observa na tabela 3. Os parâmetros para a geração dos perfis de cada componente do computador é dado como entrada para o simulador Sim-Panalyzer que juntamente com o SimpleScalar gera os padrões de consumo de energia.

Tabela 3: Áreas de consumo apresentadas no Sim-Panalyzer.

| Áreas de Consumo | Significado | Áreas de Consumo | Significado |
|------------------|--|------------------|-------------------------------------|
| aio | Endereço da unidade de I/O | dio | Dados da unidade I/O |
| irf | Registradores de inteiro | fprf | Registradores de ponto flutuante |
| il1 | cache de instruções nível 1 | dl1 | cache de dados nível 1 |
| dl2 | cache de dados nível 2 | itlb | Tabela de instruções do TLB |
| dtlb | Tabela de dados do TLB | btb | Branch Target Buffer |
| bimod | Previsor bimodal | ras | Retorno da pilha de endereço |
| logic | Circuito lógico aleatório | clock | Relógio gerador de clock do sistema |
| uarch | Microarquitetura, a maneira com que a ISA é implementada em um processador | | |

Segundo Austin *et al* (2010) o Sim-Panalyzer foi escrito originalmente com base na arquitetura de instruções (ISA) da família de processadores ARM (ARM 1995), obtendo excelentes resultados em simulações deste tipo. Assim, durante a execução dos testes, usaram-se configurações e parâmetros de uma arquitetura ARM, que é usada e configurada como padrão no Sim-Panalyzer.

A aplicação do Sim-Panalyzer é feita de forma simples, necessitando apenas executar o simulador passando como parâmetro a arquitetura a ser simulada, juntamente com as características do sistema, e o programa compilado para a arquitetura ARM, juntamente com os parâmetros para a execução do mesmo.

Para se obter o consumo total do algoritmo é necessário apenas somar consumo energético de cada componente simulado pelo Sim-Panalyzer, obtendo então o consumo geral do dispositivo simulado.

Existem outras ferramentas para realizar a medição, tais como mostrados em (Neto 2010): Netsim, PowerScope, Jouletrack, SNU Energy Explorer *Web*; mas a maioria delas não está disponível ou são de uso restrito.

4.4 Medição Real usando Osciloscópio

Para a medição do consumo de energia real pode ser usado um osciloscópio para medir a corrente consumida pelo processador ao executar o algoritmo desejado. Para medir a corrente deve-se inserir um resistor o qual deve estar conectado em série com o cabo de alimentação da plataforma computacional (Lin 2006)(Lee 2004) e mede-se a diferença de voltagem da entrada e saída do resistor de *shunt* conforme se observa na Figura 10. Com esses dados pode-se calcular a corrente e potência consumidas pelo processador, da seguinte forma:

- (a) Corrente (A) = Variação de voltagem (V)/Resistência em Ohm;
- (b) Potência (W) = Corrente (A) x Voltagem de saída (V);

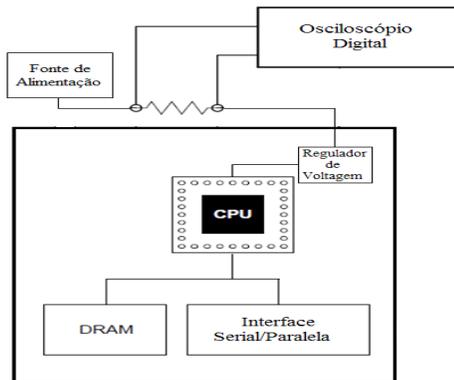


Figura 10: Esquema utilizado para as medições com o Osciloscópio.

O cálculo do consumo do dispositivo durante a execução do algoritmo pode ser feito ao analisar o gráfico gerado pelo osciloscópio e multiplicar o tempo gasto para cada execução do algoritmo pela potência média usada pelo mesmo, calculada com as variáveis: voltagem e corrente reais consumidas pela plataforma.

Como exemplo, os gráficos da Figura 11 mostram mais detalhes do consumo de energia para uma situação específica do algoritmo de hashing SHA, e as diferentes medidas que podem ser obtidas usando osciloscópio, as quais permitem quantificar o real consumo de energia da

aplicação.

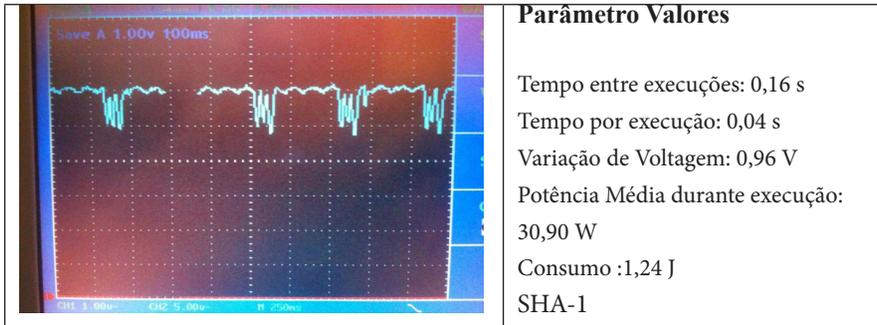


Figura 11: Gráfico gerado para a execução do algoritmo SHA para uma entrada.

Os resultados da medição real do consumo de energia usando um osciloscópio são obtidos a partir dos gráficos gerados pelo mesmo, como o mostrado na figura 12, onde se observam várias execuções, e cada uma delas apresenta o mesmo comportamento no consumo uma vez que se executa o mesmo algoritmo com os mesmos parâmetros.

No caso específico da Figura 12, se observa que o programa SHA demanda mais consumo no começo da execução, o qual se mantém quase constante, com variações pequenas, o que corresponde ao cálculo da mudança das constantes A, B, C, D e E do algoritmo, cada uma delas com tamanho de 32 bits, para compor a palavra especial HASH procurado da mensagem de entrada, composta de 160 bits. No final, se faz somente uma atualização do último cálculo, o que demanda menos processamento e, portanto, menor consumo de energia (ver círculo pontilhado, interno ao círculo total de consumo).

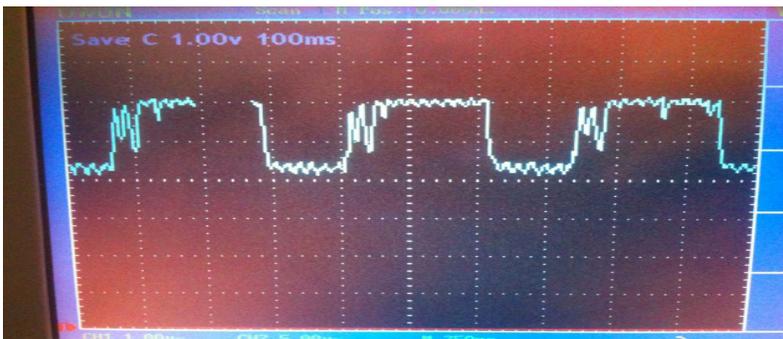


Figura 12: Detalhes do consumo de energia na execução do algoritmo SHA.

4.4 Questões em Aberto sobre consumo de energia

Considerando que os dois primeiros métodos não são eficientes e nem precisos, a comunidade vem usando muito técnicas de simulação e também medições experimentais com equipamentos como osciloscópios, medidores e outros. Apesar disso, ainda há campo aberto para realizar pesquisas nesta área, como por exemplo:

- Criar simuladores para diferentes plataformas computacionais;
- Criar ferramentas que permitam a medição real do consumo de energia na execução de algoritmos e aplicações provenientes do benchmark de sistemas embarcados Mibench (Guthaus 2002), e programas de outros *benchmarks* de aplicações científicas (Ex. o SPEC), Java (*benchmarks*

de Java), jogos, multimídia, entre outros.

- Criar ferramentas que detectem os gargalos no consumo de energia, identificando as funções e/ou trechos de código que mais demandam corrente;
- Otimizar código visando diminuir o consumo de energia;
- Criar novas estratégias de projeto e comunicação, tanto em *hardware* quanto em *software*, que permitam um menor consumo de energia.

5 Conclusões

Neste capítulo foi apresentado um overview sobre os principais aspectos que esta sendo investigados na área de arquitetura de sistemas computacionais no DCOMP. Alguns trabalhos já foram iniciados como o levantamento e estudo comparativo de diferentes técnicas para medir o consumo de energia de aplicações computacionais, conforme descrito em Neto (2010). Outros trabalhos estão em andamento com alunos de graduação e o estudo do impacto da eficiência de algumas opções de compilador no consumo de energia em algumas aplicações do Mibench. Outro trabalho em andamento, refere-se a construção de uma ferramenta que permita mensurar, de forma automática, o consumo de energia de algoritmos e aplicações. Na ferramenta se oferecerá também uma descrição de quais os componentes arquiteturais que mais consumo possui, mostrando assim quais as partes do *hardware* que poderiam ser mais bem projetadas. Outra parte interessante é que a ferramenta oferecerá quais as opções de compilador que mais colaboram com a diminuição do consumo de energia.

Outros trabalhos em andamento referem-se ao projeto em FPGAs dos principais algoritmos do SHA-3. Outro trabalho em andamento no DCOMP/UFS é o projeto de uma solução baseada em microcontrolador que ajudará a otimizar a emissão de gases e líquidos especiais em algumas plataformas da Petrobrás.

Outro trabalho também em andamento é o estudo de ferramentas e o projeto de um *software* que ajude no projeto de sistemas de iluminação de prédios residenciais e comerciais, visando ter um melhor uso da energia elétrica, diminuindo e otimizando as opções de consumo.

Com relação a idéias para trabalhos futuros, cada seção do capítulo oferece uma série de idéias e perguntas que merecem atenção para pesquisa, e que são apresentadas no final de cada seção, no item questões em aberto. Assim, o capítulo deixa alguns questionamentos que merecem dedicação e poderão trazer bons trabalhos de pesquisa e desenvolvimento nas áreas de arquiteturas e *hardware* específico para soluções de segurança em sistemas computacionais e sistemas embarcados; metodologias de projeto para sistemas embarcados; projeto de *hardware* específico para aplicações específicas, aceleradores de *hardware* para aplicações críticas em desempenho, paralelização de aplicações e projeto usando soluções baseadas em sistemas multicore; paralelização e desempenho de aplicações em GPUs, assim como técnicas e ferramentas para mensurar, caracterizar, diminuir e otimizar o consumo de energia.

Referências Bibliográficas

- ARM Versatile Platform, plataforma de desenvolvimento da empresa ARM, 2010. Link www.arm.com/products/tools/versatile.php. Acesso em: 01/03/2011.
- ARM. An Introduction to Thumb. Advanced RISC Machines Ltd., March 1995.
- AUSTIN, T. *et al.* SimpleScalar: An Infrastructure for Computer System Modeling IEEE Computer, Vol. 35, pp. 59-67, Fevereiro de 2002.
- AUSTIN, T.; MUDGE, T.; Grunwald, D. Sim-Panalyzer. Link: www.eecs.umich.edu/~panalyzer/. Acessado em Julho 2010.
- BOBDA, C. *et al.* Design of adaptive multiprocessor on *chip* systems. In: Proceedings of the 20th

- Annual Conference on Integrated Circuits and Systems *Design*. p. 177-183. ACM, 2007.
- CHUN-MING, H. *et al.* Implementation and prototyping of a complex multi-project system-on-a-chip. In: ISCAS 2009. IEEE International Symposium on Circuits and Systems, p. 2321-2324. May 2009.
- COSTA, Ricardo A. G. Desempenho e Consumo de Energia de Algoritmos Criptográficos do MiBench em Sistemas Móveis. UEA - Amazonas. Nov. 2007.
- DOMEIKKA, M. *Software* development for embedded multi-core systems: a practical guide using embedded Intel architecture. 420p. Ed. Elsevier, 2008.
- ECC - Elliptic Curve Cryptography. 10th Workshop on ECC. Fields Institute, Toronto, Canada; September 2006. Link: www.cacr.math.uwaterloo.ca/.../ecc2006/announcement.html. Acessado em 07/2011.
- FAN, J.; GUO, X.; DeMULDER, J.; SCHAUMONT, P.; VERBAUWHEDE, Ingrid. "State-of-the-art of secure ECC implementations: a survey on known side-channel attacks and countermeasures", IEEE Intl. Symposium on *Hardware-Oriented Security and Trust* (HOST2010), Jun. 2010.
- GAIESLER, J. The LEON processor user's manual. Link: www.gaisler.com/cms/index.php?option=com_content&task=section&id=5&Itemid=51. Acesso em 01/03/2011.
- GERICOTA, M. G. O. Metodologias de teste para FPGAs (Field Programmable Gate Arrays) integradas em sistemas reconfiguráveis. Tese de Doutorado apresentada a Faculdade de Engenharia da Universidade do Porto, Porto, Portugal, 2003.
- GONZALEZ, J. Uma metodologia de projetos para circuitos com reconfiguração dinâmica de *hardware* aplicada a support vector machines. Tese de Doutorado, Escola Politécnica da Universidade de São Paulo, São Paulo, Brasil, 2006.
- GUTHAUS, M. *et al.* MiBench: A free, commercially representative embedded suite. Workload Characterization. WWC-4. IEEE Intl. Workshop, 2001.
- HAO, Feng; ANDERSON, Ross and DAUGMAN, John. Combining Crypto with Biometrics Effectively. IEEE Trans. on Computers. Vol. 55, No. 9, Sep. 2006.
- HUFFMIRE, Ted; BROTHERTON, Brett; CALLEGARI, Nicholas; VALAMEHR, Jonathan; WHITE, Jeff; KASTNER, Ryan and SHERWOOD, Tim. "Designing Secure Systems on Reconfigurable *Hardware*", ACM Transactions on *Design Automation of Electronic Systems* (TODAES), vol. 13, issue 3, July 2008.
- HUFFMIRE, Ted; BROTHERTON, Brett; WANG, Gang; SHERWOOD, Timothy; KASTNER, Ryan; LEVIN, Timothy; NGUYEN, Thuy D; IRVINE, Cynthia. "Security Primitives for Reconfigurable *Hardware* Based Systems", ACM Trans. on Reconfigurable Technology and Systems, vol. 3, issue 2, May 2010.
- IHRIG, C. J. *et al.* Automated modeling and emulation of interconnect *designs* for many-core chip multiprocessors. In: Proceedings of the 47th *Design Automation Conference*. p. 431-436. ACM, 2010.
- INOUE, H. A multi-core processor platform for open embedded systems. Dissertação de mestrado apresentada a Graduate School of Science and Technology of Keio University, Setembro 2009.
- KASTNER, Ryan; FALLAH, Farzan and HOSANGADI, Anup. "Arithmetic Optimization Techniques for *Hardware* and *Software Design*", Cambridge University Press, ISBN-13: 9780521880992, May 2010.
- KASTNER, Ryan; KAPLAN, adam and SARRAFZADEH, Majid. "Synthesis Techniques and Optimizations for Reconfigurable Systems", Kluwer Academic Publishers, November 2003, ISBN 1402075983. 2003.
- KAVADIAS, S. G. *et al.* On-chip communication and synchronization mechanisms with cache-

- integrated network interfaces. In: Proceedings of the 7th ACM Intl. conference on Computing frontiers. ACM, p. 217-226, 2010.
- KOC, Cetin Kaya. *Hardware Security*. Material na página [HTTP://islab.oregonstate.edu/koc/](http://islab.oregonstate.edu/koc/). Acessado e 07/2011.
- LEE, I. *et al.* Web-Based Energy exploration tool for embedded systems, IEEE *Design & Test of Computers*, 2004.
- LIN, C. *et al.* Energy Efficiency Measurement for Multimedia Audio Decoding on Embedded Systems. Tunghai University, 2006.
- McCalpin, J. STREAM benchmark, 1995. Link: www.cs.virginia.edu/stream/ref.html#what. Acesso em 01/03/2011.
- MEAKIN, B.; GOPALAKRISHNAN, G. *Hardware Design, Synthesis, and Verification of a Multicore Communication API*. Univ. of Utah. TECHCON 2009.
- MENEZES, Alfred *et al.* Handbook of Applied Cryptography. 2006. www.cacr.math.uwaterloo.ca/hac/authors/ajm.html. Acesso em 07/2011.
- MENEZES, J. *et al.* Guide to Elliptic Curve Cryptography. 2010. www.cacr.math.uwaterloo.ca/hac. Acesso em 07/2011.
- MINHAS, W. H. *et al.* Design and implementation of a plesiochronous multi-core 4x4 network-on-chip FPGA platform with MPI HAL support. In: Proc. of the 6th FPGA world Conference. Stockholm, Sweden: p. 52-57, ACM, 2009.
- NETO, Lealdo Santos; CHELLA, M. T.; MORENO, Edward David. Estudo e Medição do Consumo de Energia de Algoritmos Criptográficos do MiBench. WIC/WSCAD 2010. p. 29-32. 2010.
- PAAR, C. Reconfigurable *Hardware* in Modern Cryptography. Material em www.crypto.ruhr-uni-bochum.de/en/paar.html, 2011.
- PAAR, Christof; CHETWYND, Brebdon; CONNOR, Thomas; DENG, Sheng Yung; MARCHANT, Steve. An Algorithm-Agile Cryptographic Coprocessor Based on FPGAs. ECE Department, Worcester Polytechnic Institute, Worcester, U.S.A, The SPIE's Symposium on Voice, and Data Communications, Sep. 1999.
- RASHEED, H.; CHOW, Y.C.R. An *Information Model for Security Integration*, 11th IEEE International Workshop on Future Trends of Distributed Computing Systems (FTDCS'07), pp.41-47 2007.
- SAVAGE, J. E.; ZUBAIR, M. A unified model for *multicore* architectures. In: Proc. of the 1st international forum on Next-generation multicore / manycore technologies. Cairo, Egypt: p. 1-12. ACM, 2008.
- SRIVATHS, Ravi. SoC: Security on Chip. In Proceedings of the MPSoC, July, 2005. Disponível em www.princeton.edu/~sravi.
- STALLINGS, W. *Cryptography and network security: principles and practice*. 2008.
- WU, Lisa; WEAVER, Chris; AUSTIN, Todd. CryptoManiac: A Fast Flexible Architecture for Secure Communication. Advanced Computer Architecture Laboratory, Univ. of Michigan, U.S.A, In Proc. of ISCA-2001 (Intl. Conf. On Computer Architecture), Goteborg, Sweden.
- YANG, S.; SAKIYAMA, K. and VERBAUWHEDE, I. "A compact and efficient fingerprint verification system for secure embedded devices". In *Proc. of the 37th Asilomar Conference on Signal Systems, and Computers*, pp. 2058-2062, Pacific Grove, Calif, USA, November 2003.
- ZEMVA, A. *et al.* A rapid prototyping environment for teaching digital logic design. *Education, IEEE Transactions*, [S.I.], v. 41, n. 4, p. 8, 1998.

O encontro da *Internet* das pessoas com a *Internet* dos objetos

Marco Túlio Chella

Resumo. A rede de computadores *Internet* e serviços como a *Web* têm evoluído de ferramenta para distribuição de documentos textuais para uma plataforma multimídia e de colaboração na qual, usuários participam ativamente na geração e compartilhamento de informações. Novas propostas nesse caminho de evolução indicam produtos e serviços que adicionam à *Web* contexto e significado as informações, e incorpora dispositivos físicos capazes de gerar e compartilhar dados acessíveis por usuários humanos e outros dispositivos e máquinas. Uma linha de pesquisa que tem sido desenvolvida na UFS (Universidade Federal de Sergipe) é a exploração dos conceitos relacionados à conexão de objetos físicos e desenvolvimento de serviços *Web* que disponibilizam informações geradas por esses equipamentos a usuários humanos de forma amigável, seguindo padrões de interoperabilidade de forma a habilitar a comunicação entre aplicativos e outros dispositivos desenvolvidos que atendam os mesmos requisitos.

Palavras-chave: *Internet*. Sistemas embarcados. Sensor. Atuador. Interoperabilidade.

1 Introdução

No começo da década de 1990 pesquisadores da universidade inglesa de Cambridge trabalhando em salas nos vários prédios do campus compartilhavam uma única cafeteira elétrica e muitas vezes ficavam frustrados ao verificar que depois de uma longa caminhada para tomar um café a cafeteira estava vazia. A solução encontrada foi configurar uma câmera conectada a rede de comunicações que transmitia imagens da cafeteira para todos os computadores permitindo monitorar o nível de café. Uma evolução do sistema foi à conexão da câmera a rede *Internet* tornando a cafeteira uma curiosidade cujas imagens em tempo real foram acessadas por milhões de pessoas ao redor do mundo.

A cafeteira foi a prova de conceito mais conhecida do que hoje é denominada *Internet* dos Objetos. Seguindo esse conceito tem sido desenvolvido um número significativo de projetos e pesquisas acadêmicas que aproveitando o custo decrescente e aumento da capacidade dos dispositivos semicondutores demonstram o potencial e vantagens obtidas ao se adicionar a

objetos diversos processadores, sensores, atuadores e recursos para comunicação Quentin (2011).

Conhecida pelo termo *Internet* dos Objetos compreende a integração de informações em sentido duplo entre objetos e as grandezas físicas que o mesmo expressa representadas e apresentadas em formato amigável e compreensível para os seres humanos, porém não se restringindo a esse cenário.

A capacidade de comunicação favorecida pela utilização de padrões para troca de dados que garantem a interoperabilidade entre plataformas heterogêneas habilita a comunicação entre: pessoas e dispositivos (ex.: monitorar o estado de sensores, acionar dispositivos remotamente), dispositivos e dispositivos (ex.: um dispositivo ao detectar um estado pode enviar mensagem a outro dispositivo que poderá executar alguma ação como acionar um alarme ou outro dispositivo), dispositivos e aplicativos (ex.: um dispositivo que detecte o código de uma etiqueta inteligente pode enviar mensagens a uma aplicação que faz controle de inventário ou estoque).

Embutir alguma capacidade de processamento e comunicação em objetos que nos cercam pode propiciar um ganho significativo na qualidade de vida individual e para a sociedade como um todo. Objetos conectados em rede podem ajudar as pessoas a mudarem seu comportamento de forma positiva ao fornecer informações adequadas e no momento oportuno, disponibilizar indicações ou alertas de condições ambientais ou de saúde antecipando e prevenindo situações de risco.

Nesse trabalho são discutidos os projetos e pesquisas em andamento na UFS, tecnologias de *software* e *hardware* para a implementação de dispositivos e objetos conectados, e perspectivas futuras da *Internet* dos Objetos.

2 Cenário atual

Pesquisas indicam que atualmente existem aproximadamente um bilhão e meio de computadores pessoais conectados a rede *Internet* e aproximadamente um bilhão de dispositivos móveis como *smartphones* e *tablets* habilitados para conexão em rede. Segundo Nelson (2011) esses números indicam que a *Internet* atual dos computadores pessoais está se movendo com um crescimento exponencial para a *Internet* dos Objetos sendo a expectativa para os próximos 5 a 10 anos de que 100 bilhões de dispositivos estarão conectados em rede.

O incremento em poucos anos da ordem de unidade para centena de bilhão dos dispositivos conectados à rede *Internet* irá causar uma demanda por soluções como as tecnologias para suporte a computação ubíqua com e sem fio, computação em nuvem, serviços para interoperabilidade, entre outros.

O salto propiciado pelo baixo custo com o aumento da capacidade computacional associados aos recursos para comunicação em rede disponibilizados nos dispositivos móveis como telefones móveis, *tablets* e *smartphones* tem movido o mercado para o que é classificado como computação e *Internet* em qualquer lugar e em qualquer momento.

De maneira geral os computadores de mesa e os móveis com arquitetura semelhante tem sido a forma primária e mais usual de acessar a *Internet*. Assim em grande parte a forma de gerar, gerenciar e distribuir conteúdo e informações tem privilegiado as características desses computadores tradicionais.

A popularização do acesso a *Internet* em dispositivos móveis e relativamente pequenos tem forçado e direcionado a indústria para inovações na área de conteúdo, entretenimento e comércio eletrônico. Muitas dessas inovações são necessárias pelo fato dos usuários de informações em dispositivos móveis apresentarem uma demanda com características diferentes dos usuários de um computador de mesa ou notebook.

Ao ecossistema de dispositivos móveis estão sendo incorporados pequenos dispositivos cujas principais características são baixo consumo, pequenas dimensões físicas, com capacidade

computacional e de comunicação. São adequados para execução de tarefas específicas como medir o consumo de energia de uma casa, informar os produtos que estão com a validade vencida na geladeira, ou monitorar sinais biológicos de um paciente nas atividades normais do dia a dia, enviando as informações para uma central, inclusive podendo alertar anomalias ou condições pré-estabelecidas. Esse tipo de dispositivo que executa processamento, troca e compartilhamento de informações, muitas vezes tem dimensões físicas que o fazem parecer invisível, tem sido enquadrado na categoria dos dispositivos para computação ubíqua ou pervasiva.

No início da década de 1990 alguns desenvolvedores começaram a desenvolver a partir de microcontroladores genéricos dispositivos com *internet* embarcada. Isto é dispositivos que tinha implementado em firmware uma pilha TCP-IP o que permitia a esses dispositivos se conectar a *Internet* sem a utilização de um computador tradicional. Cabe salientar que a tecnologia de microcontroladores nessa época era limitada em termos de capacidade de processamento e memória.

Um exemplo é o projeto *iready* (2011) baseado em um microcontrolador de 8 bits. Com pequenas dimensões (figura 1) tem implementado um servidor *Web* em pouco mais de 100 bytes. Nesse servidor páginas *Web* em HTML eram armazenadas em uma memória EEPROM auxiliar, apresentando funcionalidades para interação como exibir no navegador *Web* valores de sensores conectados ao microcontrolador e permitir a alteração dos níveis lógicos nos pinos de entrada e saída do dispositivo.

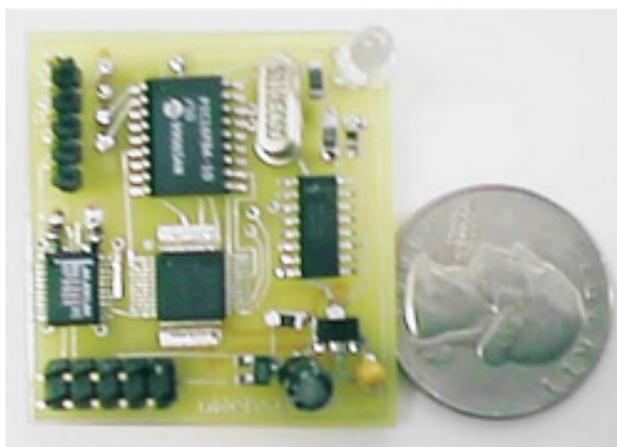


Figura 1: Placa com circuito do *uWebserver Iready* (2011).

Dispositivos com processamento e recursos para comunicação em rede contendo sensores de diversos tipos e atuadores podem detectar grandezas físicas e reagir em função de determinadas variáveis ou parâmetros. A capacidade de comunicação provê a oportunidade de monitorar e analisar fluxo constante de dados em tempo próximo ao real implicando em uma nova categoria de sistema de buscas capazes de atuar com dados gerados em tempo real.

A consultoria Gardner (2011) prevê que até o final de 2012, sensores físicos conectados em rede irão gerar até vinte por cento do tráfego de rede, excluindo dessa estimativa o tráfego de vídeo: *“The extent and diversity of real-time environmental sensing is growing rapidly as our ability to act on and interpret the growing volumes of data to capture valuable information increases. Organisations should consider how use of the growing volume of real-time data might create new business models and enable better informed business decision making.”*

Esses dispositivos com dimensões relativamente pequenas e baixo consumo de energia

habilitam a inserção em objetos e equipamentos que são utilizados no dia a dia favorecendo o conforto, segurança e muitas vezes oferecendo uma experiência de uso mais satisfatória.

Um contexto de aplicação é no trânsito de veículos nas cidades em especial as brasileiras onde a cada dia as ruas recebem mais veículos como carros, caminhões e motocicletas. Aplicações tradicionais como geolocalização e redes sociais tradicionais com algumas adaptações podem transformar veículos em objetos conectados em rede. As tecnologias base já são conhecidas como GPS (*Global Position System*), aplicações e ferramentas para desenvolvimento para os serviços de redes sociais e a infraestrutura de comunicação para *Internet* móvel sem fio com tecnologias 3G e *Wi-Fi* e a próxima evolução 4G que promete maior velocidade e confiabilidade.

Veículos atuando como objetos em rede podem interagir dentro de uma organização que inclui os objetos de rede internos ao veículo (objeto para objeto), comunicação dos objetos do veículo com pessoas, utilizando os mecanismos de rede sociais, veículo para veículo e veículo com a infraestrutura.

Esses veículos enquanto trafegam podem gerar uma quantidade significativa de dados em tempo real. Jarvis (2009) observa que dados coletados e aplicados de forma adequada podem beneficiar motoristas, corporações, entidades públicas responsáveis pela manutenção operacional das vias, e serviços de emergência e atendimento ao usuário. Os dados coletados em grande volume e analisados poderão revelar relações e os diversos fatores que afetam o fluxo de veículos. Para exemplo as condições de temperatura e umidade em uma via podem afetar e alterar os sistemas de informação indicando aos usuários as condições de velocidade adequadas para determinada situação ambiental, podendo ainda alertar os sistemas de emergência.

Veículos conectados formando uma rede de comunicação podem trocar dados entre si informando em tempo real as condições do trânsito, sinalizando emergências, indicando alterações na velocidade do fluxo, sugerindo ao motorista uma desaceleração ou em alguns casos podendo atuar diretamente nos sistemas de parada do veículo sem intervenção do condutor.

Os objetos na rede interna do veículo podem ser integrados a sistemas da infraestrutura oferecendo serviços e informações ao motorista sobre as condições do trânsito e do clima. Serviços de RSS (*Really Simple Syndication*), um subconjunto de tags XML como função de agregador de informações geralmente utilizado para informar ao usuário atualizações em blog e serviços de notícias, pode ser integrado aos sistemas de navegação como o GPS indicando vias em manutenção, acidentes, ou congestionamentos. Essas informações associadas ao sistema de posicionamento pode após processamento gerar um novo roteiro que evite as situações adversas.

Um exemplo da topologia para comunicação veículo para veículo é o serviço Trapster(2011) que inicialmente se conectava a dispositivos GPS de alguns fabricantes e atualmente tem aplicações disponíveis para equipamentos como smartphones com sistemas operacionais como o da empresa Apple e o Android iniciativa da empresa Google.

O sistema da Trapster emprega um modo de operação semelhante a wiki, onde os usuários colaboram gerando conteúdo relativo a condições de trânsito como radares, limites de velocidade e condições que podem representar risco a segurança dos motoristas. Os dados são enviados e recebidos pelos usuários em tempo real. Esse serviço atualmente utilizando conteúdo gerado por usuários e dispositivos cria serviços que beneficiam o trânsito de forma geral, mas que ainda explora um pequeno conjunto de possibilidades.

Serviços que ainda estão em fase de projeto propõem a utilização de sensores para indicar ocupação por veículo nos estacionamentos, associados aos sistemas de GPS irão indicar ao usuário por meio de um mapa as posições livres em determinado espaço.

Os objetos conectados em rede têm reduzido a fronteira entre as grandezas físicas ou átomos e os bits de informação digital. Os objetos físicos da mesma forma que os serviços de *internet* disponibilizados para as pessoas podem também assumir a representação de um avatar, ter um

apelido e ser capaz de comunicar informações para outros dispositivos ou pessoas. Segundo Kuniavsky (2010) objetos do dia a dia como calçados são expandidos pela tecnologia, e se tornam componentes de aplicações acessíveis a partir de um computador.

Um exemplo é o tênis Nike+iPhone (2010), que contém um sensor de aceleração e um dispositivo de rádio que comunica essa grandeza física registrada pelo sensor para um *smartphone*, possibilitando medir a distância e quantidade de passos da corrida ou caminhada. Enquanto o objeto físico gera valores físicos analógicos o valor adicionado está na informação digital gerada, armazenada e que após ser disponibilizada na rede pode ser compartilhada. O aplicativo que acompanha o tênis cria um registro contendo informações adicionais como quantidade de calorias consumidas, informação que é sobreposta às informações dos sensores. A combinação de tecnologias faz com que as informações de um tênis possam ser acessadas por um mecanismo de busca. Recentemente, uma camada de ligação com redes sociais permite que os dados sejam compartilhados e atividades como desafios possam ser agendados e acompanhados via rede (figura 2).



Figura 2: Interface gráfica do aplicativo Nike+Iphone (2010)

Na área de saúde e telemedicina algumas empresas começam a oferecer produtos e serviços em que os dispositivos conectados são um elo fundamental. A empresa Vitality criou o produto GlowCap (2011) (figura. 3), um recipiente para acondicionar medicamentos que são usados no dia a dia pelo paciente. Fisicamente sua aparência não difere dos recipientes comuns, porém um sistema embarcado com conexão a rede *internet* sem fio provê um conjunto de serviços. Luzes e sons indicam a hora que o medicamento deve ser administrado.

Quando o recipiente é aberto uma mensagem com o seu estado é enviada por meio de uma rede de comunicação. Se o recipiente não for aberto no prazo de duas horas ao estipulado o paciente é avisado por meio de uma chamada telefônica. O suporte social é fornecido com o recurso de envio de email periódico contendo um sumário com informações sobre a utilização do equipamento.

Esse email pode ser configurado para recebimento pelo paciente, uma pessoa próxima como um amigo ou parente e o médico. Uma outra funcionalidade é a capacidade de registrar a quantidade de medicamento disponível e em caso de falta uma mensagem para uma farmácia é disparada solicitando a reposição.

Estudos iniciais da empresa que desenvolveu o produto indicam que a utilização do glowcap tem colaborado na elevação dos índices de aderência a tratamentos a doenças que dependem da administração de medicamentos por longo prazo.



Figura 3: Dispositivo Glowcap.

Um outro produto que agrega ao dispositivo conectado o conceito de rede social é o Bodytrace (2011), uma balança cujos valores de massa medidos, também por meio de conexão a rede envia dados para um *Website*. Com base nos dados enviados o sistema do site oferece um conjunto de funcionalidades e recomendações com foco em dieta alimentar e monitoramento de peso. Um componente de rede social possibilita compartilhar as informações e troca de mensagens entre os usuários.

Numa escala maior, segundo prevê Sterling (2005), os recursos tecnológicos que permitem hoje a identificação única e comunicação de cada objeto produzido pela indústria, permitirá que cada objeto se integre a sistemas de gerenciamento de informação e também operando de forma autônoma serão capazes de indicar a sua localização, sua forma de atuar dentro de um contexto, gerar um registro de operações, interação com dispositivos próximos a partir de uma política de localidade e esse conjunto de informações indexado em um sistema de busca permitirá que esses objetos possam ser buscados, encontrados e suas características ou informações obtidas a partir de uma ferramenta como um buscador tradicional.

A esse conceito Sterling deu o nome de *Spime* que de forma genérica são objetos inseridos em um espaço físico real com capacidade para processamento de dados, comunicação e controle para atuar no meio em que está inserido.

Um dos cenários de aplicação desse conceito é em uma loja de varejo, que hoje em dia ultrapassa facilmente em milhares a quantidade de itens em estoque. Cada item já que tem sua identidade única pode ser acompanhado desde o início da produção até o seu descarte e eventual reciclagem o que pode de certa forma colaborar com as ações de sustentabilidade, já que todo o ciclo do produto do início ao fim pode ser acompanhado e registrado.

Um dos contextos que tem uma grande perspectiva de exploração dos *spimes* são as ações de marketing em pontos de venda ao consumidor como as gôndolas de um supermercado em que cada produto armazenando informações e a sua posição podem se comunicar e controlar alguns recursos do ambiente. A figura 4 ilustra as várias possibilidades que um sistema com essas características pode oferecer. Observar que algumas interações ocorrem entre o objeto e o cliente, outras entre os objetos, e outras interessam aos sistemas de gerenciamento para administração e controles como o estoque.

Todas as tecnologias disponíveis para implementação de um sistema desse tipo estão disponíveis atualmente. Algumas ainda possuem a restrição de alto custo, outras ainda necessitam atenção em aspectos como robustez, confiabilidade e segurança.

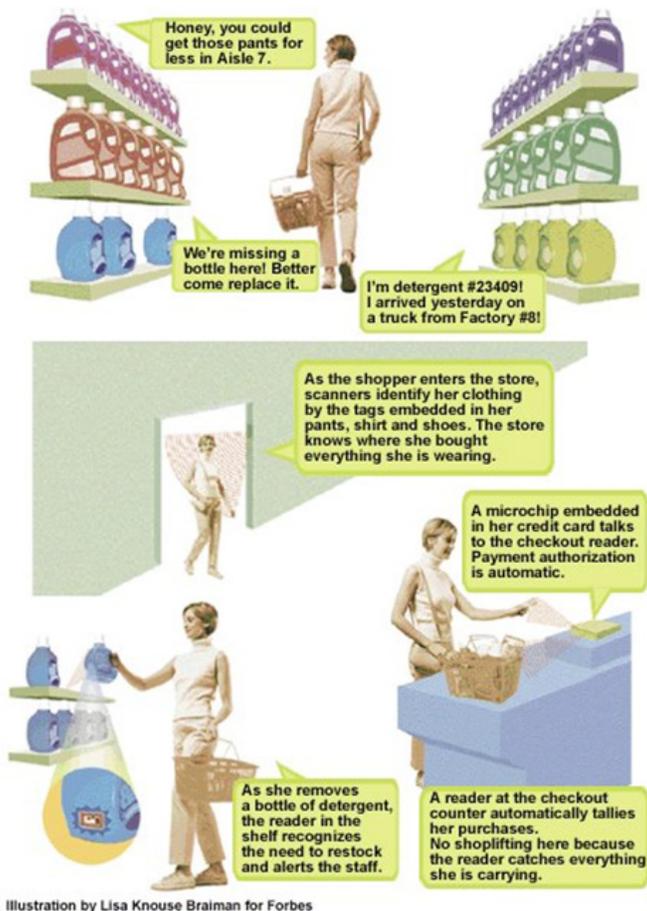


Illustration by Lisa Knouse Braiman for Forbes

Figura 4: Ilustração com o conceito de *Spime* Sterling (2005).

O conceito pioneiro com propostas para a identificação eletrônica de objetos teve origem no MIT Auto-ID Center e os trabalhos de Brock (2001) e Asthon (2011). O termo "Auto-ID" refere-se às várias classes de tecnologias para identificação incluindo código de barra, smart cards, sensores diversos, reconhecimento de voz e biometria. A partir de 2003 a tecnologia de identificação por radiofrequência (Radio Frequency Identification - RFID) foi reconhecida com ferramenta com características tecnológicas mais adequadas para emprego em identificação de objetos

Nesse ano de 2003 o simpósio executivo do instituto americano EPC (*Electronic Product Code*) lançou o programa EPC Network, uma infraestrutura de tecnologia aberta que permitiria aos computadores identificar produtos manufaturados possibilitando o rastreamento da etapa de fabricação até o fim do seu ciclo de vida.

O simpósio obteve apoio de 90 das maiores empresas do mundo em áreas como alimentação, bens de consumo, transporte, indústria farmacêutica, entre outras. Houve o consenso de que a tecnologia RFID seria a tecnologia chave aplicável nas várias atividades produtivas. Do ponto de vista histórico o simpósio pode ser considerado o marco em que o computador incorpora a funcionalidade de se comunicar com os objetos do mundo real além da já tradicional função de processar dados.

A EPC Network e o Auto-ID não buscaram criar outra rede global paralela a *Internet*, mas desenvolver os elementos tendo a *Internet* como uma das camadas que permitisse a interação com objetos físicos e o compartilhamento de dados em escala planetária.

Outro marco do desenvolvimento da *Internet* dos Objetos ocorreu em 2008 quando um grupo de empresas se reuniram sob a coordenação da IPSO (2011) alliance para promover o IP (*Internet Protocol*) como a rede dos objetos inteligentes. A IPSO é apoiada por grandes empresas como Bosh, Cisco, Intel, SAP, Google e Fujitsu.

Objetos inteligentes são definidos pela IPSO como pequenos computadores com sensores e atuadores com recursos de comunicação, embarcado em termômetros, motores de carros, interruptores domésticos, e máquinas industriais, para citar alguns exemplos, habilitando uma ampla faixa de aplicações em áreas como automação residencial, monitoramento na indústria, infraestrutura para cidades inteligentes, gerenciamento de consumo de energia e transporte.

Atualmente o sistema de endereçamento utiliza o sistema IPv4 capaz de fornecer quatro bilhões de endereços. É previsto um novo sistema que irá adotar o protocolo IPv6 com espaço de endereçamento de 3.4×10^{38} , número que para fazer uma analogia, provê endereços suficientes para endereçar todos os grãos de areia de todas as praias do mundo. Esse dimensionamento se faz necessário à medida que forem sendo atendidas as previsões pelas quais em breve cada pessoa estará em contato com pelo menos 5000 objetos inteligentes no seu dia a dia.

Um sistema com a tecnologia RFID utiliza a comunicação por radiofrequência para identificar, categorizar, localizar e rastrear objetos. Dois elementos são necessários em um sistema mínimo: um *transponder* e o leitor.

O *transponder* normalmente é chamada de etiqueta (*tag*) e é um dispositivo que gera os sinais elétricos ou pulsos que são recebidos pelo leitor. O leitor é uma combinação de receptor e transmissor (*transceiver*) composto de um circuito para comunicação por rádio, microprocessador, responsável por codificar e decodificar mensagens, executar algumas funções de processamento e prover uma interface para comunicação de dados com um computador.

O sistema de comunicação é definido em classes de frequência de operação. São definidas bandas que se dividem em alta, média e baixa. Os sistemas são projetados de acordo com as características específicas de cada faixa de frequência.

Um sistema que opera na faixa de baixa frequência, entre 30kHz a 500kHz, tem o alcance limitado a pouco mais de alguns centímetros. A faixa de média e alta frequência opera na faixa frequência entre 13.5MHz e 2.4GHz, apresenta maior alcance podendo chegar a vários metros, porém dependem para o seu uso de autorização dos mecanismos reguladores de cada país já que compartilham espectros de frequência com outros serviços de emissão de radiofrequência.

Embutir em objetos alguma capacidade de processamento e controle pode criar condições que beneficiam a qualidade de vida das pessoas, muitas vezes reduzindo o consumo de recursos naturais. O valor agregado a esses objetos está na capacidade de servir melhor aos seus usuários, possibilitando a mudança de comportamento na forma como se obtém informações, alertando sobre compromissos ou ações relacionadas ao objeto e outras necessidades comuns ao dia a dia.

Por exemplo, algumas universidades e condomínios residenciais no exterior têm conectado recursos de uso comum como lavanderia, vagas de estacionamento, equipamentos de ginástica aos sistemas de informação do local. Dessa forma os usuários sabem da disponibilidade dos equipamentos e podem inclusive alocar o tempo de utilização dos mesmos. Alguns desses dispositivos podem ainda ter um recurso para comunicação que os permita se comunicar com serviços de mensagem de texto (SMS) ou mesmo serviços *Web* de mensagens curtas como o Twitter, enviando mensagens que informam o estado, o fim da execução de uma tarefa ou a sua disponibilidade.

Serviços de suporte podem receber um email diretamente de um equipamento indicando

alguma anomalia. Em muitos casos, o registro contínuo dos parâmetros de operação do equipamento pode gerar alertas que antecipam problemas que são resolvidos com manutenção preventiva.

Um das áreas que começa a ser beneficiada pela *internet* dos objetos é a pesquisa científica. A ciência básica por sua natureza é construída em metodologias nas quais a coleta de dados diversos é fundamental, e geralmente o limite de amostras possível de obter é insuficiente para análises. Tecnologias atuais para coleta e sensoriamento possibilitam realizar pesquisas científicas em uma abordagem que combina sistemas para coletas de dados com baixo custo e capacidade computacional e de comunicação, conectados a sistemas com grande capacidade de armazenamento.

Sensores desenvolvidos por empresas ou universidades não mais ficam restritos a pequenos períodos de coleta e dependentes da presença de pessoas para a sua operação, assim é possível um tempo de aquisição maior, em resoluções e escala cada vez menores e com uma variedade de grandezas analisadas maiores. Esses dados podem ser compartilhados e estudados por uma comunidade maior o que pode resultar numa maior qualidade dos trabalhos e melhores análises. Os resultados gerados por esses fatores propiciam um fluxo de dados cada vez maior.

Para caracterizar a ordem de grandeza que a coleta de dados pode assumir, um exemplo é o conjunto de dados provido pelo serviço Fluxnet (2011) contendo 966 locais para coleta anual de dados de sensores. Alguns desses dados estão armazenados em arquivos com vários Terabytes e cujo processamento pode ocorrer em computadores distribuídos nos vários centros de pesquisa já que muitas bases são compartilhadas pelos pesquisadores.

Essa forma de fazer ciência é descrita por Gray (2009) como quarto paradigma científico e usa o termo *e-science* para descrever uma forma de fazer ciência que resolve as limitações impostas a problemas e pesquisas de cientistas que não podem testar suas idéias e teorias pela limitação dos dados disponíveis.

O quarto paradigma caracterizado pela disponibilidade extremamente grande de dados experimentais obtidos a partir dos mais diversos instrumentos excedendo muitas vezes a capacidade de análise com os métodos atuais, cria a oportunidade e necessidade de novas ferramentas para apoiar os cientistas na análise e obtenção de resultados.

À medida que o custo do *hardware* e os sistemas para desenvolver aplicações baseados em sistemas embarcados como *smartphones*, *tablets* ou mesmo microcontroladores, se tornam mais populares é preciso prover uma infraestrutura de serviços capaz de conectar as mais diversas formas desses dispositivos. Com essa proposta foram desenvolvidos serviços como o Pachube (2011). Baseado na *Web* o Pachube tem como propósito habilitar a conexão em modalidades como objetos e objetos, objetos e ambientes e prover armazenamento e compartilhamento de dados com usuários humanos e aplicações.

Os dados fornecidos pelos dispositivos físicos podem ser visualizados em tempo real em gráficos que mantém um histórico das leituras (fig 5.) Esses dados podem ser utilizados como informação para controlar e atuar em outros dispositivos remotamente caracterizando a comunicação objeto com objeto. O modelo para troca de dados e comunicação é baseado em protocolos padrão *Web* e formatos como XML que garante a interoperabilidade entre os diversos sistemas. O serviço ainda provê aos desenvolvedores uma interface de programação com *Webservices* baseada na arquitetura REST (Fielding,2000).

A arquitetura que garante a interoperabilidade do serviço permite que dispositivos e aplicativos desenvolvidos com as mais diversas arquiteturas possam se comunicar e trocar dados. Uma interface gráfica *Web* possibilita o desenvolvimento de serviços como, gerenciamento de sensores remotos, exibição de gráficos e controle de ambientes remotamente, possibilidade de compartilhar dados de dispositivos físicos e criar comunidades.

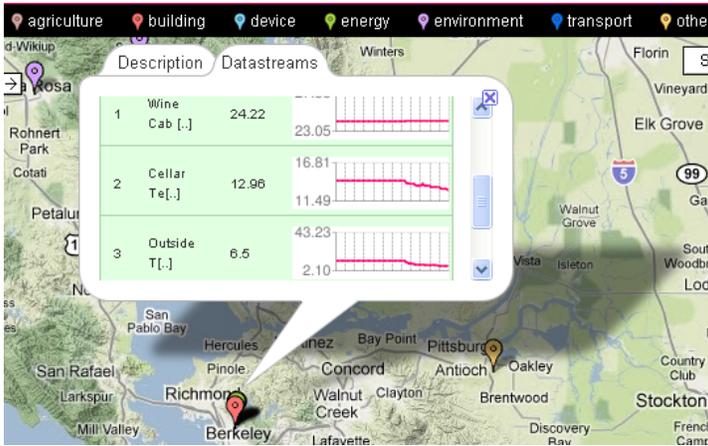


Figura 5: Interface gráfica do serviço Pachube.

O termo NED (*Networked Embedded Devices*) tem sido utilizado para designar a extensa diversidade de aplicações para dispositivos com capacidade computacional e recursos para comunicação em rede, capazes de auto reconhecimento, informar sua descrição ao sistema em que está inserido e coordenação para prover uma experiência de uso e aplicação integrados na rede em que está inserido.

O futuro da *Internet* consistirá de dois núcleos em duas camadas (Presser, 2010): o núcleo será composto pela evolução da infraestrutura da *Internet* atual, a camada externa será composta por uma nova geração de terminais com recursos de comunicação a rede e possivelmente dotado de recursos para auto organizar-se em redes, a segunda camada baseada em dispositivos conectados irá permitir a combinação do mundo real com o mundo digital (figura 6).

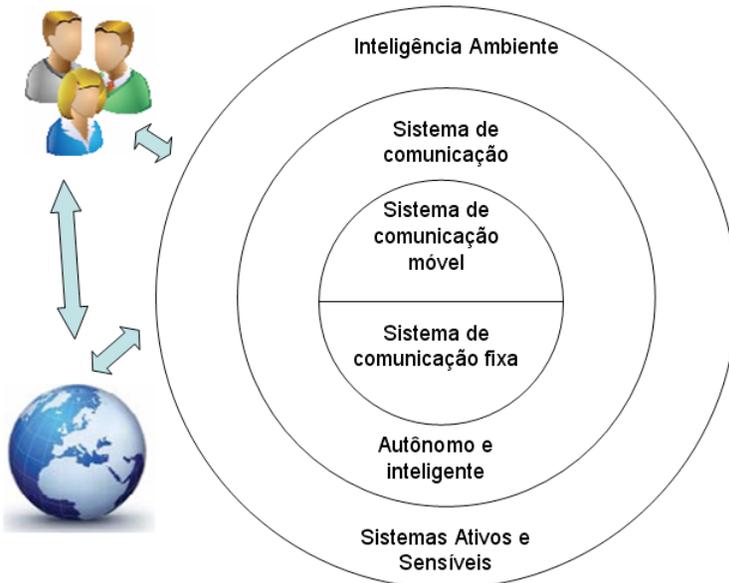


Figura 6: Representação com evolução da *Internet*.
Presser, (2010).

3 Aspectos e desafios para projeto e integração da *Internet* dos Objetos

O ecossistema que compõe a denominada *Internet* dos Objetos é constituído por um sistema de *hardware* dotado em grande parte de sensores capazes de capturar grandezas físicas diversas, atuadores capazes, por exemplo, de acionar outros dispositivos como máquinas e sistemas de iluminação, e recursos para comunicação em rede. Esses diversos objetos devem operar por longo tempo sem depender de intervenção humana, com alta confiabilidade e tolerância a falha tanto do ponto de vista de *hardware* quanto de *software*.

Alguns requisitos relacionados aos sistemas de *hardware* enumerados a seguir e depois discutidos de forma breve. São eles:

- Gerenciamento de energia em nível de objeto.
- Encapsulamento e integração de sensores, atuadores e sistemas de comunicação.
- Desenvolvimento e calibração dos sensores e atuadores.
- Tecnologias para a comunicação, que de forma geral pode ser com e sem fio.
- Aspectos de confiabilidade, segurança e robustez.
- Capacidades para auto reconfigurar *hardware* e *software*.

Tendências indicam um crescimento significativo das redes de objetos físicos conectados em futuro próximo. Esses objetos, nós de uma rede, poderão ter topologias que podem variar de umas poucas unidades a milhares, podendo estar posicionadas em locais de difícil acesso para manutenção. Considerando essas características um ponto importante é que esses módulos ou nós devem ser auto suficientes quanto a alimentação elétrica, muitas das vantagens dos dispositivos em redes sem fio é que não existam fios também para a alimentação. Uma das opções utilizadas são as baterias recarregáveis, que podem oferecer uma densidade de energia alta a baixo custo. Porém, atualmente a evolução dos recursos computacionais dos circuitos integrados foi muito mais rápida se compararmos com a evolução das tecnologias de bateria. Como consequência observa-se que a percentagem em ocupação de tamanho e peso das baterias tem aumentado de maneira significativa, sem que isso represente um maior tempo de autonomia. Na especificação das baterias para um sistema devem ser levados em conta alguns outros pontos como questão ambiental caso o sistema seja perdido, o econômico caso seja necessário realizar a troca das baterias, impossibilidade de trocar as baterias caso sejam embutidas no sistema durante o processo de fabricação ou os utilizados em implantes biomédicos.

Uma solução possível é a substituição da bateria por sistemas que capturam energia do meio ambiente, criando a possibilidade de energia disponível durante todo o ciclo de vida dos dispositivos, recebendo a denominação de tecnologia, “instalar e esquecer”. As grandezas físicas mais comuns empregadas nessa modalidade de alimentação são a luminosa, fluxo eletromagnético e movimentos mecânicos. A desvantagem dessa solução é a necessidade de um dispositivo transdutor capaz de converter a grandeza física em energia elétrica dentro das especificações do equipamento e com um nível de eficiência de conversão viável para a aplicação, dependendo da grandeza escolhida aspectos ambientais também devem ser considerados.

Para que os limitados recursos de alimentação possam ser utilizados de forma efetiva e confiável um sistema de gerenciamento que monitore a geração e consumo do equipamento deve ser previsto. Como os sistemas devem operar de forma autônoma o gerenciamento da energia é um aspecto crucial no projeto e requer uma análise global de operação do equipamento levando em conta os diversos cenários de operação do equipamento. As estratégias adotadas nessa etapa do desenvolvimento demandam a utilização de sistemas que fazem a simulação baseados em modelos precisos para análise e otimização da utilização de energia.

Os aspectos relacionados ao encapsulamento dos sensores e atuadores nos materiais que compõe o sistema requerem atenção especial em função das especificidades e cuidados

necessários a manipulação e operação de dispositivos eletrônicos. Dependendo do ambiente em que o dispositivo irá operar alterações mecânicas poderão causar fadiga e levar ao rompimento de conexões inutilizando ou comprometendo seu funcionamento. Elementos químicos no ambiente que reajam com os materiais que compõe o dispositivo também podem comprometer seriamente a operação dos mesmos.

Por exemplo, um sensor integrado em uma estrutura quimicamente agressiva como cimento ou concreto pode provocar a corrosão, nesse caso um encapsulamento adequado deve ser previsto no projeto.

A fim de evitar a fadiga por stress físico um projeto adequado acompanhado de uma cuidadosa metodologia de testes práticos pode garantir a confiabilidade do sistema.

Sensores como dispositivos que fazem a transdução de uma grandeza física em sinais elétricos devem representar de forma fiel a grandeza que está sendo capturada. Alguns tipos de sensores fabricados dentro de padrões de qualidade têm seus parâmetros de uso atestados pelo fabricante. Nesse caso basta que os projetistas sigam as especificações do fabricante ao desenvolver o projeto. Porém em alguns casos os parâmetros do sensor não são conhecidos e, portanto devem ser aferidos e calibrados. Muitas vezes as condições ambientais com faixa de variação extensa em suas características exigem que o procedimento ocorra de forma dinâmica. As principais características que são corrigidas em um sistema de calibração são *bias*, nível de ruído e *drift*. Essas imperfeições se não forem corrigidas podem gerar medições incorretas e dispersões nos valores medidos.

Aspectos econômicos devem ser considerados quando se planeja uma estrutura massiva de sensores e pontos como redundância, resiliência, a forma pela qual serão conectados entre si e posicionamento se tornam importantes como requisitos do projeto.

Sistemas de dispositivos com atuadores e sensores devem ser capazes estabelecer comunicação com funcionalidade para transportar dados de um ponto a outro da rede. Do ponto de vista físico os sistemas têm que prover comunicação estando inseridos ou próximos a materiais com características de condutividade e permeabilidade eletromagnética dos mais diversos tipos e níveis. Muito tem sido investido na pesquisa e desenvolvimento de sistemas de baixa potência de transmissão e protocolos eficientes. As pesquisas tem se concentrado em resolver questões de consumo, dimensões dos equipamentos e desempenho. A confiabilidade é um fato determinante em aplicações como em dispositivos médicos, veiculares e segurança. Sistemas podem explorar as particularidades das características físicas das ondas de rádio para prover de forma precisa serviços de localização. Os projetos devem ser executados com uma visão sistêmica como forma de garantir a interoperabilidade dos sistemas nas várias camadas.

Sistemas de rádio na faixa de 100 MHz a 2.4 GHz têm sido amplamente pesquisados e aplicados nas comunicações sem fio, não podem ser esquecidas as formas de comunicação ótica geralmente usando fibras óticas, ou sem fio por meio da faixa de infravermelho.

Vários estudos e pesquisas (Vasseur, 2010) confirmam que os circuitos de comunicação são os que mais contribuem no consumo de um sistema. Dessa forma o consumo de energia quando a comunicação está ativa é a que tem o maior impacto nas definições do sistema de gerenciamento e seleção da tecnologia usada para prover a alimentação elétrica. De forma geral os nós dos sistemas operam durante 99% do tempo no modo que reduz o consumo para aproximadamente 100 nA, contudo o sistema de energia deve ser capaz de fornecer a corrente de pico do transmissor, normalmente na ordem de dezenas de miliampères por um curto período de tempo.

A necessidade de miniaturização conduz a integração em encapsulamento extremamente denso dos sistemas incluindo os circuitos de comunicação por rádio. Como os circuitos devem ocupar um espaço limitado, propriedades físicas que determinam a dimensão de antenas têm impulsionado a seleção de faixa de frequências altas, mais adequadas a aplicações de baixo

consumo e possibilitando reduzir a dimensão das antenas.

Atualmente uma grande parcela dos sistemas de comunicação opera nas faixas de 800 MHz a 2.4 GHz, faixa de frequência que dispensa o licenciamento e regulamentação na maioria dos países. Cabe salientar que pelo fato de serem livres para o uso são faixas de frequência utilizadas por muitos dispositivos e serviços, portanto, com a inserção de novos dispositivos problemas de congestionamento poderão ocorrer.

As questões de segurança, confiabilidade e robustez devem ser tratadas como ponto vital já nas etapas de validação e certificação dos sistemas. Deve-se garantir que as situações que coloquem em risco a operação do sistema sejam identificadas e os requisitos de segurança sejam de forma antecipada corretamente implementados. Esse processo de analisar e detectar potenciais problemas ocorre durante a certificação normalmente sendo executado por uma comissão de especialistas que em um processo manual estudam um grande conjunto de documentos. O processo de certificação em alguns casos consome mais tempo e recursos financeiros que o desenvolvimento do produto. Ainda que os avanços no desenvolvimento dos dispositivos de *hardware* ocorram em um ciclo rápido, os dispositivos com aplicação em sistemas ubíquos e pervasivos ainda são limitados em termos de funcionalidade e tem que evoluir em vários pontos antes de atender os rígidos requisitos demandados por usuários.

Um dos pontos chave ainda com limitações é a confiabilidade, pois o sistema deve oferecer um serviço que atenda condições como:

- Gerenciamento de energia em nível de objeto.
- Tenha proteção contra intervenções mal intencionadas, propositais causadas por usuários.
- Não possibilite condições que causem danos físicos ao usuário, danos ou prejuízos ao equipamento ou ambiente no qual está inserido.
- Opere como o esperado, sem apresentar erros durante o ciclo de vida planejado.

Os sistemas para *Internet* dos Objetos podem ser acessados de várias formas e muitas vezes por pessoas não autorizadas ou habilitadas. Os sistemas de *hardware* e *software*, incluindo os de acesso à rede devem ser desenvolvidos de forma a garantir um nível de segurança compatível com o planejado para o dispositivo e sistema no qual está inserido.

A complexidade dos sistemas é uma característica inerente já que pode incluir um volume significativo de *software* embarcado que irá controlar sensores, atuadores, gerenciar e efetuar a comunicação com a rede.

O *software* embarcado cumpre funções de coletar os dados e prover computação adicional como filtragem em *software*, correção de erros, correlação dos dados capturados com modelos, entre outros.

Adiciona-se a complexidade o fator de incerteza e restrições de tempo que devem ser efetivamente gerenciados. Considerar as restrições de tempo é importante já que cada dispositivo interage com outros sistemas dinâmicos. A incerteza ocorre pelo fato do ambiente e as interações não poderem ser modelados de forma completa e precisa.

Um novo serviço ou alterações no código em um dos nós pode modificar o comportamento de outros componentes, tornar inoperante alguns ou todo o sistema.

O sistema de *software* para esses sistemas autônomos deve prover robustez suficiente para administrar comportamentos não previstos. Da perspectiva do *software*, mecanismos devem ser providos para assegurar que estas situações possam estar sob controle e não levar o sistema e ambiente a condições danosas.

As técnicas e ferramentas para o desenvolvimento dos sistemas devem permitir a construção de sistemas complexos a partir de componentes de *software* heterogêneos, ser estruturados de forma a encapsular as propriedades funcionais com métodos que garantam a agregabilidade dos

componentes e análises que prevejam as características comportamentais do sistema.

Em adição aos requisitos que garantem a operação dos sistemas embarcados os objetos devem ter capacidade para que possam ser reconfigurados utilizando recursos próprios, já que serão inseridos ou retirados em redes com características *ad hoc*.

Ao ser inserido em uma área contendo uma rede o dispositivo deve ser capaz de entrar e fazer parte da rede de forma transparente dispensando qualquer ação externa por parte do usuário. Habilidades que possibilitam a realização dessa ação pelo dispositivo incluem rastrear o ambiente e detectar objetos na vizinhança de forma a se reconfigurar de acordo com o protocolo de comunicação disponível.

Reconfiguração do dispositivo também pode ser necessária à medida que dispositivos são conectados, funcionalidades são requeridas, falhas são detectadas, assim um dispositivo pode em dado momento assumir identidades diferentes como um roteador, estação centralizadora ou coordenador de informações.

As características de reconfiguração têm impacto direto no consumo de energia já que mais recursos computacionais são requeridos para dar suporte a essa funcionalidade. Assim o desenvolvimento de um dispositivo deve buscar o equilíbrio entre suas capacidades de reconfiguração e flexibilidade enquanto mantém o consumo de energia dentro dos limites estabelecidos para o projeto.

No contexto do que é designado como *Internet* dos Objetos, múltiplas tecnologias estão sendo desenvolvidas e estarão disponíveis para conectar objetos em rede sendo importante investigar a interação e resultantes dessas ações, algumas já conhecidas e tratadas como mobilidade e conectividade, outras novas e mais relacionadas às particularidades desses dispositivos como reconfiguração dinâmica, uso eficiente de energia, abordagens baseadas em camadas que se sobrepõe, entre outras.

As funcionalidades para identificação e localização da posição de dispositivos no espaço físico caracterizam a geolocalização, a questão que se coloca a partir dessas habilidades dos dispositivos é a privacidade e os aspectos de segurança que tais informações podem provocar. Geoprivacidade é o termo que tem sido utilizado para gerenciar o acesso a informações que indicam o passado, presente e futuro da localização de uma pessoa ou de objetos conectados. Os métodos para garantir a geoprivacidade podem ser classificados em três grupos:

- Temporal e espacial, pode ser necessário proteger a privacidade do usuário ou dispositivo quando ele estiver *on-line*, isto é conectado fisicamente ou *off-line* para no caso de acesso posterior ser possível o acesso a registros gravados durante operações *on-line*.
- Definição da política de proteção, as regras de proteção podem aplicadas de forma única a cada usuário, ou a um grupo com alguma similaridade. Os níveis de segurança podem ser ajustados e personalizados de acordo com a necessidade de forma semelhante aos procedimentos executados nas redes de comunicação tradicionais.
- Definição das técnicas, a preservação de dados com informações espaciais e temporais de indivíduos podendo ser realizada a partir de técnicas como criptografia, controle de acesso por meio de sistemas de autenticação, entre outras.

Para que a *Internet* dos Objetos possa ser efetivamente concretizada uma série de serviços precisa ser desenvolvida considerando todo o ciclo que pode compreender, por exemplo, da aquisição de uma grandeza física por um sensor, até a apresentação dessa informação para o usuário humano, ou como a informação que irá ser compartilhada com outras aplicações ou outros objetos físicos. Para oferecer um serviço desse tipo é necessário haver camadas de serviços que atendam demandas como:

- Reconhecimento de objetos e suas capacidades em níveis semânticos diversos, como por exemplo, protocolo de comunicação e serviços disponíveis.

- Interoperabilidade entre protocolos heterogêneos baseados em tecnologias e com aplicações diversas, como telefonia, infra-estrutura, automação residencial e industrial.
- Soluções para garantir a segurança durante o fluxo da informação pelos vários mecanismos que fazem parte do sistema, deve ser garantida a confiabilidade e segurança dos serviços, proteção para dados, mecanismos de autenticação e rastreamento das operações; Operação mútua de aplicações com diferentes características quanto aos requisitos de segurança e confiabilidade, por exemplo, uma aplicação de segurança patrimonial deve ser capaz de operar no mesmo ambiente da aplicação de entretenimento, sempre considerando as limitações de que esses sistemas em grande parte estarão sendo executados em sistemas embarcados com recursos computacionais limitados e com restrições quanto ao consumo de energia.
- Exposição dos serviços a *Web* de forma amigável. O acesso a serviços como acionamento de dispositivos remotos, monitoramento de estado de sensores e atuadores deve ser feita de forma que não envolva por parte do usuário dificuldades ou o aprendizado na utilização de novos aplicativos. O navegador *Web*, hoje plataforma universal para apresentação de informações, tem se mostrado também como uma opção para a interação com dispositivos em rede.

4 *Web 4.0 a Internet do Objetos*

Tecnologias que habilitaram a criação de aplicações ricas para a *Internet* capazes de serem executadas em navegadores padrões para a *Web* e a combinação de computação em nuvem que fornece grandes capacidades computacionais e de armazenamento permitiram a emergência da que foi denominada e reconhecida na mídia e em muitos ambientes acadêmicos como a *Web 2.0* caracterizada pelas redes sociais e conteúdo gerado pelo usuário. Redes como Facebook e MySpace atraem milhões de visitantes por mês com empresas e instituições as utilizando para interação e troca de experiências com usuários e clientes. A *Web 2.0* é suportada por tecnologias que popularizaram serviços como *blogs*, *wikis*, *podcasts*, *tagging* e as redes sociais, todas valorizando a geração e compartilhamento de conteúdo entre os usuários.

A *Web 3.0* é geralmente conhecida como a *Web* semântica (Berners, 2001). Duas tecnologias utilizadas no desenvolvimento da *Web* semântica são o XML (*eXtensible Markup Language*) e o RDF (*Resource Description Framework*). Com XML podem ser criadas tags específicas para cada domínio de aplicação. Arquivos com informações contidas em *tags* são distribuídas pela *Web* possibilitando que *scripts* e programas possam executar algum tipo de processamento de forma flexível podendo gerar resultados bastante elaborados. O XML permite adicionar estrutura aos documentos, mas não informa sobre o significado dessas estruturas. O significado é expresso pelo RDF, um arquivo que também pode ser montado com XML contendo palavras que indicam a relação entre sujeito, verbo e objeto.

Aplicativos com acesso a coleções de dados estruturados podem criar ou inferir regras que são usados para gerar conhecimento. Pesquisadores em áreas como inteligência artificial tem estudado e desenvolvido agentes de *software* que coletando informações em bases de dados estruturados são capazes de criar significado para as informações.

A utilização de representações semânticas com base em informações geradas por objetos possibilita elaborar contextos que permitem conectar pessoas com bases em seus interesses ou atividades. A detecção de padrões anormais em sinais vitais coletados por dispositivos pode alertar um sistema de saúde. A partir das informações geradas por sensores biomédicos, o profissional da saúde, assim como familiares e pessoas próximas podem ser avisados e executar as ações necessárias. Pesquisas em ciências sociais e médicas têm demonstrado que o agregamento social é um importante aliado para prever e indicar o estado de saúde .

A *Web 4.0* é o termo que foi proposto para designar a *Internet* dos Objetos (Valhouli, 2011). Essa nomenclatura baseada em números que expressam versões não representa seqüências sendo que na verdade as três classificações de *Web* estão sendo desenvolvidas de forma paralela e o avanço em cada uma é muitas vezes incorporada em cada modelo e de forma geral com benefícios para todas.

O ponto central da *Web 4.0* ou *Internet* dos Objetos é a incorporação e sinergia com recursos já disponíveis na *Web* pelos diversos serviços *Web 2.0* com os objetos físicos conectados em rede (figura 7).

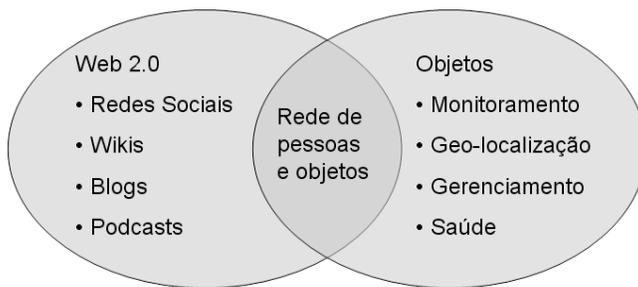


Figura 7: Convergência de serviços *Web*.

Um exemplo de integração é utilizar um mecanismo de busca como o Google para localizar onde se encontra dentro de casa a chave do carro. A associação entre as várias versões e tecnologias da *Web* permitirá transpor para o mundo físico serviços de recomendação como o da Amazon, no qual com base em visitas feitas ao site, são geradas recomendações de produtos em novas visitas. Sistemas de identificação com tecnologia de RFID pode registrar em um supermercado as seções visitadas, os produtos manipulados, com essas informações um sistema de *software* para recomendação pode extrair informações que indiquem a preferência por um produto. Em uma segunda visita atuadores em objetos conectados como sintetizadores de voz, displays para vídeo ou outros dispositivos de interação podem chamar a atenção do usuário para determinado produto, indicando características, condições de pagamento, promoções entre outras informações.

Serviços para armazenamento e compartilhamento de fotos como Google e Flickr oferecem recursos para que o usuário forneça durante o *upload* da imagem a coordenada geográfica do local, obtida por meio de um sistema de GPS. Algumas câmeras fotográficas já trazem embutido módulo de GPS sendo possível que as coordenadas para cada foto sejam registradas de forma autônoma e informadas aos serviços *Web* sem a intervenção do usuário. As informações nesses sistemas são organizadas na forma de metadados, essa característica habilita que outras informações físicas além de posicionamento, incluindo, data, horário, temperatura, pressão atmosférica, luminosidade, entre outras sejam incorporadas para descrever um objeto e o seu estado. Milhares ou milhões desses metadados podem ser agrupados e tratados por aplicações de *Web* semântica criando, por exemplo, um cenário que informe as condições ambientais mais comuns para uma determinada época do ano em um local. Informações desse tipo podem ser utilizadas por usuários no planejamento de uma viagem ou para estudar as diversas interações entre as grandezas ambientais.

5 Combinando as versões da *Web*

Na UFS (Universidade Federal de Sergipe) dentro dos cursos de Ciência da Computação e

Engenharia da Computação tem desenvolvida uma linha de pesquisa que tem o propósito de projetar, implementar e avaliar sistemas que combinam *software* e *hardware* conectados a rede *Internet* constituindo um ambiente no qual aplicações de *Web 2.0*, *Web Semântica* e a *Web 4.0*, também conhecida como a *Internet* dos Objetos são integradas. Participam dessa linha de pesquisa professores/pesquisadores, alunos de graduação e pós-graduação. A organização das atividades se dá por meio de disciplinas optativas e os programas de iniciação científica.

A metodologia de trabalho nessa pesquisa, em linhas gerais, passa por estudar e realizar levantamentos do estado da arte do que está realizado na academia e indústria, estudo das normas e padrões que ainda estão em construção, das tecnologias e técnicas disponíveis para desenvolvimento validação e implementação dos sistemas. Com base nesses levantamentos e estudos são realizadas propostas de projeto que procuram explorar conceitos sempre relacionados à conexão de objetos físicos à rede *Internet* e prover a comunicação entre dispositivos e pessoas, ou entre dispositivos e dispositivos.

As atividades para desenvolvimento dos projetos são extensas no sentido que envolve competências que incluem desenvolvimento de sensores, programação de firmware de sistemas embarcados, sistemas *Web* com interfaces gráficas interativas, algumas vezes apoiadas pela arquitetura de *Webservices*. São adotadas técnicas de particionamento e encapsulamento de funcionalidade em todos os níveis do desenvolvimento a fim de garantir que cada módulo possa ser testado e depurado de forma independente e que o módulo gerado possa também ser reusado em sistemas diversos.

Do ponto de vista de *hardware* são utilizados, sensores, atuadores, microcontroladores e outros dispositivos eletrônicos que permitem o desenvolvimento dos diversos protótipos. Na seleção do *hardware* há preferência por dispositivos de baixo custo e que possam ser adquiridos com facilidade no mercado nacional.

Com essa estratégia de trabalho têm sido realizados trabalhos que em sua diversidade procuram explorar as diversas nuances, e explorar as diversas questões que se colocam e ainda estão em aberto quando se conecta dispositivos físicos à rede *Internet* e incluem aspectos de segurança, interoperabilidade, confiabilidade entre outros.

A seguir são descritos dois trabalhos desenvolvidos que de certa forma são representativos ao expor os desafios e soluções propostas.

5.1 Instante Robô

Os serviços de comunicação instantânea, isto é, aplicações que permitem a pessoas conectadas em rede trocar mensagens em tempo real são extremamente populares e usadas por milhões de pessoas atualmente. Um dos primeiros serviços desenvolvidos foi o ICQ, pela empresa Mirabilis. O conceito básico é uma aplicação com uma janela para enviar mensagens de textos, outra para receber as mensagens e uma lista de contatos. Diversas empresas como Microsoft, Google e Yahoo têm oferecido esse tipo de serviço acrescentado recursos a essas funcionalidades básicas como troca de arquivos, comunicação por voz, gerenciamento de contatos, entre outros.

O que levou a desenvolver o conceito do que chamamos Instante Robô foi a possibilidade de utilizar um comunicador de mensagem instantânea, normalmente utilizado para a comunicação pessoa-pessoa, para realizar a comunicação de pessoa para um dispositivo. Para esse trabalho foi escolhido como dispositivo a ser controlado uma plataforma para robótica móvel (Chella, 2010), conectado a um link de rádio que se comunica com um computador recebendo comandos para controle dos motores de direção e câmera de vídeo, captura de grandezas físicas como temperatura e posição (figura 8). A idéia central da pesquisa foi conectar o robô à rede e permitir que um usuário conectado a um cliente de mensagem instantânea pudesse controlar os movimentos do robô e receber dados provenientes dos sensores.

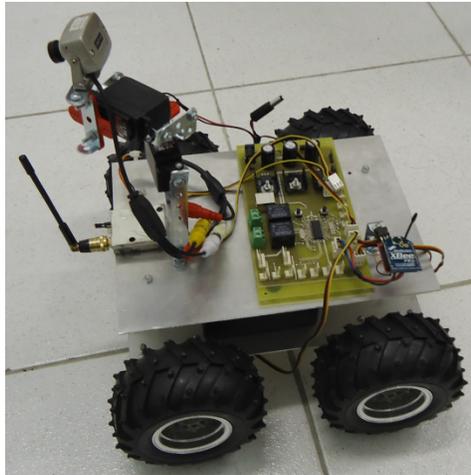


Figura 8: Plataforma Robótica Móvel (Chella, 2010).

Os estudos iniciais para o desenvolvimento da aplicação indicaram a abordagem baseada na arquitetura cliente-servidor (fig 9) como viável. O servidor é o módulo responsável por realizar a comunicação com o cliente do comunicador, tratar as mensagens de forma que possam ser enviadas ao robô e gerenciar o acesso dos usuários.

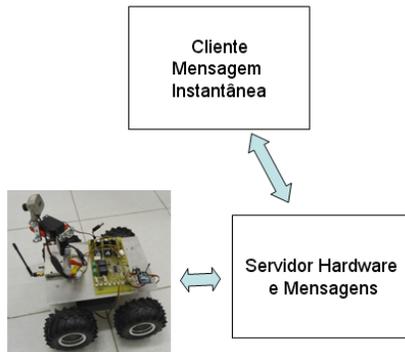


Figura 9: Diagrama da arquitetura do sistema (Chella, 2010).

Quanto ao cliente existia a opção de desenvolver um cliente específico ou utilizar um dos serviços ou aplicações disponíveis no mercado. A opção foi por utilizar um dos sistemas disponíveis por duas razões principais, a primeira permitiria avaliar a facilidade de integração com serviços produzidos por terceiro, a segunda é o fato que a utilização de um serviço já existente dispensaria ao usuário a instalação e configuração de mais um aplicativo em seu computador. Com essa opção selecionada o passo seguinte foi a escolha do cliente a ser utilizado. Como o servidor se comunica com o cliente para troca de informações é importante que o protocolo de comunicação seja aberto e documentado, dessa forma aplicações e serviços com protocolo fechado foram descartados como candidatos nesse trabalho. Esse requisito levou a pesquisa desses protocolos abertos, sendo que o XMPP (2011) se mostrou adequado em função de características como:

- Uso livre, O núcleo do sistema é desenvolvido pela comunidade *software* em processo rigoroso de revisão com atestado da IETF (*Engineering Task Force*), atualizações são

publicadas periodicamente com código e documentação de acordo com as licenças mais liberais de *software* livre.

- Extensível, núcleo é baseado em XML, padrão para interoperabilidade de dados, permite o desenvolvimento dos mais diversos tipos de aplicação como comunicação instantânea, jogos em rede, comunicação máquina- máquina, voz sobre IP, entre outras.
- Escalável, o modelo de transferência de informação do tipo “*push*” resolve questões de escalabilidade tradicionais com o modelo http.

A partir da definição do protocolo de comunicação foi realizada a pesquisa para selecionar a ferramenta de desenvolvimento para implementação da aplicação. Os requisitos para a ferramenta foram a disponibilidade para acesso a comunicação com *hardware* e API funcional. Dentre as opções foram avaliados a plataforma .NET, Java e Python. Todas se mostraram equivalentes, a escolha ficou com o .NET por atender melhor o compromisso com a comunicação com *hardware* e facilidade no desenvolvimento de interfaces gráficas.

Para a implementação foram levantados alguns requisitos mínimos para operação e os mesmos em seguida foram implementados, são eles:

- Gerenciamento de conexões e usuários, uma tabela é referência para indicar quais usuários estão autorizados a acessar o sistema.
- Vocabulário e decodificação, os usuários utilizam um cliente de mensagens para efetuar o controle do robô, para facilitar essa interação foi definido um vocabulário de palavras que indicam as ações que serão executadas, por exemplo, “ler sensor 1” faz a captura dos dados do respectivo sensor. A palavra “ajuda” envia ao cliente uma pequena lista com os comandos disponíveis, dispensando a memorização de todo o vocabulário
- Comunicação com o *hardware*, após receber os comandos do usuário as mensagens são traduzidas para o protocolo de comunicação do robô, e enviadas para o driver de dispositivo que trata a comunicação com o *hardware*, no caso uma porta do barramento USB;
- Tratamento de exceções, com o propósito de garantir a continuidade da operação em situações como queda da rede, falha na comunicação com a rede, falha nos serviços, foram desenvolvidas rotinas que capturam essas situações e fazem operações na tentativa de recuperar o funcionamento normal e não sendo possível informar o usuário.

A aplicação servidor é constituída pelas funcionalidades descritas e a interface gráfica fornece informações sobre a operação, mensagens de erro e visualização dos acessos (fig 10).

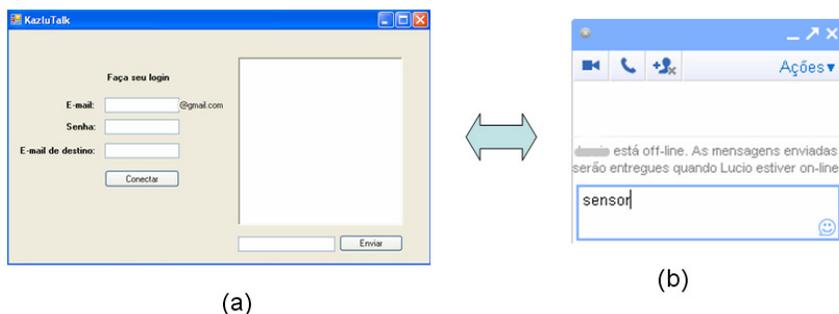


Figura 10: Interface gráfica do servidor (a) e Cliente web (b) do Instante Robô.

O protótipo desenvolvido e os testes realizados mostraram a viabilidade de utilizar um serviço de mensagem instantânea baseado em protocolos abertos para controlar um dispositivo de

hardware. O controle remoto de um robô móvel e o monitoramento do estado de seus sensores permite a criação de aplicações para telepresença onde o usuário se projeta em um espaço com capacidades para controlar e obter dados nesse ambiente, potenciais usos desses recursos são em ambientes hostis para o ser humano ou de alguma forma inacessíveis. Outra possibilidade é o compartilhamento de dispositivos que muitas vezes por terem um custo financeiro alto não estão facilmente acessíveis. Pode-se observar que as características de confiabilidade e robustez em grande parte estão relacionadas às redes de comunicação. Um trabalho adicional para prever e tratar exceções de *hardware* e *software* pode aumentar a confiabilidade do sistema de forma significativa.

5.2 Real Tweet

Serviços para troca de mensagens curtas, como o *Twitter*, tem se tornado um serviço muito popular na *Internet* com milhões de usuários cadastrados, sendo que alguns usuários chegam a ter milhares de seguidores que acompanham as mensagens. A característica de concisão em razão do limite de 140 caracteres para cada mensagem pode ser particularmente oportuna para comunicação de mensagens provenientes de dispositivos físicos como sensores, ou indicadores diversos de estado. A possibilidade de envio contínuo dos dados pode fornecer um registro de eventos que estejam acontecendo, por exemplo, em um ambiente. A comunicação pode ocorrer nos dois sentidos, dessa forma uma mensagem pode ser recebida por outro dispositivo criando a oportunidade para que dispositivos diversos possam trocar dados entre si, alguns monitorando os estados disponíveis e dependendo da configuração enviando comando para atuadores que podem estar dispersos geograficamente em qualquer lugar onde exista uma conexão a rede.

Para avaliar esse conceito foi desenvolvido um projeto que conecta à rede uma interface de *hardware* com um sensor de temperatura (fig 11). Também baseado na arquitetura cliente servidor, é a aplicação servidor que captura os dados provenientes do sensor, converte o mesmo para a escala de graus Celsius e cria uma mensagem de texto que é postada no *Twitter*.

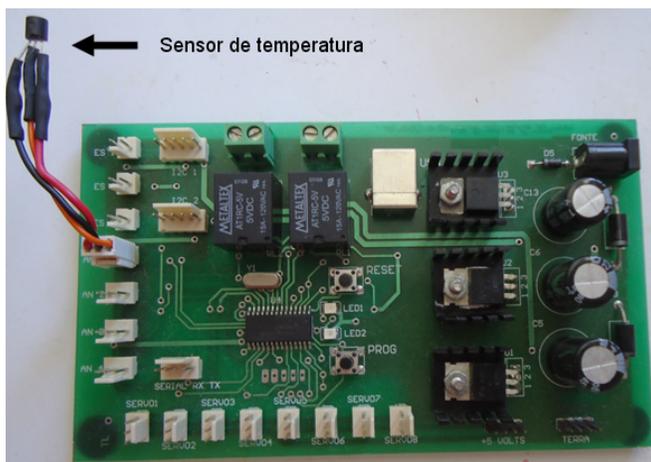


Figura 11: Interface de *hardware* com sensor de temperatura.

A aplicação servidor foi desenvolvida na plataforma .NET utilizando a API *Yedda* um *wrapper* para o C# para a API do *Twitter*. A aplicação captura os dados do sensor e gera a mensagem. Para cada acesso a aplicação realiza o acesso com informações como usuário e senha. Uma interface gráfica indica as mensagens de estado do sistema e permite executar as operações de conexão,

ativação e recepção e envio das mensagens (figura 12).

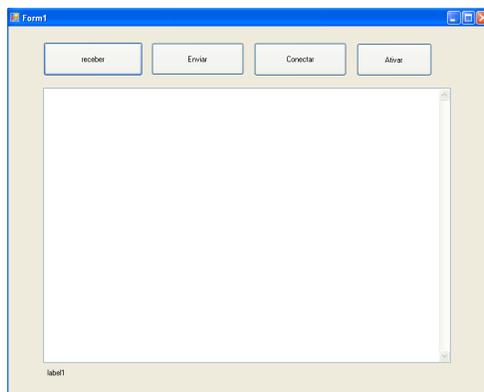


Figura 12: Interface gráfica da aplicação Real Tweet.

O trabalho realizado mostrou a viabilidade de usar serviços como o *Twitter* para o envio de mensagens em tempo próximo ao real com informações provenientes de dispositivos físicos. Esses recursos permitem planejar serviços em áreas como telemedicina em que sensores biomédicos podem enviar para uma central informações acessíveis a médicos e demais profissionais auxiliando no acompanhamento de tratamento. A prevenção de riscos e desastres ambientais pode ser uma área de aplicação, onde uma rede de sensores pode enviar mensagens aos usuários quando uma situação de perigo ocorrer.

As pesquisas desenvolvidas nos projetos demonstrados têm propiciado material e experiência real que nos permitem avaliar, propor soluções e inserir técnicas e resultados acadêmicos ao que alguns meios têm chamado de *Internet* dos Objetos ou a *Internet* do futuro, na qual milhares de objetos físicos com capacidade de comunicação possam operar e cooperar trazendo benefícios a sociedade de forma segura e confiável.

6 Conclusão

A *Internet* dos Objetos não é apenas uma evolução da *Internet* das pessoas, onde se conectam dispositivos físicos, mas também envolve novos modos de pensar e projetar tendo o foco nas particularidades desses dispositivos.

Nesse caso, tópicos relacionados a conectividade, transferência eficiente de dados, facilidade e acesso aos serviços serão requisitos essenciais. As limitações não estão relacionadas apenas às restrições dos recursos dos circuitos que compõem o objeto, mas também relacionadas à heterogeneidade, escalabilidade em termos de números de objetos conectados e segurança.

Existem questões em aberto que justificam pesquisas como se objetos devem ser integrados em um esquema de convergência total para IP com implementações reduzidas da pilha e protocolos IP, ou devem ser conectados em sub-redes com técnicas e protocolos de comunicação diferentes, com *gateways* específicos interconectando as sub-redes às redes baseadas em IP.

Rede de comunicação com topologia ponto a ponto, comunidades suportadas por redes sociais, dispositivos conectados em rede que fazem a computação pervasiva e ubíqua estão sendo conectados para criar novos modos de colaboração e tomada de decisão. Pessoas, informações e dispositivos conectados habilitam a convergência entre o mundo virtual e físico. A *internet* que está a caminho irá integrar conhecimento, pessoas e objetos.

A pesquisa desenvolvida no DCOMP (Departamento de Computação) da UFS sobre *Internet*

dos Objetos atuará em temas relevantes para essa área, como definição e implementação da arquitetura de *hardware* e *software*, integração com API's de terceiros e desenvolvimento de serviços *web* específicos. Aspectos de segurança, confiabilidade e robustez serão alvo de estudos, tendo em vista a criação de uma infra estrutura genérica que permita o desenvolvimento dos mais diversos tipos de aplicação.

Referências Bibliográficas

- ASHTON, K. That 'Internet of Things' Thing. USA: RFID Journal. <http://www.rfidjournal.com/article/view/4986>. Acesso em 02/2011.
- BERNERS, L. *et al.* The Semantic Web. USA: Scientific American Magazine, 2001.
- BODYTRACE. BodyTrace eScale. <http://www.bodytrace.com/>. Acesso em 02/2011.
- BROCK, D. L. The Electronic Product Code. USA: Journal IEEE Micro Volume 21 Issue 6, 2001.
- CHELLA, M. Robotic tool with scratch language. 4^o Workshop in Applied Robotics and Collaborative Systems Engineering, Bauru, SP, 2010.
- FIELDING, R. T. Architectural styles and the *design* of network-based *software* architectures. Doctoral Thesis, 2000.
- FLUXNET. Fluxnet LaThuile. <http://www.fluxdata.org>. Acesso em 02/2011.
- GARDNER, G. Gartner's Top 10 Predictions for 2009. <http://www.gartner.com/it/page.jsp?id=876512>. Acesso em 02/2011.
- GLOWCAP. Intelligent Devices That Care. <http://www.vitality.net/index.html>. Acesso em 02/2011.
- GRAY, J. The Fourth Paradigm: Data-Intensive Scientific Discovery. First Edition. USA: Microsoft Research, 2009.
- IPSO. IPSO Alliance. <http://www.ipv6forum.com/index.php>. Acesso em 02/2011.
- IREADY. uWebserver. <http://www.mycal.net/old/wsweb/design/descript.html>. Acesso em 02/2011.
- JARVIS, J. What Would Google Do?. First Edition. USA: HarperBusiness, 2009.
- TRAPSTER. Speed trap share system. <http://www.trapster.com>. Acesso em 02/2011.
- KUNIAVSKY, M. Smart Things: Ubiquitous Computing User Experience *Design*. First Edition. USA: Morgan Kaufmann, 2010.
- MERCER, S. The Fourth Paradigm: Data-Intensive Scientific Discovery. First Edition. USA: Microsoft Research, 2009.
- NELSON, M. The Cloud, the Crowd, and the 3D *Internet*: What's Next for Collaboration Online. <http://www.ndu.edu/iCollege/fcvw/images/2009/Apr24slides/nelson.pdf>. Acesso em 02/2011.
- Nike+iPhone. Apple - Run or workout with Nike + iPod. Acesso em 02/2011.
- PATCHUBE. Data infrastructure for the *Internet* of Things. <http://www.pachube.com/>. Acesso em 02/2011.
- PRESSER, M. *et al.* Publishing Linked Sensor Data. The 3rd International workshop on Semantic Sensor Networks, 2010
- QUENTIN, S. P. The Trojan Room Coffee. <http://www.cl.cam.ac.uk/coffee/qsf/coffee.html>. Acesso em 02/2011.
- STELING, B. Shaping Things. First Edition. USA: MIT Press, 2005.
- VALHOULI, C. A. Clicks & Mortar: *Web* 4.0, The *Internet* of Things. <http://www.thehammersmithgroup.com/images/reports/web4.pdf>. Acesso em 02/2011.
- VASSEUR, J. *et al.* Interconnecting Smart Objects with IP: The Next *Internet*. First edition. USA: Morgan Kaufmann, 2010.

XMPP. Extensible Messaging and Presence Protocol. www.xmpp.org. Acesso em 02/2011.

Pesquisa em Serviços de Transações para Computação Móvel e Middlewares Abertos

Tarcísio da Rocha

Resumo. Técnicas de processamento de transações têm sido de grande importância no que diz respeito à preservação da correção de dados em diversas áreas da computação. Contudo, desenvolver técnicas de apoio a transações para ambientes dinâmicos pode ser uma tarefa complexa pelos seguintes fatores: o próprio dinamismo dos ambientes móveis e a grande variedade dos requisitos transacionais exigidos pelas aplicações atuais. Nesse contexto, este capítulo revisa os principais trabalhos apresentados na literatura que fornecem bases para a definição de serviços de transações adequados às características dos ambientes móveis. Além disso, este capítulo apresenta o conceito de *middleware* aberto – uma nova geração de *middlewares* capazes de trazer a serviços como o de transação a capacidade de lidar com os obstáculos desses ambientes.

Palavras-chave: Serviços de Transações. Computação Móvel. Middleware

1 Introdução

Transações se tornaram uma técnica largamente usada para garantir a consistência de dados, a confiabilidade e a recuperabilidade de aplicações distribuídas. Devido a funções como, garantir a consistência de dados, a recuperação de falhas e o controle de concorrência, transações são consideradas blocos de construção apropriados para a estruturação de sistemas confiáveis (Rocha, 2008).

Dentre as áreas que mais têm influenciado o estudo de novas técnicas de processamento de transações, a computação móvel merece uma posição de destaque. Apesar da sua crescente popularidade por trazer a computadores portáteis a capacidade de se tornarem onipresentes, a computação móvel trouxe consigo uma série de novos desafios ao processamento de transações (Venkatraman, 2005). Esses desafios são causados principalmente pelo seu alto dinamismo que é refletido na prática por: desconexões inesperadas, erros de transmissão, variações na largura de banda e escassez de recursos nos dispositivos de computação portáteis. Diante desse dinamismo, transações também agregam o importante papel de garantir que as variações ocorridas não comprometam a confiabilidade das aplicações.

Contudo, desenvolver técnicas de apoio a transações para ambientes dinâmicos é uma tarefa complexa. O primeiro obstáculo está no próprio dinamismo dos ambientes – a disponibilidade de recursos pode variar inesperadamente. Isso pode causar dois efeitos diretos no processamento de transações: altas taxas de cancelamento de transações e grandes atrasos na execução das tarefas transacionais. O segundo obstáculo está na atual flexibilização do conceito de transações. Isso ocorre porque os requisitos transacionais exigidos pelas aplicações atuais estão tão variados que extrapolam as propriedades tradicionalmente definidas para uma transação (atomicidade, consistência, isolamento e durabilidade).

A complexidade em se projetar serviços de transações para esses ambientes não está somente em encontrar formas de superar esses obstáculos, mas principalmente no fato de que essas formas podem ser dependentes das aplicações que utilizam os serviços. Por exemplo, considerando o caso das desconexões, o interessado em determinar quais serão as medidas adotadas em caso de uma interrupção abrupta causada por uma desconexão inesperada é o desenvolvedor da aplicação e não o desenvolvedor do serviço de transação por si só. O problema é que as medidas para lidar com o dinamismo e as propriedades transacionais requeridas pelas aplicações podem ser tão diferentes que é uma tarefa muito complexa implementar um serviço de transação que consiga contemplar todas as possíveis alternativas exigidas pelas aplicações.

Este capítulo apresenta uma revisão dos trabalhos apresentados na literatura que fornecem uma base para o estudo de técnicas adequadas ao processamento de transações em ambientes de computação móvel. Além disso, este capítulo apresenta o conceito de *middleware* aberto – uma nova geração de *middlewares* capazes de trazer a serviços como o de transação a capacidade a lidar com os obstáculos desses ambientes.

Este capítulo está organizado da seguinte maneira. A seção 2 descreve os conceitos básicos relacionados com a computação móvel e com a definição de transação. A seção 3 apresenta os principais padrões tradicionais que definem transações distribuídas. A seção 4 apresenta os principais modelos de transações para ambientes de computação móvel destacando suas limitações e contribuições. A seção 5 discute os principais *frameworks* para definição de transações. A seção 6 revisa alguns dos principais *middlewares* abertos da literatura. A seção 7 apresenta um modelo de componentes aberto que pode ser usado no desenvolvimento de *middlewares* altamente configuráveis. Por fim, como considerações finais, a seção 8 apresenta os trabalhos relacionados que já foram realizados no Departamento de Computação da Universidade Federal de Sergipe assim como os os possíveis trabalhos futuros que podem vir a ser realizados.

2 Conceito Básico de Transações

Transação é um conceito importante para a simplificação da construção de aplicações confiáveis, especialmente aquelas que requerem acesso concorrente a dados compartilhados. O conceito de transações foi desenvolvido primeiramente junto a aplicações comerciais como um mecanismo de proteção aos dados em bases centralizadas (Gray e Reuter, 1992). Posteriormente, o conceito de transações foi estendido para o mais amplo contexto da computação distribuída. Atualmente, há uma grande aceitação de transações como um elemento chave para a construção de aplicações confiáveis em novas áreas como, serviços *web*, comércio eletrônico e computação móvel.

2.1 Propriedades Tradicionais de Transações

Tradicionalmente, uma transação pode ser definida como um conjunto de operações que é executado honrando as propriedades ACID (Gray, 1992). ACID é um acrônimo para Atomicidade, Consistência, Isolamento e Durabilidade. Segue a descrição de cada uma destas propriedades:

- **Atomicidade.** O conjunto de operações de uma transação deverá constituir uma unidade indivisível, ou seja, ou todas as operações são executadas ou nenhuma das operações é executada. Isto implica que, se a transação for interrompida por uma falha, todos os seus efeitos sobre os itens manipulados têm que ser desfeitos.
- **Consistência.** Uma transação deve manter as regras de integridade dos dados que ela acessa.
- **Isolamento.** Os resultados de uma transação não deverão sofrer interferências de outras transações concorrentes. Se duas ou mais transações estão executando concorrentemente e de forma isolada, o resultado final obtido será o mesmo se elas fossem executadas uma após a outra em alguma ordem.
- **Durabilidade.** Assim que uma transação for efetivada, seus resultados se tornam permanentes, ou seja, nenhuma falha depois da efetivação poderá desfazer esses resultados.

Devido ao seu conceito simples, rigoroso e eficaz, as propriedades ACID são muito usadas em operações de sistemas de banco de dados. Modelos de transação que seguem estas propriedades com rigor são adequados para aplicações que fazem transformações de estado simples, como realizar uma transferência de valores entre duas contas bancárias. Entretanto, a aplicação rígida das propriedades ACID impõe limitações que tornam o modelo inadequado para outros tipos de aplicações.

2.2 Relaxamento das Propriedades Transacionais

Vários outros modelos de transação que relaxam estas propriedades foram propostos na literatura. Esses modelos têm por objetivo melhor atender a requisitos de aplicações avançadas. Por exemplo, transações aninhadas (Moss, 1985) relaxam a propriedade da atomicidade. Uma contribuição das transações aninhadas é evitar a necessidade de se desfazer a transação por completo se algo der errado durante a sua execução. Para isso, uma transação é dividida em subtransações e cada subtransação pode ser desfeita sem que a transação seja desfeita por completo.

Um outro exemplo do relaxamento das propriedades ACID é o uso de níveis de isolamento (Gray, 1992). Para a definição de níveis de isolamento, graus de interferência entre transações foram definidos e vão desde um isolamento total (sem interferências) até a ausência de isolamento (permite todo tipo de interferência). O uso de níveis de isolamento é muito útil para melhorar o desempenho de aplicações que não requerem uma garantia total de isolamento. Isto se deve ao fato de que as restrições impostas para se garantir o isolamento total de transações podem acarretar grandes perdas de desempenho (Adya, 1999).

Na literatura também há exemplos de modelos de transações que relaxam as propriedades da consistência e da durabilidade (Lee e Helal, 2002). Uma forma de se relaxar a propriedade da consistência é permitir que as aplicações definam regras de consistência personalizadas para o acesso a dados. Uma forma de relaxar a propriedade de durabilidade é tolerar o descarte de parte de atualizações de dados no processo de efetivação de transações.

2.3 Discussão

Diante da variedade dos ambientes em que transações são executadas e da variedade dos requisitos transacionais das aplicações destinadas a esses ambientes, pode-se dizer que, ao implementar um serviço de transações, poderia-se adotar uma das duas estratégias:

1. o serviço seria implementado levando em conta somente os mecanismos necessários para

- atender aos requisitos comuns de aplicações de alguns domínios correlatos;
- o serviço seria implementado com um grande número de mecanismos capazes de atender aos requisitos de aplicações do maior número possível de domínios.

Cada uma dessas duas estratégias possui suas desvantagens. Uma das desvantagens da primeira estratégia é que o serviço poderá se tornar muito específico e incapaz de atender aos requisitos de diferentes tipos de aplicações. Uma das desvantagens da segunda estratégia é que nenhuma aplicação necessitará de todos mecanismos disponibilizados pelo serviço, mas, mesmo assim, pagará por eles (ex.: grande porte, custo monetário alto, baixo desempenho).

Uma resposta para esse problema pode estar na possibilidade de personalização dos serviços de transações. Dessa forma, um serviço poderia ser composto somente com aqueles mecanismos necessários às aplicações interessadas. Nas seções a seguir são apresentados alguns conceitos que podem fornecer as bases para a definição de serviços de transação personalizáveis: reflexão computacional, componentes de *software* e *framework* de componentes.

3 Padrões Tradicionais de Transações Distribuídas

Padrões de processamento de transações distribuídas surgiram como consequência dos esforços mantidos por grandes representantes da indústria de *software* em aumentar a confiabilidade de aplicações em domínios como, financeiro, bancário e de comércio eletrônico. Entre esses padrões estão *X/Open Distributed Transaction Processing*, *Java Transaction Service*, *OMG Object Transaction Server* e *Microsoft Transaction Server* que, na prática, visam manter a integridade de dados acessados por aplicações concorrentes em ambientes distribuídos.

Nas seções a seguir, três dos principais padrões são apresentados ressaltando suas arquiteturas e funcionamento. Na primeira seção, os conceitos básicos do processamento de transações distribuídas e o padrão X/Open DTP são discutidos. A segunda seção apresenta o padrão Object Transaction Service. A terceira seção apresenta o padrão Java Transaction Service. Na última seção, são resumidos os padrões descritos fazendo um paralelo entre os componentes envolvidos e discutindo o apoio a ambientes móveis.

3.1 X/Open Distributed Transaction Processing

O X/Open DTP (*Distributed Transaction Processing*) (X/OPEN COMPANY, 1996) é uma arquitetura proposta pelo *Open Group*¹ que possibilita o compartilhamento de recursos entre diversas aplicações através da coordenação de transações globais. O X/Open DTP tem sido adotado como padrão por grandes representantes da indústria de *software* para o desenvolvimento de sistemas de banco de dados e de processamento de transações.

A Figura 1 mostra uma arquitetura simplificada do X/Open DTP. Essa arquitetura é representada por três principais componentes:

- Resource Managers (RM)*. Representam bancos de dados ou sistemas gerenciadores de arquivos que proveem acesso aos recursos compartilhados.
- Transaction Manager (TM)*. É responsável por gerenciar transações globais, associar identificadores a transações, monitorar seu progresso e coordenar os processos de efetivação e recuperação.
- Application Programs (AP)*. Representam aplicações que implementam as funcionalidades desejadas do usuário. Cada aplicação especifica a sequência de operações que envolvem recursos compartilhados junto aos RM's. Uma aplicação define o início e o fim de transações globais, acessa recursos dentro do escopo de transações e normalmente toma

1: Um consórcio entre empresas de desenvolvimento de *software*.

decisões como, efetivar ou abortar transações.

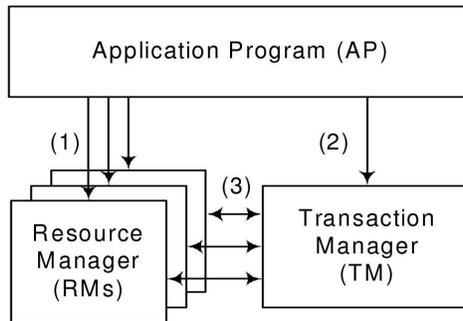


Figura 1: Arquitetura do X/Open DTP (X/Open Company, 1996).

A interação entre os elementos que compõem o padrão X/Open é feita através de um conjunto de interfaces bem definidas. Dentre elas estão as interfaces TX (X/Open Company, 1995), XA e AX (X/Open Company, 1991).

- Interface TX – define as operações do TM que são chamadas por aplicações. Algumas das principais operações:
 - tx_begin – chamada pela aplicação para iniciar uma nova transação.
 - tx_commit – chamada pela aplicação para efetivar a transação.
 - tx_rollback – chamada pela aplicação para abortar a transação.
- Interface XA – define as operações dos RM's que são chamadas pelo TM. Algumas das principais operações:
 - tx_commit – chamada pela aplicação para efetivar a transação.
 - tx_rollback – chamada pela aplicação para abortar a transação.
 - xa_prepare – chamada pelo TM na primeira fase do protocolo de efetivação² para pedir um voto ao RM. Nesse voto o RM indica ao TM se a transação deve ser efetivada ou cancelada.
 - xa_commit – chamada pelo TM na segunda fase do protocolo de efetivação para efetivar as operações transacionais junto ao RM.
- Interface AX – define as operações do TM chamadas pelo RM. Algumas das principais operações:
 - ax_reg – usada pelo RM para se registrar junto ao TM.
 - ax_unreg – remove o registro do RM junto ao TM.

Em geral, o apoio transacional provido pelo padrão X/Open DTP funciona da seguinte maneira.

1. A aplicação chama a operação tx_begin do TM para iniciar uma transação.
2. A aplicação executa as suas operações sobre os recursos compartilhados junto a um ou mais RM's.
3. Ao receber operações sobre os recursos, cada RM requisitado se registra junto ao TM chamando a operação ax_reg do TM.
4. A aplicação chama a operação tx_commit ou tx_rollback do TM para finalizar a transação.
5. Ao receber a chamada da operação tx_commit, o TM inicia o protocolo de efetivação. Na primeira fase do protocolo, o TM chama a operação xa_prepare em cada RM registrado. Se todos os RM's responderem positivamente, o TM inicia a segunda fase do protocolo.

Na segunda fase, o TM chama a operação `xa_commit` para cada RM registrado. Depois de executada a segunda fase, a transação é considerada efetivada. No caso de término com `tx_rollback`, a transação é abortada.

3.2 Object Transaction Service

O Object Transaction Service (OTS) (Object Management Group, 2003) é um serviço de processamento de transações distribuídas baseado no X/Open DTP que foi especificado pela OMG (Object Management Group). O OTS define um conjunto de interfaces que proveem um serviço de processamento de transações para aplicações que acessam objetos CORBA.

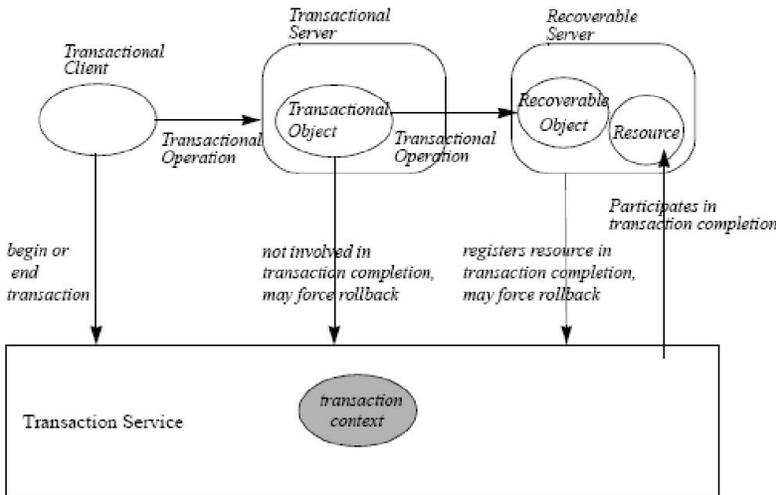


Figura 2: Arquitetura do OTS
Fonte: Object Management Group (2003).

A arquitetura do OTS é apresentada na Figura 2. Uma breve descrição dos seus elementos é apresentada a seguir:

1. *Transactional Client*. Representa as aplicações que utilizam o serviço de transações provido pelo OTS para acessar objetos transacionais.
2. *Transactional Object*. Representa os objetos transacionais, ou seja, objetos que estão aptos a serem controlados pelo serviço de transação quando acessados dentro do escopo de uma transação.
3. *Recoverable Object*. Representa objetos transacionais que possuem um estado persistente que deve ser mantido pelo serviço de transações.
4. *Resource Objects*. Representa objetos que são registrados pelo serviço de transações para que possam fazer parte dos protocolos de efetivação e de recuperação.
5. *Transactional Server*. Representa uma coleção de um ou mais objetos transacionais.
6. *Recoverable Server*. Representa uma coleção de objetos em que pelo menos um deles é um objeto recuperável. o *Recoverable Server* participa dos protocolos do serviço de transações registrando um ou mais *Resource Objects*.
7. *Transaction Service*. Representa o gerenciador de transações que provê operações transacionais (ex: *begin*, *end*) para aplicações transacionais (*Transactional Client*) e gerencia a execução de protocolos de efetivação e de recuperação junto aos participantes da transação (*Resource*).

3.3 Java Transaction Service

O Java Transaction Service (JTS) (Sun Microsystems, 1999) especifica a implementação de um gerenciador de transações capaz de prover um serviço de transações a aplicações através de uma API chamada JTA (Sun Microsystems, 1999). Visando uma maior interoperabilidade e portabilidade, o JTS foi especificado com base nas interface definidas pelo OTS.

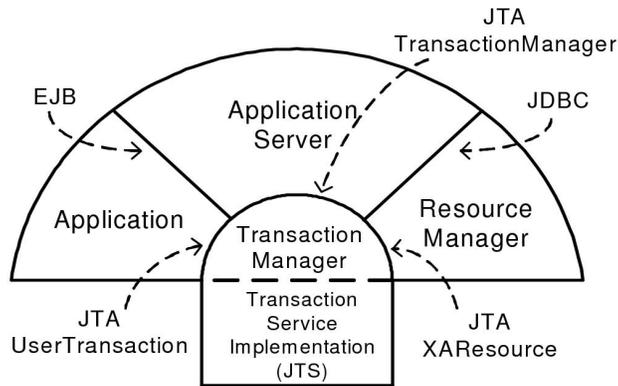


Figura 3: Arquitetura do JTS (Sun Microsystems, 1999).

A arquitetura do JTS é apresentada na Figura 3. Uma breve descrição dos seus elementos é apresentada a seguir.

1. *Transaction Manager*. Representa o gerenciador de transações que provê operações transacionais de demarcação a aplicações, gerenciamento de recursos, sincronização e propagação do contexto de transações.
2. *Application Server*. Representa o servidor de aplicação que provê a infraestrutura necessária para apoiar um ambiente de execução de aplicações. Um exemplo de servidor de aplicação é um servidor EJB.
3. *Resource Manager*. Representa o gerenciador de recursos – gerencia os recursos acessados por aplicações durante a execução de transações. Participa dos protocolos de efetivação e de recuperação de transações.
4. *Application*. Representa as aplicações que utilizam o serviço de transações para acessar os recursos gerenciados pelo gerenciador de recursos.

3.4 Discussão

Dada a grande importância dos serviços de processamento de transações na implementação de aplicações que acessam dados distribuídos, várias plataformas de processamento de transação diferentes foram propostas. Para tentar manter uma interoperabilidade entre as mesmas é que os padrões descritos acima foram especificados. Assim, as plataformas de processamento de transação desenvolvidas em um mesmo padrão poderiam interoperar.

Apesar da vantagem da interoperabilidade, esses padrões foram especificados com base nos requisitos de aplicações tradicionais. Esses requisitos são baseados nas propriedades ACID tradicionais. Dessa forma, esses padrões não são aptos a atender os requisitos de aplicações de novos domínios como, a computação móvel, aplicações multimídia e de tempo real. Esses domínios possuem requisitos como: flexibilização das propriedades ACID, configurabilidade e extensibilidade. Dados os requisitos dos novos domínios de aplicação, diversos modelos de transação especializados foram propostos. Na seção a seguir, são apresentados alguns desses

modelos especializados que consideram os requisitos de ambientes de computação móvel.

4 Modelos de Transação para Computação Móvel

Nesta seção é apresentada uma breve descrição dos principais modelos de transação projetados para o ambiente de computação móvel. Uma descrição mais detalhada desses e outros modelos pode ser encontrada em (Rocha e Toledo, 2005).

4.1 Coda-IOT

O Coda (Satyanarayanan, 2002) é um dos trabalhos pioneiros e relevantes na área da computação móvel. Apesar de não ter sido esse um dos seus objetivos iniciais, o Coda é um sistema que provê acesso a dados a partir de máquinas móveis. Para isto, esse sistema utiliza técnicas como operação desconectada e operação fracamente conectada.

Os primeiros enfoques do Coda foram voltados ao tratamento de falhas de partição que afetam os usuários de sistemas distribuídos. Para contornar esses problemas foram propostas basicamente duas técnicas: a *replicação de servidor* e a *operação desconectada*. A *replicação de servidor* é uma técnica fundamental para a garantia da disponibilidade de dados durante falhas que ocasionam partição da rede. O Coda utiliza um conjunto de servidores que contém réplicas dos dados úteis aos usuários da rede. Assim, quando ocorre a partição da rede, um usuário poderá acessar os dados de um outro servidor na mesma partição em que ele se encontra.

A *operação desconectada* é uma técnica usada principalmente para garantir a disponibilidade em circunstâncias de falhas em que a replicação de servidor não resolve. Um exemplo disso seria a perda de comunicação com todos os servidores da rede. A ideia central da operação desconectada é manter na cache da máquina do usuário uma cópia dos dados úteis ao mesmo. Assim, com a desconexão, o usuário poderia continuar as suas operações sobre a cópia em cache.

Estudos posteriores deram ao Coda a capacidade de operar em ambientes com comunicação sem fio, caracterizados pela largura de banda baixa e intermitente. Para isto, foi utilizada uma técnica chamada operação fracamente conectada (Mummert, 1995). Primeiro, o protocolo de manutenção dos dados em cache foi modificado para permitir uma rápida revalidação de grandes volumes de dados em cache depois de um período de desconexão. Além disso, foi desenvolvido um mecanismo chamado reintegração por partes que utiliza, de forma flexível, a largura de banda da rede na propagação da atualização dos dados em cache para os servidores.

Os modelos iniciais de replicação de dados no Coda só visavam a resolução de conflitos do tipo escrita/escrita. Os conflitos do tipo leitura/escrita eram ignorados visto que a possibilidade de sua ocorrência era tolerada pelos usuários de sistemas de arquivos compartilhados. Porém, com a operação desconectada a ocorrência desses conflitos aumentou de forma considerável.

Para lidar com esse problema, o Coda foi estendido com um mecanismo chamado Isolation-Only Transaction (IOT) (Lu, 1994). Esse mecanismo permite ao sistema admitir somente os conflitos leitura/escrita que satisfaçam certos critérios de serialização. Assim, o IOT busca dar uma melhor garantia de consistência a aplicações que executam no ambiente móvel.

4.2 PRO-MOTION

O PRO-MOTION (Walborn, 1999) é um sistema de processamento de transações projetado para lidar com os problemas introduzidos pela desconexão e pelos recursos limitados do ambiente móvel. Seu principal bloco de formação é o *compact* que funciona como uma unidade básica de replicação de dados.

Um *compact* é representado como um objeto que encapsula: dados, operações para acesso aos

dados, operações para o gerenciamento do *compact*, informação a respeito do estado corrente do *compact*, regras de consistência e obrigações. O *compact* representa um acordo entre o cliente móvel que requisitou o dado e o servidor de banco de dados.

Nesse acordo, o servidor de banco de dados delega à unidade móvel o controle de alguns dados que serão usados por transações locais. O cliente móvel, em contrapartida, concorda em honrar com as condições específicas do uso dos dados de maneira que sua consistência seja mantida quando as atualizações forem propagadas de volta para o servidor de banco de dados. Cada *compact* é responsável pelo controle de concorrência e recuperação dos seus dados.

O PRO-MOTION sugere quatro fases de processamento de transações:

- *Armazenamento* – A unidade móvel está conectada à rede e os *compacts* estão sendo armazenados localmente na preparação para uma eventual desconexão.
- *Processamento conectado* – A estação móvel está conectada à rede e as transações estão acessando dados diretamente do servidor.
- *Processamento desconectado* – A estação móvel está desconectada da rede e as transações estão acessando dados locais contidos nos *compacts*.
- *Ressincronização* – A estação móvel se reconectou à rede e as atualizações feitas durante a desconexão estão sendo reconciliadas com o banco de dados no servidor.

Para aumentar a disponibilidade de dados, o PRO-MOTION permite a efetivação (*commit*) local de transações. Caso uma transação opte pela efetivação local, seus resultados se tornarão visíveis para outras transações da unidade móvel. Se a transação não optar pela efetivação local, ela só será efetivada localmente após a efetivação global realizada no servidor depois da reconexão.

A consistência das operações sobre os dados no PRO-MOTION pode ser caracterizada através de uma escala com dez níveis de isolamento baseados no padrão ANSI-SQL. O nível 9 garante a execução sequencial de transações e o nível 8 garante uma execução serializável. Cada nível inferior representa um menor nível de isolamento. O nível 0 não garante nenhum tipo de consistência. Flexibilizando a propriedade de isolamento, o PRO-MOTION aumenta a concorrência no acesso a dados, o que implica em uma melhoria de desempenho das transações.

4.3 Kangaroo

O modelo de transações Kangaroo (Dunham, 1997) enfoca a questão da mobilidade de transações, permitindo que elas possam executar mesmo a partir de uma unidade móvel em movimento. Nesse modelo, o local de origem de uma transação pode ser diferente do local de finalização da mesma. As operações de uma transação Kangaroo sempre são executadas em máquinas da rede fixa a partir de uma unidade móvel.

A execução de uma transação Kangaroo pode consistir na execução de um conjunto de transações chamadas *Joey*. Uma transação Kangaroo inicia a sua execução através de uma transação *Joey* em uma determinada estação base. À medida que a unidade móvel onde se encontra a transação Kangaroo se move de uma célula para outra, uma transação *Joey* da célula anterior é efetivada e é criada uma nova transação *Joey* na estação base da nova célula. Cada transação *Joey* representa uma parte da transação Kangaroo que foi executada em uma determinada célula.

Uma transação Kangaroo pode ser executada no modo *compensating* ou no modo *split*. Quando uma transação Kangaroo executa no modo *compensating*, se ocorrer falha na transação *Joey* corrente, todas as transações *Joey*s que já tiverem sido executadas são desfeitas juntamente com a transação *Joey* que falhou. No modo *split*, quando ocorre uma falha, a decisão de efetivar ou de abortar as transações *Joey* passa a ser responsabilidade do DBMS (Database Management System).

4.4 AMT

AMT (SERRANO-ALVARADO, 2003) é um modelo que tem como objetivo dar a transações a capacidade de se adaptar à alta instabilidade do ambiente de computação móvel caracterizada por largura de banda variável, desconexões e custo de comunicação variável. Para isso, ele provê um mecanismo de monitoramento capaz de informar as transações sobre as variações ocorridas no ambiente. Ciente do estado do ambiente de execução, uma transação AMT é apta a escolher a melhor forma de execução dentre um conjunto de alternativas previamente programadas. O sucesso de pelo menos uma das alternativas representa a execução correta da transação.

Uma transação AMT pode conter descritores de ambiente que expressam os estados no qual o ambiente deve estar para que cada alternativa seja executada. Quando uma transação é iniciada, o estado do ambiente é verificado e a alternativa de execução apropriada é escolhida. Se não existir nenhuma alternativa para o estado do ambiente, a execução da transação pode ser adiada.

Uma alternativa de execução será executada assim que um estado aceitável do ambiente for detectado. O modelo AMT utiliza um protocolo de efetivação chamado CO2PC que é uma combinação entre o protocolo Two-Phase Commit (de abordagem pessimista) e mecanismos de uma abordagem otimista. Esse protocolo permite que alguns tipos de transações possam ser efetivadas localmente dentro da abordagem otimista enquanto que outros tipos de transações só sejam efetivadas globalmente juntamente com a efetivação da EA.

4.5 Moflex

Moflex (KU, 2000) é um modelo que visa trazer mais mobilidade, heterogeneidade e flexibilidade para a execução de transações em ambientes de computação móvel. Esse modelo se propõe a permitir que transações possam ser executadas mesmo enquanto a unidade móvel se move entre diferentes células.

Uma transação Moflex é definida como uma coleção de subtransações que podem estar inter-relacionadas por um conjunto de dependências de execução. Podem ser definidos três tipos de dependências entre subtransações: dependências de sucesso (*ds*), dependências de falhas (*df*) e dependências externas (*de*). Uma transação *A* que possui uma *ds* em relação a uma transação *B* só pode ser executada se *B* tiver sido efetivada com sucesso. Se *A* possui uma *df* em relação a *B*, *A* só pode ser executada depois que *B* falhar. Quando uma transação possui uma *de*, ela só pode executar quando um predicado externo (tempo, custo ou localização) for satisfeito.

Uma transação Moflex pode conter regras de mudança de célula. Estas regras definem as políticas de execução de uma subtransação quando há mudança de célula em decorrência da movimentação da unidade móvel. Podem ser definidas as seguintes regras para a mudança de célula de uma transação: *continue(ti)* – a transação *ti* continua a sua execução na nova célula; *restart(ti)* – a transação *ti* é abortada na célula anterior e reiniciada na nova célula; *resume(ti)* – a transação *ti* é dividida em uma subtransação efetivada na célula anterior e uma segunda subtransação que continua a executar na nova célula; *split_restart(ti)* – a transação *ti* é dividida em uma subtransação efetivada na célula anterior e uma segunda subtransação é reiniciada na nova célula. As transações que executam com as regras *split_resume(ti)* ou *split_restart(ti)*, ou seja, que executam a operação de divisão da transação em duas subtransações, são acompanhadas por uma regra de junção que é usada para verificar se a concatenação das subtransações resultantes de *ti* é computacionalmente equivalente à *ti*.

4.6 HiCoMo

HiCoMo (Lee e Helal, 2002) é um modelo de transações voltado a aplicações de tomada de decisão que acessam dados do tipo estatístico ou agregados. O ambiente é composto por tabelas

base na rede fixa e agregados de dados das tabelas base (*data warehouse*) nas unidades móveis. Esse modelo se propõe a manter altas taxas de efetivação global de transações que acessam esses dados agregados durante os períodos de desconexão. Para isto, o modelo impõe restrições como: (i) os dados agregados a serem armazenados nas unidades móveis só devem conter valores como média, soma, mínimo e máximo; (ii) uma transação HiCoMo só pode realizar operações comutativas (adição e subtração) sobre os dados agregados; (iii) deve ser especificada uma margem de erro para a execução de uma transação HiCoMo.

O resultado do processamento de transações HiCoMo sobre os dados agregados durante a desconexão é transferido para as tabelas da rede fixa quando ocorre a reconexão. Para isso, transações base são criadas e executadas na rede fixa para gerar sobre as tabelas base o mesmo resultado que foi obtido durante a desconexão. Esse processo é realizado pelo seguinte algoritmo:

- **Passo 1.** Detecção de conflitos: verifica-se se há conflitos entre as transações HiCoMo e as transações base que executaram na rede fixa. Se algum conflito for detectado, a transação HiCoMo é abortada. Esta detecção de conflitos é implementada por um controle de concorrência otimista baseado em *timestamps*.
- **Passo 2.** Criação das transações base: se não existem conflitos, é criada uma transação base para cada tabela base que foi usada na formação dos dados agregados. Uma vez criadas, estas transações base são executadas como subtransações de uma transação aninhada estendida aplicadas sobre as tabelas base.
- **Passo 3.** Cancelamento (*abort*) de subtransações bases: quando submetidas à execução, algumas subtransações geradas podem abortar devido a regras de manutenção de integridade das tabelas base. Se a diferença que as subtransações abortadas criam estiver dentro da margem de erro, pode-se considerar que as subtransações obtiveram sucesso. Senão, novas subtransações podem ser geradas e executadas com uma nova estratégia. Se a transformação não resultar em sucesso em algum ponto, a transação HiCoMo é abortada.

4.7 MobileTrans

MobileTrans (Santos, Veiga e Ferreira, 2004) é um modelo que provê meios de uma transação se adaptar a diferentes cenários de ambientes de computação móvel. Nesse modelo, o desenvolvedor deve especificar uma política de adaptação para cada transação. Uma política de adaptação consiste em associar valores a um conjunto de atributos que irão determinar o comportamento da transação. Os atributos que podem ser definidos são os seguintes:

- **Consistência** – é possível especificar regras de consistência para se permitir o uso de versões desatualizadas de objetos se a cópia remota correspondente não estiver disponível.
- **Busca (fetching)** – é possível selecionar os objetos que devem ser copiados para a cache da unidade móvel antes da execução da transação; pode-se especificar também se os objetos devem ser copiados para cache previamente ou sob demanda.
- **Delegação** – é possível delegar a responsabilidade de efetivação de uma transação para outra transação.
- **Atomicidade** – é possível especificar se a transação pode ser efetivada mesmo se nem todos os participantes envolvidos estão acessíveis.
- **Caching** – é possível especificar que os objetos copiados para a cache e os objetos modificados por transações locais serão armazenados localmente de forma que possam ser acessados durante os períodos de desconexão.
- **Tratamento de falhas** – é possível determinar o que deve ser feito quando as condições de consistência, de busca e de atomicidade especificadas não puderem ser garantidas devido ao estado da rede. Também é possível mudar a configuração da transação em

tempo de execução como um meio de tratamento da falha ocorrida.

4.8 Modelo Baseado em Pré-Serialização

Pre-serialization (Dirckze e Gruenwald, 2000) é uma técnica de gerenciamento de transações móveis sobre um sistema de bases de dados múltiplas (SBDM). Esta técnica permite que uma transação móvel estabeleça a sua ordem de serialização³ antes mesmo de chegar à fase de efetivação. Nesse modelo, uma transação que executa na unidade móvel (chamada transação global) é composta por um conjunto de subtransações compensáveis na rede fixa. Cada subtransação engloba as operações da transação global realizadas sobre uma determinada base de dados do SBDM. Como todas as subtransações locais são compensáveis, elas podem ser efetivadas antes mesmo que a transação global tome a decisão de entrar na fase de efetivação. Isto permite que os recursos alocados para a execução da transação global possam ser liberados à medida que cada subtransação vai sendo finalizada e efetivada.

Em cada estação de suporte do cenário para o qual a técnica é proposta existe um coordenador de transações globais (GTC) responsável por submeter as subtransações aos servidores da rede fixa, gerenciar a migração da unidade móvel e a desconexão e verificar as propriedades de isolamento e atomicidade. Quando uma unidade móvel migra de uma estação de suporte para outra, o GTC da primeira estação se encarrega de enviar à segunda estação as informações necessárias para que a transação global possa continuar a sua execução.

Quando uma unidade móvel é desconectada, a transação global não é totalmente interrompida, as subtransações que já tiverem sido submetidas continuam a executar na rede fixa. Nesse período, todas as respostas geradas pelas transações locais são inseridas em uma lista que é enviada para a transação global quando ocorre a reconexão. Para evitar que os recursos de uma transação global T_x que não voltou a se reconectar fiquem indefinidamente alocados, quando outra transação T_y deseja acessar tais recursos, esses são automaticamente liberados para o uso de T_y e T_x pode ser considerada abortada.

Para verificar a garantia das propriedades de atomicidade e isolamento da transação global, o GTC, em determinados períodos, executa um algoritmo de serialização global em grafos. Para garantir a atomicidade, ou todas subtransações são efetivadas ou todas são abortadas ou compensadas (atomicidade semântica). Para garantir a propriedade de isolamento, as transações globais conflitantes são serializadas na mesma ordem da serialização de suas subtransações que executam em cada servidor.

4.9 Discussão

Para lidar com obstáculos presentes em ambientes de computação móvel, cada modelo descrito acima introduz um conjunto próprio de mecanismos. A tabela 1 destaca, para cada modelo, os mecanismos utilizados para lidar com os seguintes obstáculos: desconexões, largura de banda baixa ou variável, suprimentos de energia limitados, escassez de espaço em disco e mobilidade (mudança de célula).

Como se pode observar, todos os modelos descritos propõem pelo menos um mecanismo para lidar com o problema da desconexão. Nesse caso, o mecanismo mais utilizado é a realização de *caching* de dados, ou seja, cópia dos dados da rede fixa para a unidade móvel. Com isso, os dados se tornam disponíveis às aplicações nos momentos de desconexão.

Para lidar com os demais obstáculos, podem-se destacar: (i) os mecanismos do *Coda-IOT* e

3 - Segundo a regra da serialização, transações que executaram concorrentemente são consideradas corretas quando os resultados obtidos são equivalentes aos resultados de uma execução serial das mesmas (uma após a outra).

do *PRO-MOTION* para lidar com a baixa largura de banda – transferir para a cache somente as partes necessárias dos dados; (ii) os mecanismos do *PRO-MOTION* para lidar com a escassez de espaço em disco – transferir para a cache somente as partes necessárias dos dados; (iv) os mecanismos do *Kangaroo* para lidar com a mobilidade – divisão de transações e transferência de informações de uma estação base para a outra.

Tabela 1: Estratégias dos Modelos de Transações.

| Modelo | DESCONEXÃO | LARGURA DE BANDA | ENERGIA | DISCO | MOBILIDADE |
|-------------------------|--|--|---|--|--|
| Coda-IOT | <i>Caching</i> de dados; efetivação local de transações | Reintegração por partes; validação rápida de dados em cache | Não especificado | Não especificado | Não especificado |
| PRO-MOTION | <i>Caching</i> de dados; efetivação local de transações | Transmissão das partes dos dados em falta ou desatualizados | Não especificado | Transferência de partes de dados para a cache; otimização do <i>log</i> de recuperação | Não especificado |
| Kangaroo | Transações interrompidas por uma desconexão podem continuar a executar depois da reconexão | Não especificado | Desligar a unidade móvel permitindo a continuação da execução das transações quando esta for religada | Não especificado | Divide a execução da transação para que esta possa continuar a executar na nova célula |
| AMT | Permite que a transação decida qual alternativa deve ser executada | Permite que a transação decida qual alternativa deve ser executada | Permite que a transação decida qual alternativa deve ser executada | Permite que a transação decida qual alternativa deve ser executada | Não especificado |
| Moflex | Permite criação de dependências entre transações | Permite criação de dependências entre transações | Permite criação de dependências entre transações | Permite criação de dependências entre transações | Permite criação de dependências entre transações |
| HiCoMo | <i>Caching</i> de dados; restrições no acesso a dados em cache | Não especificado | Não especificado | Não especificado | Não especificado |
| MobileTrans | <i>Caching</i> de dados | A transação pode decidir quais serão as medidas de adaptação | A transação pode decidir quais serão as medidas de adaptação | Não especificado | Não especificado |
| Pré-serialização | As operações são sempre realizadas na rede fixa; as subtransações que já tiverem sido submetidas na rede fixa continuam a executar | Não especificado | Não especificado | Não especificado | Não especificado |

Nos modelos de transação descritos, notou-se um relaxamento das propriedades ACID. Isso foi feito por pelo menos um dos dois motivos: (i) para atender requisitos específicos de aplicações e (ii) para lidar com as características dos ambientes de computação móvel. A Tabela 2 mostra, para cada modelo, os tipos de relaxamento das propriedades ACID que foram adotados.

Tabela 2: Relaxamento das Propriedades ACID (ROCHA e TOLEDO, 2005).

| Modelo | ATOMICIDADE | CONSISTÊNCIA | ISOLAMENTO | DURABILIDADE |
|-------------------------|---|---|--|--|
| Coda-IOT | Sem flexibilidade | Sem flexibilidade | Liberção de resultados parciais antes da efetivação global | Sem garantia de durabilidade |
| PRO-MOTION | Transações aninhadas | Regras de consistência para acesso a dados | Provê dez níveis de isolamento | Sem garantia de durabilidade |
| Kangaroo | Divisão de transações no processo de mudança de célula | Não Especificado | Sem garantia de isolamento | Não especificado |
| AMT | Transações aninhadas | Não especificado | Liberção de resultados parciais antes da efetivação global | Não especificado |
| Moflex | Divisão de transações no processo de mudança de célula | Regras de junção de subtransações que definem a correteude de uma transação | Sem garantia de isolamento | Não especificado |
| HiCoMo | Divisão de transações no processo de mudança de célula | Pode-se especificar uma margem de erro para execução de transações | Liberção de resultados parciais antes da efetivação global | Tolera o descarte de atualizações dentro de uma margem de erro |
| MobileTrans | Permite efetivação de transações mesmo se determinados objetos não puderem ser efetivados | Regras de consistência para acesso a dados | Permite a leitura de dados não atualizados | Tolera a efetivação de transações mesmo se todas as atualizações não puderem ser propagadas para a rede fixa |
| Pre-serialização | Sem flexibilidade | Sem flexibilidade | Liberção de resultados parciais antes da efetivação global | Sem flexibilidade |

Fonte: Rocha e Toledo (2005)

Apesar da variedade de mecanismos de adaptação e da variedade das propriedades transacionais providos pelo conjunto dos modelos de transação apresentados, quando visto de forma isolada, cada modelo provê somente um subconjunto limitado desses mecanismos e propriedades. Essa limitação faz com que cada modelo atenda somente aos requisitos de pequenos grupos de aplicação. Além disso, esses modelos não foram estruturados de forma extensível, ou seja, se a aplicação possuir requisitos um pouco diferentes daqueles previstos pelo modelo, ele não é capaz de prover.

5 Frameworks para Transações

Considere os seguintes dois fatos: (1) cada aplicação pode possuir um conjunto de requisitos distintos com relação às propriedades transacionais; (2) os modelos de transação propostos normalmente proveem um conjunto fixo e limitado de propriedades transacionais (tomem-se como exemplo os modelos apresentados na seção anterior). Esses fatos podem tornar difícil a tarefa de encontrar um modelo transacional que se enquadre perfeitamente nos requisitos desejados de uma determinada aplicação.

Diante dessa dificuldade, surgiram várias pesquisas que propuseram métodos de flexibilização do apoio transacional. A ideia principal foi, ao invés de se propor um apoio transacional com um conjunto fixo de propriedades, a de permitir a personalização do apoio transacional de forma a atender aos requisitos específicos de cada aplicação. Esta seção apresenta alguns trabalhos que foram guiados nesse sentido.

5.1 ACTA

O ACTA (CHRYSANTHIS e RAMAMRITHAM, 2004) é dos trabalhos que forneceram bases para a definição de um apoio transacional flexível. Esse trabalho demonstrou como a estrutura e o comportamento de transações estendidas poderiam ser analisados e formalmente especificados. Para a especificação sistemática de modelos de transação estendidos, esse trabalho propôs o *framework* ACTA. Nesse *framework*, a especificação de um modelo estendido consiste na definição dos seguintes elementos:

- **Histórico.** Representa a execução concorrente de um conjunto de transações T . Um histórico define a ordem parcial dos eventos significantes (gerados por operações de controle como, *begin*, *commit* e *abort*) e dos eventos sobre objetos (gerados por operações sobre objetos) invocados por transações em T . Segundo o ACTA, através da imposição de restrições a um histórico, pode-se definir o efeito das transações sobre outras transações e sobre os objetos.
- **Dependências entre transações.** São de dois tipos: dependência de efetivação (*de*) – se uma transação t_i possuir uma *de* em relação a uma outra t_j , t_i só poderia ser efetivada quando t_j fosse efetivada ou cancelada; dependência de cancelamento (*dc*) – se t_i tivesse uma *dc* em relação a t_j , se t_j fosse cancelada, t_i teria que ser também cancelada.
- **Visão.** Para cada transação t deve ser definida uma visão. Uma visão determina os objetos e estados de objetos que são visíveis a t em um dado ponto no tempo. Uma visão normalmente é definida como um subconjunto do histórico que satisfaz a um dado predicado.
- **Conjunto de operações conflitantes.** O conjunto de operações conflitantes de uma transação t (*ConflictSet*) são todas aquelas operações pertencentes ao histórico que podem ser conflitantes em relação a operações invocadas por t . Quando t invoca uma operação, os conflitos desta operação são verificados junto ao *ConflictSet*.
- **Delegação.** Uma transação t_a pode delegar operações para uma outra transação t_b . Quando t_a delega uma operação *op* para t_b , t_b passa a ser responsável por efetivar ou cancelar *op*.

Esse trabalho demonstrou que através do uso desses cinco blocos, vários modelos de transação estendidos podem ser especificados. Como exemplo, esse trabalho especificou modelos como, transações aninhadas, aninhadas abertas e *split/join*. Apesar de ter sido um trabalho teórico, esse *framework* tem sido usado como base para diversos outros trabalhos que implementam *frameworks* flexíveis de transação.

5.2 Transações Bourgoigne

Este trabalho (Prochazka, 2001) apresenta uma extensão para o apoio transacional do Enterprise JavaBeans. As limitações do apoio a transações provido pelo Enterprise JavaBeans foram identificadas e um modelo chamado transações Bourgoigne foi proposto para sobrepor-las. As transações Bourgoigne visam dar ao Enterprise JavaBeans o apoio a modelos avançados de transações.

Para permitir a definição de transações avançadas, esse trabalho tomou como base a ideia do *framework* ACTA – de que um modelo de transação arbitrário pode ser especificado usando

a definição de eventos significantes da transação (tal como a criação, o início, a efetivação e o cancelamento da transação), o estabelecimento de dependências entre transações, delegação e compartilhamento de recursos entre transações. Na prática, o modelo Bourgoigne estende o apoio a transações do Enterprise JavaBeans através das seguintes premissas:

- **Delegação recursos de uma transação para outra.** Uma transação pode mover objetos associados a ela para uma outra transação tal que a transação receptora se torna responsável pelos processos de efetivação e cancelamento dos mesmos.
- **Compartilhamento de recursos entre transações.** Uma transação pode dar a uma outra o direito de acessar dados reservados para ela. Esta permissão pode ser para escrita ou leitura e pode se restringir a somente um subconjunto das operações de um objeto.
- **Estabelecimento de dependências entre transações.** Dependências são vínculos condicionais entre transações que podem depender de eventos significantes como a efetivação ou o cancelamento de uma transação. Por exemplo, se uma transação *A* possui uma dependência de cancelamento em relação a uma transação *B*, se *B* for cancelada, *A* também deve ser cancelada.

5.3 Síntese de Plataformas Transacionais

Este trabalho (Zarras e Issarny, 1998) apresentou um método que, dados os requisitos transacionais de uma aplicação, sintetiza uma plataforma que atende a esses requisitos baseado no uso de serviços já existentes. Esse processo de sintetização de plataformas transacionais é realizado em três etapas: especificação dos requisitos transacionais, seleção dos serviços existentes e composição da plataforma.

Para a especificação dos requisitos transacionais deve ser provida uma descrição da arquitetura da aplicação e uma especificação formal das propriedades transacionais usadas pela aplicação. A descrição da arquitetura identifica as interfaces providas e requeridas para cada componente da aplicação assim como as interconexões entre os componentes. Na especificação dos requisitos transacionais, esse método provê a flexibilização das propriedades ACID. Cada propriedade ACID pode ser gradualmente refinada em propriedades mais especializadas. As propriedades de um dado modelo transacional corresponderá à conjunção de um refinamento de cada uma das propriedades ACID. A especificação destas propriedades deve ser feita formalmente através do uso de lógica temporal.

No processo de seleção, dada a especificação formal das propriedades transacionais requeridas, um procedimento verifica e seleciona os serviços de *middleware* existentes que atendem a cada propriedade especificada. Esse procedimento só é executado com sucesso se, para cada propriedade especificada, existe pelo menos um serviço disponível que a atende.

Depois de selecionados os serviços de *middleware*, o método usa um procedimento de composição que utiliza esses serviços e a descrição da arquitetura da aplicação para sintetizar a plataforma desejada. Com base nestas informações, esse procedimento gera um conjunto de *stubs* que intermediará a interconexão entre os componentes da aplicação e os serviços selecionados. Depois de gerados os *stubs*, o método sintetiza a plataforma transacional combinando os serviços selecionados com os componentes funcionais da aplicação.

5.4 Monitor de Processamento de Transações Extensível

Este trabalho (Barga e Pu, 1996) teve como objetivo permitir a implementação de modelos de transações estendidos a partir de um monitor de processamento de transações (TP-Monitor) comercial existente. No geral, a definição de transações estendidas é feita em uma camada de mais alto nível utilizando operações primitivas providas pelo monitor de processamento de

transações.

Nesse trabalho, a reflexão computacional é usada na definição de adaptadores de transações – módulos de *software* reflexivos que têm a tarefa de revelar aspectos normalmente não visíveis da funcionalidade do TP-Monitor. Cada adaptador corresponde a um aspecto particular do TP-Monitor como, a execução da transação, gerenciamento de trancas, detecção de conflitos e gerenciamento de *logs*. Um adaptador de transação contém metaobjetos que representam o comportamento desejado para o componente funcional do TP monitor e uma metainterface para controlar o comportamento desse componente. O comportamento do TP-Monitor pode ser mudado (para a definição de modelos de transação estendidos) atualizando-se esses metaobjetos através da metainterface do adaptador. Dessa forma, programadores poderiam implementar modelos de transações alternativos e criar novas interfaces de aplicação de acordo com os requisitos de suas aplicações.

A arquitetura proposta é dividida em dois níveis: o nível base e o metanível. O nível base é onde fica o TP-Monitor e seus componentes funcionais. No metanível ficam os adaptadores de transação. Como o nível base não foi desenvolvido com apoio a reflexão, ela foi implementada no metanível com base no uso de eventos propagados pelo nível base.

Um evento do nível base é propagado para um dos adaptadores do metanível toda vez que o estado de uma transação está para ser mudado, por exemplo, uma transação está para ser abortada ou efetivada. Ao receber a notificação do evento, o adaptador toma o controle da transação juntamente com as informações que descrevem o evento. A partir destas informações, o adaptador, a partir dos seus metaobjetos, verifica qual o modelo estendido que está sendo executado e responde ao evento com chamadas à API do TP-Monitor de acordo com o modelo correspondente.

5.5 Transações Estendidas com o Reflective Java

Este trabalho (Wu, 1998) descreve uma arquitetura flexível de apoio a transações baseada no uso do Reflective Java. Nessa arquitetura, serviços de transação podem ser personalizados para atender necessidades específicas de aplicações de *internet*. Para se personalizar um serviço de transação, estratégias de implementação são passadas para um *container* na forma de metaobjetos. A troca de um metaobjeto por outro pode representar uma reconfiguração do serviço de transação para atender necessidades específicas das aplicações.

A arquitetura de apoio a transações organiza os metaobjetos em uma estrutura de dois níveis. No primeiro nível ficam os metaobjetos relacionados com a coordenação dos objetos do segundo nível e no segundo nível ficam os metaobjetos responsáveis por tarefas básicas do serviço de transação como, controle de concorrência e gerenciamento da persistência.

Para prover flexibilidade e adaptabilidade de serviços de transação, cada tarefa básica pode possuir diversas implementações representadas através de diferentes metaobjetos. Dessa forma, um serviço de transação pode ser personalizado selecionando-se os metaobjetos mais apropriados para cada tarefa do serviço.

5.6 CAGISTrans

CAGISTrans (Ramampiaro e Nygard, 2004) é um *framework* de apoio a transações que, além de prover flexibilização de serviços de transação, visa a adaptação de transações em relação a aspectos dinâmicos de ambientes cooperativos. Esse trabalho assume que o apoio à adaptação dinâmica é essencial para se obter um maior sucesso na execução de transações em ambientes que podem sofrer mudanças em tempo de execução.

O CAGISTrans divide a especificação de transações em duas partes. A primeira parte é chamada

de especificação das características – em que são definidos todos os aspectos fixos da execução de uma transação como, propriedades ACID, estrutura de relacionamento entre transações, critério de corretude, mecanismos e regras que definem o uso dos critérios de corretude. A segunda parte é chamada de especificação da execução – onde são definidos todos os aspectos da transação que poderão apresentar mudança em tempo de execução. A especificação da execução é dividida em três blocos:

- **Operações de gerenciamento.** Permite a reestruturação dinâmica de transações dando a estas a capacidade de delegar operações entre si.
- **Operações avançadas.** Permite a definição de operações de mais alto nível baseadas no uso de operações simples de leitura e escrita.
- **Gerenciamento e controle do comportamento transacional.** Permite a definição de regras de corretude pelo usuário. Basicamente estas regras definem: (i) operações que não podem ser executadas concorrentemente; (ii) operações que, embora sejam consideradas conflitantes com as operações definidas em (i), ainda podem ser executadas concorrentemente; (iii) seqüências de operações que devem aparecer para se alcançar execuções corretas.

5.7 ReflecTS

ReflecTS (Arntsen e Karlsen, 2007) é uma plataforma reflexiva de apoio serviços de transações que provê apoio concorrente de diferentes serviços de transação. O principal objetivo dessa plataforma é o de atender a diversidade de requisitos das aplicações permitindo que eles escolham o serviço mais adequado.

É através de uma arquitetura chamada TSEnvironment (Karlsen e Jakobsen, 2003) que o ReflecTS permite que diferentes serviços de transação possam ser instalados, modificados e usados concorrentemente de acordo com as necessidades das aplicações. Cada serviço de transação é considerado um componente que pode ser desenvolvido independentemente e que pode ser composto por componentes menores chamados componentes de serviço. Um componente de serviço normalmente implementa uma tarefa bem definida de um serviço de transação como, controle de concorrência, efetivação ou recuperação.

Quando uma aplicação requisita apoio transacional ao ReflecTS, ela fornece a especificação formal do serviço de transação a ser usado. A partir desta especificação, o ReflecTS realiza as seguintes etapas: (i) seleciona o serviço de transação adequado dentre os serviços disponíveis, (ii) verifica a compatibilidade do serviço com relação aos gerenciadores de recursos e com relação aos demais serviços de transação correntemente ativos. Se estas etapas forem cumpridas com sucesso, o ReflecTS ativa o serviço de transação requisitado e o disponibiliza à aplicação requisitante.

Para permitir introspecção e reconfiguração, o ReflecTS disponibiliza um conjunto de metainterfaces. Através dessas metainterfaces podem-se realizar as seguintes ações: (i) adicionar, remover ou modificar os componentes do ReflecTS; (ii) adicionar, remover e reconfigurar serviços de transação; e (iii) permitir a seleção de serviços de transação dentre os disponíveis. Para evitar que estas reconfigurações comprometam a integridade da plataforma, cada mudança realizada é comparada a um conjunto de arquiteturas válidas pré-definidas.

5.8 Discussão

Apesar de os trabalhos descritos acima permitirem a definição de propriedades transacionais personalizadas, eles possuem algumas limitações. O ACTA é um modelo meramente formal. Os demais trabalhos, com exceção do CAGISTrans, não consideram os requisitos transacionais de

ambientes dinâmicos – que requer a capacidade de reconfiguração em tempo de execução. Em especial, o CAGISTrans se limita a atender requisitos específicos de ambientes cooperativos.

Embora esses trabalhos tentem prover um apoio transacional personalizável, ainda há uma lacuna no que diz respeito ao apoio à reconfiguração em tempo de execução necessária em ambientes dinâmicos. Um dos elementos que podem ajudar na busca de uma solução é o estudo de *middlewares* abertos.

6 Middlewares Abertos

Middlewares têm emergido como um componente arquitetural importante no apoio a aplicações distribuídas. A função do *middleware* é apresentar um modelo de programação unificado aos desenvolvedores e mascarar problemas de heterogeneidade e de distribuição (Blair *et al.*, 1998). Além disso, *middlewares* normalmente disponibilizam um conjunto de serviços de alto nível que podem ser usados para facilitar o desenvolvimento de aplicações. Dentre esses serviços está o de transações. A importância do conceito de *middleware* é refletida na crescente popularidade de seus representantes como, CORBA, DCOM/.NET e Java RMI/J2EE.

Com os recentes avanços em sistemas distribuídos, móveis e onipresentes, surgiram novas infra-estruturas computacionais caracterizadas por um alto dinamismo. Esse dinamismo exige das aplicações a capacidade de se adaptar às mudanças ocorridas no ambiente (Kon, 2002). O problema é que *middlewares* convencionais não têm a capacidade de adaptação e reconfiguração necessárias para lidar com os aspectos introduzidos por estas novas infra-estruturas. Uma das soluções propostas para transpor estas limitações foi o uso da chamada implementação aberta. Através dela, aspectos da implementação do *middleware* seriam revelados dando a esses a capacidade de reconfiguração, tanto para atender aos requisitos das aplicações quanto para melhor se adequar ao ambiente. Esta é a característica principal dos chamados *middlewares* abertos (Kordon e Pautet., 2005).

São vários os trabalhos que podem ser considerados parte integrante desse novo conceito de *middleware*, dentre eles estão o Quarterware, o OpenCORBA, o DynamicTAO, o OpenORB, e o ReMMoC. Nas seções a seguir, será apresentada uma breve descrição desses trabalhos.

6.1 Quarterware

O Quarterware (SINGHAI, 1999) propõe um *framework* para a construção de *middleware* flexível e de alto desempenho. Para permitir apoio a diferentes funcionalidades, o Quarterware abstrai as propriedades comuns presentes em *middlewares* convencionais permitindo que versões especializadas possam ser implementadas para satisfazer necessidades específicas das aplicações. Para atender os requisitos de desempenho das aplicações, o Quarterware permite otimizações com relação ao gerenciamento de memória e a comunicação. O Quarterware também disponibiliza interfaces reflexivas que tornam as aplicações aptas a interagir com o *middleware* e reconfigurá-lo dinamicamente.

6.2 OpenCORBA

O OpenCORBA (LEDOUX, 1999) é um ORB que revela aspectos da implementação do CORBA através da reflexão para permitir a reconfiguração dinâmica. Esses aspectos são representados por meio de metaclasses, cuja mudança dinâmica permite modificar os mecanismos do ORB por elas representados. Os aspectos do ORB que podem ser reconfigurados são: os mecanismos de invocação remota, a verificação de tipos do IDL e o gerenciamento de exceções.

6.3 DynamicTAO

O DynamicTAO (Kon *et al.*, 2000) é um trabalho que utilizou reflexão computacional para permitir a reconfiguração dinâmica a partir de um *middleware* chamado TAO. O DynamicTAO exporta uma interface reflexiva que pode ser usada por objetos locais ou remotos para instalar, desinstalar ou configurar componentes. O DynamicTAO também propõe mecanismos que visam evitar que estas reconfigurações comprometam a integridade do *middleware* como um todo.

6.4 OpenORB

O OpenORB (Blair *et al.*, 2001) é um *middleware* que foi projetado para ser altamente configurável e reconfigurável dinamicamente de forma a prover apoio a aplicações com requisitos dinâmicos. Diferentemente do DynamicTAO e do OpenCORBA que foram projetados com base na implementação de *middlewares* já existentes, o OpenORB foi inteiramente projetado e implementado independentemente de estruturas de *middleware* já existentes. Instâncias personalizadas do OpenORB podem ser configuradas selecionando os componentes apropriados. A arquitetura do OpenORB permite que componentes possam ser identificados e reconfigurados em tempo de execução.

A reconfigurabilidade dinâmica do OpenORB é obtida através de um extensivo uso de reflexão com uma clara separação entre o nível base e o metanível. Enquanto o nível base consiste de componentes que implementam os serviços usuais do *middleware*, o metanível expõe a implementação desses componentes através da reflexão permitindo inspeção e adaptação.

6.5 ReMMoC

O ReMMoC (GRACE, BLAIR e SAMUEL, 2005) é um *middleware* direcionado a lidar com aspectos do ambiente de computação móvel. O ReMMoC foi projetado para ser configurável, reconfigurável e para dar a clientes móveis a capacidade de interoperar transparentemente com diversos serviços independentemente do protocolo de comunicação. Para permitir configuração e reconfiguração, o ReMMoC foi projetado com base no modelo de componentes OpenCOM, reflexão e *framework* de componentes.

A estrutura do ReMMoC usa dois *frameworks* de componentes: *binding framework* – responsável pela interoperabilidade entre os serviços implementados em diferentes paradigmas de *middleware* (IIOP, SOAP); *discovery framework* – responsável pela localização de serviços móveis independentemente do protocolo de localização (ex: SLP, UPnP e Jini). Esses *frameworks* são compostos por grupos de componentes e cada grupo implementa um tipo de protocolo. Outros protocolos também podem ser implementados e adicionados ao *framework*.

Para permitir que clientes sejam desenvolvidos independentemente do tipo de *middleware* com o qual eles interagirão, o ReMMoC disponibiliza uma API baseada em uma abstração dos diferentes tipos de serviço. Esta API pode ser usada tanto para a localização de serviços quanto para o acesso a serviços conhecidos. Quando é utilizada uma operação de localização da API, o ReMMoC utiliza transparentemente os diversos protocolos de localização implementados e devolve a lista de serviços encontrados. Ao receber uma operação de acesso a serviços através da API, o ReMMoC mapeia essa chamada para o protocolo do *middleware* correspondente.

6.6 Discussão

Uma característica em comum que pode ser encontrada entre esses *middlewares* abertos apresentados é a combinação do uso de reflexão, componentes e *framework* de componentes como técnicas para se obter implementações mais flexíveis e com capacidade de configuração

dinâmica. Com isso, a partir de uma única implementação podem ser derivadas várias especializações capazes de atender os requisitos de uma gama maior de aplicações.

Apesar dos avanços alcançados pelo conceito de *middleware* aberto, algumas limitações ainda precisam ser superadas. As soluções apresentadas restringem-se basicamente aos problemas da heterogeneidade e da distribuição. Soluções para outros serviços importantes que podem estar integradas a plataformas de *middleware*, como serviços de transações, não são apresentadas.

Outra limitação das soluções correntes em *middlewares* abertos está na estrutura utilizada para representá-los. No geral, a representação de *middlewares* abertos conta somente com elementos tradicionais dos modelos de desenvolvimento baseados em componentes: componentes e conexões. Essa representação é exibida ao usuário do *middleware* (desenvolvedor da aplicação) através de interfaces reflexivas para que ele possa fazer as requeridas configurações.

Porém, possuir uma estrutura baseada em componentes que pode ser manipulada via reflexão não é o suficiente para encorajar os usuários a configurar o *middleware* aberto. A principal preocupação desse usuário está em utilizar os serviços do *middleware* para desenvolver aplicações. Se o processo de configuração do *middleware* for complexo, o usuário poderá perder o interesse em utilizá-lo, já que um dos principais motivos para se adotar o *middleware* é justamente o de facilitar o processo de implementação de aplicações.

Uma tentativa de facilitar o processo de configuração de *middlewares* comumente adotada pelos trabalhos em *middleware* abertos apresentados é a utilização de *framework* de componentes (Grace, Blair e Samuel, 2005; Jakobsen e Karlsen, 2005; Blair *et al.*, 2002). Porém, nesses trabalhos, um *framework* de componentes é simplesmente um invólucro que contém uma sub-arquitetura composta de componentes e conexões. Dessa maneira, essa noção de *framework* de componentes pode ser resumida na simples noção de componente composto. É certo que componentes compostos podem ajudar na organização da estrutura de um *middleware*, mas eles não são o suficiente para torná-lo facilmente configurável para o usuário.

Por fim, o estudo de *middlewares* abertos pode ser um caminho promissor para se chegar a serviços de transações flexíveis e configuráveis o bastante para atender os variáveis requisitos de ambientes dinâmicos, como a computação móvel. Porém, novas formas de representação de *middleware* devem ser estudadas para que o processo de configuração esteja ao alcance dos interesses dos usuários.

7 O Modelo Motherboard

Inspirado na solução adotada em placas mãe em *hardware*, foi criado o modelo *motherboard* (Rocha e Toledo, 2008; Rocha *et al.*, 2007). Esse modelo visa lidar com aspectos relativos à complexidade existente em se configurar *middlewares* abertos. O modelo *motherboard* foi incorporado na implementação de uma plataforma para o desenvolvimento de *middleware* chamada MotherFrame (Rocha, 2008). Essa plataforma foi desenvolvida com base em um modelo para desenvolvimento de *middlewares* abertos chamado OpenCOM (Clarke, 2001).

A ideia chave para a elaboração do modelo *motherboard* está na separação de atribuições entre os principais grupos interessados: o desenvolvedor do *middleware* e o usuário do *middleware* (programador de aplicações). Essas atribuições estão listadas a seguir:

Atribuições do Desenvolvedor

- Definir todos os detalhes quanto à estrutura e funcionalidade do *middleware*.
- Definir precisamente os aspectos relacionados com a reconfiguração do *middleware*: quais partes são reconfiguráveis e como fazer a reconfiguração.

Atribuições do Usuário

- Identificar as possibilidades de reconfiguração que o *middleware* pode assumir.
- Configurar o *middleware* de forma a atender os requisitos desejados.

- Usar os serviços do *middleware* no desenvolvimento de aplicações.

Considerando as diferentes atribuições que há entre esses dois grupos de interessados, o modelo *motherboard* possibilita a definição de *middleware* aberto em duas diferentes visões: (i) a visão do desenvolvedor do *middleware* (MD-view) e (ii) a visão do usuário do *middleware* (MU-view) descritas a seguir.

7.1 Visão do Desenvolvedor de Middleware

A visão do desenvolvedor do *middleware* (MD-view) é a parte do modelo *motherboard* usado para se definir as atribuições do desenvolvedor do *middleware*: definir os detalhes da arquitetura e os pontos de configuração. Em particular, uma MD-view define um nível base de *middleware* a partir do qual se podem derivar diferentes configurações. Fazendo uma analogia às placas mãe em *hardware*, o MD-view é a placa mãe do *middleware*.

O modelo *motherboard* provê os seguintes elementos para a definição do MD-view: componentes, conexões, interfaces, *jumpers*, *slots* e demultiplexadores. Os três primeiros são elementos comumente encontrados em definições tradicionais de *middleware* baseado em componentes. Os demais são elementos introduzidos pelo modelo *motherboard* para definir os pontos de configurabilidade do *middleware*. Uma breve definição de cada um desses elementos é dada como segue:

- **Componente.** Um componente é uma unidade de computação que implementa uma tarefa bem definida. Um componente é dito simples quando é monolítico e composto quando possui uma arquitetura interna formada por outros componentes interligados.
- **Interface.** Uma interface define operações a partir das quais componentes interagem. Em um componente, as interfaces podem ser providas ou requeridas. Interfaces providas são aquelas pelas quais um componente provê seus serviços (a outros componentes ou a usuários). Interfaces requeridas são aquelas pelas quais um componente usa serviços de outros componentes.
- **Conexão.** Uma conexão é definida pela interligação entre a interface requerida de um componente e a interface provida de um outro componente. Através de uma conexão, o primeiro componente é capaz de utilizar serviços providos pelo segundo.
- **Jumper.** Um *jumper* é um tipo especial de conexão que pode ser ativado ou desativado. Um *jumper* ativado cria uma conexão entre os dois componentes envolvidos. No *jumper* desativado, essa conexão é desfeita.
- **Slot.** Um *slot* é um elemento que equivale a um espaço reservado à conexão de componentes prontos. Um *slot* é capaz de prover um ponto de configurabilidade no qual diferentes variações implementadas como componentes independentes podem ser conectadas. Um *slot* pode ser simples ou múltiplo. No primeiro caso, somente um componente pode ser conectado por vez. No segundo caso, mais de um componente podem ser conectados ao mesmo tempo para serem usados em paralelo ou selecionados em tempo de execução. A definição de um *slot* pode incluir interfaces providas e requeridas. É o conjunto dessas interfaces que irá definir se um componente é compatível com um *slot*. Um componente é compatível com um dado *slot* se o conjunto formado por suas interfaces providas e requeridas é equivalente ao do *slot*.
- **Demultiplexador.** O demultiplexador é um elemento especial que mapeia uma interface requerida de um componente com à interface provida de dois ou mais componentes. Assim definido, o demultiplexador permite que, uma vez fornecida uma chave de seleção, a interface provida correspondente seja conectada à interface requerida.

7.2 Notação Gráfica da MD-View

Para facilitar o projeto da visão do desenvolvedor do *middleware*, foi definida uma notação gráfica. Essa notação inclui uma representação para cada elemento de uma arquitetura do tipo MD-View: componente simples, componente composto, interface provida, interface requerida, demultiplexador, *slot*, *jumper* ativado e *jumper* desativado (Figura 4). Nessa notação, para representar um *slot* simples ou múltiplo coloca-se o número indicador “1” ou o símbolo indicador “*” sobrescrito logo após o seu identificador, respectivamente. Se após o identificador do *slot* não houver nenhum dos indicadores (“1” ou “*”) sobrescritos, o *slot* é considerado simples.

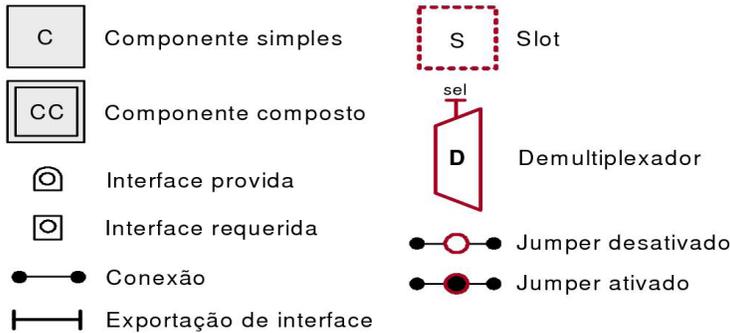


Figura 4: Notação Gráfica da MD-View.

Fonte: Rocha (2008).

Para ilustrar o uso da notação gráfica da MD-View, a Figura 5(a) apresenta um exemplo simples. Nesse exemplo, uma arquitetura do tipo MD-View é definida pelos seguintes elementos: componentes simples C_1, C_2, C_3 e C_4 , componente composto CC_1 , *slot* S_1 , *jumper* J_1 e conexões B_1, B_2, B_3, B_4 e B_5 . Como se pode observar, nessa arquitetura foram definidos três elementos abertos a partir dos quais podem ser derivadas diferentes configurações – pode-se conectar um componente no *slot* S_1 , ativar ou desativar o *jumper* J_1 e selecionar a porta de saída do demultiplexador D_1 .

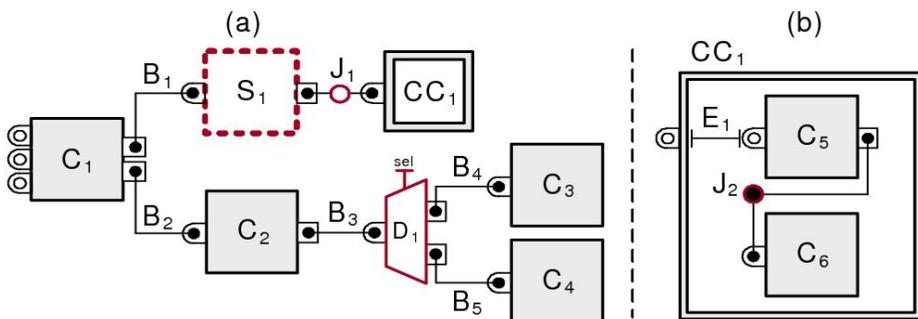


Figura 5: Exemplo de Uso da Notação Gráfica da MD-View.

Fonte: Rocha (2008).

A Figura 5(b) mostra como a arquitetura interna de um componente composto pode ser representada graficamente. Em particular, foi representada a arquitetura do componente CC_1 – composto pelos componentes C_5 e C_6 e pelo *jumper* J_2 . Pode-se notar que nessa arquitetura, o componente C_5 exporta uma interface – isso significa que essa interface do componente C_5 passa

a ser uma interface do componente composto CC_1 .

7.3 Visão do Usuário de Middleware

A visão do usuário do *middleware* (MU-View) é a parte do modelo *motherboard* em que o usuário (programador de aplicação) configura o *middleware* de acordo com os requisitos desejados. Na MU-View, ao invés de ter que lidar com detalhes complexos da estrutura interna do *middleware*, o usuário só precisa manipular os pontos abertos previamente definidos pelos desenvolvedores do *middleware* como uma MD-View.

Na prática, a definição de uma MU-View envolve um conjunto das seguintes tarefas:

- **Instanciar componentes.** O processo de definição de uma MU-View pode envolver a instanciação de componentes. Normalmente, antes da tarefa de conectar um novo componente em um determinado *slot*, esse componente precisa ser instanciado anteriormente.
- **Conectar componentes em slots.** Depois de instanciados, os componentes que implementam os requisitos desejados podem ser conectados nos *slots* disponíveis.
- **Ativar ou desativar jumpers.** Os *jumpers* disponíveis podem ser ativados ou desativados de acordo com os requisitos desejados para o *middleware*.
- **Selecionar saída de demultiplexadores.** Para cada demultiplexador disponível pode-se definir qual a conexão de saída que deverá ser usada na arquitetura. Na prática, selecionar uma conexão de saída, significa conectar uma interface requerida de um determinado componente com a interface provida de um outro componente específico.

7.4 Visão do Desenvolvedor de Middleware

A notação gráfica da MU-View enfoca um subconjunto da notação da MD-View representado pelos elementos abertos disponíveis (*jumpers*, *slots* e demultiplexadores). Na prática, a notação da MU-View visa representar graficamente o resultado da manipulação dos elementos abertos previamente definidos na MD-View. Essa notação é apresentada na Figura 6.

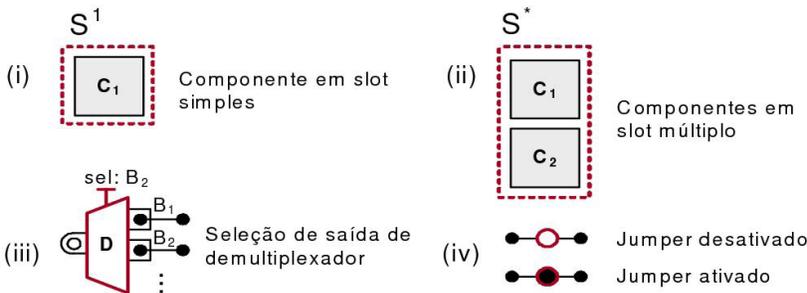


Figura 6: Notação Gráfica da MU-View.

Fonte: Rocha (2008).

A notação da MU-View mostra como devem ser representados: (i) um componente conectado em um *slot* simples; (ii) mais de um componente conectado em um *slot* múltiplo; (iii) seleção de conexão de saída de um demultiplexador (representada pelo símbolo "sel:" seguido do identificador da conexão selecionada); (iv) ativação ou desativação de *jumpers*.

A Figura 7 e a Figura 8 apresentam dois exemplos de como a manipulação dos elementos abertos de uma arquitetura pode ser representada graficamente. A MD-View usada como base para esses exemplos é a apresentada na Figura 5.

No primeiro exemplo é mostrada uma MU-View com a seguinte configuração: componente C_7 conectado no slot S_1 , *jumper* J_1 ativado e saída B_4 do demultiplexador ativada. Além disso, esse exemplo mostra a configuração da arquitetura interna do componente composto CC_1 . Nesse caso, o *jumper* J_2 do componente CC_1 foi desativado.

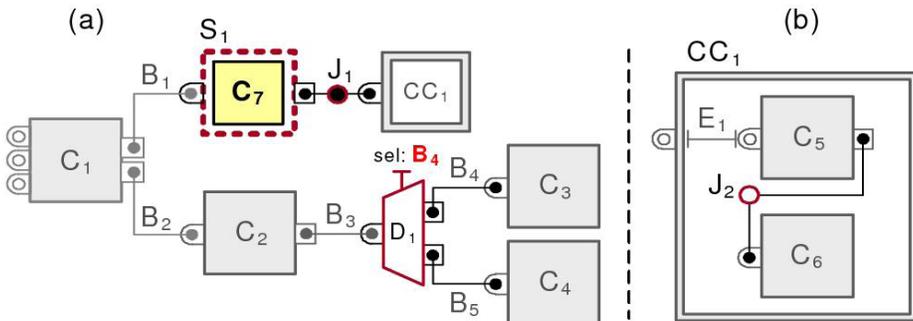


Figura 7: Notação Gráfica da MU-View (Primeiro Exemplo).

Fonte: Rocha (2008).

No segundo exemplo é mostrada uma outra MU-View com a seguinte configuração: componente C_8 conectado no slot S_1 , *jumper* J_1 desativado e saída B_5 do demultiplexador ativada. Nesse exemplo, o *jumper* J_2 do componente CC_1 foi mantido desativado.

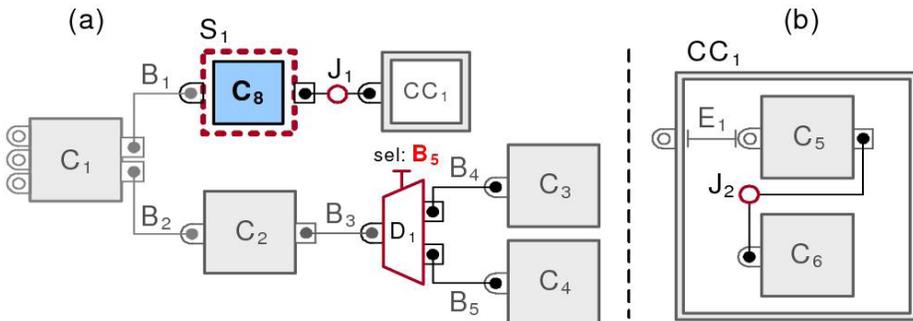


Figura 8: Notação Gráfica da MU-View (Segundo Exemplo).

Fonte: Rocha (2008).

8 Considerações Finais

Com o surgimento da computação móvel, novos obstáculos foram introduzidos em ambientes de sistemas distribuídos. Para permitir que aplicações de acesso a dados distribuídos possam executar neste ambiente de maneira satisfatória, vários modelos de transações para a computação móvel foram propostos. Porém, estes modelos possuem limitações como: (i) alguns deles são voltados para um domínio de aplicação específico; (ii) alguns modelos não possuem a flexibilidade das propriedades ACID necessária para atender requisitos mais específicos de aplicações avançadas; (iii) alguns modelos não tratam problemas relevantes do ambiente móvel. Para isso, o conceito de *middleware* aberto utilizado pode ser utilizado. Em particular, o modelo MotherBoard fornece bases para o desenvolvimento de serviços de *middleware* (como os de transação) flexíveis e configuráveis conforme as características de ambientes dinâmicos como a computação móvel.

Este trabalho apresentou um estudo dos principais modelos de transação para a computação móvel destacando algumas de suas contribuições e limitações. Estas limitações podem ser utilizadas como base em diversos novos trabalhos a serem realizados no Departamento de Computação (DCOMP) da Universidade Federal de Sergipe incluindo Trabalhos de Conclusão de Curso de Graduação, de Iniciação Científica e Dissertações de Mestrado.

Atualmente, o DCOMP já possui trabalhos concluídos e em andamento que aborda o tema apresentado neste capítulo. Estes incluem:

- **O projeto de Iniciação Científica “MiddleC – Apoio ao Desenvolvimento de Middlewares para Dispositivos Móveis”.** Como parte deste projeto, o modelo de componentes OpenCOM, com base no qual a plataforma MotherFrame foi desenvolvida, está atualmente sendo portado para a plataforma Android (Android, 2011). Essa é uma das etapas iniciais a serem cumpridas com o objetivo de se obter um suporte mais adequado ao projeto e desenvolvimento de *middlewares* e serviços configuráveis para dispositivos móveis.
- **Os trabalhos de conclusão de curso de graduação intitulados “Uma Plataforma de Agentes Móveis para Dispositivos Portáteis Utilizando J2ME” e “Mobilidade de Código em Dispositivos Móveis com Java ME”.** Esses trabalhos estudaram e desenvolveram uma plataforma de agentes móveis capaz de mover código e estado de aplicações Java em execução de um dispositivo móvel para outro. Seus resultados fornecem bases para o desenvolvimento de *middlewares* e serviços configuráveis capazes de terem seus componentes realocados dinamicamente entre os dispositivos móveis como parte de processos de reconfiguração dinâmica.

Vários trabalhos futuros podem ser desenvolvidos como atividades de pesquisa no DCOMP. Esses trabalhos incluem:

- **Propor serviços de transações flexíveis e configuráveis com base nos requisitos das aplicações e dos dispositivos envolvidos nos ambientes de computação móvel.** Para alcançar esse objetivo, o modelo *motherboard* poderia ser utilizado como base para a especificação de serviços baseados em componentes. Com isso, estes serviços poderiam ser compostos estaticamente e reconfigurados dinamicamente para refletir diferentes níveis de flexibilização das propriedades ACID ou de outros requisitos envolvendo o serviço.
- **Estender o modelo de componente OpenCOM e a plataforma MotherFrame para permitir conexão entre componentes localizados em máquinas diferentes.** Apesar do OpenCOM e da MotherFrame terem sido projetados para o desenvolvimento de *middleware*, eles não permitem uma conexão transparente entre componentes distribuídos do mesmo. Como uma das propostas do conceito de *middleware* é oferecer serviços distribuídos, sem esse tipo de suporte, desenvolver esses serviços envolveria um considerável esforço extra do desenvolvedor com implementação dos processos de comunicação entre as partes distribuídas.
- **Portar a plataforma MotherFrame para o Android. Essa etapa completa a tarefa do projeto de iniciação científica descrito acima que trata de portar o OpenCOM para o Android.** Como a plataforma MotherFrame usa o modelo de componentes OpenCOM, com ambas portadas para o Android, ter-se-ia um suporte adequado para desenvolvimento de *middlewares* para dispositivos móveis capaz de prover as abstrações apresentadas pelo modelo *motherboard*. A solução OpenCOM + MotherFrame + Android poderia ser usada como base para outros trabalhos que visam o desenvolvimento de serviços adequados a ambientes de computação móvel.

- **Propor e desenvolver meios mais simples de reconfiguração de *middlewares* além do método tradicional via linguagem de programação.** Uma dessas formas pode ser via interface gráfica. Com ela, *middlewares* poderiam ser compostos e reconfigurados dinamicamente através de elementos gráficos que poderiam representar componentes e conexões. Uma outra alternativa é a utilização de notações declarativas via XML. Essas notações poderiam ser criadas e submetidas dinamicamente a um interpretador que faria as alterações descritas no *middleware* sem a necessidade de criar e compilar código fonte em linguagem de programação.

Referências Bibliográficas

- ADYA, A. *Weak consistency: A generalized theory and optimistic implementations for distributed transactions*. Technical report, Massachusetts Institute of Technology, Cambridge, MA, USA, 1999.
- ANDROID. *Android Developers*. Disponível em: <http://developer.android.com/> Acesso em: 15 de Junho de 2011.
- ARNTSEN, A. e Karlsen, R. Dynamic transaction service composition. In *Proceedings of the IADIS International Conference on Applied Computing*, 2007.
- BARGA, R. e Pu, C. Reflection on a legacy transaction processing monitor. In *Proceedings of Reflection '96*, San Francisco, CA, USA, April 1996.
- BLAIR, G. *et al.* An architecture for next generation middleware. In *Proceedings of the International Conference on Distributed Systems Platforms and Open Distributed Processing*. Springer-Verlag, 1998.
- BLAIR, G. *et al.* The design and implementation of OpenORB v2. *IEEE DS Online – Special Issue on Reflective Middleware*, 2(6), 2001.
- BLAIR, G. *et al.* Reflection, self-awareness and self-healing in OpenORB. In *WOSS '02: Proceedings of the first workshop on Self-healing systems*, pages 9-14, New York, NY, USA, 2002.
- CHRYSANTHIS, P. e Ramamritham, K. Synthesis of extended transaction models using ACTA. *ACM Trans. Database Syst.*, 19(3):450-491, 1994.
- CLARKE, M. *et al.* An efficient component model for the construction of adaptive middleware. In *Proceedings of the IFIP/ACM International Conference on Distributed Systems Platforms Heidelberg*, pages 160-178, London, UK, 2001.
- DIRCKZE, R. e Gruenwald, L. A pre-serialization transaction management technique for mobile multidatabases. *MONET-Mobile Networks and Applications*, 5:311-321, 2000.
- DUNHAM, M. Helal, A. e Balakrishnan, S. A mobile transaction model that captures both the data and movement behavior. *Mob. Netw. Appl.*, 2(2):149-162, 1997.
- GRACE, P., Blair, G. e Samuel, S. A reflective framework for discovery and interaction in heterogeneous mobile environments. *SIGMOBILE Mob. Comput. Commun. Rev.*, 9(1):2-14, 2005.
- GRAY, J. *et al.* Granularity of locks and degrees of consistency in a shared database. In *IFIP Working Conference on Modeling in Data Base Management Systems*, pages 365-394, 1976.
- GRAY, J. e Reuter, A. *Transaction Processing: Concepts and Techniques*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1992.
- JAKOBSEN, A. e Karlsen, R. ReflecTS: a Flexible transaction service framework. In *ARM '05: Proceedings of the 4th workshop on Reflective and adaptive middleware systems*, New York, NY, USA, 2005.
- KARLSEN, R. e Jakobsen, A. Transaction service management - an approach towards a reflective transactional service. In *Proceedings of the 2nd Workshop on Reflective and Adaptive*

- Middleware*, Rio de Janeiro, Brazil, 2003.
- KON, F. *et al.* Monitoring, Security, and Dynamic Configuration with the dynamicTAO Reflective ORB. In *Proceedings of the IFIP/ACM International Conference on Distributed Systems Platforms and Open Distributed Processing (Middleware'2000)*, pages 121-143, New York, April 2000.
- KON, F. *et al.* The case for reflective middleware. *Commun. ACM*, 45(6):33-38, 2002.
- KORDON, F. e Pautet, L. Toward nex-generation middleware? *IEEE Distributed Systems Online*, 6(3):2, 2005.
- KU, K. e Kim. Y. Mofex transaction model for mobile heterogeneous multidatabase systems. In *IEEE Workshop on Research Issues in Data Engineering*, San Diego, USA, 2000.
- LEDoux, T. OpenCorba: A reflective open broker. *Proceedings of the 2nd International Conference on Reflection '99*, 1616:197-??, 1999.
- LEE, M. e Helal, S. Hicomo: High commit mobile transactions. *Distrib. Parallel Databases*, 11(1):73-92, 2002.
- LU, Q. e Satyanarayanan, M. Isolation-only transactions for mobile computing. *SIGOPS Oper. Syst. Rev.*, 28(2):81{87, 1994.
- MOSS, J. Nested transactions: an approach to reliable distributed computing. *Massachusetts Institute of Technology*, Cambridge, MA, USA, 1985.
- MUMMERT, L., Ebling, M. e Satyanarayanan, M. Exploring weak connectivity for mobile file access. In *Proceedings of the 15th ACM Symposium on Operation Systems Principles*, Colorado, 1998.
- OBJECT MANAGEMENT GROUP. *Transaction service specification - version 1.4*, September 2003.
- PROCHAZKA, M. Advanced transactions in Enterprise JavaBeans. In *EDO '00: Revised Papers from the Second International Workshop on Engineering Distributed Objects*, pages 215-230, London, UK, 2001.
- RAMAMPIARO, H. e Nygard, M. CAGISTrans: Providing adaptable transactional support for cooperative work - an extended treatment. *Inf. Tech. and Management*, 5(1-2):23-64, 2004.
- ROCHA, T e Toledo, M. *Estudo de modelos de transação para o ambiente de computação móvel*. Technical Report IC-05-025, Instituto de Computação - UNICAMP, Setembro 2005.
- ROCHA, T. *et al.* Promoting levels of openness on component-based adaptable middleware. In *Proceedings of the 6th Workshop on Adaptive and Reflective Middleware*, Newport Beach, United States, 2007.
- ROCHA, T. e Toledo, M.B. Managing complexity in open adaptive middleware. In *Proceedings of the 2008 International Symposium on Scientific and Engineering Computing*, São Paulo, Brazil, 2008.
- ROCHA, T. *Serviços de Transações Abertos para Ambientes Dinâmicos*. Tese de Doutorado, Instituto de Computação, Universidade Estadual de Campinas, Agosto, 2008.
- SANTOS, N., Veiga, L. e Ferreira, P. Transaction policies for mobile networks. In *POLICY '04: Proceedings of the Fifth IEEE International Workshop on Policies for Distributed Systems and Networks*, page 55, Washington, DC, USA, 2004.
- SATYANARAYANAN, M. The evolution of CODA. *ACM Trans. Comput. Syst.*, 20(2):85-124, 2002.
- SERRANO-ALVARADO, P. *et al.* Adaptable mobile transactions. In *Proceedings of the 29th VLDB Conference*, Berlin, Germany, 2003.
- SINGHAI, A. *Quarterware: A Middleware Toolkit of Software Risc Components*. PhD Thesis, University of Illinois at Urbana-Champaign, 1999.
- SUN MICROSYSTEMS. *Java Transaction Service (JTS) version 1.0*, December 1999.

- Sun Microsystems. *Java Transaction API (JTA) version 1.0.1*, April 1999.
- VENKATRAMAN, S. Mobile computing models - are they meeting the mobile computing challenges? *Association for Computing Machinery, New Zealand Bulletin*, 1(1), 2005.
- X/OPEN COMPANY. *Distributed transaction specification: Reference model, version 3*, February 1996.
- X/OPEN COMPANY. *Distributed transaction specification: The TX (transaction demarcation) specification*, April 1995.
- X/OPEN COMPANY. *Distributed transaction specification: The XA specification*, December 1991.
- WALBORN, G. e Chrysanthis, P. Transaction processing in PRO-MOTION. In *SAC '99: Proceedings of the 1999 ACM symposium on Applied computing*, pages 389-398, New York, NY, USA, 1999.
- WU, Z. Reflective Java and a reflective component-based transaction architecture. In *Proceedings of the ACM OOPSLA'98 Workshop on Reflective Programming in Java and C++*, October 1998.
- ZARRAS, A. e Issarny, V. A framework for systematic synthesis of transactional middleware. In *Proceedings of MIDDLEWARE '98 - IFIP/ACM International Conference on Distributed Systems Platforms and Open Distributed Processing*, pages 257-272, 1998.