

Adaptando o Middleware EXEHDA para o Tratamento de Atividades Clínicas

Giuliano Ferreira, Giovani Librelotto, Fábio da Silva, Iara Augustin

Programa de Pós-Graduação em Informática
Universidade Federal de Santa Maria
{giuliano,lorenzi,librelotto,august}@inf.ufsm.br

Abstract. Currently in its second generation, the Pervasive Computing is focused in 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.

Keywords: Middleware, task-driven computing, end-user programming, clinical activities.

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 [1]. Os recursos computacionais estarão totalmente integrados ao ambiente físico, da forma mais transparente possível [2].

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 [3] [4] [5]. 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; porém, tarefas exigem um esforço maior, pois estão em um estágio mais recente, o de definição do que se deseja [6].

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 [7].

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 [6]. 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 [8] [9].

Projetos de pesquisa em Computação Pervasiva, como Aura [10] e Gaia [11], 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) [12].

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, que é o foco motivacional do projeto ClinicSpaces.

Dessa forma, o desafio deste trabalho foi adaptar um middleware de gerenciamento do ambiente pervasivo para gerenciar também as tarefas dos usuários, 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 trabalho desenvolvido; a sessão 6 apresenta os trabalhos relacionados; e finalmente, a sessão 7 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 [13].

Segundo a Teoria da Atividade Humana [14], 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 são construídas (programadas) pelo usuário de forma personalizada, ou seja, de acordo com a maneira que o médico realiza sua atividade. Desse modo, para cada atividade do seu dia-a-dia, o médico programa uma tarefa para auxiliá-lo, e esta irá executar conforme ele está acostumado a proceder nas suas atividades, diferentemente dos sistemas automatizados, nos quais o usuário tem que se adaptar à forma pré-programada no sistema.

Uma Interface de Edição de Tarefas foi desenvolvida para permitir a programação orientada ao usuário-final. Nela, uma tarefa é “programada” intuitivamente pelo médico, através do agrupamento de subtarefas, numa seqüência que reflete a forma do usuário realizar sua atividade (personalização). Além disso, a personalização também pode ocorrer em nível de subtarefa, com a configuração de parâmetros específicos que mudam a forma da subtarefa processar sua função.

Subtarefas são pequenas operações implementadas como aplicações pervasivas e dependem do tipo de atividade que se deseja modelar. Portanto, do mesmo modo como ações (na teoria da atividade) são formadas por operações, as tarefas são formadas por subtarefas. Além disso, para representar atividades mais complexas, que envolvem mais de uma tarefa, foi inserido o conceito de workflow, que é um conjunto de tarefas seqüenciais.

Para o middleware, tanto tarefas como subtarefas são aplicações pervasivas. As subtarefas são aplicações compiladas e parametrizáveis, cujas funções, muitas vezes, dependem da execução de outras subtarefas, ou seja, a saída de uma é entrada para outra. Porém, diferentemente das subtarefas, as tarefas são aplicações construídas pelo

usuário através do agrupamento de outras aplicações (subtarefas). Por isso, tarefas não são aplicações compiladas. Tarefas são descrições ontológicas que representam o agrupamento entre as subtarefas, indicando a seqüência de execução e a forma de acoplamento (como as informações são compartilhadas) entre elas.

Portanto, foi necessário adaptar o middleware, tanto para gerenciar esse tipo de aplicação (não compilada), como para disponibilizar meios do usuário ter controle sobre suas tarefas, como agendar, executar, interromper, continuar, cancelar. Além disso, foi necessário adaptar o mecanismo de migração do middleware para permitir que esse tipo de aplicação (tarefa) “siga” o usuário quando ele troca de dispositivo.

3 Middleware de Gerenciamento do Ambiente Pervasivo

O middleware EXEHDA – Execution Environment for Highly Distributed Applications [15] 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” [16]. 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; (d) acesso pervasivo aos recursos e serviços. [17]. O 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. O Subsistema de Execução Distribuída é 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. O Subsistema de Reconhecimento de Contexto e Adaptação é 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. Finalmente, o Subsistema de Comunicação disponibiliza mecanismos para atender, principalmente, a aspectos relacionados às desconexões, muito comuns em ambientes pervasivos, devido tanto à existência de enlaces sem fio como às estratégias de economia de energia dos dispositivos móveis.

Como visto, 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 [17]. 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 figura 1, 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 [17].

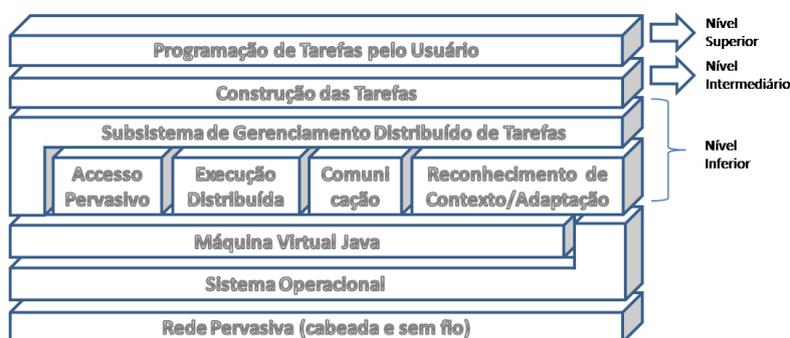


Fig. 1. Arquitetura do sistema para programação e gerenciamento de tarefas

As tarefas são criadas pelos usuários-finais, a partir da interface de edição de tarefas, 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 [18] 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 de 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 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 SGGT e sua integração com os outros serviços do EXEHDA (ver figura 2).

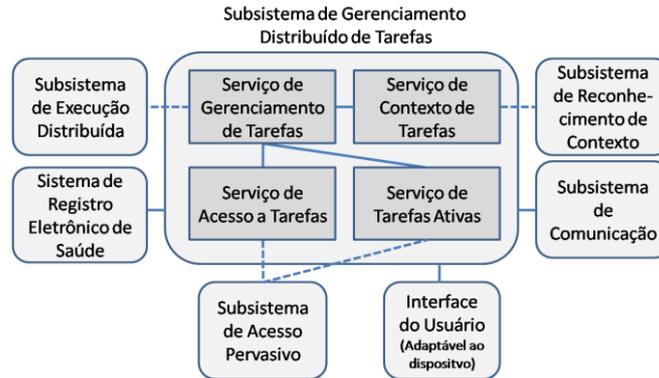


Fig. 2. Arquitetura do Subsistema de Gerenciamento Distribuído de Tarefas

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. Através desse serviço, o SGGT busca a lista de tarefas disponíveis ao usuário, quando a sessão dele é iniciada. Também é com esse serviço que a descrição ontológica [18] das tarefas é buscada para ser instanciada em aplicações pervasivas e iniciarem sua execução. A descrição ontológica [18] das tarefas é armazenada no ambiente virtual do usuário, gerenciado pelo EXEHDA.

O **Serviço de Contexto de Tarefas** 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 que ele julgar úteis 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** 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 haver um host para onde as tarefas devem migrar se o usuário não estiver usando um dispositivo.

Conseqüentemente, o Serviço de Tarefas Ativas 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. O Serviço de Tarefas Ativas foi baseado em requisições HTTP devido a já existir uma infra-estrutura no middleware para esse tipo de comunicação. Dessa forma, o Serviço de Tarefas Ativas utiliza as funcionalidades do HTTPService do EXEHDA. Um tratador (Handler) específico é usado para converter as requisições ao serviço em requisições HTTP. Portanto, o uso de comunicação HTTP é transparente do ponto de vista dos outros serviços que usam o Serviço de Tarefas Ativas, o que permite mudar a implementação sem modificar os “clientes” deste serviço.

5 Protótipo e 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 SGDT, 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, durante a execução dos testes, 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.

Como pode ser visto na figura 3, o tempo médio para inicialização e controle das tarefas estão abaixo de um milissegundo. No gráfico pode ser notado que a migração das tarefas não tem influência significativa no tempo de gerenciamento. Porém, na execução da primeira tarefa (após a inicialização do sistema), o tempo de gerenciamento é influenciado pela inicialização dos serviços do middleware. Além disso, o gráfico mostra que, mesmo no gerenciamento de tarefas concorrentes (de 5 a 10 tarefas), a sobrecarga do sistema é mínima.

A avaliação dos testes mostrou 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.

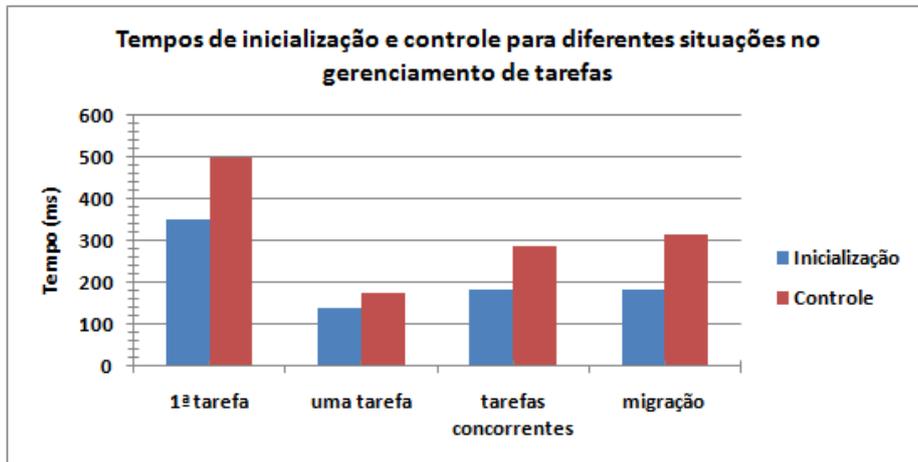


Fig. 3. Avaliação do SGGT

Portanto, de modo geral, a avaliação do protótipo 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 Trabalhos Relacionados

O Projeto Gaia [11] 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 [19], 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 [17] 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 [10] 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 [8] 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 para este trabalho, 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 ClinicSpaces. Este projeto apresenta o diferencial de permitir aos clínicos a programação personalizada das tarefas, e de possibilitar o balanceamento entre a execução automatizada e o controle sobre a execução das tarefas. O primeiro conseguido através do agrupamento de subtarefas representando os passos que o médico costuma realizar para desenvolver sua atividade. O segundo, através da disponibilização de meios para o médico agendar suas tarefas ou relacioná-las a determinados contextos (automatização), bem como, para controlar manualmente as tarefas (executar, interromper, continuar, cancelar).

7 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 pode ser auxiliada pela existência de *middlewares* que dêem suporte ao estudo e avaliação 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 cotidianas). Próximos passos da pesquisa visam avaliar a usabilidade dessa solução em um ambiente real.

Referências

1. Saha, D., Mukherjee, A.: Pervasive Computing: a Paradigm for the 21st Century. In: IEEE Computer, vol. 36, issue 3, p. 25-31 (2003)
2. Weiser, M.: The Computer of the 21st Century. In: ACM SIGMOBILE Mobile Computing and Communications Review, Volume 3, Issue 3, p. 3-11 (1999)
3. Distraction-free Ubiquitous Computing, <http://www.cs.cmu.edu/~aura>
4. Active Spaces for Ubiquitous Computing, <http://gaia.cs.uiuc.edu>
5. ISAM - Infra-estrutura de Suporte às Aplicações Móveis, <http://www.inf.ufrgs.br/~isam>
6. Augustin, I., Lima, J.C.D., Yamin, A.C.: Computação Pervasiva: como Programar Aplicações. In: X Simpósio Brasileiro de Linguagens de Programação, Itatiaia, RJ (2006)
7. Varshney, U.: Pervasive Healthcare. In: IEEE Computer, vol. 36(12), p. 138-140 (2003)
8. Bardram, J.E., Christensen, H.B.: Pervasive Computing Support for Hospitals: An overview of the Activity-Based Computing Project. In: IEEE Pervasive Computing, vol. 6, issue 1, p. 44-51 (2007)
9. Bardram, J.E.: Hospitals of the Future: Ubiquitous Computing support for Medical Work in Hospitals. In: 5th International Conference on Ubiquitous Computing. Seattle (2003)
10. Garlan, D., Siewiorek, D.P., Smailagic, A., Steenkiste, P.: Project Aura: toward distraction-free pervasive computing. In: IEEE Pervasive Computing, vol. 1, issue 2, p. 22-31 (2002)
11. Roman, M., Hess, C., Cerqueira, R., Ranganathan, A., Campbell, R.H., Nahrstedt, k.: A Middleware Infrastructure for Active Spaces. In: IEEE Pervasive Computing, vol. 1, issue 4, p. 74-83 (2002)
12. Christensen, H.B., Bardram, J.: Supporting Human Activities - Exploring Activity-Centered Computing. In: 4th International Conference on Ubiquitous Computing, p. 107-116 (2002)
13. Ranganathan, A., Campbell, R.H.: Supporting Tasks in a Programmable Smart Home. In: From Smart Homes to Smart Care, vol. 15, p. 3-10. IOS Press, Amsterdam (2005)
14. Kaenampornpan, M., O'Neill, E.: Integrating History and Activity Theory in Context Aware System Design". In: 1ST International Workshop on Exploiting Context Histories in Smart Environments. Munich, (2005)
15. EXEHDA, <http://www.inf.ufrgs.br/~exehda>
16. Augustin, I., Yamin, A.C., Geyer, C.F.R.: Managing the follow-me semantics to build large-scale pervasive applications. In: 3rd International Workshop on Middleware for Pervasive and Ad-hoc Computing. (2005)
17. Yamin, A.C.: Arquitetura para um Ambiente de Grade Computacional Direcionado às Aplicações Distribuídas, Móveis e Conscientes do Contexto da Computação Pervasiva. Tese de Doutorado em Ciência da Computação. Universidade Federal do Rio Grande do Sul, Porto Alegre (2004)
18. Librelotto, G.R., Gassen, J.B., Silveira, M.C., Freitas, L.O.: OntoHealth - Um framework para o gerenciamento de ontologias em ambientes hospitalares pervasivos. In: II Workshop on Pervasive and Ubiquitous Computing, 20th International Symposium on Computer Architecture and High Performance Computing. (2008)
19. Augustin, I., Yamin, A., Silva, L.C., Real, R., Frainer, G., Cavalheiro, G., Geyer, C.: ISAM, joining context-awareness and mobility to building pervasive applications. In: Mahgoub, I., Ilyas, M. (eds.) Mobile Computing Handbook. I. CRC Press, Florida (2004)