

Introduzindo o Gerenciando de Tarefas Clínicas em um Middleware da Computação Pervasiva

Giuliano Geraldo Lopes Ferreira¹, Fabio Lorenzi da Silva¹,
Giovani Rubert Librelotto¹, Iara Augustin¹, Adenauer Correa Yamin²

¹ Programa de Pós-Graduação em Informática – Universidade Federal de Santa Maria
Av. Roraima, 1000 – Bairro Camobi – 97.105-900 – Santa Maria – RS – Brasil

² Escola de Informática – Universidade Católica de Pelotas
Rua Félix da Cunha, 412 – 96.010-000 – Pelotas – RS – Brasil

{giuliano, lorenzi, librelotto, august}@inf.ufsm.br,
adenauer@ucpel.edu.br

Abstract: *Currently in its second generation, the Pervasive Computing is focused on development of interactive and programmable environments, which assist the user in their daily activities. The health care system of the future addresses the use of Pervasive Computing as a way to automate and optimize the clinical activities. This study tried to adapt a middleware for management of pervasive environment to support and manage the execution of clinical tasks (pervasive applications that help the clinician to carry out their activities), satisfying some requirements of the activity-driven computing, and creating a wizard to assist clinicians in their daily activities.*

Resumo: *Atualmente, em sua segunda geração, a Computação Pervasiva está direcionada ao desenvolvimento de ambientes programáveis e interativos, os quais auxiliarão o usuário em suas atividades diárias. O sistema de saúde do futuro prevê o uso da Computação Pervasiva como forma de otimizar e automatizar as atividades clínicas. Tendo em vista essa perspectiva, neste trabalho buscou-se adaptar um middleware de gerenciamento do ambiente pervasivo para suportar e gerenciar a execução de tarefas clínicas (aplicações pervasivas que auxiliam o clínico a realizar suas atividades), atendendo a alguns requisitos da computação orientada a atividades, e criando um assistente para auxiliar os clínicos em suas atividades diárias.*

1. Introdução

Computação Pervasiva (*Pervasive Computing*) é um novo paradigma computacional com tecnologia de comunicação e informação em qualquer lugar, acessível por qualquer pessoa, disponível todo o tempo [Saha & Mukherjee 2003]. Os recursos computacionais estarão totalmente integrados ao ambiente físico, da forma mais transparente possível [Weiser 1999].

A primeira geração da computação pervasiva concentrou-se em produzir *middlewares* para gerenciamento e disponibilização do ambiente com objetivo de entender os requisitos e necessidades destes [Aura 2002] [Gaia 2002] [Isam 2001]. Os sistemas existentes permitiram levantar alguns dos conceitos inerentes à natureza do espaço pervasivo: comunicação, mobilidade, contexto e tarefas cotidianas. Comunicação e mobilidade já são tratadas há muito tempo; contexto está sendo tratado

atualmente e tarefas exigem um esforço maior, pois estão no estágio mais recente, o de definição do que se deseja [Augustin, Lima & Yamin 2006].

O atual foco da Computação Pervasiva, segunda geração, volta-se ao processamento das **atividades humanas** da forma mais integrada possível ao ambiente real conhecido pelo profissional. Nessa perspectiva, uma das grandes áreas de aplicação é a Saúde, já que esta possui muitas situações onde há necessidade de que as informações do mundo físico sejam adquiridas pró-ativamente e integradas automaticamente às aplicações (mundo virtual). Estudos revelam que a *Pervasive Healthcare Computing* oferece vantagens competitivas aos provedores de serviços de saúde; em particular, aumenta a eficiência do serviço; a qualidade e melhora o gerenciamento da relação com o paciente [Varshney 2003].

Neste momento, pode-se considerar que a Computação Pervasiva ou Ubíqua na Saúde (*pHealth* ou *UbiHealth*) está em sua primeira geração, a qual procura entender as necessidades, características e tecnologias para projetar sistemas que criarão o hospital do futuro [Augustin, Lima & Yamin 2006]. No Brasil, observa-se que o maior foco das pesquisas em Informática na Saúde ainda é em registros de pacientes e medicamentos, prontuário eletrônico e informação sobre cursos clínicos (www.sbis.org.br). Porém, o hospital do futuro prevê o uso de tecnologias da Computação Pervasiva que formarão um espaço inteligente, reativo e pró-ativo, onde os sistemas de gerenciamento de informações tomarão decisões e adaptar-se-ão às situações detectadas [Bardram & Christensen 2007] [Bardram 2003].

Projetos de pesquisa em Computação Pervasiva, como Aura [Garlan et al 2002] e Gaia [Roman et al 2002], que introduziram os conceitos de Computação Orientada a Tarefas/Atividades (*Task-Oriented or Activity-Driven Computing*), abordam os problemas relativos ao gerenciamento das tarefas do usuário de forma pró-ativa, o sistema age em nome do usuário. Nesses, o sistema pervasivo induz o usuário a realizar as suas tarefas de acordo com a forma pré-definida e pré-programada.

Porém, argumenta-se que a pró-atividade do sistema não deve ser tão rigorosa, uma vez que este é projetado de forma genérica e não personalizada. Se o usuário desejar fazer de forma diferente, ele deve ter possibilidade de interagir, comandar e influenciar a execução das suas tarefas/atividades gerenciadas pelo sistema pervasivo. Portanto, para se definir um sistema pervasivo é necessário que este esteja centrado no usuário-final (clínicos) e forneça suporte às tarefas de seu trabalho diário, equilibrando a pró-atividade (agir em nome do usuário) com a personalização (forma individual de cada um realizar sua atividade) [Christensen & Bardram 2002].

Como se pode observar, a segunda geração de sistemas ubíquos traz novos desafios em diversas áreas. Entre elas está o desenvolvimento de novos *middlewares* que, além de gerenciar o ambiente pervasivo (primeira geração), devem gerenciar as tarefas cotidianas dos usuários. O *middleware* deve permitir, aos usuários, personalizar (programar), executar, interromper, retomar, e agendar suas tarefas, e ainda controlar o disparo delas em resposta a alterações no contexto.

Nesse escopo, o projeto “ClinicSpaces: auxílio às tarefas clínicas em um ambiente hospitalar do futuro baseado em tecnologias da Computação Ubíqua/Pervasiva”, em desenvolvimento na Universidade Federal de Santa Maria (UFSM), grupo GMob (www.inf.ufsm.br/gmob), propõe o desenvolvimento de uma ferramenta-piloto que permita aos clínicos a programação personalizada de suas tarefas,

as quais são gerenciadas e executadas, por um *middleware*, em um ambiente pervasivo. O objetivo central da personalização das tarefas pelo clínico é reduzir o impacto da interferência do sistema automatizado no ambiente clínico, e com isso, diminuir a rejeição aos sistemas computacionais que essa interferência pode causar.

Dessa forma, o desafio deste trabalho foi adaptar um *middleware* de gerenciamento do ambiente pervasivo (1ªG) para gerenciar também as tarefas dos usuários (clínicos), atendendo aos requisitos da segunda geração da computação pervasiva e do ambiente clínico.

Este trabalho é descrito da seguinte forma: a sessão 2 discute a modelagem de atividades/tarefas clínicas; a sessão 3 contém uma breve apresentação do *middleware* de gerenciamento do ambiente pervasivo adotado como base para este trabalho; a sessão 4 apresenta a solução proposta para adaptação do *middleware* aos requisitos da computação orientada a atividades e ao ambiente clínico; a sessão 5 discute a avaliação do protótipo implementado; a sessão 6 apresenta um caso de uso do sistema de gerenciamento de tarefas; a sessão 7 apresenta os trabalhos relacionados; e finalmente, a sessão 8 contém as considerações finais sobre o trabalho.

2. Computação Orientada a Tarefas

Atividades clínicas, como atendimento a pacientes são processos realizados de forma colaborativa, coordenada e distribuída em um espaço determinado, e podem ser auxiliadas por aplicações computacionais [Ranganathan & Campbell 2005].

Segundo a Teoria da Atividade Humana [Kaenampornpan & O'Neill, 2005], atividades humanas podem ser modeladas com sujeito (quem pratica a atividade), objetivo (o que fazer), ação (processos direcionados ao objetivo) e operações (como a ação é executada). Dessa teoria, derivaram os conceitos adotados para a modelagem das tarefas no escopo deste trabalho.

Assim, as atividades clínicas foram modeladas em tarefas (ações da teoria da atividade). Por exemplo, a atividade de “atendimento a pacientes” pode ser modelada com uma tarefa que identifica o paciente e busca seu prontuário eletrônico, apresentando-o ao médico e permitindo que este adicione as informações do atendimento. Dessa forma, cria-se uma tarefa simples que irá auxiliar o médico no atendimento a pacientes. Essa tarefa ainda pode ser ligada a outras tarefas, para modelar a atividade de forma mais completa. Por exemplo, poderia ser ligada a uma tarefa de solicitação de exames e/ou prescrição de tratamento.

As tarefas podem ser personalizadas de acordo com a forma de o usuário (clínico) realizar sua atividade. E, do mesmo modo como ações são formadas por operações, as tarefas são formadas por subtarefas, que são pequenas operações implementadas como aplicações pervasivas, as quais dependem do tipo de atividade que se deseja modelar. Além disso, foi acrescentado o conceito de *workflow* (conjunto de tarefas seqüenciais), representando atividades mais complexas.

Uma Interface de Edição de Tarefas [Lorenzi et al 2009] permite a programação orientada ao usuário-final. Dessa forma, uma tarefa é “programada” intuitivamente pelo clínico através do agrupamento de subtarefas parametrizáveis, numa seqüência que reflete a forma de o usuário realizar sua atividade (personalização).

Para o *middleware*, tanto as tarefas como suas subtarefas são aplicações pervasivas. Porém, sendo somente aplicações pervasivas, o usuário não teria uma forma prática de personalizar e controlar a execução delas. Por isso, foi necessário adaptar o *middleware* para disponibilizar meios de o usuário ter controle sobre suas tarefas, como agendar, executar, interromper, continuar, cancelar; além de criar um mecanismo que permita que as tarefas “sigam” o usuário quando ele troca de dispositivo.

3. Middleware de gerenciamento do ambiente pervasivo

O *middleware* EXEHDA – *Execution Environment for Highly Distributed Applications* [Yamin 2004] foi utilizado para gerenciar o ambiente pervasivo no qual as tarefas irão executar. O EXEHDA é um *middleware* para suporte a Computação Pervasiva que visa criar e gerenciar um ambiente pervasivo, provendo um ambiente virtual para o usuário, no qual as aplicações têm o estilo “siga-me” [Augustin; Yamin & Geyer 2005]. Assim, o EXEHDA permite ao usuário ter acesso ao seu ambiente computacional independentemente de localização e de tempo.

O EXEHDA é estruturado em um núcleo mínimo (necessário a sua inicialização), e serviços carregados sob demanda, os quais estão organizados em subsistemas que gerenciam: (a) execução distribuída; (b) comunicação; (c) reconhecimento de contexto e adaptação; e (d) acesso pervasivo aos recursos e serviços [Yamin 2004]. Os subsistemas que compõem o EXEHDA são descritos a seguir:

- Subsistema de Acesso Pervasivo, tem por finalidade dar suporte à premissa da Computação Pervasiva de acesso em qualquer lugar e todo o tempo a dados e código;
- Subsistema de Execução Distribuída, responsável pelo suporte ao processamento distribuído no EXEHDA. Esse subsistema interage com outros subsistemas, como o de reconhecimento do contexto e o de adaptação, para promover uma execução efetivamente pervasiva;
- Subsistema de Reconhecimento de Contexto e Adaptação, responsável pela aquisição de informações sobre o ambiente pervasivo, pela identificação em alto nível dos elementos de contexto e pelo disparo das ações de adaptação ao contexto;
- Subsistema de Comunicação, o qual disponibiliza mecanismos para atender, principalmente, a aspectos relacionados às desconexões, muito comum em ambientes pervasivos, devido tanto à existência de enlaces sem fio como às estratégias de economia de energia dos dispositivos móveis.

Como visto anteriormente, os subsistemas do EXEHDA são formados por serviços. Cada serviço define uma interface (funcionalidade) e pode possuir várias implementações, adequadas aos tipos de dispositivo que se pretende suportar. Os serviços providos pelo EXEHDA são personalizáveis em nível de host, sendo determinadas pelo perfil de execução [Yamin 2004]. O perfil de execução de cada host define o conjunto de serviços a ser ativado e os parâmetros para sua execução, além de associar a cada serviço uma implementação específica. Dessa forma, serviços podem ser “plugados” sem que haja necessidade de modificar o núcleo do *middleware*.

4. Adaptando o EXEHDA para gerenciar tarefas clínicas

A arquitetura para a programação e o gerenciamento das tarefas, apresentada na **Erro! Fonte de referência não encontrada.**, é organizada em níveis que refletem as visões

do sistema: (i) nível superior, composto pelo usuário-final (clínico) que interage com a ferramenta para (re)definir suas tarefas que executarão de forma reativa – disparadas por alterações no contexto; (ii) nível intermediário, composto pelo mapeamento e conversão das tarefas definidas pelo usuário e pelo gerenciamento destas; (iii) nível inferior, composto pelo conjunto de serviços do *middleware* de gerenciamento do ambiente pervasivo e de suporte à execução das aplicações pervasivas: EXEHDA [Yamin 2004].

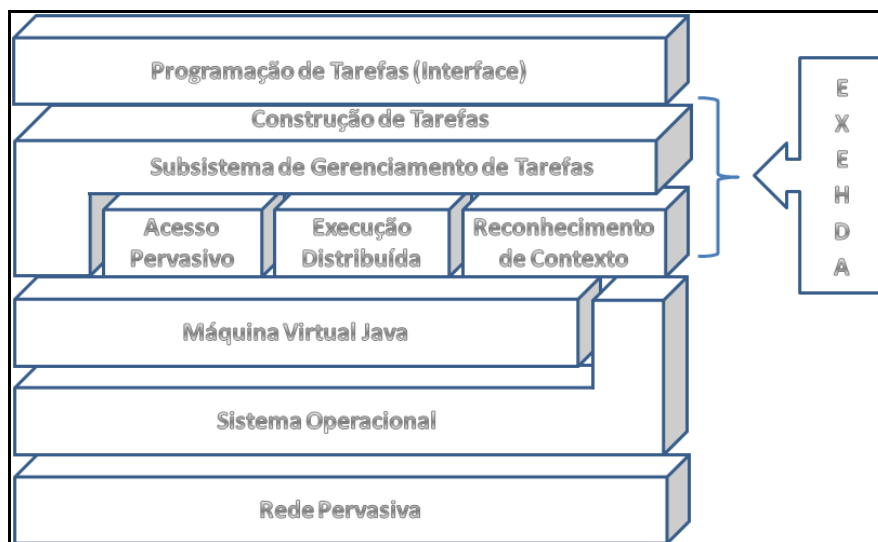


Figura 1. Arquitetura do sistema para programação e gerenciamento de tarefas

As tarefas são criadas pelos usuários-finais, a partir da interface de edição de tarefas [Lorenzi et al 2009], a qual disponibiliza as subtarefas e/ou tarefas já criadas pelos usuários. As subtarefas são implementadas diretamente como aplicações gerenciadas pelo *middleware* EXEHDA. Ambas possuem uma descrição ontológica [Librelotto et al 2008] contendo informações como: o criador, o estado atual, a descrição, os recursos, as pré-condições, a especialidade clínica e o contexto utilizado.

Devido às características flexíveis do EXEHDA quanto à integração de novos serviços, a adaptação do *middleware* para o gerenciamento de tarefas foi modelado na forma de um novo subsistema, chamado Subsistema de Gerenciamento Distribuído de Tarefas (SGDT), que funciona como um assistente que auxilia o clínico na realização e suas atividades diárias. O novo subsistema segue a mesma forma de modelagem dos outros subsistemas do EXEHDA. Portanto, também foi dividido em serviços, os quais implementam as funcionalidades necessárias ao gerenciamento das tarefas.

Os serviços do SGDT, mostrados na Figura 2, são: (i) Serviço de Gerenciamento de Tarefas, responsável pelo gerenciamento da execução das tarefas; (ii) Serviço de Acesso a Tarefas, responsável pelo acesso pervasivo ao repositório de tarefas do usuário; (iii) Serviço de Contexto de Tarefas, responsável por disponibilizar as informações de contexto relevantes para as tarefas; (iv) Serviço de Tarefas Ativas, responsável pelo acesso pervasivo às tarefas ativas do usuário.

4.1. O Subsistema de Gerenciamento Distribuído de Tarefas

Essa sessão detalha o funcionamento dos serviços do SGDT e sua integração com os outros serviços do EXEHDA (ver Figura 2).

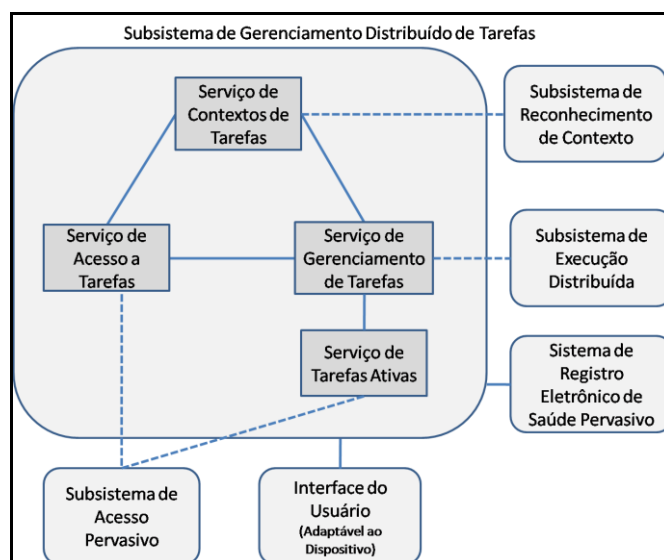


Figura 2. Arquitetura do Subsistema de Gerenciamento Distribuído de Tarefas

O Serviço de Acesso a Tarefas (SAT) tem a função de prover acesso pervasivo ao repositório de tarefas de cada usuário, bem como ao repositório de subtarefas, que é único para todo o sistema, já que estas são imutáveis. Através desse serviço, o SGDT busca a lista de tarefas disponíveis ao usuário, quando a sessão dele é iniciada. Também é com esse serviço que as tarefas (descrição ontológica personalizada) são buscadas para serem instanciadas em aplicações pervasivas e iniciarem sua execução.

O Serviço de Contexto de Tarefas (SCT) encapsula, em objetos usados pelas tarefas e subtarefas, informações de contexto obtidas pelo Subsistema de Reconhecimento de Contexto do EXEHDA. Dessa forma, o tratamento de contexto fica mais simples do ponto de vista do programador de subtarefas, e melhor compreensível do ponto de vista do usuário, que irá associar os contextos às suas tarefas.

Em uma primeira análise, observou-se que, de forma geral, o contexto de tarefas está relacionado a usuários, localização, tempo e recursos. Porém, o detalhamento do contexto depende da área de aplicação das tarefas, e para cada caso é necessário um estudo aprofundado sobre as informações necessárias à formação do contexto.

O Serviço de Gerenciamento de Tarefas (SGT) tem a função de controlar a execução das tarefas do usuário, permitindo que este possa (i) disparar a execução de determinada tarefa ou agendar a execução; (ii) interromper e, posteriormente, continuar uma tarefa que estava sendo executada; (iii) cancelar uma tarefa que não precisa mais ser continuada. Para tanto, o SGT disponibiliza uma API para que o usuário possa controlar suas tarefas. Essa API pode ser encapsulada por uma interface gráfica adequada ao tipo de dispositivo e ao tipo de atividade que se está modelando. Além disso, o SGT gerencia a migração das tarefas, para que elas possam “acompanhar” o usuário quando ele troca de dispositivo.

O Serviço de Tarefas Ativas (STA) foi criado com o propósito de manter o controle centralizado das tarefas ativas de cada usuário. Pois, se esse controle fosse distribuído (realizado por cada instância do SGT), exigiria muita comunicação e processamento para manter as tarefas sincronizadas, tornando o gerenciamento complexo e até inviável para alguns dispositivos portáteis. Além disso, é necessário um host para o qual as tarefas devem migrar se o usuário não estiver usando um dispositivo.

Conseqüentemente, o STA trabalha em uma arquitetura cliente/servidor, através de requisições HTTP. Um host (servidor) mantém as informações sobre as tarefas ativas (para cada usuário), e os clientes fazem requisições para obter, acrescentar, excluir e atualizar tarefas.

5. Avaliação

Como forma de avaliar a proposta desenvolvida neste trabalho, bem como, o impacto do gerenciamento de tarefas no *middleware* e na execução das aplicações pervasivas (tarefas), foi implementado um protótipo do Subsistema de Gerenciamento Distribuído de Tarefas, contendo o Serviço de Gerenciamento de Tarefas (SGT), o Serviço de Acesso a Tarefas (SAT) e o Serviço de Tarefas Ativas (STA). Esses serviços foram integrados ao EXEHDA e instanciados em dois nodos: um nodo base, responsável pelo gerenciamento da célula; e um nodo comum (cliente), representando um dispositivo de usuário, o qual faz parte da célula.

A aplicação do usuário, através da API do SGGT, busca as tarefas disponíveis ao usuário, bem como, as tarefas que foram iniciadas e não concluídas em outra sessão (tarefas ativas). Como essas informações estão localizadas no nodo base, as instâncias locais dos serviços SAT e STA geram requisições às respectivas instâncias remotas. Por isso, as operações desses dois serviços foram monitoradas nas simulações para determinar o impacto deles no sistema, em termos de tempo para inicialização das tarefas e número de requisições remotas. Já o gerenciamento da execução das tarefas é feito, localmente, pelo SGT. Desse modo, esse serviço foi monitorado em termos de processamento extra para controle das tarefas.

As simulações mostraram que o impacto dos novos serviços, necessários ao gerenciamento das tarefas, foi mínimo, tanto do ponto de vista do *middleware*, como do usuário que está executando as aplicações. Além disso, a avaliação dos serviços indicou pontos que podem ser melhorados no protótipo, como número de requisições remotas, que apesar de estarem dentro de valores aceitáveis, podem ser reduzidas.

Portanto, de modo geral, a avaliação das simulações comprovou que a proposta de estender um *middleware* da computação pervasiva para gerenciar as tarefas cotidianas de profissionais da saúde é promissora e praticável. Os resultados foram bastante satisfatórios, uma vez que o protótipo funcionou em conjunto com EXEHDA, desempenhando suas funções sem gerar sobrecarga ao *middleware*.

6. Cenário de Execução de uma Tarefa

Supondo que o médico irá atender a um paciente (atividade humana modelada como um conjunto de tarefas), na interface do sistema estão disponibilizadas suas tarefas. Ele dispara a execução da tarefa de atendimento a pacientes, previamente programada por ele (personalizada). O Serviço de Gerenciamento de Tarefas (SGT) acessa o Serviço de Contexto de tarefas (SCT) para obter a identificação do profissional, a configuração do dispositivo e outras informações necessárias à instanciação da tarefa. Processando a descrição ontológica da tarefa, o SGT encontra as subtarefas que a compõem e busca o código delas.

A primeira subtarefa a ser executada é a de identificação do paciente. Essa utiliza o SCT, o qual usa sensores gerenciados pelo *middleware* EXEHDA [Yamin 2004], para identificar o paciente que está sendo atendido. A informação retornada por

essa subtarefa é repassada para as outras subtarefas. Em seguida, é disparada a subtarefa de busca de informações do paciente no Sistema de Registro Eletrônico de Saúde Pervasivo (EHRp) [Vicentini *et al.* 2009] acoplado ao sistema ClinicSpaces. A informação retornada é, então, usada como parâmetro para a próxima subtarefa, que exibe os registros do prontuário no formato escolhido pelo usuário. É somente a partir desse momento que começa a interação do usuário (médico) com a tarefa, pois a execução das subtarefas anteriores é transparente.

Após o médico revisar as informações do paciente, ele encerra a aplicação (subtarefa), e o SGT dispara a próxima subtarefa, que irá armazenar as anotações do médico no prontuário (EHRp). Essa subtarefa invoca uma aplicação específica do EHRp, parametrizando-a com a identificação do profissional e do paciente. Ao término desta subtarefa, a próxima tarefa - solicitação de exames - é disparada. Aqui vale salientar que se trata de uma tarefa criada pelo usuário ou obtida do sistema. Essa tarefa é independente, podendo ser executada isoladamente.

A tarefa de solicitação de exames armazena a solicitação do médico no prontuário do paciente e encaminha-a para o laboratório, desde que o sistema do laboratório esteja integrado ao EHRp. Ao término dessa tarefa, o SGT dispara tarefa de prescrição de medicamento ou tratamento, que registra a prescrição do médico no prontuário e a envia para a impressora mais próxima, obtida através do SCT.

7. Trabalhos relacionados

O Projeto Gaia [Roman *et al.* 2002] deu origem ao *middleware* Gaia. Esse *middleware* estende as funcionalidades de um sistema operacional tradicional, provendo gerenciamento do contexto, suporte a computação móvel e gerenciamento de atuadores. Dessa forma, as aplicações podem ser construídas de forma genérica (sem suposições sobre hardware ou ambiente) e o *middleware* encarrega-se de adaptá-las de acordo com os recursos de cada ambiente.

O projeto ISAM [Augustin *et al.* 2004], assim como o Gaia, deu origem a um *middleware* de gerenciamento do ambiente pervasivo, que integra as premissas da computação em grade, da computação móvel, e da computação consciente do contexto. O *middleware* EXEHDA [Yamin 2004] cria e gerencia um ambiente virtual para o usuário, no qual as aplicações executam de forma distribuída e adaptável ao contexto. Assim, o EXEHDA permite ao usuário ter acesso ao seu ambiente computacional independentemente de localização e de tempo.

O conceito de Computação Baseada em Tarefas foi introduzido pelo Projeto Aura [Garlan *et al.* 2002] como um meio de o *middleware* gerenciar o ambiente pervasivo de forma que o usuário possa manter a continuidade de suas atividades, ao mesmo tempo em que ele desloca-se de um lugar a outro. Nesse projeto, tarefas são modeladas como coleções de serviços; a descrição do serviço é usada para encontrar os recursos necessários ou reconfigurar o *middleware* para executar a tarefa. Porém, o sistema Aura é automatizado e pró-ativo, e não permite a personalização das tarefas (serviços), o que leva ao aumento da interferência do sistema no ambiente.

O projeto *Activity-based Computing* [Bardran & Christensen 2007] apresenta uma proposta de utilização da Computação Baseada em Tarefas em ambientes da saúde. Nesse projeto, foi desenvolvido um *framework* que provê a infra-estrutura necessária para execução de serviços que dão suporte às características inerentes ao trabalho dos

profissionais da saúde. Dessa forma, os serviços podem ser inicializados, suspensos, armazenados, retomados em qualquer dispositivo e a qualquer tempo, encaminhados para outros usuários ou compartilhados entre diversos usuários. O projeto visa, ainda, permitir que **desenvolvedores** de aplicações clínicas possam incorporar, aos seus programas, suporte a mobilidade, interrupções, atividades paralelas e cooperação.

Muitas das idéias apresentadas por esses projetos influenciaram este trabalho. Destacando-se o EXEHDA, que foi utilizado como plataforma-base, ou seja, *middleware* para gerenciar o ambiente pervasivo; e o projeto *Activity-based Computing*, que norteou a definição de conceitos relativos a tarefas para a área da saúde. Porém, como se pode observar, nenhum dos projetos apresenta a visão centrada no usuário final defendida pelo Projeto ClinicSpace. Este projeto apresenta o diferencial de permitir aos clínicos a programação de forma personalizada das tarefas, e de possibilitar o balanceamento entre a execução automatizada e o controle sobre a execução das tarefas.

8. Conclusão

A primeira geração da Computação Pervasiva estudou, principalmente, *middlewares* para gerenciamento do ambiente pervasivo. O atual foco da Computação Pervasiva volta-se ao processamento de Atividades Humanas. Nessa perspectiva, este trabalho buscou adaptar um *middleware* da primeira geração para suportar a execução de tarefas cotidianas (aplicações pervasivas que auxiliam o usuário a realizar suas atividades), atendendo aos requisitos da segunda geração da computação pervasiva, da computação orientada a atividades e ao usuário-final e do ambiente clínico.

Certamente, há muito que se evoluir nessa área. E, assim como na primeira geração, a pesquisa na área de computação orientada a atividades humanas (como interfaces gráficas voltadas ao usuário-final - *End-user Programming*, contexto, inferência de atividades, etc.) pode ser auxiliada pela existência de *middlewares* que dêem suporte ao estudo e simulações das teorias propostas em torno do assunto, para serem comprovadas e consolidadas.

Portanto, a principal contribuição desse trabalho é uma arquitetura de serviços necessários para adaptar um *middleware* de gerenciamento do ambiente pervasivo ao gerenciamento de tarefas nesse ambiente, atendendo a alguns requisitos da computação orientada a atividades e ao usuário-final. Além disso, outro resultado deste trabalho é o protótipo de um assistente (implementado como um subsistema do *middleware* de gerenciamento do ambiente pervasivo) que auxilia os clínicos no gerenciamento de suas tarefas (atividades diárias). Próximos passos da pesquisa visam avaliar a usabilidade dessa solução em um ambiente real.

Referências Bibliográficas

- Augustin, Iara et al. (2004) "ISAM joining context-awareness and mobility to building pervasive applications". In: *Mobile Computing Handbook*. I. New York: CRC Press.
- Augustin, I.; Yamin, A. & Geyer, C. F. R. (2005) "Managing the follow-me semantics to build large-scale pervasive applications". *Proceedings of the 3rd international workshop on Middleware for pervasive and ad-hoc computing*.
- Augustin, I.; Lima, J. C. D. & Yamin, A. C. (2006) "Computação Pervasiva: como Programar Aplicações". In: *X SBLP, 2006, Itatiaia, RJ. Anais...*

- Aura. (2002) "Distraction-free Ubiquitous Computing". Disponível em: <<http://www.cs.cmu.edu/~aura>>. Acesso em: 17 mar. 2009.
- Bardram, J. E. (2003) "Hospitals of the Future: Ubiquitous Computing support for Medical Work in Hospitals", 5th International Conference on Ubiquitous Computing. Proceedings... Seattle, USA.
- Bardram, J. E. & Christensen, H. B. (2007) "Pervasive Computing Support for Hospitals: An overview of the Activity-Based Computing Project", IEEE Pervasive Computing, vol. 6, issue 1, p. 44-51.
- Christensen, H. B. & Bardram, J. (2002) "Supporting Human Activities - Exploring Activity-Centered Computing". Proceedings of the 4th international conference on Ubiquitous Computing, p. 107-116.
- Gaia. (2002) "Active Spaces for Ubiquitous Computing". Disponível em: <<http://gaia.cs.uiuc.edu>>. Acesso em: 17 mar. 2009.
- Garlan, D. et al. (2002) "Project Aura: toward distraction-free pervasive computing". IEEE Pervasive Computing, vol. 1, issue 2, p. 22-31.
- Isam. (2001) "Infra-estrutura de Suporte às Aplicações Móveis". Disponível em: <<http://www.inf.ufrgs.br/~isam>>. Acesso em: 14 jun. 2008.
- Kaenampornpan, M. & O'Neill, E. (2005) "Integrating History and Activity Theory in Context Aware System Design". In: 1ST International Workshop on Exploiting Context Histories in Smart Environments, 2005. Anais eletrônicos...
- Librelotto, G. R. et al. (2008) "OntoHealth - Um framework para o gerenciamento de ontologias em ambientes hospitalares pervasivos". In: II Workshop on Pervasive and Ubiquitous Computing, 2008, Campo Grande. Anais...
- Lorenzi, Fábio et al. (2009) "Inserindo o usuário na modelagem de tarefas clínicas em um ambiente da computação ubíqua". (submetido à publicação).
- Ranganathan, A. & Campbell, R. H. (2005) "Supporting Tasks in a Programmable Smart Home". In: From Smart Homes to Smart Care, vol. 15. Amsterdam: IOS Press, p. 3-10.
- Roman, M. et al. (2002) "A Middleware Infrastructure for Active Spaces". IEEE Pervasive Computing, vol. 1, issue 4, p. 74-83.
- Saha, D. & Mukherjee, A. (2003) "Pervasive Computing: a Paradigm for the 21st Century". IEEE Computer, vol. 36, issue 3, p. 25-31.
- Varshney, U. (2003) "Pervasive Healthcare", IEEE Computer, 36(12): 138-140.
- Vicentini, Caroline et al. (2009) "Proposta de Arquitetura para um Sistema de Registro de Saúde Pervasivo (EHRp)". (submetido a publicação).
- Weiser, Mark. (1999) "The Computer of the 21st Century". ACM SIGMOBILE Mobile Computing and Communications Review, Volume 3, Issue 3, p. 3-11.
- Yamin, Adenauer Corrêa. (2004) "Arquitetura para um Ambiente de Grade Computacional Direcionado às Aplicações Distribuídas, Móveis e Conscientes do Contexto da Computação Pervasiva". 194f. Tese (Doutorado em Ciência da Computação) - Universidade Federal do Rio Grande do Sul, Porto Alegre, 2004.