

# Inferência de Atividades Clínicas a partir de Propriedades do Contexto

**Marcos V. B. Souza, Iara Augustin**

Programa de Pós-Graduação em Informática (PPGI) – Universidade Federal de Santa Maria (UFSM) CEP 97.105-900 – Santa Maria – RS – Brasil

[marvin.souza@gmail.com](mailto:marvin.souza@gmail.com), [august@inf.ufsm.br](mailto:august@inf.ufsm.br)

*Abstract. In the scope of the pervasive and context-sensitive computing, is inserted the ClinicSpace project with an architecture focused to the clinical environment. Among the pervasive computing principles is the invisibility of the system, in the sense of this act without being noticed, using, for example, the task inference as way of aid in the system utilization. To perform inference of clinical tasks, in the ClinicSpace, were developed components that store the task execution history of the user together with the state of context. Therefore, by monitoring the context, the system can suggest the next task to the user to perform and improve its usability.*

**Resumo.** No âmbito da computação pervasiva e sensível ao contexto, insere-se o projeto ClinicSpace com uma arquitetura voltada ao ambiente hospitalar. Dentre os princípios da computação pervasiva está a invisibilidade do sistema, no sentido de esse agir sem ser notado pelo usuário, podendo-se utilizar a inferência de tarefas como forma de auxílio na utilização do sistema. Para a realização de inferência de tarefas clínicas, no ClinicSpace, foram desenvolvidos componentes que armazenam o histórico de execução de tarefas do usuário juntamente com estado dos elementos do contexto. Dessa forma, a partir do monitoramento do contexto, o sistema pode sugerir ao usuário a próxima tarefa a ser executada e melhorar a sua usabilidade.

## 1. Introdução

Atualmente, a utilização de serviços como e-mail, mensagens eletrônicas, etc, estão fortemente presentes no cotidiano das pessoas. Dessa forma, prover o acesso a tais serviços de forma transparente e constantemente presente torna-se imprescindível. Essas características são premissas da computação pervasiva, aonde o acesso às informações acontece de qualquer lugar (“anywhere”), a qualquer momento (“anytime”) e a partir de qualquer dispositivo (“any device”), a ponto de a computação não ser diretamente percebida [Weiser, 1991].

Em um ambiente hospitalar, os benefícios da utilização da computação pervasiva tornam-se evidentes, devido à natureza altamente dinâmica e móvel das atividades clínicas. Nota-se que dificilmente os usuários possuem um local fixo para acesso ao sistema e estão em freqüente movimentação entre os diferentes setores do hospital [Bardram, 2004].

Devido a essa grande mobilidade, é notável que o acesso às informações deve poder ser realizado de diversos locais, na maioria das vezes, distantes de computadores fixos. Através da utilização de dispositivos móveis, os clínicos podem realizar as tarefas no sistema informatizado sem que haja a necessidade da sua interrupção para o acesso, aumentando o grau de confiança das informações, uma vez que elas estão prontamente disponíveis para consultas.

A computação pervasiva, ainda, tem a possibilidade de desenvolver aplicações sensíveis ao contexto (*context-awareness*), nas quais, o comportamento do sistema computacional afeta e é afetado pelo ambiente no qual está inserido. As aplicações podem perceber dispositivos presentes, através de sensores, no ambiente e utilizá-los para a distribuição de tarefas ou para captar determinadas características de interesse. Devido a isso, torna-se evidente um alto grau de heterogeneidade, pois dispositivos diferentes podem ser utilizados para prover um único serviço [Kalapriva, 2004].

Como forma de melhorar a usabilidade do sistema a partir do seu próprio uso, pode-se recorrer à inferência das atividades que o usuário virá a utilizar. O monitoramento do ambiente no qual está inserida a aplicação pode auxiliar a inferência, fazendo com que a combinação de certos fatores habilite uma resposta do sistema.

Baseado nas tarefas executadas anteriormente, pode-se traçar o perfil individual de utilização do sistema e, a partir disso, a descoberta das tarefas futuras pode ser realizado. Unindo-se as características do contexto durante a execução das tarefas e o próprio histórico de execução, têm-se mais informações para a realização da inferência e, com isso, aumenta-se a sua confiabilidade.

A ação automática do sistema pode levar a um problema de usabilidade, uma vez que o usuário pode perder o controle sobre o sistema, o que aumenta o grau de rejeição dos sistemas informatizados. Por isso, o sistema deve realizar a inferência de tarefas, mas não pode tomar decisões no lugar do usuário, deixando sobre ele o poder de decisão sobre quais funcionalidades deseja executar. Uma das alternativas não-intrusivas para a realização da inferência é apresentar o resultado da inferência ao usuário na forma de sugestão, necessitando a sua confirmação para a realização das tarefas.

No âmbito da computação pervasiva, está presente o projeto ClinicSpace provendo uma plataforma orientada a tarefas voltada a cada usuário. O ClinicSpace possui componentes que gerenciam a execução das tarefas em um ambiente pervasivo e permitem a inferência das tarefas dos usuários baseada no seu histórico de execução. A premissa do projeto é permitir a personalização das tarefas visando diminuir a rejeição do seu público-alvo (usuários clínicos), pois torna o sistema flexível, atenuando o impacto da utilização de um sistema computacional [Silva, 2009a].

Para prover o suporte a inferência de tarefas, no desenvolvimento do presente trabalho, foram implementados componentes para integrar a arquitetura do projeto ClinicSpace. Desenvolveu-se um componente para o armazenamento do histórico de execução de tarefas de cada usuário, antes não presente na arquitetura. A partir do histórico de execução dos usuários, foi desenvolvido um componente que cria as regras de inferências e as sugere na interface gráfica do usuário.

O presente artigo está organizado da seguinte forma: Na seção 1 é abordada a arquitetura do projeto ClinicSpace, juntamente com os seus componentes já existentes e

os desenvolvidos no trabalho atual. Na seção 2 é tratada a inferência de tarefas realizada no ClinicSpace, bem como o algoritmo utilizado. A seção 3 apresenta alguns trabalhos relacionados à inferência de tarefas e a seção 4 apresenta as conclusões do trabalho realizado.

## **1. ClinicSpace**

A arquitetura ClinicSpace visa proporcionar uma plataforma na qual os usuários clínicos podem estruturar tarefas para realizar as suas atividades diárias. A partir do modelo computação orientado a tarefas, foram construídos componentes que permitem que as atividades clínicas cotidianas possam ser mapeadas em tarefas no sistema computacional.

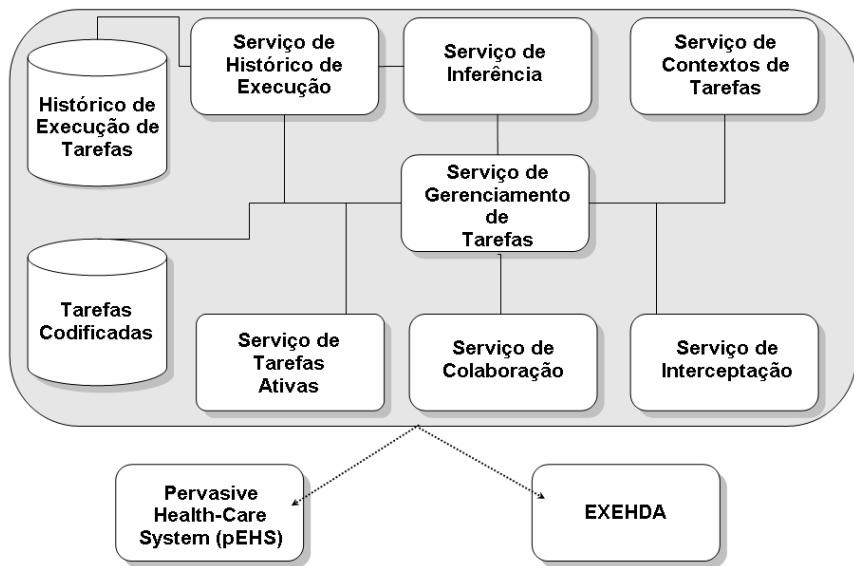
Com base em subtarefas, que são ligadas a funcionalidades do sistema EHR, o usuário, através da interface gráfica de modelagem do sistema, diagrama a melhor seqüência de passos para a realização de uma tarefa maior, possibilitando que cada usuário tenha configurações diferentes [Lorenzi, 2009b]. Além disso, estão previstos módulos no sistema que permitirão o compartilhamento de tarefas entre diferentes usuários a fim de realizar uma tarefa em conjunto.

Para o gerenciamento do contexto e execução pervasiva das aplicações, o ClinicSpace, utiliza o *middleware* EXEHDA. A partir de componentes que permitem o monitoramento de sensores dos elementos do contexto de interesse, entre outros requisitos, o EXEHDA, faz com que as aplicações possam registrar-se para receber notificações de mudança no ambiente, fazendo com que não seja necessário o desenvolvimento de tais funcionalidades. Além disso, o EXEHDA permite que as aplicações possam executar em diferentes locais, de acordo com o deslocamento do usuário, deixando-o sempre em contato com as suas tarefas em execução [Yamin, 2004].

### **1.1 Componentes da Arquitetura ClinicSpace**

Na seção atual, serão descritos os principais componentes da arquitetura ClinicSpace, bem como as suas interações com os demais sistemas externos.

Para realizar os objetivos a que se propõe, a arquitetura ClinicSpace está organizada conforme a Figura 1.



**Figura 1. Arquitetura de componentes ClinicSpace**

Na Figura 1, no centro, podemos observar o Sistema de Gerenciamento de Tarefas (SGT), que é sistema responsável por gerenciar o ciclo de vida das tarefas e subtarefas presentes no ClinicSpace. O SGT utiliza todos os demais serviços da arquitetura, de modo a acessar ou disponibilizar as funcionalidades necessárias ao sistema como um todo. De maneira a reconhecer as tarefas disponíveis ao usuário, o SGT utiliza a base de dados de Tarefas Codificadas. No momento de execução de uma tarefa, o SGT busca a tarefa da base de dados e inicia o seu processo de criação para que, finalmente, a sua execução inicie [Ferreira, 2009].

O Serviço de Contextos de Tarefas (SCT) é responsável por fazer a ligação entre as tarefas modeladas pelo usuário com o seu contexto especificado [Rizzetti, 2009]. O Serviço de Interceptação é responsável por interceptar e gerenciar a troca de mensagens entre os componentes internos à arquitetura e os componentes externos como, por exemplo, o pEHS (Sistema Pervasivo de Saúde – *Pervasive Health-Care System*).

O Serviço de Colaboração é responsável por gerenciar o processo de colaboração entre profissionais para a realização de tarefas. Esse serviço deve ser capaz de interromper a execução de uma tarefa, por exemplo, para que ela prossiga executando através de outro usuário<sup>1</sup>.

O Serviço de Tarefas Ativas tem a responsabilidade de manter o estado das atividades em corrente execução. Dessa forma, caso o usuário mude de localização, passando a utilizar outro dispositivo, o serviço repassa o estado da tarefa anterior à interrupção para o novo dispositivo [Ferreira, 2009].

O Serviço de Histórico de Execução (SHE) é responsável por armazenar as tarefas executadas pelos usuários bem como o estado do contexto no momento de sua execução. Em conjunto com o SGT, no momento em que um usuário solicita a execução

<sup>1</sup> O Serviço de Colaboração está sendo desenvolvido no trabalho de dissertação do mestrando Marcelo Lopes Kroth (PPGI-UFSM)

de uma tarefa, o SHE armazena a tarefa que está sendo executada juntamente com o estado dos elementos do contexto, permitindo que a inferência de tarefas, mais tarde, possa ser realizada a partir do seu ambiente de execução.

O Serviço de Inferência é responsável por reunir o histórico de execução de tarefas do usuário e construir uma árvore de inferência para a execução futura de tarefas. O desenvolvimento desse serviço é a principal contribuição do presente trabalho à arquitetura ClinicSpace. A partir do Serviço de Inferência, a usabilidade do sistema como um todo torna-se melhor, uma vez que, os clínicos não precisarão localizar as próximas tarefas a serem executadas.

Para fornecer as funcionalidades de gerenciamento do contexto, utilizou-se o Serviço de Gerenciamento de Contexto (*Context Manager*) do *middleware* EXEHDA. Durante a sua configuração, são especificados quais sensores do ambiente estarão ativos, bem como os valores esperados para cada sensor [Yamin, 2004]. Devido a esse serviço, a arquitetura ClinicSpace não necessita englobar funcionalidades para o gerenciamento de contexto, uma vez que elas já estão disponíveis.

## **1.2 Serviço de Histórico de Execução (SHE)**

O Serviço de Gerenciamento de Tarefas possui uma ligação com o Serviço de Histórico de Execução para que, no momento de início de uma tarefa, seja captado o seu ambiente de execução. Da mesma forma, quando a tarefa encerra, o SHE é informado para que seja atualizado o momento de término da tarefa e, a partir disso, possa ser calculado a duração da sua execução.

Para a captação dos elementos do contexto, o SHE utiliza o serviço Context Manager do EXEHDA para monitorar os seguintes sensores: dispositivo de execução, setor (localização), time frame (momento de execução), pessoas ao redor e dispositivos ao redor.

O sensor “dispositivo de execução” permite conhecer quais os dispositivos o usuário frequentemente opera para realizar as suas tarefas. Os valores esperados para tal sensor são: “PDA” (*Personal Digital Assistant*) e “Desktop”. Tais valores indicam se o usuário está executando a tarefa de modo móvel (PDA) ou a partir de um computador de mesa (*desktop*).

O sensor de setor auxilia na captação da localização física do usuário no momento da execução das tarefas. Como exemplo, pode-se citar a consulta de um paciente, na qual o clínico provavelmente estará em seu consultório realizando essa tarefa.

O sensor “*time frame*”, permite captar o momento da execução da tarefa, porém, para fins da inferência, esse momento de execução é guardado na forma de intervalos de 10min. das 24h de um dia inteiro, por se tratar do tempo médio para a execução das principais atividades clínicas [Laerum, 2004]. Para calcular o *time frame* correspondente ao momento, divide-se a quantidade de minutos passados desde a zero hora do dia pelo tamanho do *time frame*, no caso, 10min. Por exemplo, se o usuário executar uma tarefa às 0h e 8 min, o *time frame* correspondente será zero, no entanto, se a execução for às 8h e 5min, o seu *time frame* será 48. A utilização de *times frames* grandes agrupa diversas tarefas, o que aumenta o grau de confiança da inferência, porém, corre-se o risco de perder particularidades da execução de tarefas do usuário. No entanto, *times*

*frames* menores diminuem a quantidade de tarefas agrupadas e prejudicam a inferência de tarefas que tenham duração maior que esse período [Kalatzis, 2008].

O sensor de “pessoas ao redor” tem como função captar o papel ou especialidade das pessoas ao redor do usuário que está executando a tarefa. Durante a tarefa “Monitoramento de paciente”, por exemplo, o sensor pode captar a presença do paciente, enfermeiros, etc, complementando as informações do contexto relacionadas à tarefa em execução. O sensor “dispositivos ao redor” capta os equipamentos presentes ao redor do usuário que executa uma tarefa como, por exemplo, aparelhos de Raio-X, macas, aparelhos de Eletro-Cardiograma, etc.

A escolha da captação dos dados de tais sensores foi devido a eles formarem o ambiente no qual o usuário está inserido, de acordo com os 3 aspectos: “Aonde você está” (*Where are you*), “Quem está com você” (*Who you are with*) e, “Quais recursos estão próximos a você” (*What resources are nearby*) [Schilit, 1994]. Dessa forma, o instante e a forma na qual o usuário executou uma tarefa no sistema são guardados, permitindo com que o seu perfil de execução seja o mais real e preciso possível.

### 1.3 Serviço de Predição/Inferência de Tarefas (SPT)

Complementando o ClinicSpace, o Serviço de Predição/Inferência de Tarefas (SPT) visa auxiliar o usuário final na utilização do sistema como um todo, identificando a execução futura de tarefas. Dessa maneira, constrói-se uma arquitetura inteligente, uma das características da computação pervasiva [Weiser, 1991].

O SPT utiliza o histórico de execução captado pelo SHE para a construção de uma árvore de inferência das futuras tarefas do usuário e, de acordo com o estado dos elementos do contexto, é realizada a navegação da árvore em busca da tarefa futura. Para o monitoramento dos elementos do contexto, o SPT utiliza o serviço *Context Manager* do middleware EXEHDA, que informa as modificações dos estado dos sensores do contexto assim que elas ocorrem. A Figura 2 exemplifica uma árvore de decisão construída a partir do histórico de execução de um usuário.

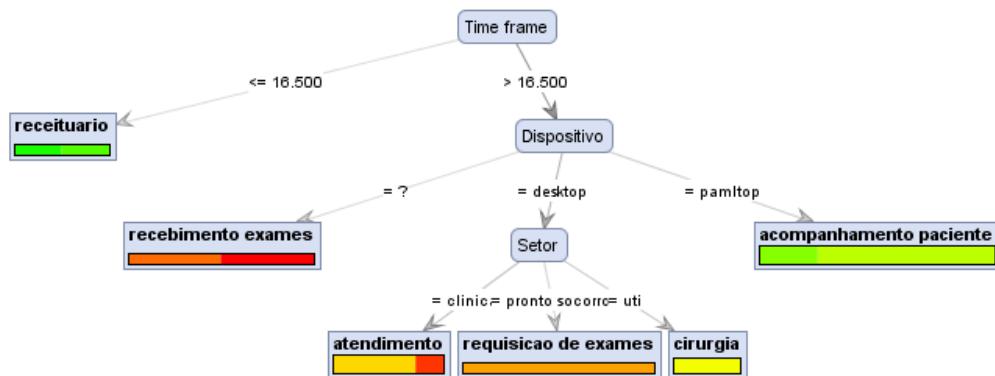


Figura 2. Árvore de decisão

Os nós da árvore representam os elementos do contexto capturados no histórico de execução e os testes a serem executados sobre o valor do elemento, enquanto as folhas representam as tarefas que podem ser inferidas. De acordo com o valor de cada elemento, a navegação inicia na raiz e prossegue em direção às folhas, até que seja

encontrada a futura tarefa do usuário. Na Figura 2, percebe-se que, caso o valor do “time frame” seja menor ou igual à “16,50”, a tarefa “Receituário” será a próxima tarefa a ser executada. Caso contrário, a navegação prossegue, e torna-se necessário verificar o valor dos demais elementos do contexto.

Para a construção da árvore de decisão, utilizou-se o *framework* RapidMiner [Mierswa, 2006]. O RapidMiner é um aplicativo *desktop* que permite a utilização de algoritmos já presentes para a mineração e extração de dados, em conjunto com uma API<sup>2</sup> que pode ser utilizada por programas de terceiros, nesse caso, utilizada pelo presente trabalho.

## 2. Inferência de Tarefas

Na seção atual serão abordados o algoritmo utilizado para a inferência de tarefas na arquitetura ClinicSpace e o fluxo de execução para a realização da inferência das tarefas.

### 2.1. Algoritmo C4.5

As árvores de decisão, utilizadas na inferência de tarefas, são construídas a partir do algoritmo C4.5, originalmente desenvolvido por Quinlan (1993). Essa escolha foi devido ao algoritmo ser um dos mais utilizados para a mineração de dados (*data mining*) [Wu, 2007]. Para a construção das árvores decisão, o SPT utilizou a API presente no RapidMiner, fazendo com houvesse somente o desenvolvimento de funcionalidades relacionadas ao foco do trabalho atual (inferência de tarefas).

Como estrutura de dados, o algoritmo C4.5 utiliza uma árvore, aonde as folhas da árvore identificam a qual classe um determinado dado pertence. A raiz da árvore e os nós intermediários representam testes a serem feitos sobre determinados atributos dos dados. Cada nó deve possuir um galho conectando a uma folha, que representa a classificação do dado ou, a uma subárvore, significando que outro atributo deve ser testado [Quinlan, 1993]. A construção da árvore envolve o uso de um conjunto de dados de testes para que, somente após essa etapa, possa haver a classificação de novos dados.

Para a construção da árvore, o conjunto de dados de teste é dividido por atributos (*time frame*, dispositivo, setor, etc), classificando-os com o uso de “*information gain*” (ganho de informação). O atributo com maior *information gain* será a raiz da árvore e possuirá um teste sobre o seu atributo, por exemplo, “*time frame*”. Os demais atributos serão as seguintes subárvore, de acordo com o seu *information gain*, em ordem decrescente. Em termos gerais, o *information gain* mede quais atributos tem maior relevância sobre o conjunto de dados, permitindo uma melhor classificação.

Após a construção da árvore, é executado o processo de “*pruning*” (poda) da árvore para retirada de nós que não contribuem significativamente na classificação dos dados. Para isso, o algoritmo re-examina a árvore construída e retira os nós que não contenham o “*information gain*” maior que um determinado valor, diminuindo a complexidade para a tomada de decisão [Quinlan, 1993]. Cabe ressaltar que, mesmo com a construção da árvore a partir dos dados de teste, pode haver um caminho que leve a decisões erradas,

---

<sup>2</sup> *Application Program Interface* (Interface de Programação de Aplicação) – Arquitetura de funcionalidades para o desenvolvimento de sistemas.

visto que são utilizados os critérios que mais influenciam na classificação e não todos os casos passíveis de acontecimento. Tais casos acontecem quando o usuário executa uma tarefa que não é muito frequente no seu cotidiano, portanto, não muito influente para a árvore de decisões.

## 2.2. Realização da Inferência de Tarefas no ClinicSpace

Para que a realização da inferência de tarefas ocorra no ClinicSpace, foi utilizado padrão de projeto *Listener*, no qual um objeto permanece em espera por um determinado evento [Gamma, 1995]. Dessa forma, mantém-se um desacoplamento entre a interface de execução do usuário e o restante da arquitetura projetada, fazendo com que mudanças na interface não afetem diretamente o modo de funcionamento do SPT. Além disso, ganha-se flexibilidade, uma vez que podem ser projetadas diferentes interfaces gráficas sem que haja a necessidade de mudanças na arquitetura do sistema.

No momento em que o usuário inicia a interface gráfica, um componente interno se registra no SPT para receber as notificações de inferência de tarefas. O SPT busca o histórico de execução de tarefas do usuário corrente e constrói a sua árvore de decisão. Após, o SPT registra-se no serviço *Context Manager* do middleware EXEHDA para receber as notificações dos elementos de contexto de interesse.

Ao receber uma notificação de mudança em algum dos elementos do contexto, o SPT navega pela árvore de execução do usuário atual, verificando se é possível inferir a tarefa futura. Caso positivo, um evento é gerado e a interface gráfica é notificada, que exibe a futura tarefa a ser executada, como sugestão. A Figura 3 exibe a interface gráfica com as tarefas disponíveis ao usuário e, ao lado, as tarefas inferidas pelo SPT. Fica evidente que as tarefas inferidas não entram em execução imediatamente, mas sim, permanecem como apoio ao usuário, não prejudicando a usabilidade do sistema.

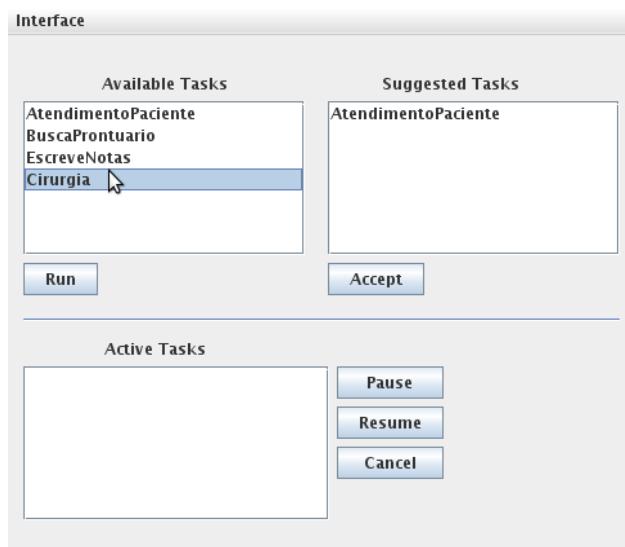


Figura 3. Interface do usuário com tarefas inferidas

### 3. Trabalhos Relacionados

Na seção atual, serão tratados os principais trabalhos relacionados a inferência de tarefas presentes.

O projeto Daidalos [Kalatzis, 2008] tem como objetivo realizar a inferência de tarefas a partir do histórico de execução do usuário em conjunto com os elementos do contexto, assim como o presente trabalho. A partir do histórico de execução são construídas tabelas de predição contendo pontuações (*scores*) que são distribuídas aos dispositivos clientes do sistema. Torna-se evidente a preocupação da arquitetura com relação à mobilidade, porém cabe aos clientes a execução da inferência das tarefas, o que pode dispositivos com recursos mais robustos.

O projeto PROACT (*Probabilistic Activity Toolkit*) [Philipose, 2003] visa identificar tarefas do dia-a-dia (*ADL – Activity of Daily Living*) de pessoas utilizando-se de dispositivos tais como etiquetas RFID. Os usuários recebem formulários no qual especificam qual tarefa estão executando, o sistema capta a seqüência de sensores ativados para a tarefa, que formam o fluxo da atividade e são armazenados. Para a inferência, o PROACT capta os dados dos sensores e, em conjunto com os passos armazenados, realiza pesquisas na internet para identificar a possível tarefa que está em realização. O objetivo do PROACT é monitorar as atividades de pacientes em internação domiciliar e, em casos de emergência, acionar os devidos procedimentos. Em comparação com o ClinicSpace, o PROACT necessita de uma conexão com a internet permanentemente para a realização da inferência de tarefas, o que pode dificultar a operação do sistema, devido a falta de conexão ou latência da rede.

### 4. Conclusões

A inferência de tarefas permite ao sistema uma melhora da sua usabilidade, uma vez que auxilia o usuário na tomada de decisões. Dessa forma, os hábitos de cada usuário servem para a inferência de futuras tarefas e diminuem a necessidade de localizar, na interface gráfica, a próxima tarefa a ser executada.

Como o histórico de execução de tarefas do usuário é usado, quanto maior for a utilização do sistema, maior será o grau de acerto da inferência de tarefas. Além disso, conforme o usuário modifica os seus hábitos, as regras de inferência também se modificam, deixando o sistema dinâmico e adaptado ao usuário final.

Como trabalho futuro, pretende-se fazer a validação do mecanismo de inferência em um ambiente real, com usuários clínicos reais, uma vez que no trabalho atual foram usados dados sintéticos. Ainda, pretende-se implementar outros algoritmos de inferência, além da árvore de decisão, para uma comparação de desempenho e taxa de acertos.

### Referências

- Bardram, J. E. (2004). “*Applications of Context-Aware Computing in Hospital Work – Examples and Design Principles*”. In *Proceedings of the 2004 ACM Symposium on Applied Computing*, Cyprus. p. 1574-1579.
- Gamma, E. et al (1995). “*Design Patterns: Elements of Reusable Object-Oriented Software*”. Addison Wesley, USA.

- Laerum, H.; Faxvagg, A (2004). “*Task-Oriented Evaluation of Electronic Medical Records Systems: Development and validation of a Questionnaire for Physicians*”. BMC Medical and Informatics Decision Making. Norway.
- Kalapriva, K., et al (2004). “*A Framework for Resource Discovery in Pervasive Computing for Mobile Aware Task Execution*”. In *Proceedings of the 1<sup>st</sup> Conference on Computing Frontiers*, USA. p 70-77.
- Kalatzis, N. et al (2008). “*User-Centric Inference Based on History of Context Data in Pervasive Environments*”. In *Proceedings of 3<sup>rd</sup> ACM International Workshop on Services Integration in Pervasive Environments*, USA. p 25-30.
- Mierswa, I. et al (2006). “*Yale (now RapidMiner): Rapid Prototyping for Complex Data Mining Tasks*”. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, USA. p. 935-940.
- Philipose, M. et al (2003). “*The Probabilistic Activity Toolkit: Towards Enabling Activity-Aware Computer Interfaces*”. Intel Research Tech Demo. USA.
- Quinlan, J. R. (1993). “*C4.5: Programs for Machine Learning*”. Morgan Kaufmann Publishers, Inc, USA.
- Raatikainen, K., Christensen, H. B., Nakajima, T. (2002). “*Application Requirements for Middleware for Mobile and Pervasive Systems*”. In *SIGMOBILE Mobile Computing and Communications Review*. vol 6, n. 4. p. 16-24.
- Schilit, B., Normal, A, Want R. (1994). “*Context-Aware Computing Applications*”. In *Proceedings of IEEE Workshop on Mobile Computing Systems an Applications*, USA.
- Silva, F. L., et al (2009a). “Ferramenta para a Programação pelo Usuário-Final de Tarefas Clínicas em um Ambiente de Saúde Ubíquo”. In XXXV Conferência Latino-Americana de Informática.
- Silva, F. L. (2009b). “*ClinicSpace: Modelagem de uma Ferramenta-Piloto para Definição de Tarefas Clínicas em um Ambiente de Computação Pervasiva Baseado em Tarefas e Direcionado ao Usuário-Final*”. Dissertação de Mestrado. Santa Maria, RS, Brasil.
- Weiser, M. (1991). “*The Computer of the 21<sup>st</sup> Century*”. In *Communications of the ACM*, n 7, vol. 36, USA, p. 94-104.
- Wu, X. et al (2007). “*Top 10 Algorithms in Data Mining*”. In *Knowledge and Information Systems*, USA. p. 1-37.
- Yamin, A. C. (2004). “Arquitetura para um Ambiente de Grade Computacional Direcionado às Aplicações Distribuídas, Móveis e Conscientes de Contexto da Computação Pervasiva. Tese de Doutorado. Porto Alegre, RS, Brasil.