



ANAIS DOS ARTIGOS DA V ESCOLA POTIGUAR DE COMPUTAÇÃO E SUAS APLICAÇÕES

De 05 a 09 de novembro de 2012, no Centro de Eventos
do Hotel Praiamar - Natal/RN

Editado por
Sílvio Roberto Fernandes de Araújo (UFERSA)
Marcelo Ferreira Siqueira (UFRN)

Sumário

| | |
|--|-----|
| Prefácio | iii |
| Organização..... | iv |
| Aplicando Controle de Energia em Redes de Sensores Sem Fio..... | 1 |
| Ontologias Educacionais | 7 |
| BestResources: Ferramenta de Apoio à Alocação de Recursos Humanos para Projetos de TI no dotProject..... | 13 |
| Protocolo Zigbee em Redes de Comunicação Intraveicular..... | 19 |
| Uma avaliação da técnica de Casamento de Padrões na construção de Agentes Conversacionais | 25 |
| Aplicação de ferramentas e técnicas de mineracao de dados em radiossonagens meteorológicas..... | 31 |
| AI Quest: Um jogo em HTML5 para aprendizado de algoritmos | 37 |
| Comparação do Desempenho entre Aplicações em C e em Python Utilizando a Biblioteca Ctypes..... | 45 |
| Gerador de CDFG com análise de dependência de dados a partir de código C..... | 61 |

Prefácio

Sejam bem-vindos à quinta edição da Escola POtiguar de Computação e Suas Aplicações (EPOCA 2012). A EPOCA é um evento regional que tem como objetivo promover a interação e aproximação de estudantes, professores, pesquisadores e profissionais de Computação do Estado do Rio Grande do Norte (RN) e vizinhança.

A EPOCA teve sua primeira edição em 2008 e foi realizada, simultaneamente, em três cidades do RN, Natal, Currais Novos e Mossoró, com o auxílio de vídeo-conferência. A programação do evento incluiu minicursos, palestras, mesa-redonda e mostra de iniciação científica. O evento contou com 250 participantes e teve como tema central “Metrópole Digital e a interação universidade-empresa-sociedade”. A EPOCA foi reeditada anualmente, de 2009 a 2011, em Natal ou Mossoró, com aproximadamente as mesmas atividades. Em todas as edições, houve o envolvimento da Universidade Federal do Rio Grande do Norte (UFRN), Universidade Estadual do Rio Grande do Norte (UERN), Instituto Federal do Rio Grande do Norte (IFRN) e Universidade Federal Rural do Semi-Árido (UFERSA).

Em 2012, a EPOCA ocorreu no Praiamar Natal Hotel, em Natal (RN), de 5 a 9 de novembro. O evento consistiu em sessões técnicas, minicursos, palestras e uma mesa-redonda que debateu o tema “empreendedorismo e inovação em tecnologia da informação”.

Natal, novembro de 2012

Edgard Correa e Marcio Kreutz

Coordenadores Gerais do EPOCA 2012

Organização

Coordenadores Gerais

Edgard de Faria Correa (Universidade Federal do Rio Grande do Norte – UFRN)
Marcio Eduardo Kreutz (Universidade Federal do Rio Grande do Norte – UFRN)

Coordenadora Local

Monica Magalhães Pereira (Universidade Federal do Rio Grande do Norte – UFRN)

Coordenadores do Comitê de Programa

Silvio Roberto Fernandes de Araújo (Universidade Federal Rural do SemiÁrido – UFERSA)
Marcelo Ferreira Siqueira (Universidade Federal do Rio Grande do Norte – UFRN)

Comitê de Programa

Adilson Lopes (Universidade Federal do Rio Grande do Norte – UFRN)
Adriana Takahashi (Universidade do Estado do Rio Grande do Norte – UERN)
Adrião Duarte Dória Neto (Universidade Federal do Rio Grande do Norte – UFRN)
Alberto Signoretti (Universidade Federal do Rio Grande do Norte – UFRN)
Alessandro Girardi (Universidade Federal do Pampa - UNIPAMPA)
Amanda Oliveira (Universidade Federal Rural do SemiÁrido – UFERSA)
Amaury Castro Jr. (Universidade Federal de Mato Grosso do Sul – UFMS)
Andre Pereira (Universidade Federal Rural do SemiÁrido – UFERSA)
Andre Silva (Universidade do Estado do Rio Grande do Norte – UERN)
Angélica Castro (Universidade Federal Rural do SemiÁrido – UFERSA)
Augusto Neto (Universidade Federal do Ceará – UFC)
Bruno Oliveira (Universidade do Estado do Rio Grande do Norte – UERN)
Bruno Motta de Carvalho (Universidade Federal do Rio Grande do Norte – UFRN)
Camila Araújo Sena (Universidade do Estado do Rio Grande do Norte – UERN)
Carlos Montez (Universidade Federal de Santa Catarina – UFSC)
Carlos Gustavo Rocha (Instituto Federal do Rio Grande do Norte – IFRN)
Cristiano Both (Universidade Federal do Rio Grande do Sul – UFRGS)
Danielle Casillo (Universidade Federal Rural do SemiÁrido – UFERSA)
Danniel Lopes (Universidade Federal Rural do SemiÁrido – UFERSA)
Debora Paiva (Universidade Federal de Mato Grosso do Sul – UFMS)
Demóstenes Sena (Instituto Federal do Rio Grande do Norte – IFRN)
Diogo Pedrosa (Universidade Federal do Rio Grande do Norte – UFRN)
Diogo Henrique Bezerra (Universidade Federal da Paraíba – UFPB)
Erica Esteves C. de Miranda (Universidade Federal do Rio do Grande do Norte – UFRN)
Everton Cavalcante (Universidade Federal do Rio do Grande do Norte – UFRN)
Fabrício Pires (Universidade Federal do Rio do Grande do Norte – UFRN)
Flávia Coelho (Universidade Federal Rural do SemiÁrido – UFERSA)
Francisco Mendes Neto (Universidade Federal Rural do SemiÁrido – UFERSA)
Frederico Lopes (Universidade Federal do Rio do Grande do Norte – UFRN)
Gibeon Júnior (Universidade Federal do Rio do Grande do Norte – UFRN)
Gustavo Girão (Universidade Federal do Rio Grande do Sul – UFRGS)
Hana Rubinsztein (Universidade Federal de Mato Grosso do Sul –UFMS)
Helcio Silva (Universidade Federal Rural do SemiÁrido – UFERSA)
Heliana Soares (Universidade Federal do Rio do Grande do Norte – UFRN)
Júlio Mattos (Universidade Federal de Pelotas – UFPel)
João Xavier-Júnior (Universidade Federal do Rio do Grande do Norte – UFRN)
Jorgiano Vidal (Instituto Federal do Rio Grande do Norte – IFRN)

| | |
|---------------------------|--|
| Judson Santiago | (Universidade Federal Rural do SemiÁrido – UFERSA) |
| Leonardo Casillo | (Universidade Federal Rural do SemiÁrido – UFERSA) |
| Leonardo Cunha de Miranda | (Universidade Federal do Rio Grande do Norte – UFRN) |
| Leonardo Lucena | (Instituto Federal do Rio Grande do Norte – IFRN) |
| Leonardo Minora | (Instituto Federal do Rio Grande do Norte – IFRN) |
| Lisane Brisolara | (Universidade Federal de Pelotas – UFPel) |
| Luciane Fraga | (Universidade Federal do Rio Grande do Norte – UFRN) |
| Luiz Guedes | (Universidade Federal do Rio Grande do Norte – UFRN) |
| Marcelo Siqueira | (Universidade Federal do Rio Grande do Norte – UFRN) |
| Marcelo Augusto Fernandes | (Universidade Federal do Rio Grande do Norte – UFRN) |
| Marcos Pinheiro | (Universidade Federal do Rio Grande do Norte – UFRN) |
| Marcos Paulo Moro | (Universidade Federal da Grande Dourados – UFGD) |
| Max Silveira | (Universidade Federal do Rio Grande do Norte – UFRN) |
| Márjory Da Costa-Abreu | (Universidade Federal do Rio Grande do Norte – UFRN) |
| Moisés Souto | (Universidade Federal do Rio Grande do Norte – UFRN) |
| Monica Pereira | (Universidade Federal do Rio Grande do Norte – UFRN) |
| Raimundo Costa Filho | (Universidade Federal Rural do SemiÁrido – UFERSA) |
| Renato Ishii | (Universidade Federal de Mato Grosso do Sul – UFMS) |
| Rivalino Matias Jr. | (Universidade Federal de Uberlândia – UFU) |
| Ronaldo Ferreira | (Universidade Federal de Mato Grosso do Sul – UFMS) |
| Rummenigge Dantas | (Universidade Federal do Rio Grande do Norte – UFRN) |
| Silvio Fernandes | (Universidade Federal Rural do SemiÁrido – UFERSA) |
| Tadeu Oliveira | (Instituto Federal do Rio Grande do Norte – IFRN) |
| Tatiana Tavares | (Universidade Federal da Paraíba – UFPB) |
| Uirá Kulesza | (Universidade Federal do Rio Grande do Norte – UFRN) |
| Valéria Soares | (Universidade Federal da Paraíba – UFPB) |
| Victor Oliveira | (Instituto Federal da Paraíba – IFPB) |

Revisores

| | |
|--------------------------|--|
| Adilson Lopes | (Universidade Federal do Rio Grande do Norte – UFRN) |
| Adriana Takahashi | (Universidade do Estado do Rio Grande do Norte – UERN) |
| Adrião Duarte Dória Neto | (Universidade Federal do Rio Grande do Norte – UFRN) |
| Alberto Signoretti | (Universidade Federal do Rio Grande do Norte – UFRN) |
| Alessandro Girardi | (Universidade Federal do Pampa - UNIPAMPA) |
| Amanda Oliveira | (Universidade Federal Rural do SemiÁrido – UFERSA) |
| Amaury Castro Jr. | (Universidade Federal de Mato Grosso do Sul – UFMS) |
| Andre Pereira | (Universidade Federal Rural do SemiÁrido – UFERSA) |
| Andre Silva | (Universidade do Estado do Rio Grande do Norte – UERN) |
| Angélica Castro | (Universidade Federal Rural do SemiÁrido – UFERSA) |
| Augusto Neto | (Universidade Federal do Ceará – UFC) |
| Bruno Oliveira | (Universidade do Estado do Rio Grande do Norte – UERN) |
| Bruno Motta de Carvalho | (Universidade Federal do Rio Grande do Norte – UFRN) |
| Camila Araújo Sena | (Universidade do Estado do Rio Grande do Norte – UERN) |
| Carlos Montez | (Universidade Federal de Santa Catarina – UFSC) |
| Carlos Gustavo Rocha | (Instituto Federal do Rio Grande do Norte – IFRN) |
| Cristiano Both | (Universidade Federal do Rio Grande do Sul – UFRGS) |
| Danielle Casillo | (Universidade Federal Rural do SemiÁrido – UFERSA) |
| Danniel Lopes | (Universidade Federal Rural do SemiÁrido – UFERSA) |
| Debora Paiva | (Universidade Federal de Mato Grosso do Sul – UFMS) |
| Demóstenes Sena | (Instituto Federal do Rio Grande do Norte – IFRN) |
| Diogo Pedrosa | (Universidade Federal do Rio Grande do Norte – UFRN) |
| Diogo Henrique Bezerra | (Universidade Federal da Paraíba – UFPB) |

| | |
|-----------------------------|--|
| Erica Esteves C. de Miranda | (Universidade Federal do Rio Grande do Norte – UFRN) |
| Everton Cavalcante | (Universidade Federal do Rio Grande do Norte – UFRN) |
| Fabrício Pires | (Universidade Federal do Rio Grande do Norte – UFRN) |
| Flávia Coelho | (Universidade Federal Rural do SemiÁrido – UFERSA) |
| Francisco Mendes Neto | (Universidade Federal Rural do SemiÁrido – UFERSA) |
| Frederico Lopes | (Universidade Federal do Rio Grande do Norte – UFRN) |
| Gianna Araújo | (Universidade Federal do Rio Grande do Norte – UFRN) |
| Gibeon Júnior | (Universidade Federal do Rio Grande do Norte – UFRN) |
| Gustavo Girão | (Universidade Federal do Rio Grande do Sul – UFRGS) |
| Hana Rubinsztejn | (Universidade Federal de Mato Grosso do Sul –UFMS) |
| Helcio Silva | (Universidade Federal Rural do SemiÁrido – UFERSA) |
| Heliana Soares | (Universidade Federal do Rio Grande do Norte – UFRN) |
| Júlio Mattos | (Universidade Federal de Pelotas – UFPel) |
| João Xavier-Júnior | (Universidade Federal do Rio Grande do Norte – UFRN) |
| Jorgiano Vidal | (Instituto Federal do Rio Grande do Norte – IFRN) |
| Judson Santiago | (Universidade Federal Rural do SemiÁrido – UFERSA) |
| Leonardo Casillo | (Universidade Federal Rural do SemiÁrido – UFERSA) |
| Leonardo Cunha de Miranda | (Universidade Federal do Rio Grande do Norte – UFRN) |
| Leonardo Lucena | (Instituto Federal do Rio Grande do Norte – IFRN) |
| Leonardo Minora | (Instituto Federal do Rio Grande do Norte – IFRN) |
| Lisane Brisolara | (Universidade Federal de Pelotas – UFPel) |
| Luciane Fraga | (Universidade Federal do Rio Grande do Norte – UFRN) |
| Luiz Guedes | (Universidade Federal do Rio Grande do Norte – UFRN) |
| Marcelo Siqueira | (Universidade Federal do Rio Grande do Norte – UFRN) |
| Marcelo Augusto Fernandes | (Universidade Federal do Rio Grande do Norte – UFRN) |
| Marconi Rodrigues | (Universidade Federal do Rio Grande do Norte – UFRN) |
| Marcos Pinheiro | (Universidade Federal do Rio Grande do Norte – UFRN) |
| Marcos Paulo Moro | (Universidade Federal da Grande Dourados – UFGD) |
| Max Silveira | (Universidade Federal do Rio Grande do Norte – UFRN) |
| Márjory Da Costa-Abreu | (Universidade Federal do Rio Grande do Norte – UFRN) |
| Moisés Souto | (Universidade Federal do Rio Grande do Norte – UFRN) |
| Monica Pereira | (Universidade Federal do Rio Grande do Norte – UFRN) |
| Rafael Aroca | (Universidade Federal do Rio Grande do Norte – UFRN) |
| Rafael Beserra Gomes | (Universidade Federal do Rio Grande do Norte – UFRN) |
| Raimundo Costa Filho | (Universidade Federal Rural do SemiÁrido – UFERSA) |
| Renato Ishii | (Universidade Federal de Mato Grosso do Sul – UFMS) |
| Rivalino Matias Jr. | (Universidade Federal de Uberlândia – UFU) |
| Ronaldo Ferreira | (Universidade Federal de Mato Grosso do Sul – UFMS) |
| Rummenigge Dantas | (Universidade Federal do Rio Grande do Norte – UFRN) |
| Samuel Azevedo | (Universidade Federal do Rio Grande do Norte – UFRN) |
| Sarah Thomaz | (Universidade Federal do Rio Grande do Norte – UFRN) |
| Silvio Fernandes | (Universidade Federal Rural do SemiÁrido – UFERSA) |
| Tadeu Oliveira | (Instituto Federal do Rio Grande do Norte – IFRN) |
| Tatiana Tavares | (Universidade Federal da Paraíba – UFPB) |
| Uirá Kulesza | (Universidade Federal do Rio Grande do Norte – UFRN) |
| Valéria Soares | (Universidade Federal da Paraíba – UFPB) |
| Victor Oliveira | (Instituto Federal da Paraíba – IFPB) |

Aplicando Controle de Energia em Redes de Sensores Sem Fio

Ismael I. Almeida, Linniker R. L. Lima, Marianny F. S. Mariano, Marlos A. S. Lima, Rommel W. de Lima

Departamento de Informática

Universidade do Estado do Rio Grande do Norte (UERN) – Mossoró, RN – Brazil

{ysmaelsr12, link.ramon, mariannyfidelis, marlos.maur,
rommel.lima}@gmail.com

Abstract. One of the most common problems in research involving Wireless Sensor Nodes is the energy consumption. The main focus of a WSN project is find ways to ensure the longest lifetime possible of the network. Thus, this article aims to describe an application developed in nesC using MICAZ sensors. This application allows nodes to operate only while the lightness and the battery level are above a predetermined value, saving energy in proportion to the nodes become idle. So, this application enables the management of certain features of the Wireless Sensor Networks, especially energy.

Resumo. Um dos problemas mais recorrentes nas pesquisas envolvendo Redes de Sensores sem Fio - RSSF é o consumo de energia. O foco principal de um projeto de RSSF é encontrar formas de garantir o maior tempo de vida possível da rede. Pensando nisso, este artigo tem como objetivo descrever uma aplicação desenvolvida em nesC utilizando sensores MICAZ. Esta aplicação permite que os nodos operem somente enquanto a luminosidade do ambiente e o nível de energia das baterias estiverem acima de um valor pré-estabelecido, visando a economia de energia na proporção que os nodos tornam-se ociosos. Desta forma, esta aplicação possibilita o gerenciamento de determinados recursos de uma Rede de Sensores sem Fio, principalmente energia.

1. Introdução

As Redes de Sensores sem Fio (RSSF) são caracterizadas pela utilização de nodos sensores com capacidade de se comunicarem entre si [10]. Tais nodos são elementos computacionais dotados de memória, processamento, energia e comunicação sem fio com capacidade de sensoriamento de diversos parâmetros físicos, tais como luminosidade, pressão, temperatura, umidade, som, gás, proximidade dentre outros. Os nodos inseridos no ambientes monitoram e coletam informações de estados e/ou fenômenos, tornando possível o monitoramento dos mais variados ambientes, inclusive ambientes inóspitos. Muitas RSSFs possuem *atuadores*, dispositivos que interagem com um processo físico a partir de sinais de entrada gerados pelo processamento dos dados coletados, lhes permitindo controlar diretamente o ambiente alvo [2]. Essas redes encontram uma diversidade de notáveis aplicações, como o controle de tráfego, assistência médica, monitoramento ambiental e de vulcão ativo, rastreamento de alvos, agricultura de precisão, monitoramento da atividade humana e mineração subterrânea, além de muitas outras aplicações [2].

Redes de sensores tem o potencial de serem empregadas em áreas como ambiente a fim de monitorar variáveis ambientais, tráfego com objetivo de monitorar tráfegos de veículos, segurança, medicina/biologia, militar com o objetivo de detectar, por exemplo, movimentos inimigos, explosões, bem como prover algum mecanismo de controle. De forma genérica, RSSF podem ser utilizadas em aplicações de segurança e monitoramento, controle, atuação de sistemas complexos e monitoramento de ambientes internos ou externos [6]. Assim, é possível visualizar o potencial científico e econômico das RSSF, tendo em vista a grande quantidade de aplicações que podem ser providas, como também oportunidades de pesquisa.

Os nodos sensores são dispositivos que apresentam diversas restrições. Em relação às redes tradicionais, é possível destacar o baixo poder de processamento, memória limitada e autonomia da energia. Na maioria das aplicações destas redes, não é impossível fazer a substituição da fonte de energia de seus nodos sensores. Assim, a capacidade de sensoriamento, processamento e comunicação são dependentes da disponibilidade de tal recurso. Dessa forma, técnicas, métodos, políticas de controle de hardware que visem otimizar o consumo de energia em nodos sensores é extremamente fundamental para maximizar o tempo de vida da rede.

Este artigo tem como objetivo prover uma aplicação de gerenciamento de energia, em uma RSSF, que seja capaz de controlar o comportamento dos nodos em resposta a determinados eventos, tornando-os em estado de latência (IDLE) com o intuito de reduzir o consumo de energia dos nodos e maximizar o tempo de vida da rede. Trabalhos deste tipo possuem grande importância devido à recorrência de problemas relativos ao consumo, economia e armazenamento de energia em RSSF. As próximas seções estão organizadas como a seguir. Na seção 2, a arquitetura de uma RSSF é descrita. A seção 3 descreve a solução proposta. Em seguida, a seção 4 apresenta os resultados obtidos. As conclusões e trabalhos futuros estão contidas na última seção.

2. Arquitetura de Redes Sensores Sem Fio

As RSSF são redes *Ad Hoc* compostas por um grande número de nodos sensores [10] capazes de interagir com o ambiente objetivando coletar informações e controlá-lo de acordo com determinados eventos. Os nodos são equipados com capacidade de sensoriamento e atuação, para medir/influenciar o ambiente, processamento e comunicação (sem fio) das informações coletadas [4]. Estes são implantados, de forma aleatória ou regular, no ambiente a ser monitorado. Despertam, descobrem sua posição por meio de algoritmos de localização, se auto-organizam e, então, estabelecem canais de comunicação dinamicamente.

Uma das principais características das RSSF é a ausência de uma infraestrutura prévia de comunicação, ao contrário das redes de telefonia, por exemplo, exigindo capacidades de auto-organização em uma rede. Por serem tipicamente redes *ad hoc*, as RSSF apresentam topologia dinâmica, uma vez que os nodos podem ficar indisponíveis, por exemplo, por motivos de defeito, falta de energia, ataques à segurança, falha de comunicação e nos componentes, alterando a topologia da rede. Outro aspecto fundamental destas redes é que o fluxo de dados é predominantemente unidirecional, onde a disseminação dos dados colhidos ocorre dos nodos sensores em direção ao(s)

ponto(s) de acesso - PA¹ utilizando nodos intermediários como roteadores. Essa comunicação multi-hop é necessária, pois muitas vezes não existe um canal de comunicação direto entre os nodos sensores e o PA. Além disso, as capacidades de transmissão limitadas dos transceptores de rádio frequência e necessidade de economia de energia exige a utilização desse mecanismo de comunicação.

Um ponto importante de tais redes é a agregação dos dados e a possibilidade de organização em grupos de nodos (*clusters*). Conhecer o estado de uma região é mais importante que conhecer o estado de cada nó, além disso, o grande número de nodos torna a coleta individual inviável. A agregação dos dados reduz a quantidade de bits transmitidos e aumenta a economia de energia. A organização em grupos permite a realização desta atividade. Em cada grupo existe um líder que recebe os dados coletados, realiza o processamento local e envia as informações ao ponto de acesso.

3. Aplicação

3.1 Recursos utilizados

A aplicação foi desenvolvida para ser executada sobre um kit de sensores MICAz da família Mica Motes da empresa *Crossbow Technology*. Os nodos são dotados com placa de sensores (*sensor board*) MTS400CC, que suporta os nodos sensores IRIS, MICAz e MICA2, com capacidades para coletar luz ambiente, umidade relativa, temperatura, acelerômetro 2D e pressão barométrica [1].

Para a implementação dos serviços utilizou-se o sistema operacional TinyOS-2.x e a linguagem nesC 1.1. O TinyOS é um sistema operacional desenvolvido pela UC Berkeley para suas várias plataformas mote (nó sensor) [7]. É um sistema cujo objetivo é o uso racional de pouca memória, é baseado em componentes, onde a interação destes se baseia em eventos assíncronos, de forma que a ligação desses componentes formam um programa, similarmente ao VHDL [9]. Tal SO foi reimplementado em nesC [3], e a programação dos nodos é feita utilizando essa linguagem. Seu código é aberto (*open source*) e pode ser obtido em [9]. Já a linguagem nesC é uma extensão de C projetada para incorporar os conceitos estruturais e modelos de execução do TinyOS [3]. Os principais conceitos acerca da nesC são [3]:

- Separação de construção e composição. Os programas são construídos a partir de *componentes*, que são montados (*wired*) para formar os programas inteiros. Componentes definem dois escopos, um para a sua especificação (contendo os nomes de suas *instâncias de interface*) e outro para a sua implementação. Os componentes possuem concorrência interna em forma de *tarefas*;
- Especificação do comportamento do componente em termos de conjunto de *interfaces*. As interfaces podem ser fornecidas - representam as funcionalidades que o componente oferece ao seu usuário - ou utilizadas pelo componente. Estas últimas representam as funcionalidades que o componente precisa para executar seu trabalho. Os componentes são ligados estaticamente uns aos outros por meio

¹ Nô com maiores capacidades de alcance de rádio e de energia, com conhecimento global da rede e não possui sensores. Sua finalidade é coletar as informações da rede e transmitir a um observador externo, conectando, assim, a rede com o mundo exterior. Pode ser uma estação base, um nó sink ou um gateway.

das suas interfaces. Isso aumenta a eficiência do tempo de execução, incentiva um projeto robusto, e permite uma melhor análise estática do programa;

- Interfaces são bidirecionais. Elas permitem que uma única interface represente uma interação complexa entre vários componentes, pois especificam um conjunto de funções a serem implementadas pelo provedor da interface (*comandos*) e um conjunto para ser implementado pelo usuário da interface (*eventos*). Isto é necessário porque os comandos e eventos do TinyOS são assíncronos [7], além disso, os comandos são sem bloqueio, pois apenas acrescentam na pilha de execução (*downcalls*), enquanto os eventos são *upcalls*, reduzindo a pilha [5]. Assim, os eventos sinalizam a conclusão de um comando por meio de sinais (por exemplo, a conclusão de um comando *enviar* é sinalizada por um evento *send done*);
- O modelo de concorrência de nesC baseia-se na corrida para conclusão das tarefas, e em tratadores de interrupção que podem interromper cada uma das tarefas, assim que sejam executadas. Quando um comando de uma tarefa é executado, um sinal é enviado ao evento associado para informar que a tarefa foi concluída.

3.2 Serviços Implementados

Por meio da linguagem nesC, foi possível implementar uma aplicação de monitoramento que possui um serviço responsável para controlar o hardware de comunicação via rádio dos nodos da rede. De forma que a transmissão de dados opere somente se o nível de energia das baterias e luminosidade do ambiente estiverem acima de um valor pré-estabelecido. Aplicações como esta são utilizadas em casos onde não há eventos a serem monitorados durante determinados horários do dia, fazendo que os nodos fiquem ociosos durante esse período, economizando energia. Pode ser utilizado, por exemplo, no monitoramento de animais que possuem hábitos noturnos e dormem durante o período diurno.

Para tal, o serviço realiza constantemente a leitura dos níveis de luz e de bateria. Por meio destes valores é possível determinar se alguma parte da rede está na iminência de falha ou enviar informações comprometidas, por falta de energia. Dessa forma é possível aplicar técnicas e políticas de gerenciamento da rede, como por exemplo, sincronizar “tempos de dormida” ou mudar o estado dos nodos, com base na disponibilidade de recurso e características do ambiente.

4. Resultados

Na execução da rede, quando o nível de bateria dos nodos atingirem o valor igual ou menor que 2.0 V (de uma autonomia total de 3.0 V) e o sensor de luminosidade registra a leitura igual ou menor que 300 lux, o rádio é desligado, porém as funções de processamento e sensoriamento continuam ativas. Outro dado coletado pelos sensores ambientais é a temperatura. Com o intuito de verificar a atuação do serviço desenvolvido, no sensor com ID 1 foi inserida uma fonte de alimentação (bateria) com um nível de energia próximo do limite de 2.0 V. Na figura 1 é possível visualizar as informações advindas de dois nodos sensores (ID: 1 e 2).

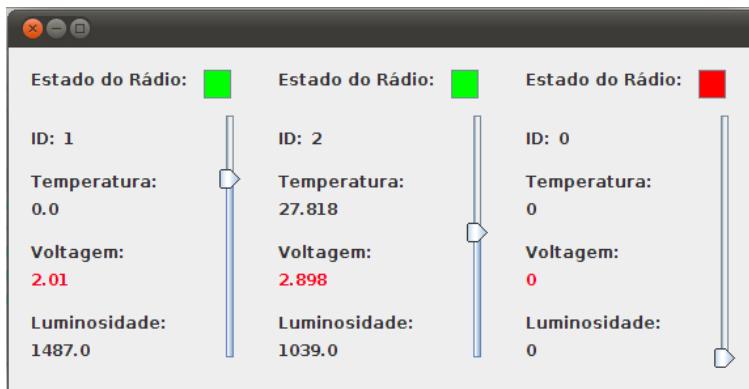


Figura 1: Exibição dos resultados do monitoramento de nodos sensores.

Já a figura 2 apresenta o momento em que ambos os níveis de energia e luminosidade atingem valores abaixo dos limites estabelecidos para a execução, fazendo com que o transmissor do nó (ID: 1) seja desligado. A partir deste estado o nodo deixa de enviar dados a estação base por um tempo definido na aplicação.

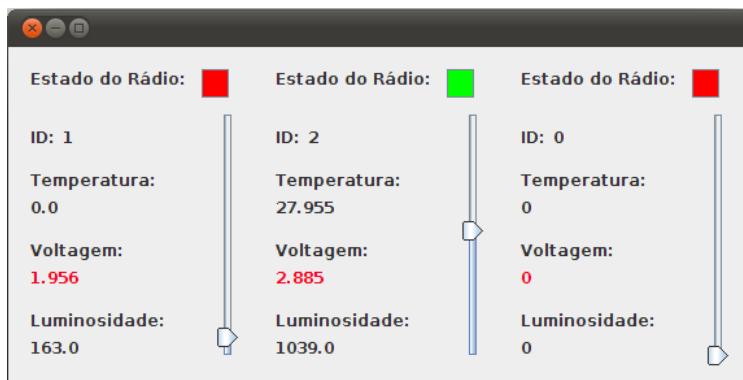


Figura 2: Momento em que os dados colhidos atingem os valores pré-estabelecidos.

Como resultados, obteve-se uma aplicação em nesC para coleta de dados do ambiente, um serviço de controle dos estados ligado/desligado do dispositivo de comunicação sem fio e um aplicativo funcional em Java (no gateway) para apresentar as informações ao usuário final por meio de interface gráfica. Esta aplicação descreve o comportamento dos estados do transceptor em cada nodo da RSSF, bem como relata os dados de sensoriamento a partir dos nodos. Um fenômeno que pode ser observado é com relação ao sensor de temperatura que deixa de funcionar corretamente quando a fonte de energia (baterias) do sensor atinge um nível muito baixo. Um fato como este pode auxiliar no tratamento de dados, oriundos da rede, de forma que não haja a interpretação errônea de certas informações advindas do nó sensor.

5. Conclusão

Com os resultados desta pesquisa foi possível constatar que a aplicação de políticas de utilização e gerenciamento de determinados hardwares dos nodos sensores podem representar uma alternativa para melhor utilização dos recursos de energia e principalmente maximizar o tempo de vida de uma RSSF. O serviço que foi desenvolvido possibilitou o controle dos estados ligado/desligado do dispositivo

responsável pela comunicação e transmissão de dados entre os nodos sensores. No entanto, existem outras abordagens que visam o controle dos estados ligado/desligado de um nó sensor como um todo, objetivando, em determinadas situações, alcançar maior economia de energia.

Para trabalhos futuros, aplicar uma política de eleição de líder de grupo em redes hierárquicas utilizando um mapa de energia, gerado a partir dos valores de energia dos sensores, que será armazenado no nó líder. Além disso, uma outra perspectiva direciona-se para as redes densas, de modo que um mecanismo de economia de energia pode ser aplicado por meio da alternância dos nodos ativos nos grupos, tendo como critério a proximidade entre si.

6. Referências

- [1] CROSSBOW. **MTS/MDA Sensor Board Users Manual.** Revision A, PN: 7430-0020-05, 2007.
- [2] DARGIE, Waltenegus; and POELLABAUER. **Fundamentals of Wireless Sensor Network: Theory and Practice.** Wiley Series on Wireless Communications and Mobile Computing, Wiley, 2010.
- [3] GAY, Luiz; LEVIS, Philip; CULLER, David and BREWER, Eric. **nesC 1.1 Language Reference Manual.** 2003.
- [4] KARL, Holger and WILLIG, Andreas. **Protocols and Architectures for Wireless Sensor Networks.** Wiley, 2005.
- [5] LEVIS, Philip and Gay, David. **TinyOS Programming.** Cambridge University Press: Cambridge, 2009.
- [6] LOUREIRO, Antonio F.; NOGUEIRA, José M. S.; RUIZ, Linnyer B.; MINI, Raquel A. F; NAKAMURA, Eduardo F. and FIGUEIREDO, Carlos M. S. **Redes de Sensores sem Fio.** Disponível em: www.dcc.ufmg.br/~loureiro/cm/docs/sbrc03.pdf. Acesso em junho de 2012.
- [7] MERLIN, Chris. **A Tutorial for Programming in TinyOS.** University of Rochester, 2009.
- [8] RUIZ, Linnyer B.; CORREIA, Luiz H. A.; VIEIRA, Luiz F.; MACEDO, Daniel F.; NAKAMURA, Eduardo F.; FIGUEIREDO, Carlos M. S.; VIEIRA, Marcos A. M.; BECHELANE, Eduardo H.; CAMARA, Daniel; LOUREIRO, ^ Antonio A. F.; NOGUEIRA, José M. S.; SILVA Jr., Diógenes C.; and FERNANDES, Antonio O. **Arquitetura para redes de sensores sem fio.** In *Minicursos do XXII Simpósio Brasileiro de Redes de Computadores, Gramado/RS*, 2004.
- [9] TINYOS. <http://www.tinyos.net/>. Acesso em junho de 2012.
- [10] VIEIRA, Marcos A.; VIEIRA, Luiz F.; RUIZ, Linnyer B.; LOUREIRO, Antonio A. and FERNANDES, Antonio O. **Como obter o mapa de energia em redes de sensores sem fio? Uma abordagem tolerante a falhas.** V WCSF - Workshop de Comunicação Sem Fio e Computação Móvel, pp. 183-189, 2003.

Ontologias Educacionais

Maria das Graças Pereira da Silva¹, Rommel Wladimir de Lima²

¹Universidade do Estado do Rio Grande do Norte (UERN)
Laboratório de Redes e Sistemas Distribuídos – (LORDI)
CEP: 59.625-620 - Mossoró – RN – Brasil

²Departamento de Informática
Universidade do Estado do Rio Grande do Norte (UERN)– Mossoró – RN – Brasil

gracinhapesilva@hotmail.com.br, rommel.lima@gmail.com

Resumo. As diferentes formas de aprendizagem e os diferentes elementos considerados no domínio educacional não apresentam uma padronização dos seus termos dificultando sua compreensão e compartilhamento. Neste contexto, as ontologias surgem como uma forma de solucionar este problema, fornecendo uma conceituação explícita sobre os elementos essenciais considerados no domínio. Neste artigo são apresentadas as principais ontologias relacionadas ao domínio educacional, considerando os diferentes elementos e algumas das formas de aprendizagem existentes.

1. Introdução

Com a crescente utilização de sistemas que deem suporte a educação e a utilização de várias formas de aprendizagem como a Cooperativa, a Baseada em Problema, a Colaborativa e a adoção de diferentes elementos nos domínios educacionais, surge a necessidade uma padronização dos conceitos relacionados a estas formas de aprendizagem e aos elementos presentes nos domínios educacionais, proporcionando uma compreensão comum e compartilhada.

Desta forma, as ontologias surgem como uma forma de descrever os conceitos relacionados as diversas formas de aprendizagem e aos elementos presentes nos domínios educacionais, promovendo sua padronização. Este fator é de grande importância, pois possibilita a comunicação entre os agentes envolvidos no processo, computadores e pessoas, na medida em que são reduzidas as diferenças conceituais ou terminológicas.

Neste contexto, este artigo apresenta as principais ontologias relacionadas a algumas das principais formas de aprendizagem como a aprendizagem cooperativa, a baseada em problema e a colaborativa e aos diferentes elementos considerados no domínio educacional.

Este artigo está organizado da seguinte maneira: na seção a seguir são apresentadas algumas definições sobre ontologias. Na segunda seção são apresentadas ontologias relacionadas ao domínio educacional. Na última seção são apresentadas as considerações finais.

2. Ontologias

O termo ontologia teve origem na filosofia e foi introduzido na Metafísica por Aristóteles. Na filosofia a ontologia trata da natureza do ser, da realidade, da existência dos entes e das questões metafísicas em geral. Visando solucionar a problemática associada ao compartilhamento de conhecimento (entre agentes), uma ontologia visa dar condições de se criar uma representação virtual e tratável computacionalmente de um modelo de conhecimento sobre conceitos e assuntos (RABELO, 2011).

Uma ontologia pode ser definida também como uma maneira de conceituar de forma explícita e formal os conceitos e restrições relacionadas a um domínio de interesse (YOKOME; ARANTES, 2010). As ontologias são utilizadas para representar conhecimento sobre um determinado domínio, por meio de conceitos e relacionamentos.

Uma ontologia esclarece a estrutura de um conhecimento. Em um domínio específico, a ontologia forma o centro de qualquer sistema de representação do conhecimento do domínio (ARAÚJO; FERREIRA, 2003). Sem uma conceituação do conhecimento, pode não existir um vocabulário que represente o conhecimento, dificultando seu compartilhamento.

3. Ontologias do domínio educacional

Frequentemente, na literatura, os termos aprendizagem colaborativa e aprendizagem cooperativa se confundem, sendo utilizados muitas vezes como sinônimos. Alguns estudiosos consideram que estes termos, apesar de possuírem definições similares, apresentam diferenças em relação às perspectivas teóricas e práticas.

O Núcleo Minerva (2012) define aprendizagem colaborativa como um conjunto de métodos e técnicas de aprendizagem para utilização em grupos estruturados, assim como de estratégias de desenvolvimento de competências mistas (aprendizagem e desenvolvimento pessoal e social), onde cada membro do grupo é responsável, quer pela sua aprendizagem quer pela aprendizagem dos elementos restantes.

Para Gava e Menezes (2003) a aprendizagem cooperativa busca promover a união de esforços para atingir um objetivo coletivo comum, sem necessariamente uma construção conjunta, mas apenas a união de esforços, que podem ser até isolados, mas

unidos ao final para atender ao objetivo traçado. Na cooperação também se pretende atingir um objetivo coletivo comum, mas subtende-se um esforço conjunto em prol do alcance deste objetivo.

Para a aprendizagem cooperativa Gava e Menezes (2003) apresentam uma ontologia cujo principal objetivo é fornecer uma conceitualização sobre os elementos presentes na aprendizagem cooperativa, proporcionando um maior conhecimento sobre esta área de conhecimento, contribuindo para a construção de ambientes cooperativos mais fundamentados.

Já para aprendizagem colaborativa Inaba et. al. (2001) visam à construção de uma ontologia através do levantamento das atuais teorias da aprendizagem e visam apoiar o processo de design instrucional para aprendizagem colaborativa através da ontologia. O design instrucional é compreendido como o planejamento do ensino-aprendizagem, incluindo atividades, estratégias, sistemas de avaliação, métodos e materiais instrucionais (FILATRO; PICONEZ, 2004).

A ontologia proposta é um sistema de conceitos, que atua como uma base de apoio para a identificação da aprendizagem colaborativa. Através de uma pesquisa sobre aprendizagem colaborativa, que apresentaram os principais conceitos que representam uma sessão de aprendizagem colaborativa.

No trabalho é apresentado o processo de identificação para a aprendizagem colaborativa e uma abordagem utilizada para apoiá-lo. São apresentados dois sistemas de apoio à identificação com a ontologia. O sistema de apoio TGF visa ajudar usuários que não tem conhecimento das teorias da aprendizagem e sistemas informatizados. O sistema de suporte TIA ajuda os usuários a analisar o processo de interação complexa de aprendizagem colaborativa.

Também para a aprendizagem colaborativa Barros et. al. (2001, 2002) apresentam uma ontologia, criada a partir da pesquisa sobre mecanismos de representação para relacionar e integrar elementos de aprendizagem colaborativa presentes nos ambientes reais de aprendizagem. A ontologia apresentada considera e relaciona estes elementos de aprendizagem, usando-os para construir novos cenários de aprendizagem colaborativa.

Outra teoria de aprendizagem é aprendizagem baseada em problemas (Problem-Based Learning - PBL), estratégia onde os estudantes trabalham com o objetivo de solucionar problemas através de estudos de caso previamente montados (GIL, 2006). Geralmente o problema não possui uma única solução correta.

A aprendizagem baseada em problema enfatiza a aprendizagem em vez da instrução, além de estar centrada no aluno. O método permite que o estudante aprenda a partir de um problema proposto, real ou simulado, interagindo, obtendo dados, formulando hipóteses, tomando decisões e emitindo julgamento. Assim, o aluno torna-se responsável por sua própria aprendizagem (KALATZIS, 2008).

Na aprendizagem baseada em problemas, a aprendizagem é centrada no estudante e o conhecimento é adquirido de forma autodirigida (FONTES et. al. 2011). O aluno torna-se um agente ativo no processo de aprendizagem.

Para a aprendizagem baseada em problema (Problem-Based Learning - PBL) Fontes et. al. (2011) propõem uma ontologia que abrange as estratégias que podem ser desempenhadas pelo facilitador, as habilidades que os estudantes devem adquirir no processo de aplicação da PBL, a capacidade de resolução de problemas, a colaboração e a motivação. Abrange também as metas relacionadas ao aprendizado dos estudantes, dentre outros conceitos.

As próximas ontologias focam a descrição da estrutura de um sistema de ensino como os cursos, as disciplinas e os alunos. Estas ontologias são disponibilizadas em repositórios de ontologias, facilitando a utilização de uma ontologia mais adequada ao domínio desejado. Neste trabalho serão apresentadas quatro ontologias encontradas em repositórios, são elas UNIVERSITY, INSTONTOLOGY, ACADEMIA e EDUCATION.

A ontologia UNIVERSITY – (<http://vocab.org/aiiso/schema>) fornece classes e propriedades para descrever a estrutura interna de uma instituição acadêmica. A ontologia representa uma instituição composta por faculdades que contém cursos com seus departamentos. Cada disciplina contém um objetivo educacional, seus programas e seus conteúdos são trabalhados em módulos. Cada professor e cada aluno podem desenvolver uma pesquisa e está associado a um grupo de pesquisa.

A ontologia UNIVERSITY – (<http://vocab.org/aiiso/schema>) fornece classes e propriedades para descrever a estrutura interna de uma instituição acadêmica. A ontologia representa uma instituição composta por faculdades que contém cursos com seus departamentos. Cada disciplina contém um objetivo educacional, seus programas e seus conteúdos são trabalhados em módulos. Cada professor e cada aluno podem desenvolver uma pesquisa e está associado a um grupo de pesquisa.

A ontologia ACADEMIA – (<http://www.aktors.org/ontology/portal>) descreve uma universidade de ensino presencial ou à distância focando o curso de ciência da computação. As classes da ontologia representam as faculdades, cursos, disciplinas, eventos, secretários, design gráfico alunos e professores. Representa a área e o grupo de pesquisa que professores e alunos estão associados. Além de suas participações em eventos, conferências e oficinas e artigos publicados em conferências, periódicos, livros ou revistas.

A ontologia EDUCATION – (<http://www.cs.umd.edu/projects/plus/DAML/onts/univ1.0.daml>) descreve uma universidade constituída por faculdades que são compostas por um ou mais cursos e cada curso com seu departamento. Cada curso é composto por disciplinas ou cadeiras que são ministradas em um horário determinado. As classes da ontologia representam

também os funcionários administrativos, assistentes, diretores, professores assistentes, professores associados, professor titular e professor visitante e por alunos de graduação e de pós-graduação e as áreas e grupos de pesquisa de professores e alunos.

Ao se definir um domínio de interesse com ontologias se proporciona ao desenvolvedor uma compreensão mais apurada do domínio, possibilitando o compartilhamento do conhecimento, a reutilização e a troca de informação.

4. Considerações Finais

As diferentes formas de aprendizagem existentes e os elementos considerados no domínio educacional são descritos de diferentes formas, não apresentando uma padronização dos seus termos dificultando sua compreensão e compartilhamento. Como uma solução para este problema é utilizada ontologias para fornecer uma conceituação explícita sobre os elementos essenciais considerados no domínio.

As ontologias fornecem um vocabulário que representa um domínio. Em um domínio específico, a ontologia forma o centro de qualquer sistema de representação do conhecimento do domínio. Sem uma ontologia ou conceituação do conhecimento, pode não existir um vocabulário que represente o conhecimento. (ARAÚJO; FERREIRA, 2003).

Neste artigo foram apresentadas ontologias relacionadas com o domínio educacional. Elas têm como objetivo fornecer uma padronização dos conceitos relacionados aos elementos presentes nos domínios educacionais e às várias formas de aprendizagem.

REFERÊNCIAS

- ACADEMIA. Disponível em <http://www.aktors.org/ontology/portal>. Acesso em junho de 2012.
- ARAÚJO, M. and FERREIRA, M. A. G. V. (2003) Educação a Distância e a Web Semântica: modelagem ontológica de materiais e objetos de aprendizagem para a plataforma CoL. Universidade de São Paulo (USP). São Paulo.
- BARROS, B., VERDEJO, M. F., READ, T. and MIZOGUCHI, R. (2002) Applications of a Collaborative Learning Ontology.
- EDUCATION. Disponível em <http://www.cs.umd.edu/projects/plus/DAML/onts/univ1.0.daml>. Acesso em junho de 2012.
- FONTES, L. M. O. F., MENDES NETO, F. M. and PONTES A. A. A. (2001) Um sistema multiagente de apoio à aprendizagem baseada em problema. Revista Brasileira de Computação Aplicada (ISSN 2176-6649), Passo Fundo, v.3, n.2, p.103-117.

- GAVA, T. B. S. and MENEZES, C. S. (2003) Uma ontologia de domínio para a aprendizagem cooperativa. XIV Simpósio Brasileiro de Informática na Educação - NCE - IM/UFRJ.
- FILATRO, A. (2007) Planejamento, design, implementação e avaliação de programas de educação on-line.
- GIL, A. C. (2006) Didática do ensino superior. São Paulo: Atlas.
- INABA, A., OHKUBO, R., IKEDA, M., MIZOGUCHI, R. and TOYODA, J. (2001) Design and Analysis of Learners' Interaction based on Collaborative Learning Ontology. In: Proc. of EuroCSCL01, pp.308-315.
- INSTONTOLOGY. Disponível em <http://drtc.isibang.ac.in/~bisu/ontology/instOntology.owl>. Acesso em junho de 2012.
- KALATAZIS, A. C. (2008) A aprendizagem baseada em problemas em uma plataforma de ensino a distância com o apoio dos estilos de aprendizagem: uma análise do aproveitamento dos estudantes de engenharia. São Carlos.
- NÚCLEO MINERVA DA UNIVERSIDADE DE ÉVORA, 2001. Disponível em: <<http://www.minerva.uevora.pt/cscl/>>. Acesso em: 23 ago. 2012.
- YOKOME, E. A. and ARANTES, F. L. (2010) Revista de Sistemas de Informação da FSMA n. 8. pp. 36-45.
- RABELO, R. J. Ontologia em Sistemas de Agentes. Disponível em <[http://www.das.ufsc.br/~rabelo/Ensino/DAS6607/Aula4%20\(Ontologia\).pdf](http://www.das.ufsc.br/~rabelo/Ensino/DAS6607/Aula4%20(Ontologia).pdf)> Acesso em: 07 mar. 2012.
- UNIVERSITY. Disponível em <http://vocab.org/aiiso/schema>. Acesso em junho de 2012.

BestResources: Ferramenta de Apoio à Alocação de Recursos Humanos para Projetos de TI no dotProject

José Osvaldo M. Chaves¹, Marcos Vinicius de A. Lima¹, Rodrigo Ronner Tertulino¹, Francisco Milton M. Neto¹, Cicília Raquel M. Leite¹

¹Mestrado Acadêmico em Ciência da Computação – PPgCC – UERN/UFERSA
Av. Francisco Mota, 572 – CEP: 59.625-900 – Mossoró – RN – Brasil

{osvaldo.mesquita, marcos.engsoft, rodrigoronner, miltonmendes, ciciliamaia}@gmail.com

Abstract. This paper presents the BestResources tool, a dotProject integrated tool, and that was developed to assist the Project Manager, giving it more quickly in choosing the best available human resources that can be allocated to run tasks in IT projects, through the use of intelligent agents and ontology.

Resumo. Este artigo apresenta a ferramenta BestResources, uma ferramenta integrada ao dotProject e desenvolvida para auxiliar o Gerente de Projetos, proporcionando a ele maior rapidez na escolha dos melhores recursos humanos disponíveis que podem ser alocados para executarem atividades em projetos de TI, por meio da utilização de agentes inteligentes e ontologia.

1. Introdução

Em um mundo globalizado é fundamental para qualquer empresa, que queira continuar com boa competitividade, a melhoria da qualidade de seus produtos. Esta máxima também é válida para as empresas desenvolvedoras de *software*. Mas como garantir qualidade para um produto intangível como *software*? De acordo com Crosby (CROSBY, 1992), “*a qualidade é conformidade aos requisitos*”. O que leva a pensar na satisfação do cliente e atendimento de suas necessidades. Mas alcançar esta qualidade em *software* não é uma tarefa trivial visto que o desenvolvimento de sistemas não possui um aspecto repetitivo, como existe na produção industrial. Outro ponto importante a ser observado é que a volatilidade dos requisitos é uma característica comum, presente nos diversos projetos de *software*.

Diante de toda a dificuldade para garantir a qualidade no desenvolvimento de *software*, é preciso buscar formas de se alcançar a excelência, esta que pode ser obtida por meio da qualidade dos processos de desenvolvimento. Um bom processo não garante que os produtos são de boa qualidade, mas é um indicador que a organização é capaz de produzir bons produtos. E quando se fala em processo de *software*, automaticamente se fala na aplicação de técnicas e ferramentas de engenharia de *software*, bem como na utilização de recursos humanos especializados para realizarem as mais variadas atividades.

No ano de 2000, o SEI – *Software Engineering Institute*, apresentou a primeira versão de um modelo de maturidade e capacidade, conhecido como CMMI – *Capability Maturity Model Integration*. Este modelo visa promover a melhoria de processos para garantia da qualidade do produto, sendo composto por cinco níveis de maturidade em sua representação por estágios, formada pelos níveis de 1 a 5 (CHRISSIS et al, 2011).

Para conseguir alcançar todas as recomendações do nível 2 (gerenciado), o CMMI sugere a adoção do Guia PMBOK®, que é um guia que identifica as boas práticas em gerenciamento de projetos. Este guia possui cinco grupos de processos (iniciação; planejamento; execução; monitoramento e controle; encerramento) e nove áreas de conhecimento (PMI, 2008); entre estas áreas está a de Gerenciamento de Recursos Humanos, que é o foco deste artigo, pois acreditamos que a escolha da melhor equipe é fundamental para garantia do sucesso em projetos de *software*.

Martins diz que o planejamento de recursos humanos “*envolve a determinação de funções, responsabilidades e a hierarquia das pessoas no projeto [...] para saber qual é o papel de cada um no projeto.*” (MARTINS, 2007). Com isto, observa-se a importância na boa definição de uma equipe, pois uma equipe ruim poderá impactar negativamente no cumprimento do prazo e custo do projeto, além da qualidade do produto de *software*. Desta forma, a ferramenta BestResources busca auxiliar o Gerente de Projetos durante o processo de alocação de recursos humanos para realização das mais diversas atividades em projetos de TI (Tecnologia da Informação), visto que a ferramenta apresenta uma lista com os profissionais capacitados para realização da atividade que está sendo cadastrada, além de, também, informar o nível de sobrecarga de trabalho no período para cada colaborador sugerido.

2. A Ferramenta BestResources

Como a ideia principal deste trabalho é de auxiliar o gerente de projetos na atividade de alocação de recursos humanos para as mais variadas atividades existentes nos projetos de TI, pensou-se em construir, ao invés de um novo sistema, uma ferramenta que pudesse ser integrada a algum *software* amplamente utilizado pelo gerente de projetos no exercício de suas atribuições. Então, BestResources é uma ferramenta que funciona integrada ao sistema de gerenciamento de projetos livre dotProject, no qual insere novas funcionalidades para tornar a atividade de alocação de recursos humanos mais rápida, fácil e inteligente, por meio da utilização de reengenharia de *software*, agentes inteligentes e ontologia.

2.1. Principais Características

A ferramenta BestResources adiciona ao dotProject novos recursos, de acordo com os requisitos funcionais listados a seguir:

- REQ01: Cadastrar competências dos colaboradores (tela de cadastro de usuário);
- REQ02: Cadastrar habilidades dos colaboradores (tela de cadastro de usuários);
- REQ03: Consultar profissionais capacitados para realização de determinada atividade (tela de cadastro de atividades, aba recursos humanos);
- REQ04: Exibir total de atividades alocadas em um dado período para os colaboradores que possuem expertise para realização da atividade (tela de cadastro de atividades, aba de recursos humanos).

3. Tecnologias Envolvidas

Para a construção da ferramenta BestResources, fez-se necessária a utilização de processos de reengenharia de *software*, além de diversas tecnologias para que as atividades do gerente de projetos pudessem ser facilitadas. Os processos de reengenharia contaram com reengenharia reversa do banco de dados (para compreender

os relacionamentos existentes entre as diversas tabelas); reestruturação dos dados (para adicionar novas informações: qualificações formais e informais dos colaboradores); reestruturação do código (somente quando possível para melhorar a clareza); e engenharia avante (para adicionar novas funcionalidades ao dotProject). As tecnologias envolvidas são explicadas nas subseções seguintes.

3.1. dotProject e PHP

Como já foi dito, dotProject é um sistema de gerenciamento de projetos livre amplamente utilizado, tem como principais características: possuir gerenciamento de usuários, e-mail, gerenciamento de empresas/clientes, relatórios de projetos, hierarquia de tarefas, repositório de arquivos, lista de contatos, calendário, fórum de discussão e controle de permissões. Outra característica importante é que o dotProject foi escrito em linguagem PHP. De acordo com The PHP Group, “*PHP (um acrônimo recursivo para PHP: Hypertext Preprocessor) é uma linguagem de script open source de uso geral, muito utilizada e especialmente garantida para o desenvolvimento de aplicações Web*”. (THE PHP GROUP, 1997).

3.2. Jena e Protégé

O “*Jena é um framework Java para criação de aplicações para a Web Semântica. Ele provê um ambiente de programação para RDF, RDFS e OWL, SPARQL e inclui um mecanismo de inferência baseado em regras*” (JENA TEAM, 2011). Seu objetivo é proporcionar um framework na linguagem Java que dê suporte à utilização da Web Semântica, possibilitando a manipulação de ontologias representadas na linguagem OWL (*Web Ontology Language*), por meio de uma API específica para esta linguagem.

No desenvolvimento da ferramenta BestResources foi utilizado o Jena para manipulação da ontologia ITCompetenceByTask, que foi criada com o editor Protégé¹, com base nas atividades e competências de um projeto de TI. Para a realização das devidas consultas à ontologia foi utilizado SPARQL, que é uma linguagem de consulta do tipo SQL-Like, que se tornou o padrão recomendado pelo W3C (W3C, 2008). A ontologia criada para a ferramenta BestResources possui duas classes principais: a classe atividades e a classe competências; onde cada uma delas possui suas respectivas subclasses. Depois de especificada cada parte da estrutura da ontologia desenvolvida é possível visualizá-la como pode ser visto na Figura 1.

3.3. JADE – Java Agent Development Framework

JADE é um *framework open source* totalmente implementado na linguagem Java que simplifica a manipulação de multiagentes por meio de *middleware*, seguindo as especificações da FIPA². Ele é distribuído sob licença LGPL – *Lesser General Public License*, sendo desenvolvido e mantido pela Universidade Parma na Itália (JADE PROJECT, 2011). A plataforma JADE possui um abrangente conjunto de serviços que facilita e possibilita a comunicação entre agentes, como: serviço de nomes (*service name*), páginas amarelas (*yellow-page service*), transporte de mensagens, serviços de

¹ Protégé – Editor de ontologia e *framework* para base de conhecimento. Disponível em: <http://protege.stanford.edu/>

² FIPA (*Foundation for Intelligent Physical Agents*) é uma organização da IEEE Computer Society que promove a tecnologia baseada em agentes.

codificação e decodificação de mensagens e uma biblioteca de interação pronta para ser utilizada. A comunicação entre os agentes só é possível por meio de troca de mensagens ACL (*Agent Communication Language*).

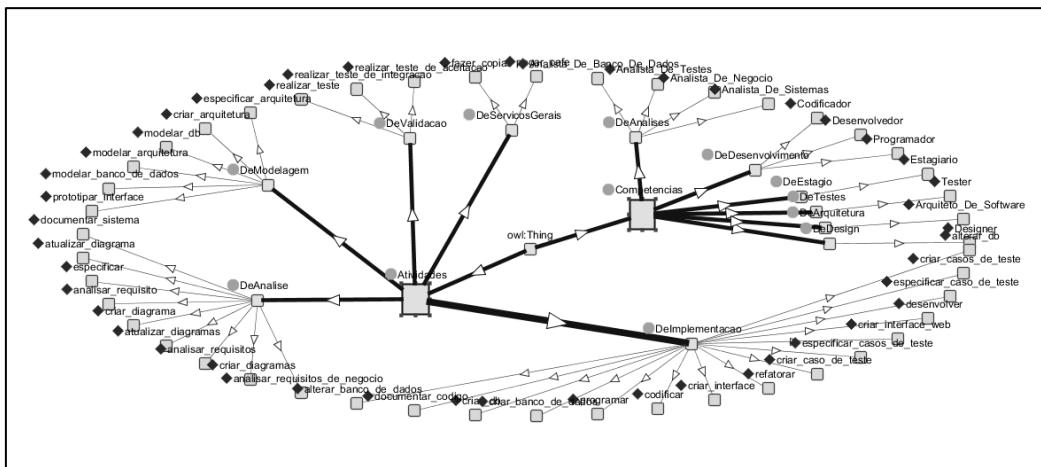


Figura 1. Estrutura da ontologia desenvolvida para ferramenta BestResources.

4. Processo de Integração

Como visto na seção anterior, para o desenvolvimento da ferramenta BestResources, foi preciso o emprego de diversas tecnologias, e como quase todas diferem em características como: tipo de linguagem utilizada, ambiente de execução e forma de acesso para troca de informações; o processo de integração provou ser a fase mais complicada a se realizar. A Figura 2 mostra a problemática para integrar as diferentes tecnologias, ferramentas e serviços (representadas por mundos), para que pudessem operar de forma simples, segura e estável. No mundo1 temos o servidor HTTP Apache, a linguagem PHP e o dotProject. No mundo2 temos a linguagem Java e os frameworks JADE e Jena. É possível ainda observar que apenas o MySQL pode ser acessado pelos dois conjuntos de tecnologias.

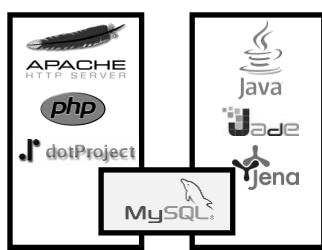


Figura 2. Dois mundos diferentes envolvidos na construção da ferramenta.

Após minuciosa análise do problema, a solução para a integração foi encontrada por meio da ideia de utilização de um *Web Service* que pudesse tratar as solicitações advindas da Web e ao mesmo tempo tivesse pleno acesso ao contêiner JADE. Desta maneira, foi preciso instalar e configurar o WSIG – *JADE Web Service Integration Gateway*, que torna possível a comunicação entre a Web e o mundo dos agentes, sendo responsável por: repassar as ações recebidas da Web para agentes, assim como devolver as devidas respostas; gerenciar o registro de agentes; disponibilizar os serviços registrados.

Para que o WSIG pudesse funcionar corretamente foi preciso instalar um contêiner Java por meio da utilização do Apache Tomcat, que permite a implementação de *servlets* e *Java Server Pages* (JSPs), sendo bastante poderoso, de larga escala e também podendo ser utilizado para aplicações *Web* de missão crítica (THE APACHE SOFTWARE FOUNDATION, 1999).

Para uma perfeita integração do ambiente do dotProject com o servidor Tomcat foi utilizada tecnologia AJAX para evitar o recarregamento da página *Web* quando uma consultar por colaboradores especializados fosse realizada. Também foi preciso codificar uma página JSP responsável por enviar uma submissão SOAP ao *servlet* WSIG para a requisição de serviços. A arquitetura final pode ser observada na Figura 3.

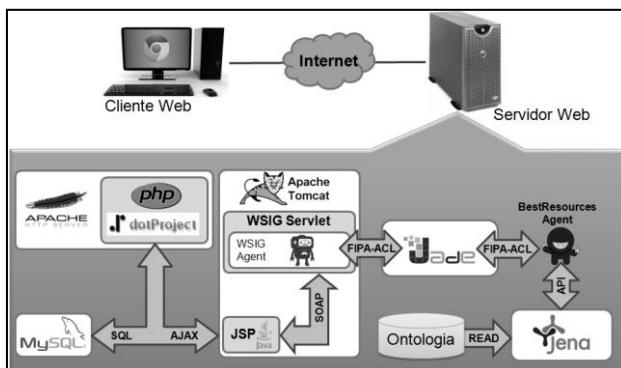


Figura 3. Arquitetura final para o funcionamento da ferramenta BestResources.

Após a instalação e configuração de todas as tecnologias necessárias para a integração da ferramenta BestResources, o dotProject ganhou modificações em duas telas: no cadastro de usuários, como pode ser observado na Figura 4.a, e no cadastro de atividades (aba de Recursos Humanos), como pode ser visto na Figura 4.b.

Figura 4.a Nova tela de cadastro de usuários

4.b Nova tela de cadastro de tarefas

5. Resultados Obtidos

Embora ainda estejam sendo realizados testes na versão alfa da ferramenta BestResources, os resultados obtidos são bastante animadores: com o uso da ferramenta o gerente de projetos não precisa mais perder tempo procurando pelo melhor recurso humano a ser alocado em uma determinada tarefa. Embora à primeira vista, em testes simples com poucos usuários, o uso da ferramenta possa não fazer sentido, em grandes organizações, com centenas ou até mesmo milhares de usuário cadastrados, a busca em uma lista que comporta todos os colaboradores pode ser um pouco estressante. Já com o uso da ferramenta, os usuários são filtrados, de acordo com a expertise necessária para

realizar a atividade. Outro resultado obtido com a integração da ferramenta foi a inclusão das competências dos diversos colaboradores, quer sejam elas formais ou informais, informação preciosa ao gerente de projetos que está alocando uma equipe.

6. Conclusão

A ferramenta BestResources foi idealizada com o intuito de auxiliar o gerente de projetos, de forma rápida, segura, estável e inteligente; na escolha dos melhores recursos humanos disponíveis para projetos de TI, sendo concebida por meio da integração de várias tecnologias, trazendo o desafio de unir: servidor Apache e linguagem PHP, com as tecnologias adotadas pelo JADE e Jena, baseadas em Java.

A solução para unir estes dois mundos foi encontrada por meio da utilização do WSIG. Esta solução trouxe um caráter profissional à ferramenta, pois toda a comunicação foi elaborada de forma a respeitar o uso racional dos recursos de infraestrutura, como conexões com os diversos servidores e utilização de banda de rede, evitando tráfego desnecessário. A ferramenta BestResources, portanto, trabalha de forma inteligente, acionando o agente apenas nos momentos que ele deve realizar alguma requisição.

Embora os resultados iniciais sejam animadores, ainda é preciso submeter à ferramenta a um número maior de projetos reais, para assim melhor avaliar seu uso e comportamento. Como trabalhos futuros esperamos poder integrar a tecnologia apresentada neste artigo a outras ideias de melhorias do dotProject, como indicação de pessoas qualificadas para ministrar treinamentos; calcular riscos nos projetos; e acompanhar o desempenhos dos diversos colaboradores.

Referências

- CROSBY, P. B. Quality is Free: the Art of Making Quality Certain. Denver: Mentor Books, 1992.
- CHRISSIS , M. B.; KONRAD, M. and SHRUM ,S. CMMI for Development – guidelines for process integration and product improvement. 3rd ed., Pearson Education, 2011.
- JADE PROJECT. Java Agent Development Framework . Disponível em:
<<http://jade.tilab.com/>>. Acesso em: 08 jul 2012.
- JENA TEAM; HEWLETT-PACKARD. Jena: A Semantic Web Framework for Java.
Disponível em: <<http://jena.sourceforge.net/>>. Acesso em: 07 jul 2012.
- MARTINS, J. C. C. Gerenciando Projetos de Desenvolvimento de Software com PMI, RUP e UML. 4rd ed., Brasport, 2007, pages 61-62.
- PMI, Project Management Institute. PMBOK – Um guia do conjunto de conhecimentos em gerenciamento de projetos. 4rd ed., Newtown Square, 2008.
- THE APACHE SOFTWARE FOUNDATION. Disponível em:
<<http://tomcat.apache.org/>>. Acesso em: 07 jul 2012.
- THE PHP GROUP. Manual do PHP. Disponível em:
<http://www.php.net/manual/pt_BR/index.php>. Acesso em: 07 jul 2012.
- W3C, World Wide Web Consortium. SPARQL Query Language for RDF. Disponível em: <<http://www.w3.org/TR/rdf-sparql-query/>>. Acesso em: 07 jul 2012.

PROTOCOLO ZIGBEE EM REDES DE COMUNICAÇÃO INTRAVEICULAR

Irlan Arley Targino Moreira¹, Karla D. N. Ramos¹, Felipe D. M. Oliveira¹

¹Universidade do Estado do Rio Grande do Norte (UERN)
Av. Ayrton Senna, 4241 – 59088-100 – Neópolis – Natal/RN - Brasil
 {irlan.arley,karla}@gmail.com, fdenis_natal@yahoo.com.br

Abstract. This article aims to study the use of the Zigbee Protocol in intra-vehicular environments, more precisely their communication aspect. A twilight sensing application was implemented to test the communication performance. Tests were performed by analyzing packet loss and interference from other networks such as Bluetooth and Wi-Fi. The results show that the Zigbee protocol is a viable alternative to be used in intra-vehicular communication networks.

Resumo. Este artigo tem como objetivo estudar o uso do protocolo Zigbee em ambientes intraveicular, mais precisamente o aspecto da comunicação. Uma aplicação de sensoriamento crepuscular foi implementada para testar o desempenho da comunicação. Foram realizados testes para analisar a perda de pacotes e interferência de outras redes como o Bluetooth e o Wi-Fi. Os resultados mostram que o protocolo Zigbee é uma alternativa viável para utilização em redes de comunicação intraveicular.

1. Introdução

Com o objetivo de monitorar um dado fenômeno, surgiram as Redes de Sensores Sem Fio (RSSF) [Loureiro *et al* 2003]. RSSF engloba algumas frentes de pesquisa tais como as redes pessoais (em inglês, *Wireless Personal Area Network - WPAN*), redes terrestres, redes subterrâneas, redes aquáticas, redes multimídia [Yick *et al* 2008] e as redes veiculares (em inglês, *Vehicle Area Network - VAN*).

As redes veiculares são formadas principalmente por veículos que interagem com outros veículos (*Vehicle-to-Vehicle – V2V*), veículo com infraestrutura para transmitir e receber dados (*Vehicle-to-roadside-Infrastructure - V2I*), veículo com a nuvem (*Vehicle-to-Broadband cloud - V2B*) e a comunicação entre os próprios dispositivos (*Intra-Vehicle – In-Vehicle*), cujo principal foco é coletar informações do comportamento do carro e do condutor, de modo que sejam analisadas e classificadas para evitar acidentes e auxiliar o motorista [Faezipour *et al* 2012].

Normalmente os veículos utilizam fios para realizar a comunicação entre seus dispositivos e o controlador central. A tendência é que mais dispositivos sejam instalados nos veículos aumentando a quantidade de fios, que deverão ser o meio de comunicação. Porém, este tipo de infraestrutura pode se tornar mais cara e aumentar o nível de complexidade. Uma alternativa seria a implantação de redes de sensores sem fio veicular [Roca 2008].

Considerando a rede veicular do tipo *Intra-Vehicle* e a problemática da complexidade do uso de fios para comunicação entre os diversos dispositivos de um carro, este trabalho pretende investigar a utilização do protocolo Zigbee, por meio de

uma aplicação semelhante ao sistema de sensoriamento crepuscular utilizado atualmente em alguns automóveis.

Este artigo está dividido da seguinte forma: a Seção 2 aborda sobre alguns trabalhos relacionados com redes de sensores sem fios e ambiente veicular. A Seção 3 apresenta uma análise sobre o protocolo Zigbee e como esta tecnologia pode ser utilizada no ambiente veicular. Na Seção 4 será mostrado o protótipo de aplicação crepuscular implementado e os testes realizados. Por fim, na Seção 5, serão apresentados os resultados e a conclusão.

2. Trabalhos Relacionados

Recentemente, em [Faezipour *et al* 2012] foram apresentados os principais desafios enfrentados pelas redes veiculares, sejam do tipo *In-Vehicle*, V2I, V2V e V2B. Em [Hu *et al* 2009] é realizado o monitoramento da qualidade do ar por meio da concentração de CO₂ (dióxido de carbono) utilizando redes de sensores veiculares baseando-se em curtas mensagens GSM e informações geográficas dos veículos.

Viana [Viana Jr, 2010] realizou estudo no ambiente das redes intraveiculares com o propósito de inovar a comunicação entre as unidades de controle veicular e os sensores espalhados pelo veículo utilizando tecnologia sem fio. Tecnologias como o Zigbee, Bluetooth, RFID e UWB foram analisadas de acordo com resultados apresentados por diversos autores.

Comparando os trabalhos citados, o presente trabalho é diferencia-se dos demais, uma vez que trata de um protótipo de uma aplicação de sensoriamento crepuscular.

3. Padrão IEEE 802.15.4 e o Protocolo Zigbee

Um dos desafios no ambiente veicular é a enorme quantidade de dispositivos que estão sendo incorporados aos veículos. Hoje existem carros com sensores para detecção de chuva, sensores de estacionamento e sensor crepuscular, por exemplo. Com a implantação de novos mecanismos, consequentemente será necessário a instalação de novos conjuntos de cabos ligados à central eletrônica, que, por sua vez, possui o número de portas limitado.

As redes de sensores sem fio se apresentam como uma solução para este problema e utilizam diferentes protocolos. A seguir será apresentado o Padrão IEEE 802.15.4, que engloba o protocolo Zigbee, o qual é um dos protocolos utilizados na redes *In-Vehicle*.

3.1 Padrão IEEE 802.15.4

O padrão IEEE 802.15.4 define as camadas MAC e PHY. A camada MAC é responsável por controlar o acesso dos canais de rádio usando o protocolo CSMA-CA. Este protocolo também é responsável pela transmissão de quadros beacons, sincronização e mecanismo de reabilitação de transmissão [Zigbee Specification 2008].

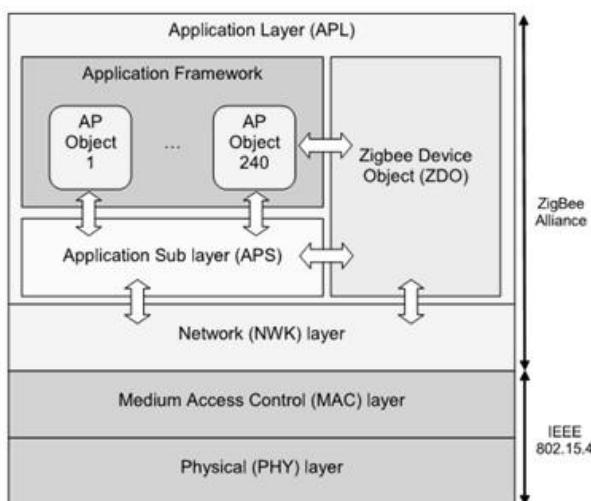
A camada PHY pode operar nas bandas não licenciadas de 868 MHz utilizada na Europa com taxa de transmissão de 20 Kb/s e com apenas um canal (canal 0); 915 MHz utilizada em países como Estados Unidos e Austrália com taxa de transferência de 40 Kb/s e possui 10 canais (canais de 1 à 10); e 2.4 Ghz utilizada mundialmente com taxa de transferência de 250Kb/s e 16 canais (canais de 11 à 26) [IEEE 802.15.4 2003].

3.2 Protocolo Zigbee

O protocolo Zigbee Foi desenvolvido pela *Zigbee Aliance* (consórcio de empresas) para ser um padrão de comunicação sem fio de baixo custo e baixo consumo energético [Zhao et al 2004]. Define as camadas de rede e de interface de aplicação e os serviços de segurança entre elas.

A pilha Zigbee é formada por diversas camadas que possuem serviços específicos para auxiliar as camadas superiores. Uma entidade de dados fornece serviços para a transmissão dos dados, e a entidade de gerenciamento fornece todos os outros serviços. Cada entidade de serviço possui uma interface para a comunicação com a camada superior por meio de um ponto de acesso de serviço (*Service Point Access - SAP*). Cada interface possui uma quantidade de serviços primitivos que podem ser suportados [Zigbee Specification 2008]. A Figura-1 ilustra a pilha de protocolos do Zigbee.

Figura 1. Pilha de protocolos do Zigbee [Baronti et al 2007]



A *Zigbee Aliance* fornece as camadas de rede (*NWK - Network*) e um *framework* para a camada de aplicação. Esse *framework* compreende a subcamada de suporte à aplicação (*APS – Application Support*) e os objetos de dispositivos Zigbee (*ZDO – Zigbee Device Objects*) [Zigbee Specification 2008], já as camadas física (*PHY – Physical*) e de controle de acesso ao meio (*MAC – Medium Access Control*) são definidas pelo padrão IEEE 802.15.4.

A camada *NWK* suporta três tipos de topologia: estrela, árvore e malha. Na topologia em estrela, a rede é controlada por apenas um nó central, responsável por inicializar e gerenciar toda a rede. Os demais componentes são conhecidos como nós terminais e se comunicam diretamente com o nó coordenador. As topologias árvore e malha possuem uma característica em comum: as redes podem ser expandidas utilizando roteadores, mas o nó coordenador continua sendo o responsável por iniciar a rede e definir os parâmetros da mesma. Na rede em árvore os roteadores trafegam os dados e controlam as mensagens de acordo com uma estratégia de roteamento hierárquico, podendo a rota de transmissão, ser baseada no mapa de energia da rede, por exemplo [Zigbee Specification 2008].

4. Aplicação e Testes

Para analisar o desempenho da comunicação do protocolo Zigbee, foi desenvolvida uma aplicação de sensoriamento crepuscular que está dividida em três módulos: um mestre, um escravo e um de tomada de decisões da aplicação. O mestre é responsável por receber as informações oriundas do nó escravo e a partir de uma comunicação USB transferir os dados para a aplicação hospedada em um computador central. O escravo é responsável por coletar os dados de acordo com a mudança do ambiente e enviá-los ao mestre. Estes dados são coletadas por meio de um sensor ligado ao mestre. E a aplicação tem por função analisar os dados recebidos e tomar decisões pré-definidas de acordo com as condições existentes.

As aplicações instaladas nos nós foram escritas em NesC e compiladas no TinyOS 2.1.1. Ambos os motes foram configurados para atuarem no canal 26. A aplicação instalada na estação base é a *BaseStation*, aplicação que acompanha o TinyOS. A aplicação presente no computador central foi escrita em Java.

Na subseção 4.1 serão relatados os equipamentos utilizados para o desenvolvimento da rede de sensores sem fio. A subseção 4.2 descreverá o funcionamento da aplicação de sensoriamento crepuscular desenvolvida e, por último, na subseção 4.5 será abordada a metodologia utilizada nos testes realizados.

4.1 Equipamentos utilizados

Para a implementação da aplicação foi utilizado o kit de redes de sensores sem fio OEM Design Kit da CrossBow. Deste kit foram utilizados os seguintes equipamentos:

- Dois motes Iris XM2110;
- Um USB Programming Pod/Gateway Mib 520;
- Uma placa sensor modelo MDA100;
- Duas baterias AA;
- Um notebook com o sistema operacional Linux (Computador Central);

A placa MDA100 está conectada fisicamente a um dos motes. Este sensor pode colher informações de iluminação e temperatura. O segundo mote é utilizado como nó central e está conectado fisicamente ao *gateway*, que por sua vez, está conectado ao computador central por uma porta USB.

4.2. Funcionamento do sistema crepuscular

A aplicação utilizou uma topologia de rede do tipo estrela, sendo a rede composta por uma estação base e um nó final. O nó sensor foi colocado a uma distância de 1,90 metros da estação base em um ambiente no qual pudesse colher as informações necessárias. Foi definido que a cada 250 milissegundos o sensor mediria o nível de luminosidade. O estado da última amostragem foi guardado em uma variável para tomada de decisões futuras.

Conforme descrito, a cada 250 milissegundos o sensor faz a coleta da informação de luminosidade e o valor obtido é comparado ao valor padrão de luminosidade. Se o valor da amostra for menor que 700 unidades (valor obtido pelo conversor AD) e o estado atual do ambiente for diferente do estado anterior, então o nó sensor enviará uma mensagem contendo o valor de id do nó e o estado atual do ambiente, “claro” ou “escuro”. O estado claro é estabelecido quando o valor de

luminosidade é superior à 700, e escuro quando este valor for menor que 700. O valor de 700 foi definido como limite entre os estados por meio de coletas analisadas em ambiente aberto ao final da tarde.

Ao receber a mensagem, a estação base encaminha para o computador central que analisa o estado do ambiente e toma a decisão de acender ou apagar as luzes do veículo. Uma característica que pode ser observada nesta aplicação é o baixo número de transferência de dados entre os motes, o que representa um benefício visto que esta é a atividade em RSSF que mais consome energia.

4.3. Teste realizado

Para validação do sistema crepuscular foram realizados testes de interferência entre equipamentos que utilizam o protocolo Zigbee e o protocolo *Bluetooth*. Como resultado desse teste foram avaliados os valores de potência de sinal por meio do valor de RSSI (*Received Signal Strength Indication*) e a porcentagem de pacotes perdidos.

Como a taxa de amostragem padrão da aplicação é muito baixa, foram necessárias alterações para que o sensor passasse a enviar dados a cada 250 milissegundos. A metodologia utilizada foi coletar três amostras, cada uma transferindo 3.500 pacotes, enquanto os dispositivos com tecnologia *Bluetooth* também realizavam transferência de dados. Com base em valores de RSSI e números de pacotes enviados e recebidos foi possível extrair a média simples da potência do sinal e a taxa de pacotes perdidos em cada amostra. Este mesmo procedimento foi realizado para um ambiente com interferência Wi-Fi.

5. Resultados e Conclusão

O primeiro teste realizado foi com a interferência *Bluetooth*. Conforme observado na tabela 1, na primeira amostragem, dos 3.500 pacotes enviados, foram recebidos 3.493, ou seja, perdeu-se 7 pacotes, resultando em 0,20% de perda. A potência média do sinal recebido foi de 43,73%. Na segunda amostragem nenhum pacote foi perdido, sendo a média da potência do sinal de 34,00%. Na terceira amostragem foram recebidos 3.499 pacotes, resultando em 0,03% de perda, com a potência do sinal de 42,62%.

A transferência dos dados entre os dispositivos *Bluetooth* teve uma pequena queda na velocidade de transferência, mas nada que afetasse consideravelmente o desempenho.

Tabela 1. Amostragem de interferência *Bluetooth* a uma distância de 1,90 metros

| Amostra | Pacotes perdidos (%) | Potência do Sinal RSSI (%) |
|---------|----------------------|----------------------------|
| 1 | 0,2 | 43,73 |
| 2 | 0 | 34 |
| 3 | 0,03 | 42,62 |

O segundo teste realizado foi o de interferência com dispositivos Wi-Fi (IEEE 802.11). Em ambas as amostragens não houve perda de pacotes. A escolha do canal de comunicação de rádio frequência número 26 (último canal do Zigbee) se mostrou ideal, já que este está fora da faixa de frequência do Wi-Fi.

Com os resultados obtidos pode-se afirmar que em condições de teste e prototipagem, a aplicação construída utilizando o protocolo Zigbee obteve bons

resultados, podendo compor sistemas intraveiculares não críticos e de baixa taxa de amostragem.

Este trabalho implementou um sistema de sensoriamento crepuscular para o ambiente intraveicular visando à substituição dos fios usados na instalação de tais dispositivos. Com isso, o número de sistemas utilizando essa tecnologia de comunicação poderia aumentar significativamente, visto que o protocolo suporta até 65.000 dispositivos. Este protocolo foi escolhido devido as suas características, tais como o grande número de nós que podem se conectar a rede, o baixo consumo de energia, baixo custo e facilidade de instalação. Os resultados mostraram que o protocolo Zigbee é uma alternativa viável para utilização em redes de comunicação intraveicular.

Referências

- Baronti, P.; Pillai, P.; Chook, V. W.C.; Chessa, S.; Gotta, A. and Hu Y. F. (2007) “Wireless sensor networks: A survey on the state of the art and the 802.15.4 and ZigBee standards”, Computer Communications, Volume 30, Issue 7, 26 May, Pages 1655-1695.
- Duarte O. C. M. B. (2008) “Redes de Sensores sem Fio”, http://www.gta.ufrj.br/grad/08_1/rssf/index.html, August.
- Faezipour, M.; Nourani, M.; Saeed, A. and Addepalli, S. (2012) “Progress and Challenges in Intelligent Vehicle Area networks”. Communications of the ACM, v. 55 n. 2, pp. 90-100.
- Hu, S.C.; Wang, Y.C; Huang, C.Y. and Tseng, Y.C. (2009) “A vehicular wireless sensor network for CO₂ monitoring”. IEEE SENSORS Conference. pp. 1498 - 1501.
- IEEE 802.15.4 (2003) “Standard Report: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)”. The Institute of Electrical and Electronics Engineers, Inc.
- Viana Jr., J. A. (2010) “Aplicação das Tecnologias Sem Fio nas Redes Intraveiculares”. 66 f. Monografia (Graduação em Ciência da Computação) – Faculdades Integradas Pedro Leopoldo, Pedro Leopoldo, Minas Gerais.
- Loureiro, A. A. F.; Nogueira, J. M. S; Ruiz, L. B. and Mini, R. A. F. (2003) “Redes de Sensores Sem Fio”. In: 21º Simpósio Brasileiro de Redes de Computadores. (Org.). Minicurso do SBRC.
- Oliveira, Felipe Denis Mendonça de. (2008) “Desenvolvimento de um software de comunicação sem fio aplicado à instrumentação de unidade de elevação de petróleo tipo plunger lift”. () 135 f. Dissertação (Mestrado) – Universidade Federal do Rio Grande do Norte.
- Roca, I. (2008) “Rede de Sensores Sem Fio em Ambientes Veiculares Baseada no Padrão Zigbee”. 136 f. Dissertação (Mestrado em Engenharia Elétrica) – Universidade de São Paulo, São Paulo.
- Tonguz, O. K.; Tsai, Hsin-Mu; Talty T.; Macdonald A. and Saraydar C. (2006) “RFID Technology for Intra-Car Communications: A New Paradigm”. In: Vehicular Technology Conference.
- Zhao, F. and GUIBAS, L. (2004) “Wireless Sensor networks: An Information Processing Approach”. Edited by Elsevier Inc. São Francisco.

Uma avaliação da técnica de Casamento de Padrões na construção de Agentes Conversacionais

Rômulo de Oliveira Nunes, André Maurício Cunha Campos, Elizama das Chagas Lemos¹

¹Departamento de Informática e Matemática Aplicada, UFRN Natal, Brasil

romulo_ciencomp@yahoo.com.br, andre@dimap.ufrn.br,
elizama@ifrn.edu.br

Abstract. This article describes presents a study about different techniques using pattern matching, by presenting a prototype used as case study. The case study uses a description language called AIML (Artificial Intelligence Markup Language) which employs pattern matching to build the chatterbot knowledge base. This study aims to make an assessment on the strengths and weaknesses that exist in using the technique of pattern matching for building conversational agents.

Resumo. Este artigo apresenta um estudo sobre diferentes técnicas que utilizam casamento de padrões na construção de agentes conversacionais, apresentando um protótipo desenvolvido para ser utilizado como estudo de caso. O estudo de caso utiliza uma linguagem de descrição chamada AIML (Artificial Intelligence Markup Language), que utiliza casamento de padrões para construir sua base de conhecimento do chatterbot. Esse estudo busca fazer uma avaliação sobre os pontos positivos e negativos que existem em utilizar a técnica de casamento de padrões na construção de agentes conversacionais.

1. Introdução

Desenvolver um agente conversacional coerente é uma tarefa complexa, principalmente por simular o comportamento humano em uma conversa. Ao interagir com um ser humano, o agente não possui conhecimento prévio das questões que serão levantadas pelo usuário. Apesar da maioria dos agentes serem voltados para um determinado domínio de aplicação, o usuário pode direcionar a conversa a um contexto totalmente diferente do esperado. Assim, uma das maiores dificuldades de se desenvolver um agente conversacional é a de utilizar todos os aspectos da comunicação verbal de forma coerente, atendendo às necessidades do usuário e sendo capaz de englobar as diversas possibilidades de conversação que venham a ocorrer.

Para desenvolver um agente conversacional, existem diversas técnicas que podem ser utilizadas. Há técnicas simples, que se baseiam apenas em casamento de padrões, e técnicas mais complexas de processamento de linguagem natural, que fazem

uma análise da sintaxe e da semântica para processar a entrada disponibilizada pelo usuário e gerar uma saída adequada como resposta.

Existem diversas formas de se utilizar a técnica de casamento de padrões no desenvolvimento de agentes conversacionais. É uma abordagem reativa, sem tratamento de semântica da mensagem. Apesar da simplicidade, como já mencionado, ela tem se mostrado bastante efetiva em diálogos simples e bem contextualizados. Há, entretanto, duas abordagens normalmente utilizadas na confecção desses agentes conversacionais: uma de mais alto nível, onde os diálogos são representados através de uma linguagem descritiva (baseada, por exemplo, em XML) outra de mais baixo nível, onde os diálogos são programados usando, por exemplo, expressões regulares para gerar as respostas.

2. Casamento de padrões

Casamento de padrões é uma técnica que consiste em realizar uma relação entre um conjunto de palavras-chaves especificadas previamente e um conjunto de respostas que serão escolhidas dependendo de qual conjunto de palavras-chaves ela se relaciona.. Diversas técnicas fazem uso de casamento de padrões para a construção de *chatterbots*, são elas:

- **Casamento de padrão simples e regras gramaticais:** o usuário digita a frase de entrada, o interpretador faz uma seleção das palavras-chaves, compara com o conhecimento do *chatterbot* e devolve para o usuário um novo conjunto de palavras de forma estruturada, no formato de frase.
- **Redes de Ativação:** Em uma Rede Conversacional, cada nó é representado por um grupo de padrões, uma resposta simples, uma lista de nós estimulados e outros inibidos. Quando uma entrada do usuário é recebida, o sistema determina qual padrão será processado e os nós que possuem este padrão serão despertados, sendo o de maior nível de estimulação escolhido, transmitindo em retorno a resposta ao usuário (JACOB JUNIOR, 2008).
- **RBC (Raciocínio Baseado em Casos):** Outra técnica de Inteligência Artificial bastante conhecida é a RBC, que consiste na busca de uma solução para uma situação atual através da recuperação e adaptação de soluções passadas semelhantes, dentro do mesmo domínio do problema (FERNANDES, MIRANDA e SANTOS, 2003).
- **AIML:** AIML é uma linguagem de descrição que permite a inserção de dados na base de conhecimento de um *chatterbot*. O casamento é

feito através das palavras e dos caracteres coringas (“*” e “_”) que funcionam como um conjunto de palavras qualquer que pode substituir esses dois caracteres. Em AIML, a busca na base de conhecimento é orientada através das *tags*, onde cada *tag* é responsável por determinar um bloco de comando responsável por uma determinada função.

3. Metodologia

Para o desenvolvimento deste trabalho foi criado para estudo de caso um *chatterbot*, cuja base de conhecimento foi implementada através da linguagem de marcação AIML. O objetivo desse *chatterbot* é ensinar as estruturas de repetição ‘PARA’ e ‘ENQUANTO’ e o condicional ‘SE’, da linguagem PORTUGOL.

Para realizar uma avaliação foram escolhidas três objetivos de avaliação aplicados ao *chatterbot* desenvolvido, foram elas:

- **Coerência na conversação:** A criação de “ilusão de vida” tem sido amplamente empregada na construção de atores sintéticos, também conhecidos como agentes credíveis (*believable agents*), ou agentes virtuais (*virtual agents*) (SILVA, 2000). Esses agentes são geralmente utilizados em ambientes educacionais ou de entretenimento (ROUSSEAU e HAYESROTH, 1997). A personalidade, incluindo atitudes e emoções, é considerada o elemento básico para a criação de “ilusão de vida” perante o usuário (REILLY e BATES, 1995; ROUSSEAU e HAYES-ROTH, 1997), processo esse que torna os agentes credíveis, outro fator fundamental é a coerência durante a conversação, utilizando expressões corretas e mantendo o sentido correto da conversação. À medida que os agentes se tornam mais capazes de simular essa ilusão, as ligações sociais formadas entre agentes e humanos tendem a se tornar mais fortes (ELLIOT e BRZEZINSKI, 1998).
- **Curva de aprendizagem:** Esse mecanismo será responsável por avaliar o grau de dificuldade que o uso da técnica de casamento de padrão, usando o AIML como estudo de caso, possui para o desenvolvimento de um *chatterbot*.
- **Manutenção do código:** muitas vezes após o desenvolvimento de um *chatterbot* é necessário a inclusão de novos conhecimentos na base de dados do agente para que o mesmo seja capaz de abordar novos assuntos e/ou elaborar novas respostas. Para avaliar isso, iremos parametrizar a

mudança de alguns conteúdos e enumerar as mudanças necessárias para que o diálogo do *chatterbot* volte a ser coerente novamente.

4. Considerações sobre os objetos de avaliação

4.1 – Coerência de conversação

Os testes foram realizados com dez diferentes usuários, com diferentes sexos, profissões e nível de escolaridade. Em relação às características os entrevistados atribuíram, em 80,7% das réplicas, adjetivos positivos ao personagem, como “esperto”, “didático”, “atencioso”, “prestativo”, “interessante” e “evolutivo”. Já 19,3% englobavam as características negativas, como “apático”, “chato” e “pouco esperto”.

Logo após foi perguntado, de modo geral, quando o usuário achou que o personagem apresentou um comportamento que atribuisse “ilusão de vida”. O resultado é mostrado no gráfico a seguir:

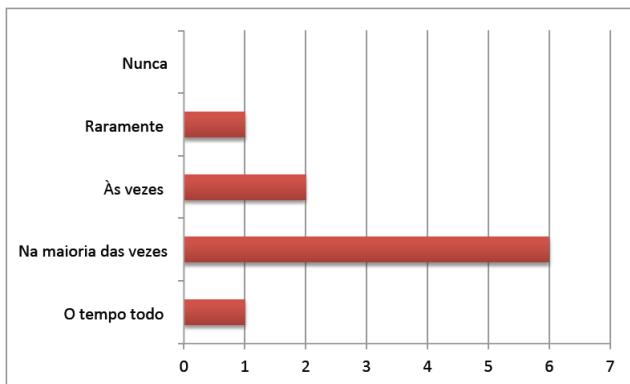


Figura 1 - Frequência com que o *chatterbot* apresentou comportamento coerente.

Por fim, foi solicitado aos entrevistados que eles classificassem os elementos identificados no personagem identificados como: representação visual, representação de emoções, objetivos claros, coerência na conversação e memória.

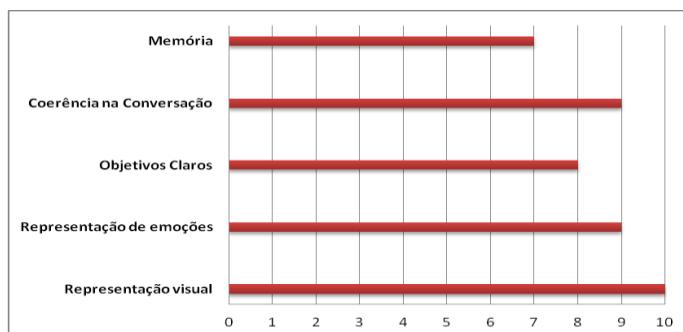


Figura 2 - Características percebidas no *chatterbot*

4.2 – Curva de aprendizagem

Para desenvolvimento do protótipo que serviu como estudo de caso, a curva de aprendizado da linguagem foi muito baixa, a linguagem utilizada é de fácil

entendimento, sendo possível criar uma base de conhecimentos básica para um *chatterbot* sem muitas dificuldades. O tempo de aprendizado da linguagem AIML foi baixo, menos de uma semana de estudo, diferente do que ocorre com outras linguagens de programação, como C, C++, Java, etc.

Para entender o funcionamento da linguagem, basta apenas aprender a funcionalidade de cada *tag*, o que não é um trabalho complicado, tendo em vista que a linguagem possui um pequeno número de *tags* e apenas 12 delas foram utilizadas. Para aprender AIML não é necessário um conhecimento avançado de programação, apenas requer muito trabalho para estruturação do conhecimento.

4.3 – Manutenção de código

Para desenvolver o *chattebot*, com as características apresentados, foi necessário a criação de 317 categorias.

Alguns *chatterbots* deixam o foco da conversação nas mãos do usuário, sem tomar iniciativa em um determinado assunto, esperando que o usuário fale alguma coisa, para buscar um padrão correspondente. Para esse tipo de *chatterbot* o custo de manutenção é basicamente nulo, não precisando alterar outros padrões quando é necessário a inserção de novas assertivas, a remoção e/ou a modificação de padrões já existentes na base de conhecimento. O grande problema desse tipo de *chatterbot* é o elevado número de padrões que se faz necessário criar, pois o mesmo precisa prever as inúmeras possibilidades de entrada do usuário.

Outros *chatterbots*, como o desenvolvido, retêm o foco da conversa para si mesmo, fazendo perguntas para o usuário e direcionando a conversa para o domínio desejado, esses *chatterbots* precisam de um número menor de padrões para representar o conhecimento, pois o mesmo limita a resposta do usuário a uma pequena quantidade de possibilidades. Porém, a manutenção do seu código não é tão simples, pois a sequência de assuntos estão interligados entre si, ou seja, a inserção, remoção e/ou modificação de padrões pode influenciar um grande número de padrões existentes, requerendo um grande esforço para tornar o código novamente coerente. Por exemplo, caso seja necessário fazer com que o *chatterbot* desenvolvido não seja mais capaz de ensinar a estrutura condicional ‘SE’, é preciso remover 70 categorias, ou seja, 70 padrões. O mesmo acontece quando se quer modificar, por exemplo, ensinar outro comando no lugar do ‘SE’.

Para esse tipo de *chatterbot*, o único caso em que a manutenção não requer um esforço grande é quando é necessário realizar a inserção de um novo padrão, por exemplo, para fazer o *chatterbot* ter a capacidade de ensinar um novo comando não é necessário modificar nenhum padrão, apenas ir inserindo novas categorias de acordo com a estrutura da conversa.

5. Considerações finais

Esse trabalho consistiu em um estudo descritivo sobre uma forma de se utilizar casamento de padrões na construção de agentes conversacionais, mais precisamente utilizando-se da linguagem de descrição AIML. Através dos resultados obtidos na avaliação, percebe-se que o uso dessa técnica constitui-se de uma maneira simples e eficaz de construir uma base de conhecimento.

Dentre os pontos positivos foram destacados a coerência da conversação e a curva de aprendizado, mostrando que esses fatores podem ser satisfatórios simultaneamente, ou seja, por mais que o uso de uma técnica de casamento de padrões, como a linguagem AIML, seja bastante simples de se utilizar, ela possui a capacidade de construir um *chatterbot* capaz de se comunicar com o usuário de maneira coerente.

Outra métrica apresentada foi a manutenção do código, que dependendo do caso pode ser realizada sem muito esforço, porém em alguns casos se torna um fator desfavorável ao uso da técnica, devido a elevada quantidade de padrões que são necessários para se criar uma conversa inteligente.

Assim como outras técnicas, o uso de casamento de padrões possui pontos fortes e fracos, que se bem trabalhados pode construir um *chatterbot* que interaja com o usuário de maneira satisfatória, fato comprovado pela apresentação do protótipo desenvolvido e de casos de sucesso de outros *chatterbots* desenvolvidos com o uso dessa técnica.

Referências

ELLIOT, C.; BRZEZINSKI, J. Autonomous Agents as Synthetic Characters. AI Magazine,19(2),1998.

FERNANDES, A. M. R.; MIRANDA, E. M.; SANTOS, F. G. Raciocínio baseado em casos. In: Inteligência artificial: noções gerais. Ed. Visual Books, Florianópolis, 2003.

LEMOS, E. ; DESENVOLVIMENTO DE CHATTERBOTS EDUCACIONAIS: UM ESTUDO DE CASO VOLTADO AO ENSINO DE ALGORITMOS. Dissertação de Mestrado,UFRN,2011.

REILLY, S.; BATES, J. Natural Negotiation for Believable Agents. School of Computer Science, Carnegie Mellon University, CMU-CS-95-164, June 1995.

ROUSSEAU, D.; HAYES-ROTH, B. A Social-Psychological Model for Synthetic Actors. Technical Report KSL-9707, Knowledge Systems Laboratory, Stanford University,1997.

SILVA, D. Atores Sintéticos em Jogos de Aventura Interativos: O Projeto Enigmas no Campus. Dissertação de Mestrado. Universidade Federal de Pernambuco, Recife, 2000.

Aplicação de mineração em dados coletados por radiosondagens meteorológicas

Francisco P. F. Neto¹, Angélica F. Castro^{1,2} Rafael C. G. Martins²

¹ Programa de Pós-Graduação em Ciência da Computação– UERN/UFERSA
Avenida Francisco Mota, 572, Costa e Silva – 59.625-900 – Mossoró – RN – Brasil

²Departamento de Ciências Exatas e Naturais – Universidade Federal
Rural do Semi-Árido (UFERSA)
Avenida Francisco Mota, 572, Costa e Silva – 59.625-900 – Mossoró – RN – Brasil

paulo.freitas.nt@gmail.com, {angelica, rcastelo}@ufersa.edu.br

Abstract. This paper presents the use of data mining techniques applied in the field of meteorology. In it is also described the process of radiosonde, method by which the data base of this study were collected, as well as the methodology used to make possible the realization of the process of knowledge discovery in database. Also presented are the results of performing some experiments that can be done with the help of the tool WEKA.

Resumo. Este artigo apresenta a utilização de técnicas para mineração de dados aplicadas no campo da meteorologia. Nele também é descrito o processo de radiosondagem, método pelo qual os dados base desse estudo foram coletados, assim como a metodologia utilizada para que fosse possível a realização do processo de descoberta de conhecimento nesse banco de dados. Além disso são apresentados os resultados da realização de alguns experimentos que podem ser feitos com o auxílio da ferramenta WEKA.

1. Introdução

Com os avanços dos estudos na área de ciência da computação, é comum a utilização das ferramentas e técnicas desse campo em diversas outras áreas como: medicina, biologia, ciências sociais, meteorologia e diversas outras. A computação auxilia os profissionais destas áreas em tarefas onerosas como análise de dados, tomada de decisões, etc. Assim, os estudiosos dessas áreas tendem a dispensar seus esforços à outras atividades mais específicas da área, tornando assim o seu trabalho mais rápido e menos oneroso.

Em meteorologia, a computação pode ser utilizada em diversas aplicações como previsão numérica de tempo, análise de fenômenos climáticos, previsão de eventos, caracterização de perfis atmosféricos, dentre outros. Uma outra possível aplicação nesse campo é a utilização de técnicas para descoberta de conhecimento em banco de dados oriundos de radiosondagens, que visa a caracterização de padrões nesses dados e pode auxiliar os meteorologistas em tarefas diversas. Dessa maneira, visa-se com este trabalho a apresentação da técnica de mineração de dados aplicado à área de estudo da meteorologia no intuito de auxiliar o profissional da área na realização dos seus trabalhos. Os dados utilizados no estudo foram obtidos através da realização de radiosondagens em três municípios distintos visando com isso a caracterização da região compreendida entre

eles. Existem diversos trabalhos relacionados nesta área, dentre os quais podemos citar [Salvador et al. 2009] que utiliza a mineração de dados para detectar inconformidades em dados meteorológicos.

O trabalho é dividido da seguinte maneira: a seção dois apresenta a metodologia utilizada para a obtenção dos dados, construção do banco e conexão da ferramenta WEKA com o banco. A terceira seção exibe os resultados obtidos com o estudo e a quarta apresenta as conclusões e trabalhos futuros que podem ser produzidos a partir deste.

2. Metodologia

A radiossondagem é uma das técnicas mais comuns para obtenção de dados relativos às camadas superiores da terra. A técnica consiste de utilizar balões dotados de sensores que medem periodicamente dados relativos à temperatura, pressão, umidade relativa, velocidade do vento, dentre outros. E enviar essas informações a uma estação base localizada em terra. Os balões utilizados neste estudo conseguem medir dados de alturas compreendidas na faixa de 0 a 27.000 metros aproximadamente, pois não existe uma altura fixa para haver a ruptura desses balões.

Foram realizadas radiossondagens nos municípios de Mossoró–RN, Fortaleza–CE e Quixeramobim–CE. O estudo tem o intuito de caracterizar a triangulação compreendida entre estas cidades Ver (Figura 1). As medições foram realizadas nos horários sinóticos (0h, 6h, 12h, 18h) *Universal Coordinated Time* (UTN) definidos pela Organização Mundial de Meteorologia (OMM). A coleta dos dados foi financiada pelo projeto CHUVA FAPESP (2009/15235-8) da Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) e que envolve diversas entidades nacionais e internacionais como: Instituto Nacional de Pesquisas Espaciais (INPE), Agência Espacial Brasileira (AEB), *National Aeronautics and Space Administration* (NASA), Fundação Cearense de Meteorologia e Recursos Hídricos (FUNCeme) e Universidade Federal Rural do Semi-Árido (UFERSA).



Figure 1. Triangulação das radiossondagens
[Google 2012]

Ao todo se tem dados de 37 radiossondagens no município de Mossoró–RN no intervalo de 08/04/2011 e 18/04/2011, 70 no município de Fortaleza–CE entre os dias

01/04/2011 e 27/04/2011, e 37 em Quixeramobim–CE no intervalo entre 06/04/2011 e 17/04/2011. As informações coletadas pelas radiossondas têm uma formatação própria, então foi necessária a formatação dos mesmos para assim, possibilitar a inserção em um banco de dados MySQL utilizando a linguagem SQL. Além disso, foi necessária a remoção de algumas variáveis que não são úteis neste estudo (mas podem ser em outros) como: tempo de subida, que está mais relacionada com informações da subida da radiosonda, além de direção e velocidade do vento que estão voltadas a representar o estado do vento na região. No total existem 238.755 instâncias neste banco de dados.

A ferramenta utilizada para a realização do processo de mineração de dados foi a *Waikato Environment Knowledge Analysis* (WEKA) que segundo [Witten et al. 2011] corresponde a um agrupamento de algoritmos e ferramentas utilizadas na aplicação mineração de dados. Trata-se de um sistema gratuito, de código fonte aberto, desenvolvida por pesquisadores da Universidade de Waikato (Nova Zelândia) que possui diversos algoritmos implementados e permite a conexão com vários softwares SGBDs como o MySQL, PostgreSQL, dentre outros.

3. Resultados

O WEKA permite a aplicação de diversas técnicas de mineração de dados, foram utilizadas nesse trabalho, técnicas de sumarização e classificação de dados, mas a ferramenta também permite diversas outras como a agrupamento e associação.

3.1. Sumarização dos dados

A sumarização dos dados no WEKA é uma tarefa relativamente simples, pois a própria interface do ambiente *Explorer* realiza esta tarefa, apresentando ao conectar com os dados informações referentes a média, mediana, moda, desvio padrão, dentre outros na aba *Process*, além de plotar as relações entre todos os atributos entre si com o uso da aba *Visualize* como pode ser verificado em (Figura 2).

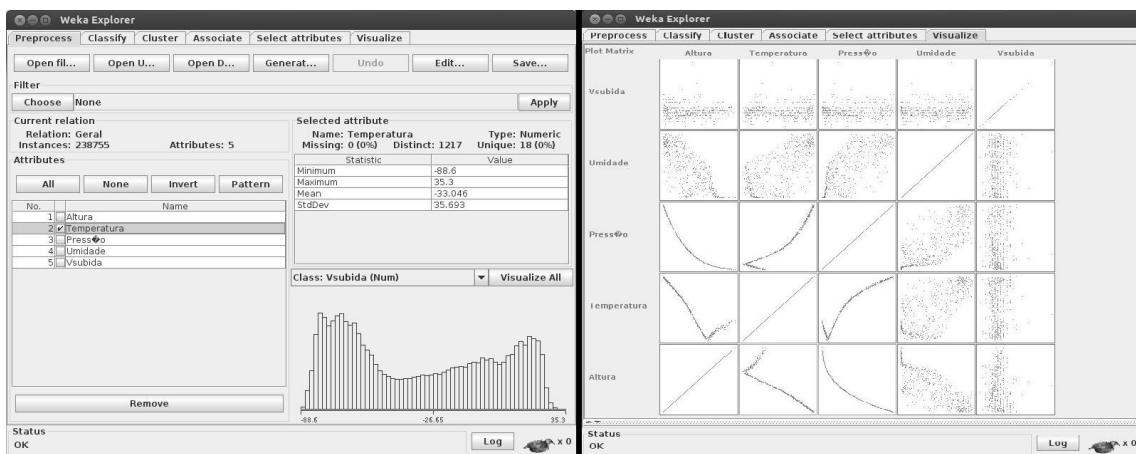


Figure 2. Abas *Explorer* e *Visualize* do WEKA

Estas informações servem como base para o início do estudo, pois tornam visível a existência de algum padrão de variação entre os parâmetros: temperatura, pressão e altitude. Outra importância da realização da sumarização é a detecção de dados ruidosos, que são informações corrompidas por alguma falha na comunicação entre o balão e a

radiossonda. Por falta de uma ferramenta específica para análise dos dados obtidos pelas radiosondagens, os estudiosos da área por muitas vezes têm que realizar a plotagem de todos os gráficos um a um até perceberem em qual leitura houve alguma informação ruidosa.

De acordo com [Wallace and Hobbs 2006] a estrutura vertical da atmosfera tipicamente apresenta um comportamento semelhante ao descrito na (Figura 3). Por outro lado, os balões utilizados neste estudo só atingem alturas na faixa de 0 a 27.000 metros, ou seja as únicas camadas pelas quais os balões passam são: Troposfera, tropopausa (região de transição entre a troposfera e a estratosfera) e o início da estratosfera. Porém, nessa região a pressão atinge valores próximos a zero, e como a pressão externa é menor que a interna do balão há uma expansão do mesmo e o material não resiste causando assim a sua ruptura, encerrando assim a radiosondagem.

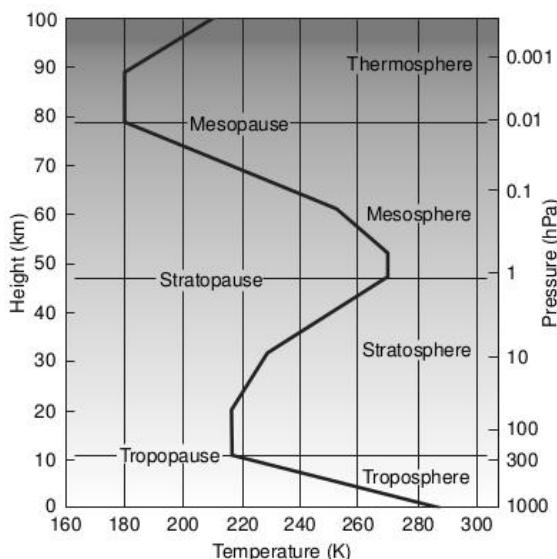


Figure 3. Estrutura vertical da atmosfera
[Wallace and Hobbs 2006]

Analizando a dispersão da temperatura em função da altura percebe-se que o comportamento é semelhante ao descrito na figura 3, porém com diferenças em relação a altitude da camada tropopausa e a sua extensão, ver (Figura 4)

Pela figura 4, temos que a tropopausa na região da triangulação possui uma altura por volta de 14 a 16 km de altitude, pois é o local onde a temperatura interrompe o decrescimento, tem uma pequena estabilidade e então começa ter os seus valores aumentados. Essa informação é importante, pois a determinação da altura da tropopausa interfere diretamente na aviação comercial, como pode ser visto em [Lyra et al. 2006] que relaciona a tropopausa à turbulências nos voos devido ao grande fluxo de ar nessa região.

3.2. Classificação dos dados

De acordo com [Wallace and Hobbs 2006] a temperatura diminui com o aumento da altitude, a uma taxa relativamente uniforme. Pois a atmosfera é aquecida por condução a

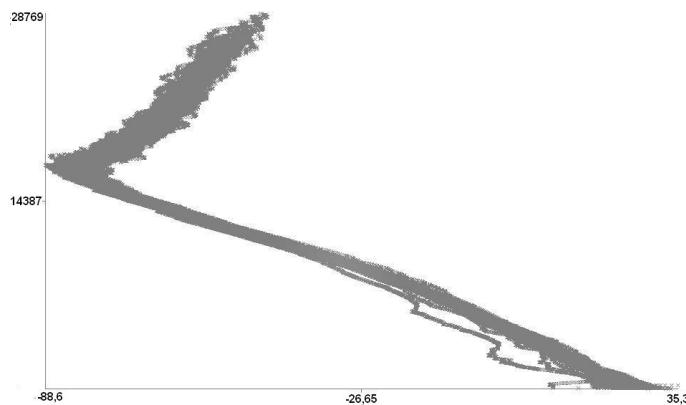


Figure 4. Análise da dispersão da temperatura em função da altura

partir da superfície da Terra. O *lapse rate* corresponde a esta taxa de decréscimo da temperatura. A definição dessa taxa é importante pois serve como entrada para softwares de previsão numérica do tempo, além de descrever como a temperatura varia na região da coleta dos dados. Existem diversas maneiras dessa informação ser determinada, uma delas é a utilização de algum algoritmo para classificação de dados como a regressão linear que determina a equação de ajuste para todos os valores de temperatura do perfil médio atmosférico da região estudada.

Aplicando-se o algoritmo de regressão linear nos dados entre 0 e 14.000 metros (troposfera na região) para as variáveis altura e temperatura, obtém-se a função $temperatura = -0,0064 * altura + 29,7674$ sendo temperatura descrita em graus Celsius ($^{\circ}\text{C}$) e altura representado em metros (m). Desse modo, pode-se definir a taxa de variação da temperatura na região como $-6,4^{\circ}\text{C/Km}$ que é um valor semelhante ao presente na literatura da área. A regressão tem um coeficiente de correlação igual a 0,9938 o que indica que a equação tem uma boa representatividade estatística.

4. Conclusões

Com a realização deste trabalho, percebe-se que as técnicas de mineração de dados podem ser utilizadas para o estudo de dados coletados por radiosondas. Além disso, existem diversas aplicações na área que ainda podem ser exploradas como a utilização de modelos minerados em sistemas de previsão numérica do tempo, construção de softwares de apoio, dentre outros. Os padrões descobertos durante o estudo são válidos pois todos já estão presentes na literatura da área, mas foram descobertos por outros métodos. Por outro lado existem diversos outros padrões que ainda não foram descobertos e podem ser através da utilização da mineração de dados. Este é o real intuito deste trabalho, que visa realizar uma introdução ao tema.

5. Agradecimentos

Os autores agradecem a CAPES pela concessão da bolsa de estudos ao primeiro autor, aos financiadores do projeto CHUVA FAPESP (2009/15235-8) que originou os dados para a realização deste estudo e ao Programa de Pós-Graduação em Ciência da Computação (UERN/UFERSA) do qual o primeiro autor é aluno.

References

- Google (2012). Google earth. Disponível em: <http://www.google.com/earth/index.html>. Acessado em: 14 set. 2012.
- Lyra, A. A., Chan, C. S., and Dereczynski, C. P. (2006). Indicadores de turbulências a partir de previsões do modelo original eta. *Revista Brasileira de Meteorologia*, 2(2):161 – 182.
- Salvador, H. G., da Cunha, A. M., and Corrêa, C. S. (2009). Vedalogic – um método de verificação de dados climatológicos apoiado em modelos minerados. *Revista Brasileira de Meteorologia*, 24(4):448 – 460.
- Wallace, J. M. and Hobbs, P. V. (2006). *Atmospheric science: an introductory survey – 2nd edition*. Elsevier.
- Witten, I. H., Frank, E., and Hall, M. A. (2011). *Data Mining – Practical machine learning tools and techniques – 3rd edition*. Elsevier.

AI Quest: Um jogo em HTML5 para aprendizado de algoritmos

Juciény D. Santos Jaime Bruno C. Oliveira Hivana A. M. Macedo Raul B. Paradeda

UERN – Universidade do Estado do Rio Grande do Norte

UFRN – Universidade Federal do Rio Grande do Norte

{juciényds, jaimecirne, hivanamaced, paradeda}@gmail.com.br

Resumo. Em diversas áreas do conhecimento as técnicas de construção de algoritmos são de grande importância para resolução de problemas. Entretanto, há um tabu entre os alunos que cursam esta disciplina, devido a alguns problemas como a carência no raciocínio lógico e a falta de conhecimento nas ferramentas para a implementação desses algoritmos. Por esse motivo, esta disciplina apresenta um dos índices mais elevados de reprovações. Esse projeto se propõe a desenvolver um jogo educacional que servirá como ferramenta para o auxílio ao ensino desta disciplina. Ele trabalhará de maneira lúdica para que o jogador possa aprender conceitos básicos da mesma. Para a construção dessa ferramenta foram empregadas tecnologias Web como a especificação do HTML5, JavaScript e CraftyJs.

Palavras-chave: HTML5, Java Script, CraftyJS, jogos, ferramenta para educação, algoritmo;

1. Introdução

A web é hoje uma plataforma poderosa na qual algumas tecnologias, protocolos e novos padrões vêm tornando esse ambiente bastante rico para o desenvolvimento de diversos tipos de aplicações. Isso se deve ao fato, de haver a união de características únicas, como a independência de sistemas operacionais e a fácil disponibilidade do acesso à internet. Desse modo, acaba se tornando em um interessante meio para se investir em novas abordagens.

Foi através dessas vantagens, encontradas no ecossistema web, que se planejou a implementação de uma ferramenta que visa ajudar de maneira prazerosa e divertida, um determinado público, a romper dificuldades no aprendizado de conceitos ligados a algoritmos. De tal modo, será expressa no decorrer deste trabalho uma proposta que apresenta uma maneira de auxiliar nessa problemática. De forma sucinta, a proposta consiste em desenvolver uma ferramenta que se resume a um jogo educacional lógico, finito e formado por níveis que aumentam sua complexidade de acordo com as vitórias obtidas. O jogador, para alcançar seu objetivo, deverá fornecer instruções algorítmicas específicas do jogo, dando assim os seus primeiros passos no desenvolvimento de sua capacidade de abstração e resolução de problemas.

Um dos desafios encontrados para o desenvolvimento do projeto proposto foi referente às questões sobre a escolha das tecnologias que seriam utilizadas. Isso se deve

ao fato, de haver nos dias atuais, uma vasta quantidade de tecnologias que poderiam ser aplicadas na criação do jogo. Entretanto, as tecnologias foram escolhidas baseadas no conhecimento a priori dos autores do projeto, que são: HTML5, e como foco principal o Java Script.

2. Problemática do ensino de algoritmos

São diversos os fatores complicadores do aprendizado de algoritmos, fatores esses, objetos de estudos de diversos trabalhos. Olsen [2005] enfatiza que há falta de habilidade para desenvolver soluções de problemas apresentados nos cursos. Acadêmicos como em Tobar [2001], elencam como problema a preocupação excessiva com os detalhes sintáticos da linguagem usada, a falta de visão de como solucionar o problema e por consequência o desenvolvimento de um plano para solucionar o mesmo e por fim a dificuldade de sintetizar um raciocínio lógico em um modelo incremental.

Há uma importância fundamental no que diz respeito ao processo de aprendizagem dos conceitos introdutórios da disciplina de algoritmos. O entendimento adequado dos problemas e a compreensão de suas possíveis soluções são nitidamente necessários no exercício do raciocínio lógico e da resolução de problemas. Como dito em Dijkstra [1982], mais que em qualquer outra atividade, programar envolve a habilidade de raciocinar.

Entretanto, as técnicas de solução de problemas em si, raramente são absorvidas pelos estudantes em cursos introdutórios, conforme McKeown [1999], nesses cursos é comum que os estudantes se deparam com dificuldades em aplicar suas habilidades prévias. O que acaba tornando-se uma fonte de frustração, incentivando a evasão. Estudos envolvendo diversas instituições de ensino brasileiras apontam para um número consideravelmente grande de reprovações em disciplinas de introdução à programação. Segundo Pereira Junior [2004], esses estudos apontam que durante o processo de ensino-aprendizagem de fundamentos de programação, grande parte dos alunos apresenta dificuldades em assimilar as abstrações envolvidas, assim como, de concretizar soluções a partir dessas abstrações. De acordo com Mayer [1988], o ensino de algoritmos e programação pode ser analisado em três fases: inicial, montagem de dados e pesquisa, cada uma dessas fases é marcada por características distintas, vê-se que nas fases iniciais é onde existe a maior ocorrência de problemas por parte dos estudantes.

Assim, diante de tais pontos, os problemas citados dão subsídios ao desenvolvimento de propostas de ensino de algoritmos designando jogos educacionais na web como ferramenta metodológica.

3. Proposta da ferramenta

Entendendo os problemas de aprendizado na construção de algoritmos, aliado ao interesse gerado em ajudar nessa dificuldade, foi pensado neste projeto uma maneira que estimule o raciocínio lógico do usuário utilizando o interesse proporcionado pelos jogos de computadores. Os jogos na Web podem ser uma ferramenta poderosa para criar motivação, as redes sociais são um exemplo disso que utilizam como estratégia o incentivo a criação e utilização de jogos a fim de manter uma gama de usuários conectados em suas páginas web. Segundo Tarouco [2005], jogos são fatores

motivadores da aprendizagem e ainda aumentam a capacidade de retenção do conhecimento.

Tendo isso em perspectiva, o desenvolvimento de um jogo educacional com essas características e a possibilidade de aprimorar a capacidade de análise e resolução de problemas seria uma solução perfeita para o treinamento e aprendizado de algoritmos. O trabalho apresentado por Johnson [2005] informa que os jogos desenvolvem a capacidade dos alunos de tomar decisões, a escolher e priorizar o melhor resultado através de experiências, ele também compara a maneira de pensar dos jogadores com as do programador, quanto a escrever um código. A função dessa ferramenta é justamente explorar essa relação similar dessas atividades mentais e também a fim de educar os jogadores quanto a resolução de problemas por algoritmos.

4. Jogos educacionais

Os jogos ao longo de toda nossa história nos acompanham desde tempos remotos, e em caráter geral tentam promover a diversão e a estimular os jogadores a reagir aos seus desafios. De certa forma, eles podem ser usados como ferramentas instrucionais que facilitam o aprendizado ao mesmo tempo em que divertem e motivam seus jogadores aumentando a capacidade de absorção do conteúdo que está sendo transmitido durante o entretenimento. Além disso, proporcionam também a melhora da flexibilidade cognitiva, aumentando a rede de conexões neurais e alternando o fluxo sanguíneo no cérebro quando em estado de concentração.

Quando utilizados dessa forma, os jogos podem ser classificados como jogos educacionais que se constituem por qualquer atividade de formato educacional, ou seja, que se propõem a ensinar, envolvendo a competição regulada por regras e restrições bem definidas. Jogos educacionais baseiam- se em uma abordagem de ensino onde o sujeito possa aprender sozinho utilizando apenas as descobertas obtidas com sua relação e interação com o software,

isso o torna interessante de certa forma, pois ajuda ao jogador a encontrar sua independência na busca do conhecimento. É importante ressaltar que, o uso dessas aplicações nas dinâmicas de ensino deve ser feito com um conhecimento prévio bem fundamentado, para que se saiba exatamente o que se deseja atingir. Atualmente, temos uma variedade de estilo de jogos que podem ser explorados para esses fins acadêmicos, como ação, aventura, lógica, estratégia e RPG (Role- Playing Game), onde em alguns casos cabe ao orientador fazer esta escolha de estilo. Um dos estilos que mais se destaca na criação de jogos educativos é o RPG, nele é possível que o educador possa criar e adaptar qualquer tipo de cenário e até mesmo adicionar outros estilos como, ação, lógica ou estratégia em seu enredo. Nesse estilo o jogador tem a liberdade para tomar qualquer atitude. Existe uma gama de jogos com propósito em ensinar temas complexos, geralmente abordando aqueles que são difíceis de serem assimilados por não existirem aplicações práticas, como conceitos em eletrização, trigonometria, conservação de energia, entre outros.

4.1 Jogos educacionais na web

Com o passar dos anos a internet tem ficado cada vez mais popular, o que a torna um ambiente propício para divulgação e desenvolvimento do conhecimento, tendo isso em

mente podemos aproveitar esse fator e utilizar novas estratégias para melhorar o processo de aprendizado. Dentre essas estratégias, existem os jogos educacionais via Web que possibilitam tanto a aprendizagem assíncrona quanto a coleta de dados, importantes para determinadas pesquisas. Para melhor aproveitamento, essas aplicações devem ser fáceis de interagir e rápidas em suas respostas, os jogos também devem manter a transparência de maneira que esconda o seu funcionamento para o usuário, assim como, também serem tolerantes a faltas. Essas recomendações são importantes para não afetar a motivação de jogar a aplicação e incrementam no valor do software.

5. Ferramentas e tecnologias envolvidas no projeto

Foi escolhida para a implementação do jogo a plataforma Web que, além de suportar as diretrizes como a fácil distribuição e portabilidade, oferecem ainda características inovadoras podendo culminar em incrementos valiosos nas perspectivas futuras do projeto. As tecnologias mais relevantes utilizadas no projeto são:

5.1 HTML e HTML5

HTML é uma linguagem de marcação interpretada por navegadores para renderização de páginas Web. Sua concepção veio a partir de duas linguagens já existentes a HyTime e SGML.. Com a internet e a difusão da Web, nesse ambiente, o HTML passou a ser a tecnologia favorita para distribuição de conteúdo na rede, com o passar do tempo sofreu uma série de atualizações e linguagens de suporte como o CSS (Cascading Style Sheets) e JavaScript. Em sua quinta versão, a linguagem de marcação sofre mudanças bem drásticas em relação às outras atualizações inserindo uma nova API para desenhos bidimensionais, controle nativo para multimídia, aprimoramentos da depuração de erros e melhor adaptação ao modo off-line.

5.2 JavaScript

JavaScript é uma linguagem de script com orientação a objetos baseada em protótipos, sua sintaxe é semelhante a C e ao C++, possui tipagem fraca e dinâmica e também suporte a programação funcional. Hoje, ela é a principal linguagem de programação para o lado cliente em navegadores web. Foi originalmente criada por Brendan Eich da Netscape, a princípio foi denominada de Mocha, porém ainda teve seu nome modificado para LiveScript e então por fim obteve o que conhecemos hoje como JavaScript. Esta é utilizada para escrever funções em páginas HTML que interagem com o Modelo de Objetos de Documentos, com ela é possível dar também interatividade com os usuários, como melhorando a acessibilidade para usuários deficientes visuais e a compatibilidade em muitos ambientes.

5.3 CraftyJs

CraftyJs é uma biblioteca de código aberto desenvolvida em JavaScript para jogos em HTML5, se encontra sob licença MIT e GLP (General Public License). Ele oferece ao programador algumas facilidades que agilizam no desenvolvimento de jogos, ou seja, focando o desenvolvedor apenas no que realmente interessa a lógica da sua aplicação.

Esta biblioteca contém características de suma importância para o desenvolvimento deste projeto, além de sua licença livre ainda oferece compatibilidade entre navegadores, suporte a mapas de imagens, customização de eventos, detecção de colisão e vários outros componentes nativos como sons, animações e efeitos.

6. Funcionamento da ferramenta

De acordo com o que foi mencionado anteriormente, o projeto trata do desenvolvimento de um jogo educacional lógico denominado de Alg Quest voltado para Web, onde o jogador ajudará o personagem a percorrer vários desafios em um determinado mapa cheio de obstáculos, a fim de conseguir concluir o principal objetivo, que é o de chegar ao tesouro. Para isso, o jogador com auxílio de comandos específicos deverá encontrar a melhor solução para resolução do problema, o jogador deverá estar atento ao tempo e também ao número de tentativas para conseguir assim uma boa pontuação.

6.1 Personagem



Figura 1: Al, o personagem principal.

O personagem principal chama-se Al (Figura 1), ele será o elemento determinante que agirá de acordo com o código que será formado a partir dos comandos inseridos pelo jogador.

6.2 Comandos

Os comandos são como as funções que encontramos em técnicas de programação estruturada, eles irão coordenar a movimentação e a lógica da resolução do problema encontrado em cada fase do jogo. Eles são sequenciais e devem ser escritos na sua maneira extensa ou usando suas abreviações, devem obrigatoriamente ser divididos através de quebras de linhas. Foram criados cinco comandos. São eles: Mover (M), Esquerda (E), Direita (D), Bater (B), Esperar (Es). Os Comandos, Esquerda e Direita, são utilizados para rotacionar o personagem para a direção desejada, deve-se sempre estar atento a perspectiva que se encontra o personagem no jogo para não confundir a direção que se deseja fazer. O comando Bater é apenas destinado para cortar arbustos e para finalizar a fase ao abrir o tesouro. Mover e Esperar são os comandos responsáveis pela gerencia do movimento do personagem.

6.3 Obstáculos e componentes

Cada nível do jogo é composto por um mapa organizado com problemas diferentes, esses mapas contêm vários obstáculos, o jogador terá que analisar qual a estratégia a se usar para driblar esses problemas. Os principais obstáculos e componentes são:

Arbusto – Bloqueia a passagem do personagem, porém pode ser devidamente cortado através do comando Bater (B). Pedra comum – Bloqueia a passagem do personagem, mas assim como o arbusto, o personagem pode interagir com este obstáculo através de comandos. Essa pedra pode ser movida de acordo com a direção do personagem ao utilizar o comando Mover (M).

Buraco – Este obstáculo não bloqueia a passagem, porém, pode levar a morte do personagem. Portal – Sua função é mover o personagem para qualquer área do mapa livre.

Pedra imóvel – Esta pedra não poderá ser movida ou quebrada, sua única função é bloquear a passagem do personagem. Tesouro - O tesouro é o objetivo principal do jogo. Para abrir o tesouro o jogador deverá utilizar o comando Bater (B).

6.4 Detalhes da interface do jogo

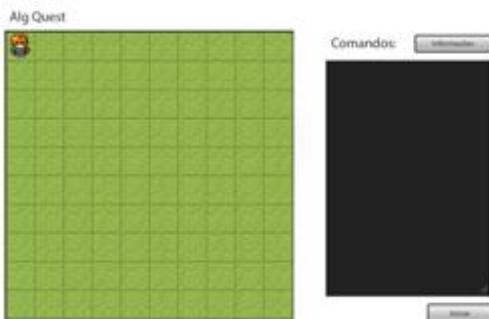


Figura 2: Tela principal do Alg Quest

A tela principal do Al Quest (Figura 2) é dividida em duas áreas distintas. A tela de jogo e a tela de comandos. Na tela de jogo, o jogador encontrará o problema que deverá ser resolvido, ele deverá analisar o terreno e os obstáculos contidos nesta tela e assim desenvolver uma solução escrita representada pelos comandos. Esses comandos deverão ser escritos na tela de comandos, onde devem ser feitos de forma sequencial para que a solução desejada seja bem compreendida pelo analisador do jogo. Ainda nesta tela, também existem dois botões, um deles é responsável em ajudar o jogador (Botão Ajudar), que ao ser clicado mostrará informações sobre os comandos e o objetivo final do jogo e o botão para rodar os comandos que foram escritos (Botão Iniciar), que, após clicado, ativará o analisador para fazer a interpretação das ações escritas do personagem.

6.5 Lógica do jogo

Alg Quest tem um cenário bastante comum aos jogos de tabuleiro ou estratégia, porém a melhor estratégia será a que prevalecerá. Para conseguir um bom desempenho no jogo, o jogador deverá estar atento em alguns requisitos para

atingir uma boa pontuação:

6.5.1 Tempo

O tempo é um dos requisitos, ele será utilizado para medir o desempenho feito, quanto mais rápido o jogador conseguir chegar ao seu objetivo melhor será sua pontuação. Para alguns mapas iniciais, o cronometro será apenas iniciado após a confirmação do jogador pelo botão de iniciar encontrado na tela principal.

6.5.2 Tentativas

As tentativas são infinitas, o jogador estará livre para recomeçar o mapa sempre que não atingir o sucesso, porem, a cada nova tentativa sua pontuação máxima

sufre uma diminuição em seu valor. Deve-se estar atento que quanto menos tentativas forem utilizadas, maior será a pontuação máxima. Cada mapa terá uma pontuação que sofrerá efeitos de acordo com o tempo e com esse número de tentativas feitas. O jogador não poderá solicitar uma nova rodada, a tentativa será entendida sempre que o código não consiga obter sucesso, ou seja, não consiga chegar ao tesouro.

7. Conclusão

A proposta do Al Quest é inovadora por haver uma serie fatores, desde a preocupação latente no ensino que de certa forma é ignorada nos cursos de algoritmos e programação, como também na utilização de tecnologias de vanguarda e ainda pela abordagem lúdica da ferramenta. Essas características inovadoras também podem ser estendidas para trabalhos futuros, podendo ser criada uma estrutura competitiva na rede similar as maratonas de programação. Ainda, o aprimoramento da interface de comandos utilizando comandos mais complexos ou com sintaxes mais próximas da programação em C ou Portugol.

Referências

DIJKSTRA, E. (1982). “On the Teaching of Programming, i.e. on the Teaching of Thinking”. In: Selected Writings on Computing: A Personal Perspective. SpringerVerlag, NY.

McKEOWN, J. e FARRELL, T. (1999). “Why We Need to Develop Success in Introductory Programming Courses”. In: CCSC – Central Plains Conference, Maryville, MO.

PEREIRA JÚNIOR, J.C.R., RAPKIEWICZ, C. “O Processo de Ensino-Aprendizagem de Fundamentos de Programação: Uma Visão Crítica da Pesquisa no Brasil”, WEI RJES 2004.

JO NSON, S., 2 . Sur reendente : a tele isão e o videogame nos tornam mais

inteligentes. RJ: Elsevier.

TAROUCO, L. M. R., CONRATH, M. L. P., GRANDO, A. R. 2006. O Aluno como Co-Construtor e Desenvolvedor de Jogos Educacionais CINTED UFRGS.V.3, No 2, novembro 2005.

Olsen, A.L. (2005). Using Pseudocode to Teach Problem Solving. In Journal of Computing Sciences in Colleges, Volume 21, Issue 2, December. Consortium for Computing Sciences in Colleges, USA.

Mayer, R. E. (1988) "Introduction to Research on Teaching and Learning Computer Programming". In: Mayer, R. E. (org.). Teaching and Learning Computer Programming: Multiple Research Perspectives, Lawrence Erlbaum Associates.

Tobar, C. M.; Rosa, J. L. G.; Coello, J. M. A.; Pannain, R. (2001) "Uma arquitetura de ambiente coletivo para o aprendizado de programação". In: Simpósio Brasileiro de Informática na Educação, 12, Vitória, 2001. Anais... Vitória: Universidade Federal do Espírito Santo, 2001.

Comparação da Eficiência entre Aplicações em C e em Python Utilizando a Biblioteca Ctypes

Edgard de Faria Correa¹, Gilles V. T. Silvano¹, Renato G. Sarmento

¹Departamento de Informática e Matemática Aplicada (DIMAp) – Universidade Federal do Rio Grande do Norte (UFRN) – Caixa Postal 1524 – 65.078-970 – Natal – RN – Brasil

gilles@lcc.ufrn.br, edgard@dimap.ufrn.br, renagsarmento@gmail.com

Abstract. This article describes the process of performance analysis of a program written in the C programming language and its implementation using the interface to foreign functions ctypes to access functions written in C through a Python script. We discuss the results for the total memory consumption, execution time and CPU occupancy rate in both implementations.

Resumo. Este artigo descreve o processo de análise do desempenho de um programa escrito na linguagem de programação C e de sua implementação utilizando a interface para funções estrangeiras ctypes de modo a acessar funções escritas em C através de um código em Python. São discutidos os resultados referentes ao consumo de memória total, tempo de execução e a taxa de ocupação da CPU em ambas as implementações.

1. Introdução

Durante muito tempo, a linguagem de programação C vem sendo melhorada e incrementada no que diz respeito na quantidade de funções e bibliotecas testadas e otimizadas [Ravitch 2009]. Este processo resultou em uma linguagem rápida e robusta, e, naturalmente, aplicações com requisitos não-funcionais rígidos recorreram a linguagem C para usufruir de suas características.

Por outro lado, surge cada vez mais aplicações escritas em linguagens de nível mais abstrato que C [Ravitch 2009], como por exemplo linguagens em scripting, tais como Python e Ruby. Este tipo de linguagem possui vantagens em relação as linguagens de mais baixo nível tais como velocidade de desenvolvimento e portabilidade. O desenvolvimento de software utilizando linguagens de scripting também trás desvantagens pois estas necessitam de um programa interpretador que traduz o script escrito na linguagem de alto nível para a um código objeto.

Então, para usufruir de ambas as técnicas, o desenvolvimento ágil somado com a portabilidade e também o desempenho das linguagens compiladas em conjunto com vários anos de testes e desenvolvimento [Kloss 2009], surgiram mecanismos que possibilitam a criação de interfaces para que estruturas de uma linguagem pudessem ser acessadas através de uma linguagem de programação estrangeira.

No caso das linguagens C e Python, existem algumas técnicas que fornecem interface de estruturas estrangeiras escritas em C para serem utilizadas em códigos Python tais como a linguagem Cython [Behnel 2011] e a interface para funções estrangeiras ctypes [Heller 2012], utilizada neste trabalho. Cython possibilita ao programador escrever código em Python utilizando tipos básicos da linguagem C, o que torna o código mais rápido devido as otimizações da linguagem. Apesar do ganho de desempenho, esta técnica não possibilita a reutilização de códigos escritos em C. Este problema é resolvido pela biblioteca ctypes [Ravitch 2009] a qual possibilita o acesso às funções compiladas em C por código estrangeiro.

O restante desse artigo está organizado como se segue. Na Sessão 2, é discutida a biblioteca Ctypes. A Sessão 3 detalha a metodologia de testes. Na Sessão 4 são apresentados os resultados obtidos. E por fim, na Sessão 5 são feitas algumas considerações e discutidos trabalhos futuros.

2. A Biblioteca Ctypes para Python

A biblioteca ctypes é uma ferramenta que fornece uma interface para que funções estrangeiras em C compiladas possam ser utilizadas em outras linguagem, no caso a linguagem Python [Heller 2012].

Para utilizar a biblioteca ctypes, basta criar um objeto compartilhado em C (Figura 1) e carrega-lo através da função CDLL, fornecida pela biblioteca ctypes, dentro do código que vai acessar as funções em C (Figura 2). Esta função retornará um objeto e, através dele, é possível acessar as funções definidas no objeto compartilhado.

```
$ gcc -shared -Wl,-soname,stdio -o stdio.so stdio.h
```

Figura 1. Produção de um objeto compartilhado, chamado stdio.so, a partir da biblioteca stdio.h.

```
#!/usr/bin/env python
import ctypes
str= 123
stdio = ctypes.CDLL('./stdio.so')
stdio.printf('Teste %d\n', str)
1,1          Tudo
```

Figura 2. Exemplo de código em Python utilizando a biblioteca ctypes.

3. Metodologia de Testes

A aplicação alvo consiste em um jogo simples escrito em C que utiliza a biblioteca gráfica Clutter a qual utiliza a API gráfica OpenGL. Os testes foram executados em um sistema com o Ubuntu 12.04 kernel versão 3.2, executado em um notebook com 2 GB de memória RAM DDR2 e um processador Intel Core2Duo 2.0 GHz e HD de 5400 RPM.

Para a obtenção dos dados sobre o desempenho da aplicação foi utilizado uma ferramenta nativa do Linux chamada time que fornece informações como tempo de execução transcorrido no espaço de usuário (sem a execução de instruções privilegiadas) e no espaço de sistema (executando instruções privilegiadas), porcentagem de ocupação da CPU, trocas de contexto involuntárias e voluntárias, entre outras informações importantes.

No caso do cálculo da quantidade de memória alocada foi utilizada a ferramenta Valgrind [Roldan 2012], muito famosa pela sua capacidade de depuração de aplicações. Para que a ferramenta Valgrind refine sua saída para os dados referentes a utilização da memória deve-se utilizar a opção tool=massif, então, a ferramenta executará a aplicação pausadamente salvando estados da memória utilizada pelo programa para obter um perfil da memória durante a execução. A saída do programa será um arquivo chamado massif.out.<PID> onde <PID> é o identificador do processo analisado. O conteúdo do arquivo possui uma formatação própria que necessita de um programa especializado para visualização, como o massif-visualizer (Figura 5 e Figura 6).

Os testes foram executados 50 vezes para ambas as implementações, em C e em Python, e obtida a média e desvio padrão para, dessa forma, isolar a influência que o estado do sistema pudesse representar em função dos recursos utilizados por outras aplicações.

4. Resultados

Foram refinados dos resultados obtidos informações como tempo de execução, porcentagem de ocupação da CPU e o consumo de memória durante a execução de cada processo. Para a implementação em C os tempos totais de execução são mostrados na Tabela 1.

Tabela 1. Tempo de execução total para a implementação em C.

| | Média | Desvio padrão |
|--|-------|---------------|
| Tempo total de execução | 17 ms | 2 ms |
| Tempo de execução no espaço de usuário | 5 ms | 1 ms |
| Tempo de execução no espaço de sistema | 6 ms | 1 ms |

Os tempos de execução para a implementação em Python utilizando ctypes estão listados na Tabela 2.

Tabela 2. Tempos de execução para a implementação em Python.

| | Média | Desvio padrão |
|--|-------|---------------|
| Tempo total de execução | 22 ms | 1 ms |
| Tempo de execução no espaço de usuário | 7 ms | 1 ms |
| Tempo de execução no espaço de sistema | 6 ms | 1 ms |

Os tempos de execução no espaço de usuário e no espaço de sistema, somados, não resultam no tempo de execução total pois há também o tempo gasto na troca de

contexto entre o espaço de usuário e o espaço de sistema, como pode ser percebido na Figura 3.

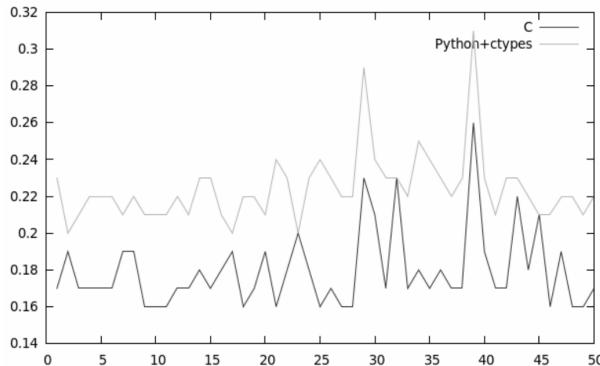


Figura 3. Tempo de execução total para ambas as implementações.

Tabela 3. Porcentagem da CPU ocupada pelo processo

| | Média | Desvio Padrão |
|--------|-------|---------------|
| C | 64% | 31% |
| Python | 67% | 30% |

Para a taxa de ocupação da CPU, os valores obtidos estão listados na Tabela 3. A Figura 4 mostra um gráfico da taxa de ocupação da CPU da aplicação em C e da aplicação em Python utilizando ctypes.

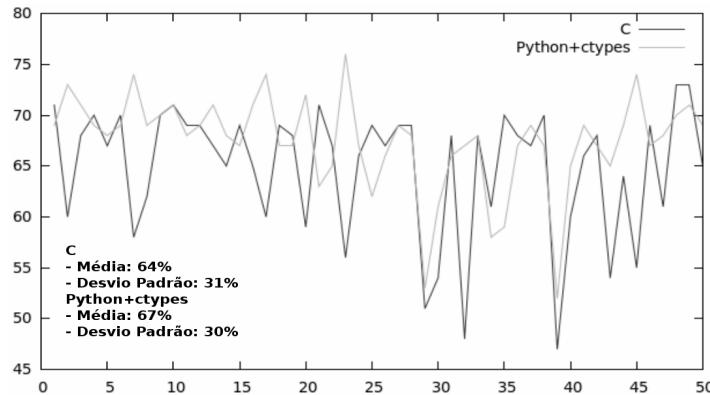


Figura 4. Taxa de ocupação da CPU para ambas as implementações.

A quantidade de memória utilizada por uma aplicação varia dependendo da quantidade de recursos alocados naquele momento da execução. As Figuras 5 e 6 mostram o perfil do consumo de memória de cada implementação, com picos de 14,8 MB para a implementação em C e 17,3 para a implementação em Python utilizando ctypes.

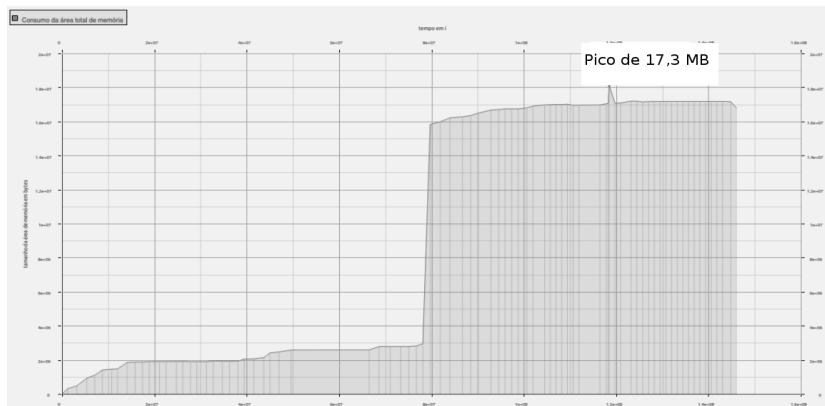


Figura 5. Gráfico visualizado com a ferramenta massif-visualizer da quantidade de memória requerida durante o tempo de execução da aplicação implementada em Python utilizando `ctypes`.

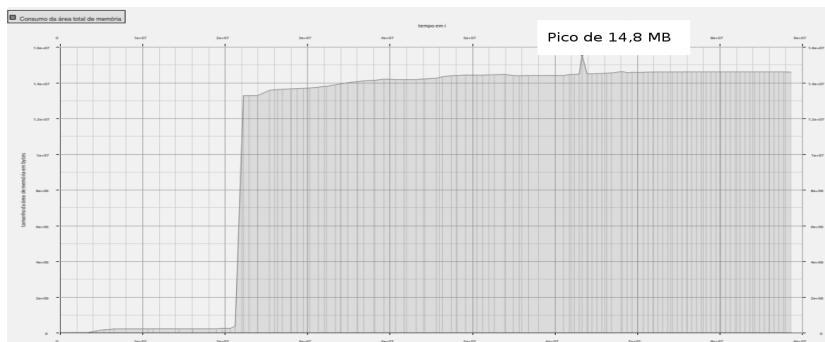


Figura 6. Gráfico visualizado com a ferramenta massif-visualizer da quantidade de memória requerida durante o tempo de execução da aplicação implementada em C.

5. Conclusões e Trabalhos Futuros

Para os testes executados para a medição do tempo de execução, os valores obtidos demonstram que a implementação em Python leva mais tempo para completar a execução, principalmente no espaço de usuário, devido a presença do elemento interpretador, necessário para a execução da aplicação. Por outro lado, os tempos obtidos no espaço de sistema são iguais. Então, em sistemas que passam a maior parte do tempo de sua execução no espaço de sistema, a implementação em Python poderia ser utilizada sem que houvesse uma grande diferença entre os tempos de execução totais.

Já nos testes para a obtenção das percentagens de alocação da CPU, obteve-se valores próximos, porém com uma pequena vantagem para a implementação somente em C. Em sistemas com requisitos não-funcionais relacionados ao consumo de energia, a implementação em C se encaixaria melhor devido ao menor uso da CPU.

Nos resultados obtidos sobre o consumo de memória, ambas as aplicações traçaram o mesmo perfil de alocação de memória, porém, com uma vantagem de aproximadamente 2,5 MB para a aplicação em C durante toda sua execução. Apesar de

memória ser um recurso relativamente barato atualmente, em certos sistemas, a quantidade de memória é limitada pelo consumo de energia ou até pelo espaço físico limitado de certos dispositivos. Em sistemas deste tipo, a implementação em C leva vantagem em relação à implementação em Python.

Embora os valores tenham sido próximos, caso estas tecnologias sejam utilizadas em uma aplicação mais complexa estes valores seriam mais sensíveis, apesar de que os tempos no espaço de sistema são bem próximos e tenderiam a continuar próximos devido a utilização das mesmas bibliotecas utilizadas em C através da interface para funções estrangeiras ctypes.

Como trabalhos futuros pretende-se realizar testes de uma aplicação em C e uma versão puramente em Python, mapeando as funções e os tipos nativos para estruturas equivalentes, de forma a analisar o quanto ctypes realmente adiciona em termos de desempenho para a aplicações escritas somente em Python.

6. Referências

- Behnel S., Bradshaw R., Ewing G., “Cython Project”, <http://cython.org/>, Acesso em: 8 set. 2012.
- Behnel S. et al. (2011), “Cython: the best of both worlds“, IEEE Computing in Science and Engineering [disponível online] <http://dx.doi.org/10.1109/MCSE.2010.118>
- Heller T., “Python ctypes Project”, <http://starship.python.net/crew/theller/ctypes/>, Acesso em: 10 set. 2012.
- Kloss, G. K. (2009), “Source Code: Automatic C Library Wrapping - ctypes from the Trenches”, *The Python Papers Source Codes* , vol. 1, pp. - , January 2009, Disponível Online] <http://ojs.pythonpapers.org/index.php/tppsc/issue/view/13>.
- Roldan, A., “Valgrind – Linux Man Pages”, <http://linux.die.net/man/1/valgrind>, Acesso em: 8 set. 2012.
- Ravitch, T. et al. (2009), “Automatic Generation of Library Bindings Using Static Analysis”, Computer Science Departament. University of Wisconsin-Madison.
- Yip, T. W., “Ctype Performance Benchmark” tungwaiyip.info/blog/2009/07/16/ctype_performance_benchmark , Acesso em: 10 set. 2012.

Gerador de CDFG com análise de dependência de dados a partir de código C

Hudson G. de Medeiros¹, Roberto B. Assunção Filho¹, Edgard de Faria Correa¹

¹Departamento de Informática e Matemática Aplicada
Universidade Federal do Rio Grande do Norte – Natal, RN – Brasil

hudsongeovane@gmail.com, robertofilhoufrn@gmail.com,
edgard@dimap.ufrn.br

Abstract. Program analyzing from a control data flow graph is an usual behavior in many computer science areas; in particular, on embedded systems development. However, these CDFGs have been done mostly manually, which makes this process time-consuming and increase error probability. This paper describes a tool that was designed to automate CDFGs creation from a language C source code, verifying data dependency between consecutive code lines, turning up parts of the program which could be run at the same time.

Resumo. Grafos de controle de fluxo são utilizados em diversas áreas da computação; em especial, no desenvolvimento de sistemas embarcados. Entretanto, geralmente são gerados manualmente para análise, o que demanda muito tempo e torna-se passível de erros. Este artigo descreve uma ferramenta desenvolvida para a criação de CDFGs (Control Data Flow Graph) a partir de um código fonte na linguagem C, analisando a dependência de dados entre as linhas de código consecutivas, tornando visíveis trechos de código que poderão ser executados paralelamente.

1. Introdução

Grafos de controle e fluxo de dados são componentes importantes no desenvolvimento de sistemas embarcados e na construção de compiladores, em que ajudam na identificação de possíveis otimizações e na tomada de decisões de projeto [Zaretsky et al. 2005]. A geração automática de CDFGs a partir de um trecho de código auxilia os profissionais dessas áreas, substituindo um trabalho manual e passível de erros por um processo automático e eficiente. Além disso, o paralelismo vem a cada dia ganhando mais espaço no mundo da computação, tendo em vista que a limitação física da velocidade dos processadores modernos está quase sendo alcançada.

A análise de dependência de dados é um passo importante na geração de CDFGs, em especial no desenvolvimento de sistemas que serão executados em paralelo. Com a realização desta análise podem ser identificados trechos de código que podem ou não ser executados simultaneamente. Os CDFGs gerados pela ferramenta descrita neste artigo destacam estes trechos.

Para apresentar este trabalho, o restante deste artigo está dividido da seguinte forma. Na Seção 2 são apresentados alguns trabalhos relacionados à geração de CDFGs. Na terceira seção a ferramenta desenvolvida é apresentada e são mostrados os resultados obtidos. A seção 4 apresenta os trabalhos futuros que virão como continuação deste, tentando de forma eficiente acabar com as restrições de uso da ferramenta.

2. Trabalhos Relacionados

A geração de CDFGs pode ser realizada tanto para fins gerais como para específicos. Rosien, Smith e Krol (2002) apresentaram um método para geração automática de CDFGs a partir de códigos em C/C++ usados para automatizar a programação de um FPFA (Field Programmable Function Array), entretanto foi utilizada a estrutura de um hipergrafo para modelar o fluxo de dados do código C/C++, entendendo cada expressão do código como uma árvore de operadores, o que não é necessário para a análise realizada.

Zaretsky et al. (2005) apresentaram uma forma de gerar CDFGs a partir de trechos de código em assembly. Os autores dividiram a construção de seus CDFGs em duas etapas: a construção do “Grafo de Controle de Fluxo” (CFG) e a do “Grafo de Controle de Dados” (DFG). Uma grande diferença entre Zaretsky (2005) e este trabalho é a complexidade das instruções interpretadas, uma vez que os saltos realizados por if-then-elses, whiles, fors e switch-cases são mais variáveis que os saltos condicionais e incondicionais do assembly, tornando o grafo gerado de um código de alto nível um pouco mais denso que o gerado a partir de instruções assembly. De fato, comparando as duas linguagens, as instruções em alto nível são bem mais robustas, e embora o fluxo do programa seja afetado por tal condição, a principal dificuldade encontrada na criação dos grafos de linguagens de alto nível não diz respeito ao fluxo, e sim à análise de dados, pois estas linguagens permitem que apenas uma instrução manipule, ou mesmo altere, grandes quantidades de dados, enquanto que as instruções assembly são em sua maioria compostas por poucas variáveis, em geral duas ou três.

Com efeito, a construção dos CDFGs na ferramenta desenvolvida dá-se principalmente nas duas etapas citadas anteriormente, tendo em vista que as análises de fluxo e de dados juntas constituem a principal característica do grafo da ferramenta: unir trechos de códigos que podem ser executados em paralelo, pois tais trechos devem ser independentes, e dessa forma, deve-se analisar a dependência de dados entre ambos os trechos, obviamente após a análise do fluxo nas instruções.

3. A ferramenta desenvolvida

O gerador de grafos de fluxo de controle de dados com análise de dependência de dados foi desenvolvido na plataforma Java em algumas etapas bem distintas: uma no que diz respeito à interpretação do código C, e outras relativas à geração do grafo em forma de dados e à criação da imagem que representará tais informações. Para tanto, foram utilizados alguns componentes que auxiliaram na análise sintática do código, bem como na criação do grafo e sua representação gráfica propriamente dita.

A figura 1 mostra o diagrama de classes do sistema, mostrando os componentes auxiliares, bem como a estrutura da ferramenta desenvolvida neste trabalho.

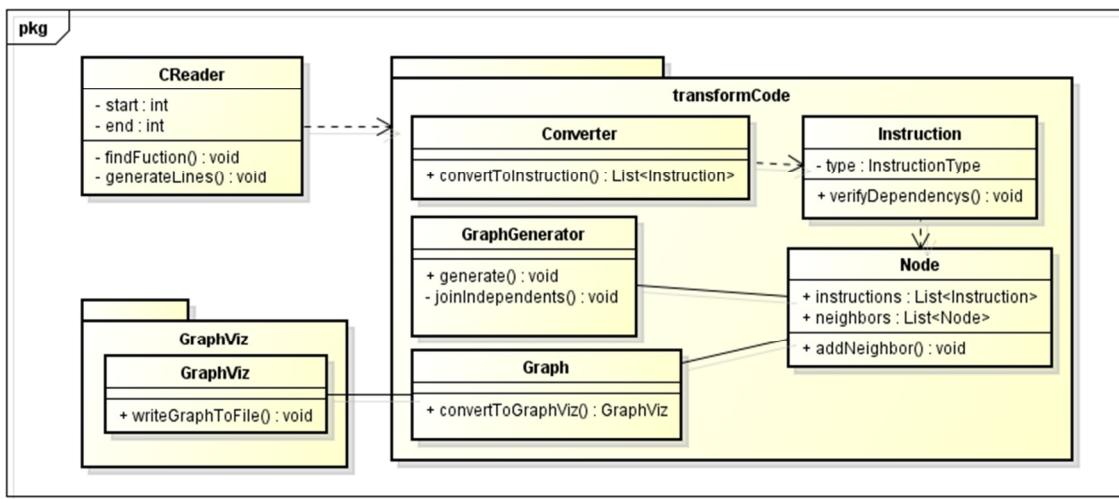


Figura 1. Diagrama de Classes

3.1. Visão Geral

O programa tem como entrada dois parâmetros: o caminho para o arquivo fonte a ser lido, e a função a partir de cujo grafo será gerado, na forma “*tipo_de_retorno nome_da_funcao*” (por exemplo, *int main*). Com tais dados, a ferramenta gerará o grafo apenas de uma função específica.

A geração do grafo, como dito anteriormente, dá-se em duas etapas. São elas a interpretação do fluxo do código e a análise das dependências das instruções, respectivamente. Entretanto, a ferramenta deve não só criar o grafo em forma de dados, mas também representá-lo graficamente para uma melhor visualização do usuário. Portanto, considera-se a existência de três macro etapas.

A partir dos parâmetros, a ferramenta deve procurar a função ou procedimento dentro do arquivo fonte e iniciar a análise do fluxo das instruções, verificando estruturas condicionais, laços de repetição e quaisquer outros saltos que possam estar contidos no programa. Instrução a instrução, o código é interpretado e só então começa-se o desenvolvimento do grafo, verificando onde cada condicional começa e termina. As instruções podem ser de diversos tipos, como atribuições, chamada de procedimento, condicionais (if-then-else), laços (while e for) e o fim de seus respectivos escopos (em outras linguagens, end if, end while; em C, apenas um fechamento de chaves “}”).

Com a lista de instruções já completamente interpretadas, pode-se esquecer então o código fonte para analisar apenas os dados gerados no primeiro passo. Cada objeto *Instruction* tem vários atributos, como tipo, lista de variáveis das quais a instrução depende, número da linha no código fonte, etc. E cada um destes atributos tem importância para alguma das fases de criação do grafo; iniciando pela análise do fluxo.

Neste passo, deve-se verificar os saltos da linha de execução do programa, ou no caso específico, da função ou procedimento. Caso o código não possua saltos condicionais, laços, e outras estruturas do tipo, haverá um grafo bambu, em que cada nó representará sua respectiva instrução, e ele sempre apontará para a instrução da linha seguinte, até a última linha de código; então não faria sentido realizar uma análise de tal grafo, afinal, ele seria apenas uma lista encadeada.

Após a análise do fluxo e de bem interpretadas as instruções com seus

respectivos tipos, haverá um grafo quase finalizado (em estrutura de dados), com cada objeto Node composto com apenas uma instrução, e seus respectivos vizinhos (atributos neighbor) bem configurados, isto é, cada nó contendo uma instrução do tipo IF deve ter dois vizinhos: a instrução correspondente em caso de TRUE, e a de FALSE (que deve ser o ELSE, se este existir, ou a instrução seguinte ao IF, caso contrário). Ocorre de maneira análoga para instruções do tipo WHILE e FOR; entretanto é válido atentar para as instruções do tipo END WHILE e END FOR, que devem apontar para o seu respectivo início de laço, ou seja, devem voltar para o teste de entrada da estrutura de repetição.

Por fim, é necessário apenas unir os nós independentes, i.e., trechos de código em sequência que poderiam ser executados paralelamente, em apenas um nó, não esquecendo de representar tal fato graficamente para o usuário. Após a união dos trechos, o grafo está gerado em forma de dados, restando representá-lo na forma de texto, na linguagem interpretada pelo GraphViz.

A Figura 2 mostra um fluxograma da ferramenta, em que estão expostas as principais macro etapas da geração do grafo, possibilitando uma visão geral do sistema.

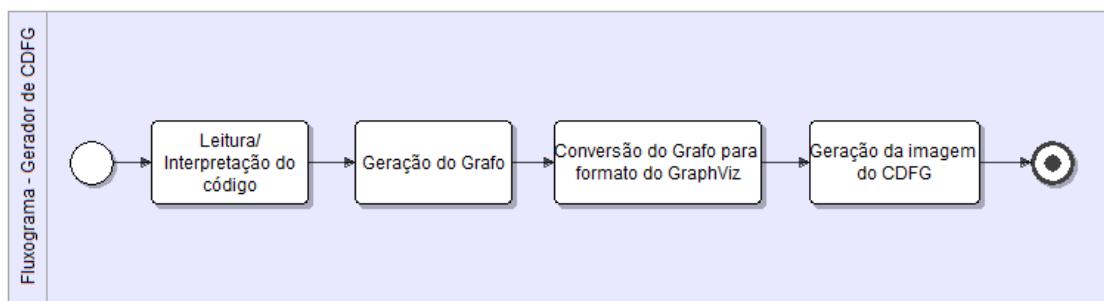


Figura 2. Fluxograma

3.2. Restrições de Uso

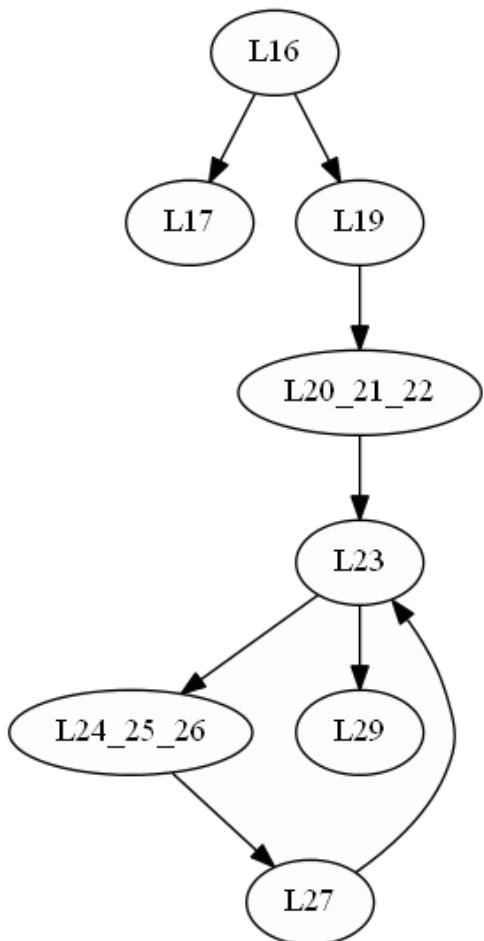
Para uma mais fácil interpretação do código C, foram tomadas algumas decisões para restringir a linguagem de programação aceita pelo gerador de CDFG, de forma que não atrapalhe o poder de computação do programa gerado pelos compiladores, porém limite a sintaxe da linguagem.

As principais restrições foram feitas devido ao tratamento linha a linha de nossa ferramenta, limitando apenas uma instrução por linha, além da obrigatoriedade da abertura de chaves de ifs, whiles e fors na mesma linha do comando. As restrições foram:

- Cada instrução deve estar em uma linha apenas, e cada linha deve conter apenas uma instrução.
- Cada *if*, *else*, *while*, e *for* deve ter sua chave de abertura, e esta deve estar na mesma linha que o respectivo comando. Além disso, cada chave (caractere '}') para fechamento da estrutura deve estar exclusiva em uma linha de código.
- Não foram tratadas atribuições em sequência, do tipo da instrução "a = b = c;".
- *Switch-cases* não foram tratados.
- Funções recursivas não foram tratadas.

3.3. Resultados

Com o desenvolvimento da ferramenta descrita, chegou-se ao resultado descrito no breve exemplo abaixo. A Figura 3 mostra um código C no formato interpretado pelo gerador, bem como o respectivo grafo gerado a partir dele. O algoritmo mostrado calcula o n-ésimo termo da sequência de Fibonacci, para um determinado n; o nome da função no **arquivo.c** é fib.



```

15 int fib(int n) {
16     if (n <= 2) {
17         return 1;
18     }
19     int f, f1, f2, i;
20     f1 = 1;
21     f2 = 1;
22     i = 3;
23     while (i <= n) {
24         i = i + 1;
25         f = f1+f2;
26         f1 = f2;
27         f2 = f;
28     }
29     return f;
30 }
  
```

Figura 3. Grafo gerado a partir de um código C.

Como mostrado na Figura 3, as linhas 20, 21 e 22 podem ser executadas paralelamente, uma vez que a ordem das instruções não é importante para o fluxo do programa. O mesmo ocorre nas linhas 24, 25 e 26, embora a linha 27 deva ser executada separadamente, pois depende dos dados do bloco de instruções anterior. A bifurcação na linha 16 representa o *if*, e na linha 23, o *while*.

4. Conclusões

A solução proposta resolveu, ainda que com limitações, o problema descrito anteriormente. Para tanto, a geração do grafo dá-se a partir da avaliação das instruções separadamente, não fazendo a análise semântica do código. Entretanto, a ferramenta proverá uma melhor análise do fluxo de dados e suas dependências, facilitando a realização de testes e suas verificações. Não obstante o sistema restringir a linguagem de entrada, ele pode ser utilizado como ferramenta posterior a um parser, que transforma

uma linguagem ANSI C na linguagem interpretada pelo gerador de CDFG, diminuindo a quase zero os problemas com C.

A ferramenta foi desenvolvida em Java, portanto é independente de plataforma, podendo ser rodada em qualquer sistema operacional, bastando que o usuário possua a máquina virtual Java instalada em seu computador. Além disso, a escolha da linguagem deve-se devido ao bom tratamento de expressões regulares por bibliotecas nativas do Java SE, bem como a utilização da biblioteca GraphViz.

4.1. Parser

Como foi dito durante este trabalho, a ferramenta não interpreta qualquer código C, pois possui algumas limitações quanto à linguagem. Contudo, tais restrições podem ser facilmente dribladas com um pouco de cuidado do programador. Apesar disso, para evitar este desconforto, um parser pode ser desenvolvido para transformar o código ANSI C para a linguagem restrita do gerador.

Para tanto, o parser deve alterar a sintaxe do arquivo de entrada para este ficar de acordo com as restrições descritas na Seção 3.2. Um problema que ainda restaria seria a recursividade, que deve ser abordado de outras formas.

4.2. Trabalhos Futuros

Para complementar a ferramenta apresentada neste artigo, poderá ser desenvolvido um parser acoplado ao gerador que percorrerá um arquivo fonte em C e o converterá para o formato reconhecido pela ferramenta, de acordo com as restrições apresentadas na Seção 3.2, solucionando o problema apresentado na Seção 4.1.

Além disso, será implementado o tratamento de instruções “switch-cases” e de atribuições em sequência, que ainda estão em desenvolvimento, assim como o tratamento de funções e procedimentos recursivos.

5. Referências

Rosien, M.A.J. and Smit, G.J.M. and Krol, T. (2002) “Generating a CDFG from C/C++ code”. Em: 3rd PROGRESS workshop on Embedded Systems, Veldhoven, The Netherlands. pp. 200-202. Technology Foundation STW. ISBN 90-73461-34-0.

Zaretsky, D.C., Mittal, G., Dick, R., Banerjee, P. (2005) “Generation of Control and Data Flow Graphs from Scheduled and Pipelined Assembly Code”. Em LCPC'05 Proceedings of the 18th international conference on Languages and Compilers for Parallel Computing, páginas 76-90.

Shibu, K. V. (2009), “Introduction to embedded systems”, Tata McGraw Hill, páginas 207-208.

Pattern (Java 2 Platform SE v 1.4.2), Disponível em <http://docs.oracle.com/javase/1.4.2/docs/api/java/util/regex/Pattern.html>. Acesso em: 2 de junho de 2012.