

PUC

ISSN 0103-9741

Monografias em Ciência da Computação

nº 06/11

Utilizando Engenharia Semiótica na Construção de uma Ferramenta de Simulação para RSSF

Adriano Branco

José Antônio Motta

Clarisse Sieckenius de Souza

Departamento de Informática

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO

RUA MARQUÊS DE SÃO VICENTE, 225 - CEP 22451-900

RIO DE JANEIRO - BRASIL

Utilizando Engenharia Semiótica na Construção de uma Ferramenta de Simulação para RSSF

Adriano Branco José Antônio Motta
Clarisse Sieckenius de Souza

{abranco , jmotta , clarisse}@inf.puc-rio.br

Resumo. A construção de uma aplicação na área de Redes de Sensores sem Fio (RSSF) exige do desenvolvedor uma visão completa do projeto, incluindo ambiente e detalhes de hardware. Consideramos que essa visão possa ser passada através de um tutorial que apresente alguns conceitos de RSSF apoiado por uma ferramenta de simulação. Nosso objetivo é aplicar os conceitos da Engenharia Semiótica na definição de uma ferramenta de simulação para RSSF e avaliar o uso do AgentSheets para a construção deste simulador. Este documento apresenta os conceitos utilizados na construção da ferramenta, detalha o processo da definição do simulador e termina com uma conclusão sobre todo o processo de desenvolvimento e o simulador construído.

Palavras-chave: Engenharia Semiótica, AgentSheets, Redes de Sensores sem Fio

Abstract. Building WSN application requires complete view of the project from the developer, including environment and hardware details. We think this view can be understood through the use of a tutorial which presents some WSN concepts supported by a simulation tool. Our goal is to use some Semiotic Engineering concepts in order to build a simulator for WSN and evaluate the use of AgentSheets as a tool to build this simulator. This report presents the concepts in the tool development, details on design decisions about the tool and ends with a conclusion about the whole process and the simulator developed.

Keywords: Semiotic Engineering, AgentSheets, Wireless Sensor Network

Responsável por publicações:

Rosane Teles Lins Castilho
Assessoria de Biblioteca, Documentação e Informação
PUC-Rio Departamento de Informática
Rua Marquês de São Vicente, 225 - Gávea
22451-900 Rio de Janeiro RJ Brasil
Tel. +55 21 3527-1516 Fax: +55 21 3527-1530
E-mail: bib-di@inf.puc-rio.br
Web site: <http://bib-di.inf.puc-rio.br/techreports/>

Sumário

1	Introdução	1
2	Conceitos Básicos	1
2.1	Engenharia Semiótica	1
2.2	Redes de sensores sem fio	2
3	Intenção de comunicação	4
3.1	Mensagem de metacomunicação Inicial	4
3.2	Processo de detalhamento da meta-mensagem	5
3.3	Interface proposta	7
4	Implementação	14
4.1	Recursos do AgentSheets	14
4.2	Versão implementada	15
5	Avaliação do trabalho	15
5.1	Consequências da dificuldade de implementação	16
5.2	Avaliação	17
6	Conclusão e trabalhos futuros	18
	Referências	20

1 Introdução

A construção de uma aplicação na área de Redes de Sensores sem Fio (RSSF) exige do desenvolvedor uma visão completa do projeto, incluindo a distribuição física dos nós sensores, os recursos de hardware disponíveis e a programação da rede como um todo [1].

Uma ferramenta de aprendizado que utilize uma linguagem visual para transmitir os conceitos de RSSF poderia permitir uma melhor compreensão do que ocorre numa rede sem a necessidade de se conhecer profundamente a teoria.

Uma plataforma de desenvolvimento que permite a criação de uma ferramenta deste tipo é o AgentSheets [2]. O AgentSheets é uma ferramenta que utiliza uma linguagem visual de fácil interação e que foi feita para ensino básico de computação. Nessa ferramenta os usuários podem criar pequenos jogos ou simulações sem a necessidade de se utilizar uma linguagem de programação textual. A programação da ferramenta de auxílio no ensino de RSSF seria feita através da criação de alguns componentes básicos para permitir que o usuário da ferramenta possa, através da interface do AgentSheets, executar, modificar ou criar novas simulações de RSSF. Inicialmente, a simulação será utilizada, por um instrutor, como ferramenta num tutorial para apresentação do conceito de RSSF e, em seguida, deverá ser utilizada pelos alunos para fixar os conceitos aprendidos.

Os objetivos principais deste trabalho são aplicar conceitos da engenharia semiótica [3] para especificar uma aplicação e também avaliar a capacidade do AgentSheets em auxiliar um professor de Redes de Sensores sem Fio a ensinar os conceitos da área. O aplicativo será utilizado na construção de um simulador de uma rede de sensores sem fio onde professores e alunos poderão interagir com o simulador e observar o comportamento de uma RSSF construída por eles mesmos. Para a construção do simulador, começamos criando uma mensagem de metacomunicação que nos guiou durante o processo de desenvolvimento. Depois, realizamos um estudo sobre o AgentSheets identificando algumas características que poderiam nos auxiliar ou atrapalhar no processo de desenvolvimento. A partir disso, incrementamos alguns trechos da metamensagem com detalhamento de alguns signos que foram usados na aplicação. Por fim, no processo de implementação tentamos representar os principais signos identificados na nossa mensagem de metacomunicação.

Na próxima seção (2) apresentamos alguns conceitos básicos que apoiam o nosso trabalho, em seguida, na seção 3, apresentamos em detalhes a nossa intenção de comunicação através das mensagens de metacomunicação. Na seção 4 apresentamos a descrição da nossa implementação no AgentSheets e em seguida, na seção 5 fazemos a nossa avaliação. Finalmente, na seção 6, apresentamos as nossas considerações finais.

2 Conceitos Básicos

2.1 Engenharia Semiótica

A Engenharia Semiótica é uma teoria baseada na visão de que a interação humano-computador é um caso particular de comunicação entre pessoas mediada por um computador. Esta comunicação é realizada em dois níveis: um é a comunicação direta entre o usuário e o sistema e o outro é o processo de *metacomunicação* que acontece entre o usuário e o designer usando o sistema como meio. É através da metacomunicação que o designer explicita a sua visão sobre a tarefa que o usuário deve realizar e tudo que influencia na

execução desta tarefa (como o perfil do usuário e suas preferências).

O conteúdo dessa da mensagem de metacomunicação pode ser caracterizado pelo seguinte template [4]:

“Eis a minha visão de quem você é, o que aprendi que você deseja ou precisa fazer, de que formas preferenciais e por quê. Este é o sistema que conseqüentemente elaborei para você, e esta é a forma como você pode ou deve usá-lo para realizar um conjunto de objetivos que se enquadram nesta visão.”

Depois que o designer concebe sua visão sobre quem é o usuário e suas necessidades, ele deve codificá-la através de palavras, gráficos, comportamentos do sistema, ajuda on-line etc. De acordo com a Semiótica, todos esses elementos que podem ser utilizados são *signos*. A Engenharia Semiótica faz uso principalmente da semiótica peirceana que diz que todo signo é algo que representa alguma coisa para alguém [5]. Peirce classificou os signos em várias categorias, mas as principais para a Engenharia Semiótica são:

- **ícone:** pode ser associado ao objeto que representa através das características que podem ser identificadas através dos sentidos (visão, audição etc).
- **índice:** pode ser associado ao objeto que por uma relação de causa ou co-ocorrência. O conhecimento para se identificar um índice, normalmente, vem através da herança cultural do indivíduo.
- **símbolo:** a relação entre o signo e o objeto se dá através de alguma norma pré-estabelecida.

Além das categorias de signos criada por Peirce, a Engenharia Semiótica criou a sua própria classificação dos signos em função do que eles expressam na interface de um sistema:

- **signos estáticos:** signos que expressam um estado do sistema e cujo significado pode ser entendido sem a necessidade de se interagir com o sistema.
- **signos dinâmicos:** signos que expressam o comportamento do sistema e cujo significado só pode ser compreendido no decorrer da interação com o sistema.
- **signos metalinguísticos:** signos fazem referência a outros signos da interface (estáticos, dinâmicos ou metalinguísticos).

A metamensagem é utilizada por alguns métodos da Engenharia Semiótica que avaliam a qualidade da metacomunicação de sistemas analisando os signos da interface, como o Método de Inspeção Semiótica [6] e o Método de Avaliação de Comunicabilidade [7].

2.2 Redes de sensores sem fio

2.2.1 Tecnologia

Uma rede de sensores é um grupo de pequenos sistemas autônomos denominados de Nós Sensores (sensor nodes). Estes nós cooperam para resolver pelo menos um problema comum. Geralmente suas funções incluem algum tipo de percepção de parâmetros físicos [8].

Um Nó Sensor sem Fio (WSN - Wireless Sensor Network) é um sistema que tem capacidade de comunicação, computação, sensoriamento e armazenagem. Esses nós miniaturizados operam com restrições severas em termos de recursos disponíveis, como a energia da bateria, memória RAM, largura de banda de comunicação disponível e poder de processamento. A figura 1 mostra um diagrama esquemático dos componentes de um nó sensor. Cada nó é composto por um micro-controlador, fonte de alimentação, transceptor de Rádio Frequência (RF), memória externa e sensores.

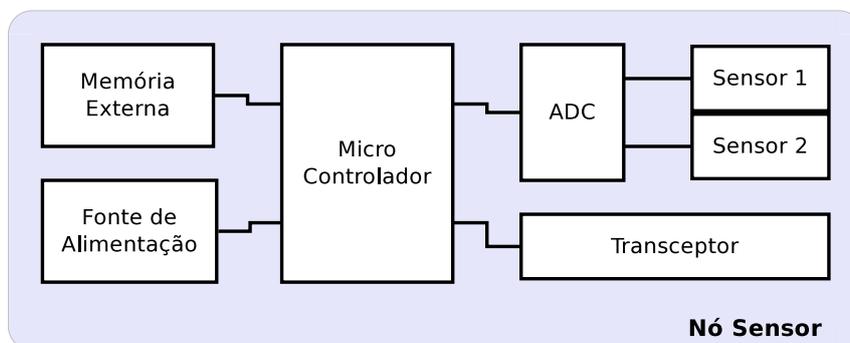


Figura 1: Módulos de um Nó Sensor

Centenas de milhares desses nós sensores são implantados numa larga variedade de aplicações, que vão desde campos vulcânicos até monitoramento ambiental. Esses nós sensores podem se comunicar uns com os outros numa rede ad-hoc usando caminhos de comunicação multi-hop, assim formando uma rede de sensores (WSN).

Em vários tipos de aplicações, após a implantação, os nós sensores são de difícil acesso. Por esse motivo essas redes devem ser autônomas e apresentar longa duração. Quase sempre os nós sensores precisam sobreviver às duras condições ambientais e conservar o máximo de energia possível.

O trabalho de desenvolvimento de aplicações para RSSF é impactado pelos mesmos desafios encontrados no desenvolvimento de aplicações para sistemas distribuídos que utilizam plataformas tradicionais. Porém, alguns desses desafios são aumentados pela escassez de recursos computacionais dos motes, pelo modelo de programação tipicamente adotado que é orientado a eventos e pela tecnologia de comunicação sem fio com características de redes ad-hoc.

A visão geral de um projeto em RSSF torna-se complexa a partir da forte dependência entre a definição do tipo de dispositivo, a topologia da rede e a complexidade do programa executado em cada dispositivo. Por exemplo, uma aplicação com centenas de nós tende a minimizar o custo utilizando dispositivos mais simples, que por consequência dispõem poucos recursos de hardware e limitando o tamanho e complexidade do programa a ser executado. Um outro exemplo é uma aplicação que necessita operar por um longo tempo e os dispositivos não são de fácil acesso, nesse caso o programa a ser executado deve minimizar ao extremo o uso dos recursos utilizados, economizando, dessa forma, o consumo de energia.

2.2.2 Visão de um projeto em RSSF

Uma característica importante no projeto de RSSF é a forte dependência entre o projeto físico e o projeto de sistema computação [1]. A maioria dos projetos de sistemas de computação são feitos para plataformas de execução padronizadas e normalmente o am-

biente físico não interfere de forma significativa na solução final. Em um projeto de RSSF essa dependência é fundamental para a solução conjunta final. O exemplo principal é a própria função da RSSF que, normalmente, trabalha com sensoriamento de grandezas físicas do ambiente como medição de temperatura e luminosidade. Um outro exemplo típico dessa dependência é o caso em que o protocolo de comunicação de dados deve minimizar o uso do rádio para economizar energia e conseqüentemente prolongar o tempo de vida da rede.

A situação muito comum e facilitadora de softwares que disponibilizam várias funcionalidades também não é uma boa estratégia em RSSF. Muitas vezes as limitações de memória e processamento permitem que só se carreguem no dispositivo a parte do software estritamente necessária. Por exemplo, não podemos carregar uma biblioteca completa para todos tipos sensores, deve-se carregar somente os componentes para os sensores que estão fisicamente ligados no dispositivo.

De forma geral o projeto de software deve considerar os tipos de sensores utilizados, a disposição dos nós na rede, a capacidade de alimentação e as limitações de recursos computacionais como memória e processamento.

Por isso entendemos que a melhor forma de introduzir o conceito de RSSF é permitir ao aluno compreender a diferença entre o ambiente, o projeto da rede e as funcionalidades executadas nos dispositivos.

3 Intenção de comunicação

Nesta seção apresentamos o resultado para o nossa intenção de comunicação. Obtemos o resultado aplicando o template de metacomunicação da Engenharia Semiótica e em seguida detalhando a meta-mensagem obtida.

3.1 Mensagem de metacomunicação Inicial

Baseado na nossa intenção, definimos como mensagem de metacomunicação inicial o seguinte texto:

“Você é um professor especialista em redes de sensores sem fio (RSSF) e dá aulas há muito tempo. Seus alunos são universitários que têm conhecimento sobre informática, mas não sabem nada sobre RSSF. Você percebeu que uma característica típica dos projetos em RSSF é a necessidade de compreensão dos conceitos envolvidos em todo processo de um projeto, não se limitando a visão típica de programação de um aplicativo. Você está buscando uma forma que seja eficaz de ensinar dois desses conceitos importantes: o da definição do ambiente simulado e da definição da rede de sensores. Sua ideia é utilizar uma ferramenta de simulação em um tutorial. Essa ferramenta usa uma linguagem visual que auxilia o aprendizado, pois ela pode comunicar ideias sem a necessidade de já saber os detalhes do que está ocorrendo.

Para lhe ajudar, estou disponibilizando um simulador de RSSF que desenvolvi no AgentSheets que é uma ferramenta para simulações simples. Eu criei um conjunto de componentes que podem ser utilizados para construir o ambiente, montar a rede de sensores e criar estímulos nessa rede. Esses componentes estão no contexto de medição de temperatura, sendo que os conceitos utilizados são os mesmos para outros contextos de utilização. Você e seus alunos poderão montar simulações simplesmente combinando alguns dos componentes disponibilizados e cada simulação permitirá acompanhar quais são as conseqüências da organização dos componentes. Durante uma simulação será possível observar as variações de temperatura. Também será possível observar situações de

alarme em cada sensor e visualizar o resultado dos cálculos periódicos definidos para cada grupo de sensor. A ideia é que observando e interagindo com a simulação, os alunos possam consolidar certos conceitos relacionados a RSSF."

3.2 Processo de detalhamento da meta-mensagem

Para podermos definir quais serão as principais idéias que queremos expressar através do simulador e quais os signos que utilizaremos para nos comunicarmos com os usuários, quebramos a metamensagem em alguns trechos que consideramos poderem ser utilizados como uma espécie de lista informal de requisitos funcionais que o sistema deve atender. Isso não quer dizer que o restante da metamensagem foi ignorada. Pelo contrário, ela foi útil para nos guiar na escolha de signos adequados tendo em vista os usuários que irão utilizar o simulador. O resultado deste processo será uma proposta de interface que deve levar em conta todas as considerações feitas nesta seção.

A seguir, estão listados os segmentos da metamensagem utilizados e o detalhamento dos signos.

3.2.1 Segmento 1

"Eu criei um conjunto de componentes que podem ser utilizados para construir o ambiente(1), montar a rede de sensores(2) e criar estímulos(3) nessa rede."

Signos (1):

- Valor da temperatura ambiente - Campo para digitar o valor, pois como os nossos usuários entendem de computação eles reconhecem um campo para digitar valores.

Signos (2):

- Agrupamentos de sensores - Representado por uma cor diferente para cada grupo, pois a diferenciação de cor explicita a variedade.
- Sensor - Representado pela imagem de um pequeno equipamento com antena. Quando o sensor estiver associado a um grupo, ele deve possuir algum indício disso, como possuir a cor do grupo ao qual está associado.
- Ativar uma ou múltiplas operações de sensor em andamento - Checkbox para ativação/desativação, pois como os nossos usuários entendem de computação eles reconhecem um Checkbox.
- Parâmetros das operações
 - Seleção da função - Drop-down Box com as opções "Alarme", "Média", "Soma", "Máximo" e "Mínimo". Como os nossos usuários entendem de computação eles reconhecem um Drop-down Box.
 - Limite para alarme e Período de atualização da monitoração - Campo para digitar o valor desses parâmetros, pois como os nossos usuários entendem de computação eles reconhecem um campo para digitar valores.

Signos (3):

- Gerador de calor - Representado pela imagem de uma chama, pois é comum a ideia de fogo como gerador de calor.

- Valor da temperatura do gerador de calor - Campo para digitar o valor, pois como os nossos usuários entendem de computação eles reconhecem um campo para digitar valores.

3.2.2 Segmento 2

“Você e seus alunos poderão montar simulações simplesmente combinando alguns dos componentes disponibilizados (4).”

Signos (4):

- Área para distribuir componentes - Janela com a identificação da área onde pode-se distribuir os componentes para a construção de uma RSSF, semelhante à area livre que se vê em aplicativos de desenho como o “Paint”. Essa idéia também é comum entre pessoas que possuem experiência em computação.
- Conjunto de componentes disponíveis - Uma galeria com os componentes que podem ser utilizados. Essa idéia é bem parecida com a de uma “toolbar” com as opções de desenho que se vê em aplicativos gráficos.

3.2.3 Segmento 3

“cada simulação permitirá acompanhar quais são as consequências da organização dos componentes. (5)”

Signo (5):

- Representações textuais, gráficas e sonoras dos acontecimentos - através de mudanças na imagem dos agentes e de dados numéricos que serão exibidos durante a simulação, torna-se possível acompanhar mudanças no ambiente configurado pelo usuário em tempo de simulação. Alguns estímulos sonoros, como a sirene de um alarme, também serão utilizados.
- Comparação entre operações de sensor diferentes - o usuário poderá visualizar, ao mesmo tempo, a execução de mais de uma operação de sensor sendo executada e poderá comparar os resultados dessas operações entre si.

3.2.4 Segmento 4

“Durante uma simulação será possível observar as variações de temperatura.(6)”

Signo (6):

- Variações de temperatura - a temperatura do ambiente será representada por uma tonalidade de cor que será aplicada à cor do grupo. Se a temperatura for ambiente, a cor do grupo não muda, mas quanto mais quente estiver, uma tonalidade vermelha será aplicada à cor.

3.2.5 Segmento 5

“Também será possível observar situações de alarme(7) em cada sensor e visualizar o resultado dos cálculos periódicos(8) definidos para cada grupo de sensor.”

Signo (7):

- Alarme - o sensor que disparar o alarme mudará de cor e disparará um som de alarme.

Signo (8):

- Resultado dos cálculos periódicos - os valores dos cálculos que dizem respeito à simulação serão exibidos em intervalos de tempo pré-definidos pelo usuário.

3.3 Interface proposta

Com os signos que deverão ser exibidos montamos uma proposta para a interface da nossa ferramenta ilustrada pela figura 2. A seguir, detalhamos as mensagens que gostaríamos de passar com esses signos e justificamos nossas decisões usando as classificações de signos de Peirce e da Engenharia Semiótica.

Tentamos seguir as indicações de [9] onde os autores sugerem que representações com ícones e índices podem facilitar o entendimento da interface de um sistema computacional.

A figura 2 mostra como queremos que seja a interface de nosso simulador, baseado no trabalho que descrevemos anteriormente. Do lado esquerdo, temos os dados que são utilizados em nossa simulação: as variáveis de ambiente e os parâmetros das operações dos sensores. Do lado direito, temos uma área para montagem do simulador e outra área para visualizar o resultados das operações que são calculados em função dos parâmetros informados nos campos do lado esquerdo da tela e do ambiente que está sendo simulado acima.

A figura 3 mostra uma parte da área onde o usuário poderá montar a sua simulação e explicita a mensagem que queremos que o usuário entenda ao ver a galeria de componentes.

Agora, detalhando um pouco mais, explicitamos a mensagem de cada componente. As figuras 4 e 5 mostram o que gostaríamos que o usuário entenda quando ver cada um dos componentes. Além dos ícones que representam a identidade do componente, utilizamos um texto que serve como um signo metalinguístico para descrever o que é o componente.

Através desses componentes, estamos tentando transmitir, além de sua identidade, também a forma que o usuário pode interagir com eles. No caso da criação de grupos de sensores, queremos que o “lápiz” seja um índice que faça o usuário ver que, para se criar um grupo de sensores, ele deve “desenhá-lo” na área de montagem da simulação. No caso da chama e do sensor, não explicitamos uma forma de interação na imagem. Nossa intenção é que o usuário pense que a forma de se criar sensores e chamadas, basta clicar na área de montagem da simulação, que é a forma default de se interagir usando o mouse. Na figura 6 mostramos a nossa mensagem sobre isso.

Em relação à montagem de simulação, temos uma grande área que possui uma cor diferente da janela. Isto é um ícone que mostra que aquela é uma área “especial”. Além

disso, essa área está vazia. Isso é um índice de que algo deve ser colocado ali. Essa área está dentro de um agrupamento cujo nome é "Área de Simulação", que é um signo metalinguístico que fala sobre tudo que está dentro do agrupamento. Nossa intenção é que o usuário junte essas informações e conclua que aquela área é uma área onde ele poderá montar sua simulação. Nossa mensagem sobre a área de simulação está descrita na figura 7.

Na figura 8 temos a mensagem sobre o botão que serve para executar a simulação.

As figuras 9 e 10 dizem respeito aos dados do ambiente e das operações. Fica claro o fato de fazemos uso de signos metalinguísticos para induzir o usuário a entender o que ele deve preencher nos campos e como deve preencher. Por exemplo, o "°C" diz para o usuário que a temperatura que deve ser informada é em graus celsius e o "segundos" diz que o intervalo de atualização das operações deve ser informado em segundos. No caso da temperatura, a notação "°C" foi utilizada pois ela é amplamente divulgada em qualquer lugar que fale sobre temperatura, como jornais e sites.

A figura 13 mostra algo que não pode ser entendido simplesmente olhando para a tela. Queremos que o usuário entenda que TODAS as operações precisam de um intervalo de atualização. Para isso, fazemos uso de um signo dinâmico, pois nossa intenção é que o usuário veja isso à medida que mude o tipo de operação que deseja ver no drop-down box que fica em cima do campo e perceber que esse parâmetro está presente em todas as operações.

Na Figura 14 fazemos uso de um outro signo dinâmico para informar que alguns parâmetros são adicionais. O usuário irá descobrir isso à medida que mudar o tipo de operação no drop-down box e ver que algumas operações como "alarme" possuem um parâmetro extra e outros, como "média" não.

Na figura 15 temos a forma que pretendemos que o usuário veja como a mudança de temperatura acontece. Aqui, fazemos uso de um índice: quanto mais vermelho, mais perto da chama. Isso quer dizer que, quanto mais vermelho, mais quente está a temperatura daquela posição. Achamos que a cor vermelha representa bem a idéia de calor. Por exemplo, em jornais que possui uma seção de previsão do tempo, as áreas mais quentes são mais avermelhadas e as mais frias são mais azuis.

Um exemplo de simulação é quando a chama se aproxima de um sensor que está configurado para trabalhar com alarme (figura 16). Quando a temperatura da área onde está o sensor ultrapassa o limite estabelecido pelo usuário, o sistema usará um índice, que é a mudança de cor do sensor, para indicar o alarme. Para reforçar a ideia de emergência, também será utilizado um ícone, que é o sinal sonoro de alarme.

Paralelo à representação visual, a simulação também deverá exibir um índice, que são dados numéricos importantes para o entendimento do que está acontecendo. Para isso, existe uma área que mostrará os resultados das operações dos grupos de sensores (figura 17).

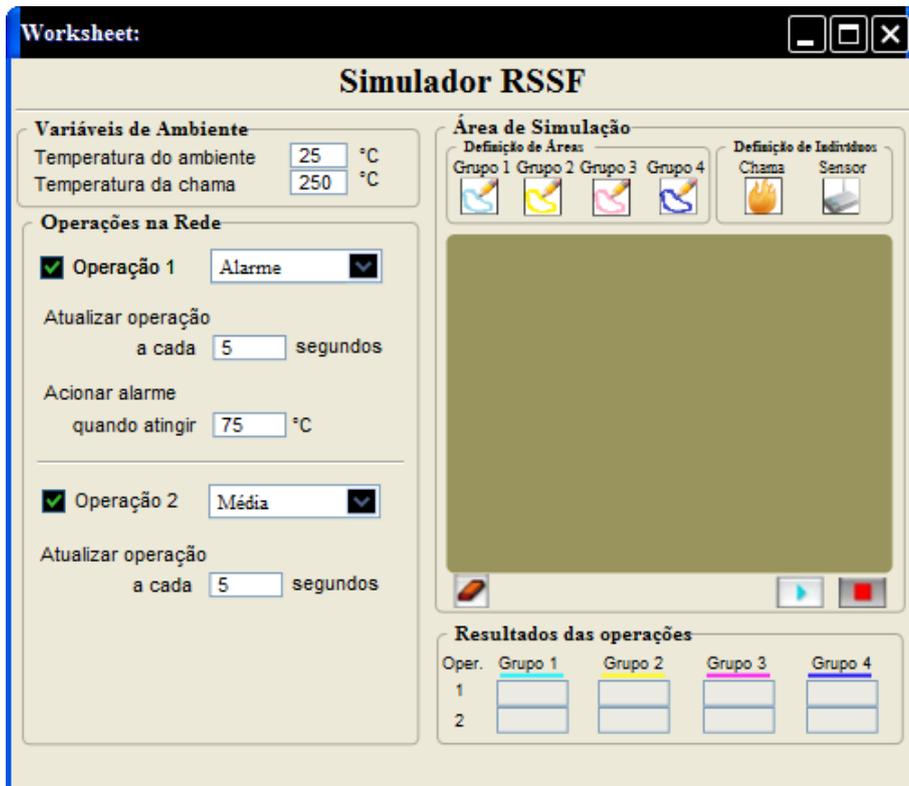


Figura 2: Interface baseada nos signos propostos

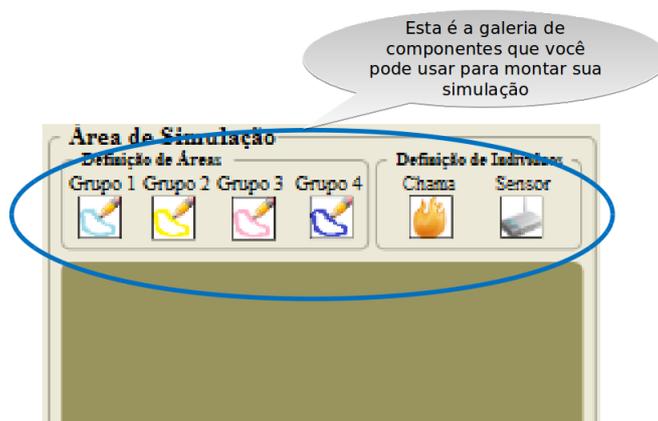


Figura 3: Galeria de componentes

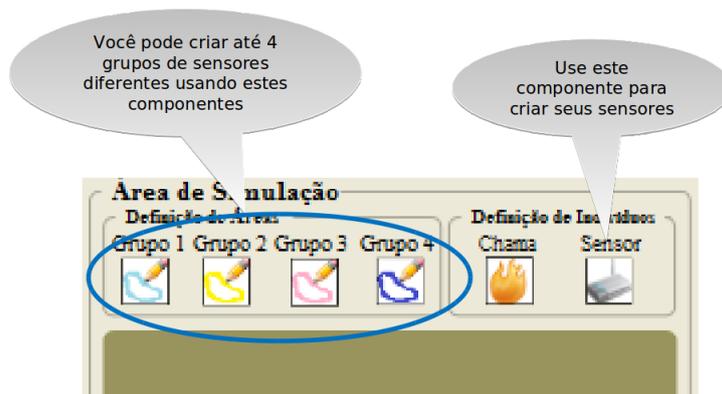


Figura 4: Agrupamentos e Sensores

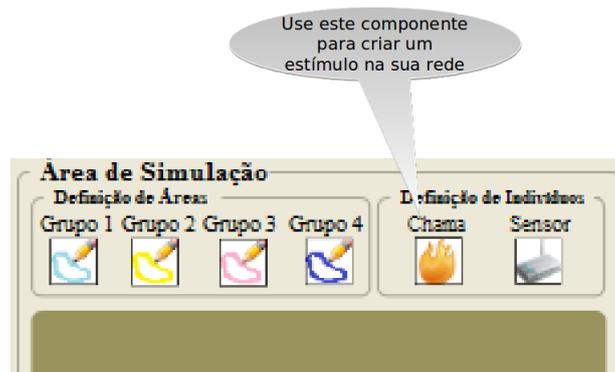


Figura 5: Chama

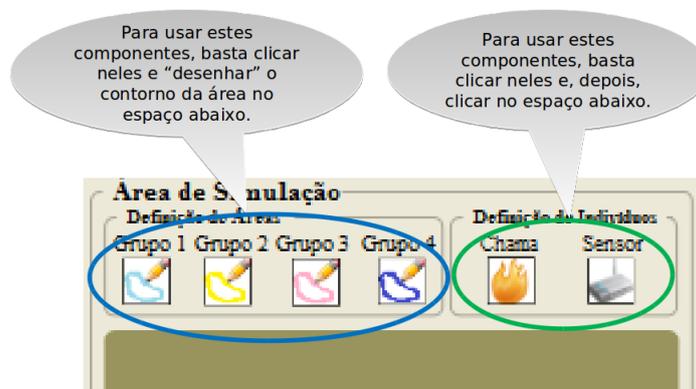


Figura 6: Edição

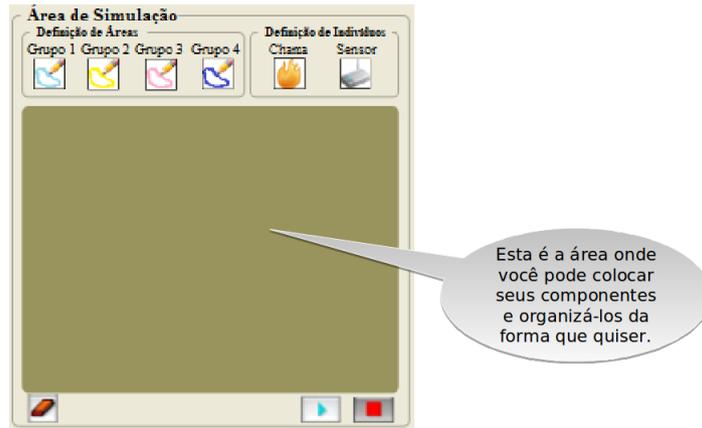


Figura 7: Área da simulação

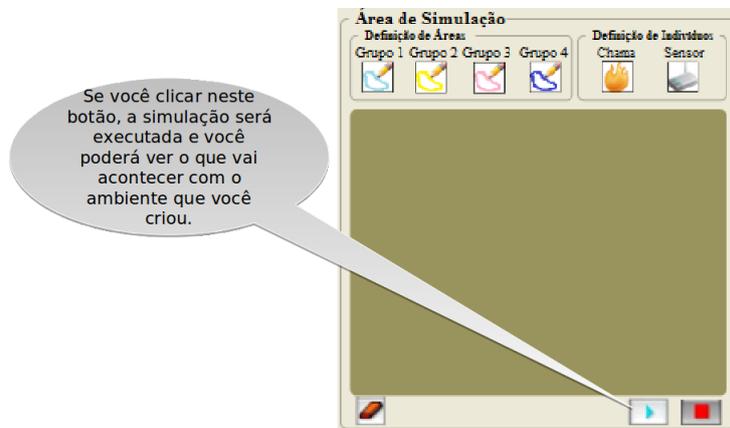


Figura 8: Iniciar uma simulação

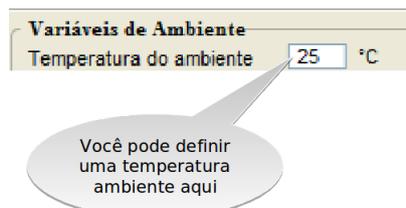


Figura 9: Temperatura ambiente

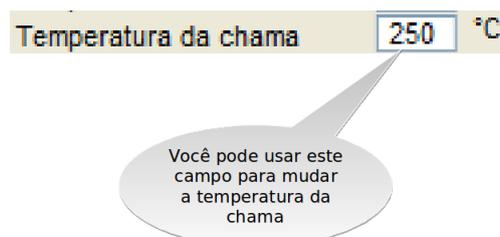


Figura 10: Ajustando a temperatura da chama

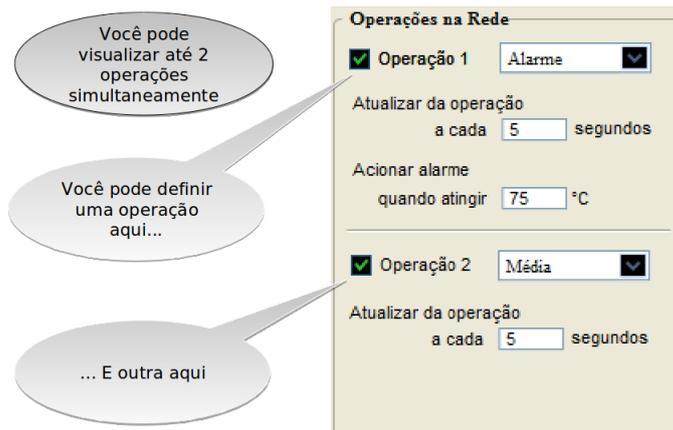


Figura 11: Seleção das operações da rede

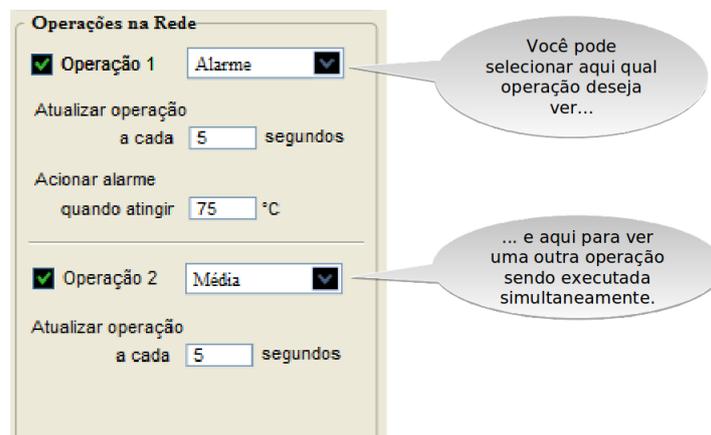


Figura 12: Seleção dos tipo de cálculo da operação

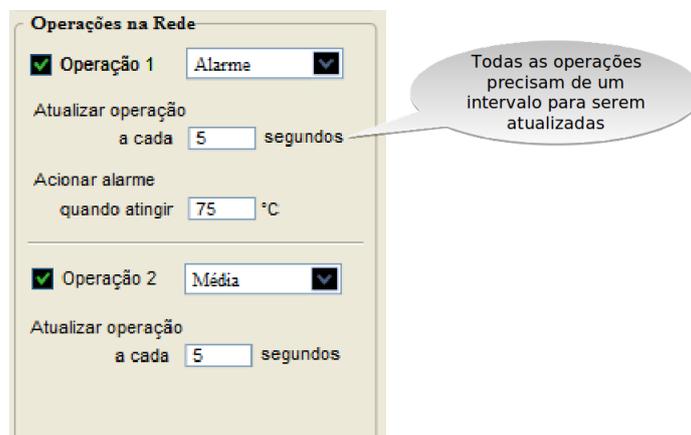


Figura 13: Definição do intervalo de operação

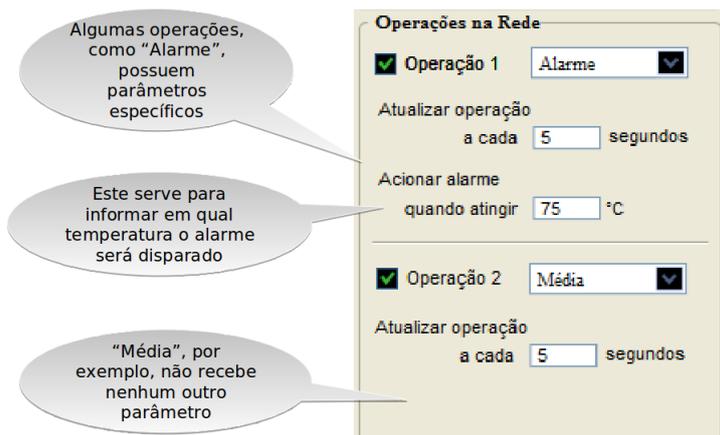


Figura 14: Parâmetros adicionais



Figura 15: Visualizando a mudança de temperatura



Figura 16: Aproximando a chama de um sensor de alarme

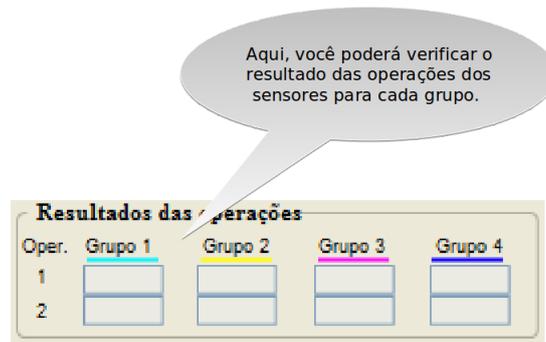


Figura 17: Apresentação dos resultados dos cálculos

4 Implementação

Nessa seção vamos apresentar os principais pontos da nossa implementação. Começamos com as facilidades e limitações do AgentSheets para a implementação dos nossos signos. Em seguida apresentamos a solução implementada.

4.1 Recursos do AgentSheets

De forma geral o AgentSheets foi um facilitador para a implementação da nossa ferramenta. Um dos principais motivos da facilitação é o fato do AgentSheets ser uma ferramenta para implementação de pequenas simulações interativas. Isso contribui positivamente, nossa aplicação é essencialmente uma simulação. Outro motivo é que o AgentSheets é baseado no conceito de agentes que podem interagir numa área de trabalho. A nossa ferramenta utiliza um conceito semelhante para os componentes que constituem um projeto em RSSF. E, finalmente, o AgentSheets disponibiliza algumas facilidades para implementar ações e reações simples como troca de imagens, movimentos, acionamentos (mouse e teclado), variações de cores, e sons básicos

Apesar das facilidades do AgentSheets, nos deparamos com diversas restrições que dificultaram a implementação de alguns signos, a seguir relacionamos algumas dessas restrições.

- Galeria única de agentes - Os agentes adicionais de suporte (agentes que não devem ser utilizados pelo usuário do simulador) ficam disponíveis para o usuário da ferramenta. Se o usuário mexer com esses agentes, o simulador pode ter um comportamento inesperado.
- Edição em todo worksheet - O usuário consegue editar toda a área de trabalho, incluindo a área de controle que não deve ser alterada, correndo o risco de corromper todo o simulador.
- Nenhuma facilidade para controles de entrada de dados e geração de texto - Não há suporte para entrada e apresentação de valores e textos
- O tamanho dos agentes é único para todos os componentes - Todos os itens na interface devem ter o mesmo tamanho e resolução.
- Agentes não são indivíduos - Não existe um suporte nativo para o programa identificar um agente específico no worksheet.

- Não há suporte para comunicação parametrizada entre agentes - Um agente pode disparar métodos em outros agentes, mas de forma *broadcast* sem nenhum filtro ou parâmetro.

Como solução de contorno às limitações acima, tivemos que:

- Criar identificadores para cada agente - Definimos um forma de criar identificadores individuais baseados na posição relativa dos agentes ou na ordem de criação na worksheet.
- Criar variáveis globais para intercomunicação entre agentes - Utilizamos variáveis globais para intercomunicação entre agentes.
- Criar eventos (métodos) de controle entre agentes - Definimos um conjunto de métodos, que combinados com as variáveis globais, pudessem resolver alguns problemas de intercomunicação entre indivíduos.
- Criar um agente que centralizasse algumas atividades de controle.
- Criar um agente improvisado para exibir valores numéricos.
- Criar um agente improvisado para aumentar e diminuir valores em substituição ao controle de entrada de dados pelo teclado.
- Criar um agente improvisado para “checkbox”.

4.2 Versão implementada

Na figura 18 apresentamos a versão atual da interface implementada no AgentSheets. Esta versão reflete as dificuldades encontradas para representar alguns signos definidos na seção 3.3. Em seguida, descrevemos como ficou a operação de alguns controles da interface.

Descrição dos controles utilizados:

Exibição de valor - Montamos uma complicada combinação entre tipo de agente e posição do agente para representar um valor numérico de uma variável global.

Checkbox - Reflete o funcionamento convencional de um controle checkbox.

Entrada de valores - Não conseguimos criar um campo editável para que o usuário pudesse digitar o valor desejado. A nossa implementação prevê que o usuário selecione um controle com o ponteiro do mouse e em seguida com as teclas de setas do teclado incremente ou decmente o valor exibido.

Alocação, movimentação e remoção de componentes - Mantivemos os mesmos controle do AgentSheets.

5 Avaliação do trabalho

Vamos separar nossa avaliação em em duas partes, primeiro apresentaremos as consequências das dificuldades encontradas na implementação e em seguida resumimos a nossa avaliação.



Figura 18: Interface implementada no AgentSheets

5.1 Consequências da dificuldade de implementação

Podemos dividir o impacto da dificuldade de implementação em dois grupos. O primeiro grupo identifica alguns pontos diretamente relacionados com a atividade de implementação. O segundo grupo lista os principais impactos na comunicação desejada.

5.1.1 Consequências na implementação

- Interface de desenvolvimento do AgentSheets não dá nenhum suporte para programação avançada e mais complexa.
- Especificação manual de todas variáveis e métodos - Para manter a consistência e evitar erros foi necessário manter um controle externo ao AgentSheets de todas variáveis e métodos criados.
- Controles de manipulação e entrada de dados não compatível com a definição dos signos - A nossa implementação para os controles de interface não conseguiu atender aos requisitos da especificação.
- Tamanho único para todos agentes dificulta a representação dos signos - A restrição do tamanho dos agentes não permitiu representar adequadamente vários signos.
- Layout da interface não segue o desenho planejado - Como consequência geral a interface implementada não reflete a interface especificada.

5.1.2 Consequências na comunicação

Descrevemos as consequências na comunicação de alguns problemas na implementação do simulador através das mensagens equivocadas que podem ser transmitidas ao usuário através da metacomunicação, que foi o que regu todo o trabalho realizado.

- Ferramenta de edição é global e galeria única de agentes
 - “Você pode reconfigurar todos os aspectos do simulador, incluindo o painel de controle.”
- Agrupamento dos elementos no painel de controle
 - “Para o simulador, não há diferenças entre as variáveis de ambiente (como a temperatura ambiente) e os parâmetros das operações da rede.”
- Resultados das operações são sempre numéricos
 - “Como o ‘0’ e ‘1’ para alarmes pode parecer estranho, criei um estímulo visual para que você saiba quando o alarme estiver ligado.”
 - “Atenção, pois a qualquer momento, o alarme pode ser disparado.”
- Os parâmetros de todas as operações são os mesmos
 - “Você pode configurar uma temperatura que irá disparar o alarme em qualquer tipo de operação.”
 - “Qualquer operação pode disparar um alarme.”

5.2 Avaliação

Nossa avaliação está dividida em duas partes. Na primeira consideramos o processo de definição da ferramenta utilizando conceitos da Engenharia Semiótica. Na segunda parte avaliamos a utilização do AgentSheets como plataforma de desenvolvimento para a nossa ferramenta de simulação.

5.2.1 Utilização da Engenharia Semiótica

A utilização de conceitos da Engenharia Semiótica contribuiu positivamente na definição da nossa ferramenta. A utilização do template de metacomunicação e o devido detalhamento da metamensagem em signos foi fundamental para um bom desenho de interface.

O exercício de verificar se um signo realmente comunicava a nossa intenção original também foi muito importante no processo de refinamento da definição dos signos e da definição da interface. Com isso foi possível verificar alguns itens que normalmente passariam despercebidos num processo convencional de especificação.

Por último ressaltamos que a definição original do nosso problema foi melhorada após o processo de definição da nossa intenção de comunicação.

5.2.2 Implementação no AgentSheets

A limitação de recursos no AgentSheets para o que queríamos fazer resultou em:

- Problemas na comunicação
 - A implementação do simulador no AgentSheets criou novas mensagens que, não só não foram previstas na criação da proposta da interface, como também, não deviam existir. Por exemplo: a mensagem que diz que o usuário pode editar o painel de controle do simulador.

- A dificuldade de se implementar uma metalinguagem no AgentSheets empobreceu as possibilidades de comunicação com o usuário. Os recursos de gale-ria, edição e configuração não puderam ser refletidos na interface da mesma forma que é disponibilizado para o desenvolvedor.
- Aumento no esforço de implementação
 - Falta de recursos para operações mais complexas - Alguns signos que depen-diam de operações mais complexas não puderam ser representados, por exem-plo a comunicação entre agentes específicos ou parametrização individual de um agente após a inserção na worksheet.
 - Falta de suporte de validação de código - Não existe nenhuma facilidade para validação das variáveis locais, variáveis globais e dos métodos entre agentes.
 - Falta de suporte para depuração - Não existe nenhum suporte auxiliar para depuração do código durante execução. Existe somente a possibilidade de vi-sualizar e alterar o conteúdo das variáveis globais e de um agente selecionado.
 - Esforço grande para resolver problemas simples - Não existem alguns signos normalmente disponibilizados em sistemas de interfaces. É necessário grande esforço de desenvolvimento das soluções de contorno que representem esses signos. Por exemplo, não existe facilidades para representação numérica ou entrada de dados durante a execução da simulação.

6 Conclusão e trabalhos futuros

Neste trabalho avaliamos a aplicação de conceitos da Engenharia Semiótica na definição de uma ferramenta de simulação. Adicionalmente avaliamos o AgentSheets como plata-forma para desenvolvimento dessa ferramenta.

A utilização do template de metacomunicação e a respectiva definição dos signos ser-viram como base para a definição da ferramenta. A definição da interface baseada nesses signos possibilitou um processo de autoavaliação constante que foi fundamental para o refinamento do desenho final.

Identificamos algumas vantagens e desvantagens em relação a utilização do Agent-Sheets como plataforma de desenvolvimento. A principal vantagem está associada ao modelo de programação do AgentSheets, esse modelo facilita a implementação de apli-cações para pequenos jogos e simulação. Por outro lado não foi possível usar o modelo de programação do AgentSheets como uma metalinguagem, dificultando dessa forma a disponibilização para o usuário de recursos importantes que estão disponíveis para o desenvolvedor. O AgentSheets também não oferece recursos típicos de outros modelos de programação que auxiliem na tarefa de configuração e programação.

Seria interessante aplicar o Método de Avaliação de Comunicabilidade para ver como o usuário recebe nossa mensagem e comparar esse resultado com a metamensagem que criamos. Porém, podemos nos adiantar e levantar a hipótese de que nós não consegui-mos atingir com total sucesso nossa meta de criar um simulador que auxilie no ensino de conceitos de RSSF, devido aos vários problemas de comunicabilidade que surgiram durante a implementação e foram descritos anteriormente. Sendo assim, a princípio, po-demos dizer que o AgentSheets não demonstrou ser a ferramenta mais adequada para a construção de um simulador dado o contexto que regeu este trabalho.

Como trabalho futuro, em cima da solução atual, podemos investir mais tempo para refinar a implementação e ficar mais próxima da Interface Proposta.

Uma outra linha que pode ser seguida é a avaliação do comportamento de outros modelos de programação utilizando o processo baseado em Metamensagem e Signos da Engenharia Semiótica. Isto é, especificar uma aplicação utilizando o processo de definição da Engenharia Semiótica e avaliar os pontos positivos e negativos ao implementarmos essa aplicação com diferentes modelos de programação.

Referências

- [1] MOTTOLA, L.; PICCO, G. P. **Programming wireless sensor networks: Fundamental concepts and state-of-the-art**. Tech. rep., University of Trento, 2010.
- [2] REPENNING, A.; IOANNIDOU, A.. **Agent-based end-user development**. Commun. ACM, 47:43–46, September 2004.
- [3] DE SOUZA, C. S.. **Semiotic engineering: bringing designers and users together at interaction time**. Interacting with Computers, 17(3):317 – 341, 2005. Special Theme - Papers from Members of the Editorial Boards.
- [4] SOUZA, C. S. D.. **The Semiotic Engineering of Human-Computer Interaction (Acting with Technology)**. The MIT Press, 2005.
- [5] PEIRCE, C.. **The essential Peirce: Selected Philosophical Writings**, volumen 2, 1893-1913. Indiana University Press, Bloomington, IN, 1998.
- [6] DE SOUZA, C. S.; LEITÃO, C. F.; PRATES, R. O. ; DA SILVA, E. J.. **The semiotic inspection method**. In: PROCEEDINGS OF VII BRAZILIAN SYMPOSIUM ON HUMAN FACTORS IN COMPUTING SYSTEMS, IHC '06, p. 148–157, New York, NY, USA, 2006. ACM.
- [7] PRATES, R. O.; DE SOUZA, C. S. ; BARBOSA, S. D. J.. **Methods and tools: a method for evaluating the communicability of user interfaces**. interactions, 7:31–38, January 2000.
- [8] SOHRABY, K.; MINOLI, D. ; ZNATI, T.. **Wireless Sensor Networks: Technology, Protocols, and Applications**. Wiley-Interscience, 2007.
- [9] FERREIRA, J.; BARR, P. ; NOBLE, J.. **The semiotics of user interface redesign**. In: PROCEEDINGS OF THE SIXTH AUSTRALASIAN CONFERENCE ON USER INTERFACE - VOLUME 40, AUIC '05, p. 47–53, Darlinghurst, Australia, Australia, 2005. Australian Computer Society, Inc.