

Uma Contribuição à Usabilidade da Sensibilidade ao Contexto na Computação Pervasiva

João Ladislau Lopes, Luiz Antônio Palazzo, Adenauer Corrêa Yamin

Universidade Católica de Pelotas, Programa de Pós-Graduação em Informática, Pelotas, Brasil

{joaolopes, lpalazzo, adenauer}@ucpel.tche.br

Iara Augustin

Universidade Federal de Santa Maria, Programa de Pós-Graduação em Informática, Santa Maria, Brasil

august@inf.ufsm.br

Claudio Geyer

Universidade Federal do Rio Grande do Sul, Programa de Pós-Graduação em Computação, Porto Alegre, Brasil

geyer@inf.ufrgs.br

Abstract

The main objective of this work is the qualification of the mechanisms used to express and to process context information, proposing for this, the use of ontologies. EXEHDA-ON makes the premise that the use of semantics of higher expressiveness than is usually practiced in collection and treatment of the obtained data provides better description levels in the information that characterizes the context of the computational environment. The main contributions of the work are the proposal of an ontological model for the pervasive environment domain provided by EXEHDA and the integration of this proposal with the software architecture of EXEHDA. The results obtained with EXEHDA-ON point to the feasibility of using ontologies for modeling pervasive environments, and for processing the corresponding ontological model in the data extraction that provides adaptation decisions in middleware and in applications.

Keywords: Pervasive Computing, Context-aware Computing, ontologies

1 Introdução

Mark Weiser idealizou ambientes físicos com dispositivos computacionais integrados que auxiliariam indivíduos na realização de suas tarefas cotidianas ao fornecer-lhes informações e serviços de forma contínua e transparente [19]. Essa visão resume o que se espera da Computação Pervasiva: acesso do usuário ao seu ambiente computacional independente de localização, tempo e equipamento [4].

Em um ambiente de Computação Pervasiva, os dispositivos, serviços e componentes de software devem ser conscientes de seus contextos e colaborativamente adaptar-se às suas mudanças, caracterizando assim sensibilidade ao contexto. Um mecanismo para sensibilidade ao contexto deve prover suporte para diferentes tarefas, dentre as quais: (i) obtenção do contexto de diversas fontes (sensores físicos e sensores lógicos); (ii) interpretação dos dados sensorados, gerando dados contextualizados e (iii) disseminação das informações para as partes interessadas, de modo distribuído e personalizado [2] [21] [14].

Considerando este cenário, este trabalho apresenta um mecanismo de sensibilidade ao contexto para coletar, processar e disseminar informações de contexto na perspectiva da Computação Pervasiva, considerando o ambiente pervasivo definido no projeto ISAM (Infra-estrutura de Suporte às Aplicações Móveis Distribuídas) e provido pelo *middleware* EXEHDA (*Execution Environment for Highly Distributed Applications*) [1] [20].

O EXEHDA-ON utiliza uma abordagem baseada em ontologias para a modelagem do contexto do ambiente pervasivo, bem como para realizar pesquisa no correspondente modelo ontológico. Este modelo ontológico tem abrangência celular, sendo alimentado por um serviço de monitoramento. Deste modo, o modelo descreve semanticamente o estado atual do ambiente, utilizando um vocabulário comum e interpretável pelos servidores de contexto existentes nas células de execução do EXEHDA.

O artigo está organizado nas seguintes seções: a seção 2 descreve as principais características do *middleware* EXEHDA; a seção 3 apresenta as tecnologias utilizadas na construção e no processamento do modelo ontológico do EXEHDA-ON; a seção 4 trata dos aspectos relacionados à modelagem do EXEHDA-ON; a seção 5 descreve a implementação do servidor de contexto do EXEHDA-ON; a seção 6 apresenta os trabalhos relacionados; a seção 7 apresenta um estudo de caso; e a seção 8 apresenta as considerações finais.

2 *Middleware* EXEHDA: Principais Características

O EXEHDA é um *middleware* adaptativo ao contexto e baseado em serviços que visa criar e gerenciar um ambiente pervasivo, bem como promover a execução das aplicações direcionadas à Computação Pervasiva. Estas aplicações são distribuídas, móveis e adaptativas ao contexto em que seu processamento ocorre, devendo estar disponíveis, a partir de qualquer lugar, todo o tempo.

O suporte à adaptação no EXEHDA está associado à operação do subsistema de reconhecimento de contexto e adaptação. Este subsistema inclui serviços que tratam desde a extração das “informações brutas” sobre as características dinâmicas e estáticas dos recursos que compõem o ambiente pervasivo, passando pela identificação em alto nível dos elementos de contexto, até o disparo das ações de adaptação em reação a modificações no estado de tais elementos de contexto.

O serviço “Collector” é responsável pela extração das informações diretamente dos recursos envolvidos, as quais, posteriormente refinadas, darão origem aos elementos de contexto. Um componente “Monitor”, localizado no recurso, gerencia um conjunto de sensores parametrizáveis. Por outro lado, entre os consumidores da informação extraída pela monitoração estão os serviços “ContextManager” e “Scheduler” [20].

3 EXEHDA-ON: Tecnologias Utilizadas

A construção e o processamento do modelo ontológico do EXEHDA-ON são baseados em tecnologias da Web Semântica. Nesta seção são caracterizadas as principais tecnologias empregadas na sua implementação.

Ontologias

Uma ontologia corresponde a uma especificação formal e explícita de uma conceituação compartilhada, onde: (i) conceituação, se refere ao modelo abstrato do mundo real; (ii) explícita, significa que os conceitos e seus requisitos são definidos explicitamente; (iii) formal, indica que a ontologia é processável por máquina, permite raciocínio automático e possui semântica lógica formal; (iv) compartilhada, significa que uma ontologia captura o conhecimento apresentado não apenas por um único indivíduo, mas por um grupo [7].

Web Ontology Language - OWL

A linguagem escolhida para construção do modelo ontológico do EXEHDA-ON foi a OWL [12], padrão para a Web Semântica. Como sub-linguagem foi adotada a OWL-DL, esta sub-linguagem corresponde à lógica de descrição e provê um maior grau de expressividade onde todas as conclusões são computáveis e todas as computações terminam em tempo finito. A OWL fornece suporte a metadados RDF, abstrações de classes, generalização, agregação, relações de transitividade, simetria e detecção de inconsistências.

API Jena

No EXEHDA-ON propõem-se o uso da API Java do toolkit Jena [12], por oferecer: (i) mecanismos para manipulação de modelos RDF em memória, bases de dados relacionais e arquivos; (ii) suporte à linguagem de consulta de dados RDF SPARQL (*SPARQL Protocol And RDF Query Language*) [17]; (iii) um conjunto de APIs para manipulação de ontologias codificadas em OWL; (iv) e máquinas de inferência baseadas em ontologias e regras, bem como um mecanismo que permite integrar máquinas de inferência de terceiros.

SPARQL

SPARQL é uma linguagem recomendada pelo W3C para realização de consultas em ontologias descritas em RDF e linguagens derivadas. É uma linguagem “orientada a dados”, ou seja, permite apenas extrair dados de documentos RDF disponíveis em rede ou armazenados em um meio físico qualquer, não possuindo

mecanismos de inferência. Para que as consultas possam ser realizadas e as informações possam ser extraídas, a linguagem SPARQL disponibiliza uma sintaxe que, embora com algumas particularidades, funciona de maneira similar à linguagem SQL (*Structured Query Language*).

4 Aspectos de Modelagem do EXEHDA-ON

A modelagem do EXEHDA-ON contemplou dois principais esforços de concepção: (i) modelagem ontológica do ambiente pervasivo, o qual prevê o uso de ontologias implementadas em OWL, sobre as quais o EXEHDA-ON realiza a representação e o processamento das informações de contexto; (ii) modelagem da arquitetura de software com a especificação dos diferentes serviços que a integram.

4.1 Modelagem Ontológica

O modelo ontológico para uso no EXEHDA-ON foi definido considerando aspectos que modelassem o domínio do ambiente pervasivo provido pelo EXEHDA. A perspectiva é que este modelo represente o estado atual do ambiente de execução pervasivo provido pelo EXEHDA, gerando deste modo um conhecimento sobre o mesmo, possibilitando assim sua manipulação pelo servidor de contexto, o qual responde às demandas introduzidas pelas aplicações dos usuários.

A natureza da informação de contexto, nas células de execução, pode ser estática ou dinâmica. Um exemplo de informação estática são os atributos relativos ao tipo do dispositivo, os quais não se alteram com o passar do tempo. Informações dinâmicas traduzem aspectos do contexto que oscilam em uma determinada frequência. Como exemplo de informações dinâmicas têm-se: a temperatura do processador, a ocupação de memória e a ocupação de disco. Estas informações são obtidas por meio de monitoramento periódico ou coletadas por procedimentos disparados por eventos [20].

A metodologia utilizada para a construção da ontologia descritiva do ambiente pervasivo é baseada na proposta de [8]. Os passos utilizados na definição da ontologia projetada para uso no EXEHDA-ON foram os seguintes: (i) definição do domínio e captura do conhecimento; (ii) conceituação do conhecimento capturado em um conjunto de representações; (iii) desenvolvimento do modelo conceitual em uma linguagem formal.

Entre os resultados da etapa de “conceituação do conhecimento capturado em um conjunto de representações”, destaca-se o desenvolvimento de um glossário de termos, contendo os conceitos pesquisados, sua descrição e atributos relacionados. Ainda nesta etapa foi construída uma árvore de classificação de conceitos, que permite visualizar as classes definidas para a ontologia do ambiente pervasivo tratado pelo EXEHDA-ON, mostrando sua hierarquia (vide Figura 1).

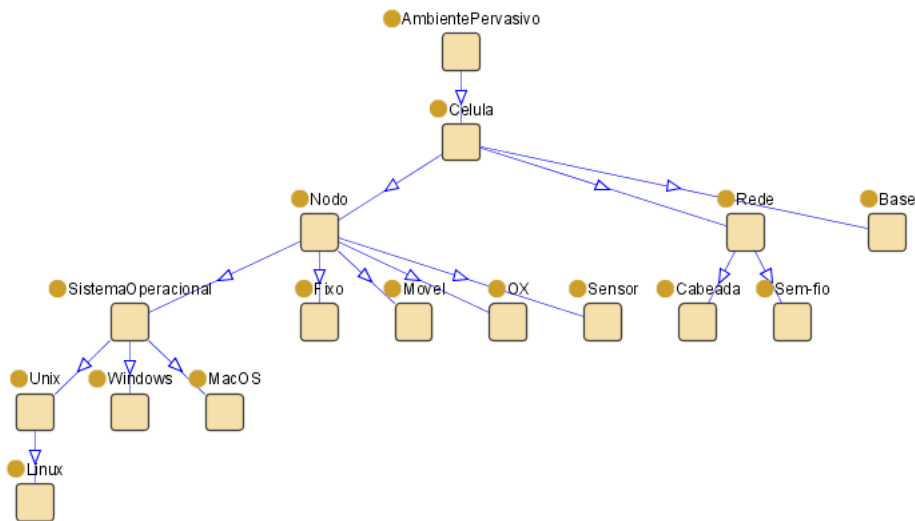


Figura 1: Árvore de conceitos da ontologia do ambiente pervasivo

4.2 Modelagem da Arquitetura de Software

Nesta seção é descrita a arquitetura de software do EXEHDA-ON e a forma de adequação da mesma ao Subsistema de Reconhecimento de Contexto e Adaptação do EXEHDA, identificando seus pontos de integração e seus serviços.

De modo geral, uma arquitetura para sistemas sensíveis ao contexto envolveria uma série de sensores, de software e/ou de hardware, que monitoram os aspectos de interesse do ambiente computacional (*middleware* e aplicações). As informações colhidas por esses sensores são passadas a um conjunto de serviços de contexto, onde são processadas e/ou modificadas para que possam ser entregues aos consumidores das informações contextualizadas [10].

A arquitetura do EXEHDA-ON prevê a inclusão de serviços e componentes nos “EXEHDA nodos” e no “EXEHDA base”, os quais são abstrações do ambiente pervasivo provido pelo EXEHDA. A Figura 2 mostra as funcionalidades do EXEHDA-ON integradas ao Subsistema de Reconhecimento de Contexto e Adaptação do EXEHDA.

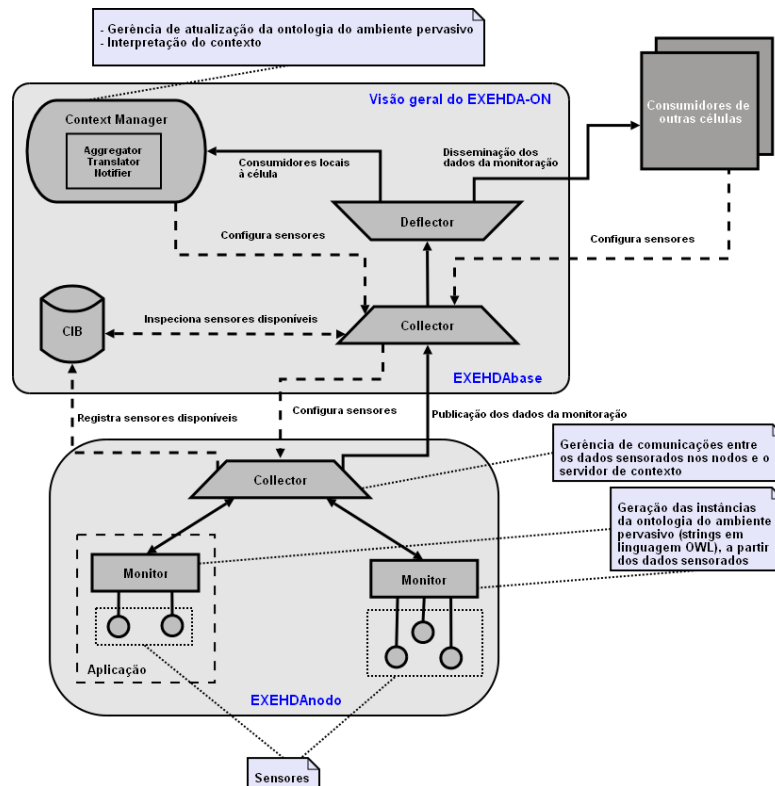


Figura 2: Integração das funcionalidades do EXEHDA-ON

Em cada EXEHDA nudo, junto ao componente “Monitor”, o serviço do EXEHDA-ON “Gerenciador de monitoramento do contexto” fica com a responsabilidade de receber os dados sensorados e gerar as instâncias da ontologia do ambiente pervasivo, representadas por *strings* em linguagem OWL. Por sua vez, o serviço “Collector” do EXEHDA é responsável pela gerência das comunicações entre os dados sensorados nos nodos e o servidor de contexto que aglutina os dados da célula como um todo.

A localização dos serviços do EXEHDA-ON dentro do EXEHDA base é junto ao “ContextManager”, que é serviço chave na construção de informações globais de contexto, mais especificamente dentro das estruturas de cadeias de detecção de contexto do “ContextManager” (*aggregator* e *translator*), que são empregadas, respectivamente, para composição dos dados de um ou mais sensores e para abstração da informação de contexto.

Assim, no EXEHDA base atua o serviço responsável pela aglutinação das instâncias na ontologia do ambiente pervasivo, denominado “Gerenciador de atualização da base ontológica”, e o serviço responsável pelas consultas sobre a base ontológica do EXEHDA-ON, denominado “Interpretador de contexto”.

4.3 Serviço de Interpretação de Contexto

Dentre os serviços que constituem o EXEHDA-ON, o serviço “Interpretador de contexto” é central para consecução dos objetivos propostos para o EXEHDA-ON. O “Interpretador de contexto” realiza o processamento das informações de contexto com o intuito de identificar na base ontológica a existência das condições de contexto de interesse dos consumidores registrados. Este serviço recebe as solicitações dos consumidores a partir do “Gerenciador de subscrições” e realiza consultas sobre a base ontológica, utilizando o componente “Gerenciador de consultas”.

Gerenciador de consultas

O serviço de interpretação do contexto possui um componente que realiza consultas à base ontológica (vide Figura 3). Este componente constrói consultas com a linguagem SPARQL, as quais são submetidas à base ontológica. Quando um consumidor registra-se no *middleware* EXEHDA, este deve informar seu contexto de interesse. As informações registradas são processadas pelo componente “Gerenciador de consultas”. Este componente, então, (i) analisa os requisitos especificados pelo consumidor; (ii) gerencia a construção das consultas; (iii) executa as consultas sobre a ontologia; e (iv) notifica as condições de contexto existentes, em função do interesse dos consumidores registrados.

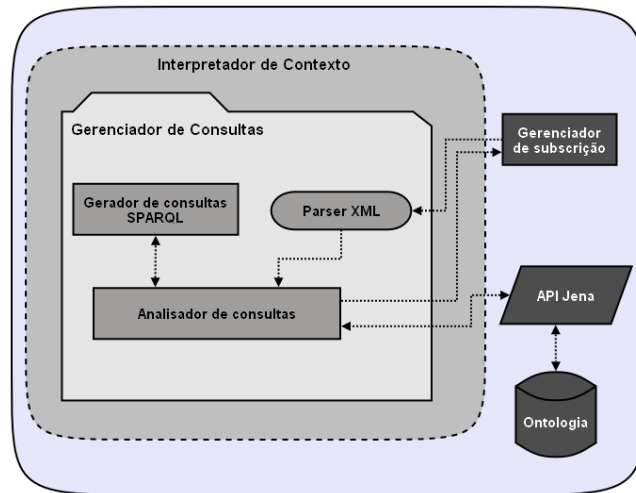


Figura 3: Gerenciador de consultas do EXEHDA-ON

5 Implementação do Servidor de Contexto

O servidor de contexto para o ambiente pervasivo foi modelado e desenvolvido com o uso da API Jena e a linguagem de programação Java. Os dados sensorados são compostos em instâncias da ontologia do ambiente pervasivo no nodo (processamento de *strings*), as quais são enviadas ao servidor de contexto, localizado no EXEHDABase. Estas instâncias contêm as informações de identificação do nodo e os dados sensorados. No servidor de contexto é feita a aglutinação destas instâncias na ontologia do ambiente pervasivo na célula com a utilização da API Jena.

Com o intuito de abstrair aspectos de baixo nível relativos ao tratamento das comunicações em rede, na implementação do servidor de contexto foram utilizados os serviços de comunicação do EXEHDA. A classe principal do servidor de contexto executa uma *thread* responsável por utilizar esses serviços para aguardar as atualizações de contexto dos nodos.

O servidor de contexto mantém uma base de dados ontológica com informações sobre o estado do ambiente pervasivo. Todos os procedimentos relativos à manipulação da ontologia são realizados com o uso da API Jena. Quando a *thread* responsável pela comunicação recebe a ontologia do ambiente pervasivo atualizada de algum nodo ela a repassa para a classe principal “*ContextServer*”, a qual possui o método

“*mergeOntologies*” responsável por verificar a ontologia recebida e realizar as atualizações necessárias na base de dados ontológica. O servidor é executado na base da célula, gravando as alterações ocorridas na ontologia dentro do arquivo *log* do EXEHDA.

Também, um serviço de consultas foi desenvolvido junto ao servidor de contexto. Este serviço possibilita a realização de pesquisas na ontologia do ambiente pervasivo utilizando a linguagem SPARQL. A linguagem de consulta SPARQL é utilizada através da biblioteca ARQ da API Jena. Os métodos da classe *QueryConnection* implementam as consultas em SPARQL que pesquisam propriedades dos nodos, tais como: temperatura do processador, memória total, tamanho do disco e número de processadores.

6 Trabalhos Relacionados

Nesta seção é feita uma comparação entre os principais projetos em Computação Sensível ao Contexto utilizados como referência para a definição deste trabalho, e a solução adotada no EXEHDA-ON. Esta comparação está organizada em três grandes categorias: arquitetura, modelagem de contexto e processamento de contexto.

Arquitetura: *CASS* [6] tem uma arquitetura baseada em *middleware* centralizado. *CoBra* [3] possui arquitetura baseada em agentes. Considerando que o EXEHDA-ON foi concebido como extensão do *middleware* EXEHDA, ele mantém os aspectos arquiteturais do mesmo. Assim, sua arquitetura é baseada em serviços, cuja integração visa fornecer a infra-estrutura necessária para suporte à sensibilidade ao contexto em um ambiente pervasivo.

Modelagem de contexto: *Context Toolkit* [5] manipula o contexto através de tuplas com atributos e valores que são codificados usando XML. *Hydrogen* [11] usa uma abordagem orientada a objetos para modelagem do contexto. A estrutura e o vocabulário da ontologia aplicada no *Context Managing Toolkit* são descritos em RDF. No Gaia [16] o contexto é representado através de predicados escritos em DAML+OIL. Abordagens baseadas em ontologias escritas em OWL para modelagem de contexto são encontradas nos projetos *SOCAM* [9] e *CoBrA*. A solução adotada no EXEHDA-ON para modelagem do contexto consiste no uso de ontologias próprias desenvolvidas em OWL, linguagem recomendada pelo W3C, visto que, revisa e incorpora melhoramentos às demais linguagens, tais como: RDF e RDF *Schema* e DAML-OIL. O modelo baseado em ontologias é o mais promissor para a modelagem de contexto em ambientes pervasivos [18].

Processamento de contexto: o processamento do contexto nos projetos *CoBrA* e *SOCAM* é baseado em modelos ontológicos. Modelos de contexto não baseados em ontologias apresentam menor expressividade, sendo esta uma das limitações dos projetos *CASS* e *Context Toolkit*. No EXEHDA-ON o processamento do contexto é baseado em um modelo ontológico, sendo realizado através de serviços e componentes de software que manipulam a base ontológica, utilizando a linguagem de consulta SPARQL e a API Jena.

7 Estudo de Caso: Suporte à Decisão para o Escalonamento de Recursos

Na perspectiva deste estudo de caso, os nodos existentes em uma célula, em função do seu estado, são selecionados para computações paralelas numericamente intensivas. Nesta seção é caracterizado o uso do EXEHDA-ON enquanto provedor de informações para a tomada de decisões por parte de um escalonador de recursos quando da gerência de uma execução paralela.

No EXEHDA, a escolha do nodo onde é instalado um componente de software denomina-se adaptação não-funcional. Neste tipo de adaptação a natureza da computação a ser realizada é mantida, independentemente do equipamento aonde o processamento será disparado [20].

Para as avaliações neste estudo de caso, foi utilizado o “CpuSteal” [15], um gerador de cargas baseado em uma distribuição de probabilidades exponencial. O “CpuSteal” opera em ciclos de ativação e desativação. Quando ativo, ele gera operações de ponto flutuante ocupando o processador, quando inativo ele entra em repouso. O controle do nível de ocupação média do processador é feito através da determinação de quanto tempo ele permanece ativo e quanto tempo ele fica em repouso em cada um dos ciclos.

Na avaliação do suporte do EXEHDA-ON à tomada de decisão para o escalonamento de recursos foi utilizado o *cluster* H3P do Centro Politécnico da UCPEL (<http://h3p.g3pd.ucpel.tche.br>), composto de 10 nodos com sistema operacional Linux, frequência de 2020MHz e memória de 185MB. Estes computadores são interligados por uma rede *FastEthernet*. As cargas computacionais ficaram distribuídas conforme Tabela 1 e foram produzidas pelo gerador de cargas “CPUSteal”. O nodo “H3P” é o gerenciador do sistema, assim

Tabela 1: Carga média dos nodos

| Nodo | Carga (%) |
|------|-----------|
| H4 | 18.48 |
| H5 | 18.82 |
| H6 | 39.30 |
| H7 | 40.26 |
| H8 | 59.95 |
| H9 | 61.45 |

não foi utilizado para o processamento. Nos nodos “H1”, “H2” e “H3” não foi imposta carga gerada pelo “CPUSteal”.

7.1 Aplicação de Teste Desenvolvida

A aplicação prevista instala computações remotas em função da disponibilidade de recursos computacionais na célula de execução. A medida que surjam recursos que atendam os requisitos mínimos estabelecidos pelo desenvolvedor no âmbito do contexto celular o componente da aplicação responsável pela gerência da execução é notificado para que execute os procedimentos pertinentes.

A aplicação desenvolvida é do tipo sintético, e tem por objetivo central permitir a exploração das funcionalidades do EXEHDA-ON. Sua finalidade matemática é calcular o número π utilizando o método de Monte Carlo [13], do ponto de vista computacional se trata de uma computação paralela do tipo *bag-of-tasks* cujo número de computações paralelas é determinado pelo programador. Considerando que o processamento de cada uma das iterações pode ocorrer de forma independente, uma das maneiras de particionar o problema é dividir o número total de iterações desejadas entre vários nodos.

7.2 Resultados Obtidos

O suporte à decisão para o escalonamento de recursos com o EXEHDA-ON foi avaliado através de quatro execuções da aplicação nos equipamentos do *cluster* H3P. Parte dos nodos foi submetido a diferentes cargas sintéticas, através de um gerador programado de contexto de carga de processador, conforme mostra a Tabela 1.

Em cada uma das quatro execuções foram selecionados cinco nodos, sendo que em três destas execuções os nodos foram escolhidos de forma aleatória. Em uma das execuções os nodos foram selecionados a partir de consulta SPARQL construída pelo EXEHDA-ON, considerando um critério de carga da CPU inferior a 70%, bem como uma ordem dos nodos crescente de carga da CPU e decrescente de poder computacional (vide Listagem 1).

Listagem 1: Nodos selecionados

Consulta SPARQL executada:

```
PREFIX exehda-on:<http://exehda.ucpel.tche.br/exehda-on#>
SELECT ?NODO ?CARGACPU ?PODERCOMP
WHERE {?NODO exehda-on:CargaCPU ?CARGACPU; exehda-on:PoderComp ?PODERCOMP}
FILTER (?CARGACPU <= 70.00)} ORDER BY ?CARGACPU DESC(?PODERCOMP)
```

Resultado da consulta:

| Nodo | CargaCPU | PoderComp |
|------|----------|-----------|
| H1 | 0.00 | 3997.69 |
| H3 | 0.00 | 3997.69 |
| H2 | 0.00 | 3325.95 |
| H4 | 18.48 | 3997.69 |
| H5 | 18.82 | 3997.69 |
| H6 | 39.30 | 3997.69 |
| H7 | 40.26 | 3997.69 |
| H8 | 59.95 | 3325.95 |
| H9 | 61.45 | 3325.95 |

Nodos selecionados: h1, h2, h3, h4, h5

Observa-se que em função do resultado ordenado da consulta torna-se possível a seleção por parte do módulo Escalonador do número necessário para o processamento do cálculo, considerando as tarefas que faltam.

Cabe ressaltar que caso o número de nodos atendendo ao critério fosse inferior a cinco, estes seriam selecionados e, o escalonador aguardaria novos nodos para dar continuidade ao restante do processamento. Quando o número de nodos necessário para o processamento é atingido o nodo mestre cancela a subscrição do contexto de interesse.

O cálculo do π pelo método de Monte Carlo foi realizado com o lançamento de 10 bilhões de pontos. Este número total de iterações foi dividido entre os cinco nodos selecionados em cada uma das execuções. Na Listagem 2 são mostrados os resultados obtidos. A forma de apresentação é a mesma registrada no *log* do EXEHDA.

Listagem 2: Resultados obtidos

```
Execução com nodos selecionados pelo EXEHDA-ON
...
O resultado de hostid:3. exehda-h3p foi 1570799271 em 812.632 segundos
O resultado de hostid:1. exehda-h3p foi 1570789020 em 813.025 segundos
O resultado de hostid:2. exehda-h3p foi 1570817787 em 983.396 segundos
O resultado de hostid:4. exehda-h3p foi 1570801460 em 991.103 segundos
O resultado de hostid:5. exehda-h3p foi 1570780264 em 995.219 segundos
Resultado do calculo do PI: 3.1415951208
...

Execução 1: nodos aleatórios
...
O resultado de hostid:1. exehda-h3p foi 1570801127 em 811.151 segundos
O resultado de hostid:4. exehda-h3p foi 1570781893 em 935.631 segundos
O resultado de hostid:2. exehda-h3p foi 1570806011 em 976.104 segundos
O resultado de hostid:5. exehda-h3p foi 1570804099 em 997.109 segundos
O resultado de hostid:7. exehda-h3p foi 1570797361 em 1221.303 segundos
Resultado do calculo do PI: 3.1415961964
...

Execução 2: nodos aleatórios
...
O resultado de hostid:3. exehda-h3p foi 1570827840 em 813.745 segundos
O resultado de hostid:5. exehda-h3p foi 1570820120 em 957.303 segundos
O resultado de hostid:7. exehda-h3p foi 1570771285 em 1192.303 segundos
O resultado de hostid:9. exehda-h3p foi 1570764426 em 1818.579 segundos
O resultado de hostid:8. exehda-h3p foi 1570813229 em 1850.741 segundos
Resultado do calculo do PI: 3.14159876
...

Execução 3: nodos aleatórios
...
O resultado de hostid:1. exehda-h3p foi 1570800575 em 812.077 segundos
O resultado de hostid:3. exehda-h3p foi 1570802723 em 814.551 segundos
O resultado de hostid:2. exehda-h3p foi 1570791423 em 971.851 segundos
O resultado de hostid:6. exehda-h3p foi 1570793860 em 1084.87 segundos
O resultado de hostid:8. exehda-h3p foi 1570801037 em 1830.695 segundos
Resultado do calculo do PI: 3.1415958472
...
```

Observando os resultados na Listagem 2 pode-se verificar a precisão do impacto do gerador de carga no tempo de execução do módulo de cálculo. Por exemplo, os nodos h1 e h5 possuem o mesmo poder computacional, tendo sido imposto ao nodo h5 uma carga média de 18.82%. Analisando os tempos destes nodos, obtidos nas diferentes execuções apresentadas na Listagem 2, pode-se observar diferenças em torno de 20%.

Os valores registrados no *log* tem a precedência definida pela ordem que ocorrem, o último a concluir a computação determina o tempo total de execução.

8 Considerações Finais

Uma questão relevante na sensibilidade ao contexto é o grau de expressividade que se pode obter na descrição dos possíveis estados do mesmo. Quanto maior a expressividade do modelo de informação do contexto, maior é a capacidade de representar a estrutura e a semântica dos conceitos. A análise dos métodos e tecnologias de Web Semântica caracteriza que quanto mais formal o modelo para descrição do contexto, maior é a capacidade de realização de pesquisa sobre o mesmo. Neste sentido, o uso de ontologias contribui para qualificar os mecanismos de sensibilidade ao contexto, em função da elevada expressividade que o uso destas pode propiciar.

O processo de tradução dos dados sensorados para contextualizados no EXEHDA vinha sendo feito por algoritmos e estruturas de dados particulares para cada tipo de aplicação. Neste sentido, a contribuição central da pesquisa EXEHDA-ON é minimizar a gerência desta personalização por parte do programador, tendo proposto a construção de um modelo ontológico que descreva semanticamente o estado atual do ambiente, utilizando um vocabulário comum e interpretável pelo servidor de contexto. Além disso, o fato

de ser utilizado um modelo ontológico também torna possível a realização de pesquisas sobre o estado do ambiente pervasivo, com a utilização de uma linguagem de alto nível.

Na perspectiva de continuidade da pesquisa desenvolvida durante a concepção do EXEHDA-ON os seguintes aspectos poderão ser explorados em trabalhos futuros: (i) analisar o desempenho do EXEHDA-ON considerando as diferentes possibilidades de modelagem ontológica do contexto; (ii) construir diferentes modelos ontológicos em função do domínio a ser tratado; (iii) expandir o mecanismo para construção de contextos que englobem várias células.

Referências

- [1] AUGUSTIN, I. *Abstrações para uma Linguagem de Programação Visando Aplicações Móveis Conscientes do Contexto em um Ambiente de Pervasive Computing*. Tese (doutorado em ciência da computação), Instituto de Informática, UFRGS, Porto Alegre, RS, 2004.
- [2] AUGUSTIN, I., YAMIN, A. C., DA SILVA, L. C., REAL, R. A., FRAINER, G., AND GEYER, C. F. R. Isamadapt: abstractions and tools for designing general-purpose pervasive applications: Experiences with auto-adaptive and reconfigurable systems. *Softw. Pract. Exper.* 36, 11-12 (2006), 1231–1256.
- [3] CHEN, H. *An Intelligent Broker Architecture for Pervasive Context-Aware Systems*. Dissertation (doctor of philosophy), University of Maryland, Baltimore, 2004.
- [4] DA COSTA, C. A., YAMIN, A. C., AND GEYER, C. F. R. Toward a general software infrastructure for ubiquitous computing. *IEEE Pervasive Computing* 7, 1 (2008), 64–73.
- [5] DEY, A., SALBER, D., AND ABOWD, G. A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. *Human-Computer Interaction* 16 (2001).
- [6] FAHY, P., AND CLARKE, S. Cass - middleware for mobile context-aware applications. *MobiSys - International Conference on Mobile Systems, Applications, and Services* (2004).
- [7] FENSEL, D. *Ontologies: Silver bullet for knowledge management and electronic commerce*. Springer - Verlag (2000).
- [8] FERNANDEZ, M., GOMES-PEREZ, A., AND JURISTO, N. Methontology: From ontological art towards ontological engineering. In *National Conference on Artificial Intelligence (AAAI) Spring Symposium on Ontological Engineering, 14*. (Stanford, EUA., 1997), Papers from the AAAI Spring Symposium, AAAI press, pp. 33–40.
- [9] GU, T., PUNG, H., AND ZHANG, D. A middleware for building context-aware mobile services. In *Proceedings of IEEE Vehicular Technology Conference* (Milão, Itália, Maio 2004), IEEE Press.
- [10] HENRICKSEN, K., AND INDULSKA, J. Developing context-aware pervasive computing applications: Models and approach. *Pervasive and Mobile Computing* 2, 2 (2006), 37–64.
- [11] HOFER, T., SCHWINGER, W., PICHLER, M., LEONHARTSBERGER, G., AND ALTMANN, J. Context-awareness on mobile devices - the hydrogen approach.
- [12] MCBRIDE, B. Jena api - a semantic web framework for java, 2007. Disponível em: <<http://jena.sourceforge.net/ontology/>>. Acesso em junho de 2009.
- [13] PRESS, W. H., FLANNERY, B. P., TEUKOLSKY, S. A., AND VETTERLING, W. T. *Numerical Recipes in C: The Art of Scientific Computing. 2nd ed.* Cambridge University Press, Cambridge, UK, 1992.
- [14] RAENTO, M., OULASVIRTA, A., PETIT, R., AND TOIVONEN, H. Contextphone: A prototyping platform for context-aware mobile applications. *IEEE Pervasive Computing* 4, 2 (April-June 2005), 51–59.
- [15] REAL, R. A. Típs, uma proposta de escalonamento direcionada à computação pervasiva, 2004.
- [16] ROMAN, M., HESS, C., CERQUEIRA, R., RANGANAT, A., CAMPBELL, R., AND NAHRSTEDT, K. Gaia: A middleware infrastructure to enable active spaces. *IEEE Pervasive Computing* (Outubro 2002).

- [17] SEABORNE, A. Sparql - a query language for rdf, 2007. Disponível em: <<http://www.w3.org/TR/rdf-sparql-query/>>. Acesso em junho de 2009.
- [18] STRANG, T., AND LINNHOFF-POPIEN, C. A context modeling survey. In *Workshop on Advanced Context Modelling, Reasoning and Management, UbiComp 2004 - The Sixth International Conference on Ubiquitous Computing* (Nottingham, England, 2004), UbiComp.
- [19] WEISER, M. The computer for the 21st century. *Scientific American* 3, 265 (Setembro 1991), 94–104.
- [20] YAMIN, A. *Arquitetura para um Ambiente de Grade Computacional Direcionado às Aplicações Distribuídas, Móveis e Conscientes do Contexto da Computação Pervasiva*. Tese (doutorado em ciência da computação), Instituto de Informática, UFRGS, Porto Alegre, RS, 2004.
- [21] YAMIN, A. C., AUGUSTIN, I., BARBOSA, J., DA SILVA, L. C., REAL, R. A., FILHO, A. S., AND GEYER, C. F. R. Exehda: Adaptive middleware for building a pervasive grid environment. *Frontiers in Artificial Intelligence and Applications - Self-Organization and Autonomic Informatics* 135 (2005), 203–219.