

Associando Contexto às Tarefas Clínicas na Arquitetura ClinicSpace

Alencar Machado¹, Iara Augustin¹

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

alencar.ufsm@gmail.com, august@inf.ufsm.br

Abstract. *This article presents a survey on modeling clinical context, seeking to introduce it in the architecture ClinicSpace. From the identification of some basic elements of context, including the patient, device and environmental resources, was created to describe this ontological domain to be associated with clinical tasks modeled by the physician. To associate context with tasks in the ClinicSpace architecture, we implemented the interface for editing tasks and context. A case of using this interface is presented to illustrate the use of contextual elements identified, thus realizing the proof of concepts. Finally related work is presented and conclusions.*

Resumo. *Este artigo apresenta a pesquisa realizada na modelagem de contexto clínico, buscando introduzi-la na arquitetura ClinicSpace. A partir da identificação de alguns elementos básicos de contexto, entre eles paciente, dispositivo e recursos ambientais, foi criada a descrição ontológica deste domínio para ser associada às tarefas clínicas modeladas pelo médico. Para introduzir contexto associado às tarefas na arquitetura ClinicSpace, foi implementada a interface de edição de tarefas e contexto. Um caso de uso desta interface é apresentada para exemplificar a utilização dos elementos de contexto identificados, assim realizando a prova dos conceitos. Por fim, são apresentados os trabalhos relacionados e as conclusões.*

1. Introdução

Atualmente, busca-se deixar os sistemas mais orientados ao usuário-final, diminuindo a distância entre a forma como o usuário realiza suas atividades e a modelagem destas nos Sistemas de Informação em Saúde, disponibilizando ao médico formas adaptativas e personalizadas de utilização, configuração e controle do sistema, baseadas em um histórico de uso ou em um perfil.

Quando uma atividade é realizada, normalmente, leva-se em consideração o meio no qual a atividade será realizada, como pessoas e objetos (contexto). Tal característica faz com que, ao realizar uma atividade humana, uma nova avaliação de como realizar essa atividade pode ser feita. Tratando-se de um ambiente hospitalar, onde a atividade de um médico sempre interage com algo voltado à saúde humana, o dinamismo da atividade torna-se mais complexo e difícil de ser gerenciado computacionalmente.

Considerando a premissa na modelagem da Arquitetura ClinicSpace de equilibrar entre a execução pró-ativa e a execução personalizável pelo médico das atividades diárias, torna-se necessário adicionar o conhecimento do meio ao qual a atividade será realizada, permitindo, assim, que esta atividade seja realizada com uma

maior sensibilidade ao contexto. Para atender tais requisitos, o projeto ClinicSpace prototipa uma arquitetura de software que é desenvolvida sob o ponto de vista do usuário (médico), orientada às atividades clínicas, consciente do contexto, baseada em tecnologias móveis e pervasivas, e utiliza técnicas da programação do usuário-final. Identificar o contexto que afeta uma atividade clínica dentro de um hospital, bem como identificar formas de o médico expressar seu interesse em um determinado contexto, associando contexto às suas atividades diárias, são os objetivos deste trabalho. Dessa forma, as atividades do médico podem sofrer alterações a cada execução, dependendo do contexto detectado no momento.

O restante do artigo está organizado da seguinte maneira. Na seção 2 são apresentados os conceitos relativos à Computação Ubíqua/Pervasiva e Computação Ciente do Contexto. Na seção 3 são apresentadas as representações do contexto através das diferentes modelagens existentes. A seção 4 apresenta a arquitetura ClinicSpace na qual esse trabalho está inserido. A seção 5 apresenta a proposta da modelagem de contexto clínico e sua associação às atividades cotidianas do usuário médico. Na seção 6 é apresentado o estudo de caso realizado para análise dos resultados, bem como os trabalhos relacionados. Finalmente, na seção 7 são destacadas as conclusões.

2. Computação Ubíqua e Computação Ciente do Contexto

Mark Weiser (1991) visualizou que no futuro se estaria vivendo em um mundo repleto de sensores espalhados através dos mais simples objetos, como xícaras, canetas, roupas, janelas, etc. A Computação Ubíqua, proposta por ele, leva em consideração que o ambiente computacional não deve impor restrições ao usuário para utilizá-lo [WEISER, 1991]. Assume-se que (i) é necessário existir um ambiente computacional invisível e transparente, com métodos intuitivos para que o usuário possa interagir com a computação sem precisar utilizar de conhecimentos da área, como ocorre hoje; (ii) o ambiente deve identificar quem (usuário) está inserido neste, e assim propor recursos para atendê-lo de forma personalizada.

Conhecer os elementos que rodeiam o usuário da aplicação faz com que ela possa interagir e agir mais proveitosamente em prol deste. Assim, a aplicação deve ser ciente do contexto, adaptando-se automaticamente às mudanças no ambiente e às necessidades correntes do usuário, sem exigir sua atenção ou intervenção. Tais aplicações devem explorar características do ambiente como localização do usuário, pessoas próximas, hora do dia, etc... Fornecendo informações adequadas à situação ou atividade.

Como o termo contexto é muito amplo em seu significado, torna-se necessário defini-lo. Dey and Abowd (2006) definem o contexto como qualquer informação que pode ser usada para caracterizar a situação de uma entidade. Uma entidade é uma pessoa, lugar ou objeto que é considerado relevante para interação entre o usuário e a aplicação, incluindo, o próprio usuário e a aplicação. Um sistema é ciente do contexto se este usa o contexto para prover informações relevantes e/ou serviços para o usuário, onde relevante depende da tarefa do usuário [DEY and ABOWD, 2006].

A definição de Dey and Abowd é a mais utilizada na bibliografia por ser a mais abrangente e dinâmica, considerando aplicações, objetos e suas relações. Porém, a visão de Augustin (2006) para o contexto é a que mais se encaixa com esse trabalho, onde o contexto é definido como toda informação relevante para a aplicação que pode ser obtida por ela, podendo se referir a informações ambientais (recursos físicos),

funcionais (recursos lógicos) ou comportamentais (perfil do usuário).

3. Modelos de Contexto

Existe um crescente interesse no uso de sensibilidade ao contexto e técnicas de desenvolvimento de aplicações que sejam flexíveis, adaptáveis e capazes de agir automaticamente em nome do usuário (pró-atividade). A modelagem de contexto é um elemento central na construção desses sistemas. Ela define tipos, nomes, propriedades e atributos de todas as entidades que são relevantes para a aplicação, onde o grau de refinamento e sensibilidade do modelo determina a “percepção do ambiente” pela aplicação. O modelo deve somente representar as entidades e relacionamentos relevantes [HENRICKSEN and INDULSKA; 2006].

Atualmente, existem alguns modelos propostos para representação de contexto, que são:

- i. modelo de pares chave-valor - a mais simples estrutura de dados para modelagem de informações contextuais, é de fácil gerenciamento e programação, porém falta a capacidade de estruturar uma forma mais sofisticada de informação para permitir algoritmos de recuperação de contexto [Strang and C.L-Popien, 2005]. Um exemplo de uso de tal modelo são as estrutura de mapas existentes no Java;
- ii. esquema de marcação - uma estrutura de dados hierárquica, constituída por marcações com atributos e conteúdos. Um exemplo deste tipo de abordagem é o padrão CC/PP [W3C, 2007];
- iii. modelos baseados em gráficos - úteis na análise do sistema, porém normalmente não são implementados. Essa categoria de modelos inclui modelos gráficos baseados em ORM (*Object Role Modeling*), grafos contextuais e em UML (*Unified Modeling Language*). Exemplos do modelo gráfico são o CMP (*Context UML Profile*) [SIMONS, 2007], e o *Context Modelling Language* (CML)[Henricksen e Indulska, 2006];
- iv. modelos baseados em objetos - utilizam orientação a objetos para definir e estruturar as informações de contexto, tais como encapsulamento e reuso, através das especificações de herança, procurando cobrir os problemas decorrentes da dinâmica do contexto em ambientes ubíquos. Os detalhes do processamento do contexto são encapsulados em objetos, ficando ocultos para o restante dos componentes do sistema. Um exemplo dessa abordagem é a pesquisa de Henricksen (2002);
- v. modelos baseados em ontologias - representação e raciocínio lógico por múltiplos propósitos [BETTINIA et al. 2010]: para descrever um dado de contexto complexo que não pode ser representado por linguagens simples; b) para prover uma semântica formal para o dado do contexto é possível tornar-se distribuída e/ou integrando contexto entre diferentes fontes; c) para avaliar o raciocínio, através de ferramentas que checam tanto a consistência quanto o conjunto de relacionamentos descrito em um cenário, e mais importante, reconhecer que um determinado conjunto de dados básicos de contexto e suas relações na verdade, revelam a presença de uma caracterização mais abstrata de contexto.

Segundo Bettinia et al. (2010), a escolha do formalismo na modelagem de informações de contexto baseados em ontologias é tipicamente vinculada à linguagem OWL-DL ou algumas de suas variações, uma vez que ela está se tornando um padrão para vários domínios de aplicação e é suportada por um grande número de servidores de raciocínio. OWL-DL possibilita uma modelagem particular de um domínio, definindo classes, indivíduos, características de um indivíduo (propriedades *datatype*), e relacionamentos entre indivíduos (propriedades dos objetos).

Não existe um consenso para a utilização de um dos modelos acima citados que seja plenamente rico na sua utilização. Algumas pesquisas descrevem propostas de modelagem de contexto genéricas para vinculação a sistemas ubíquos. Porém, as abordagens propostas não suprem todas as necessidades existentes em um ambiente clínico.

4. Projeto ClinicSpace

Atuais Sistemas Eletrônicos de Saúde (Electronic Health System, EHS) são modelados e construídos com a visão corporativa hospitalar, a qual busca prover um gerenciamento em termos de negócios. Pesquisas demonstram [BARDRAM et al, 2007] que existe uma grande rejeição por parte dos clínicos aos EHS devido a essa estruturação de negócio vinculado aos sistemas de saúde. Esse boicote ao sistema traz inúmeros problemas à realização das atividades diárias, como por exemplo, a falta ou minimização de informações históricas de consultas dos pacientes, prejudicando diagnósticos que necessitam de maiores informações históricas para serem providos.

Considera-se que a construção de EHS com a visão centrada no usuário clínico (médico) um requisito para diminuição da rejeição atual aos sistemas de saúde. Essa visão é explorada no projeto ClinicSpace, em desenvolvimento (www.inf.ufsm.br/gmob). Devido à complexidade computacional necessária para suprir as características dinâmicas do ambiente clínico, o projeto ClinicSpace procura realizar rotinas pró-ativas como a entrada de dados via captura de contexto, minimizando a complexidade para o usuário clínico. Sendo assim, o projeto usa conceitos da Computação Ubíqua para gerenciamento e adaptabilidade dos sistemas ao meio clínico, buscando equilibrar a pró-atividade (agir em nome do usuário) com a personalização (forma individual de cada usuário realizar uma atividade), visando diminuir a rejeição encontrada nos sistemas EHS atuais.

Para permitir o suporte computacional à situação descrita e aos requisitos identificados, está-se desenvolvendo uma arquitetura de software, chamada ClinicSpace, integrando tecnologias e conceitos *da Mobile, Pervasive and Ubiquitous Computing, End-User Programming e Context-Aware Computing*. A descrição das camadas dessa arquitetura se dá, a partir da visão do usuário-final (médico), até a camada de execução e gerenciamento do ambiente pervasivo pelo *middleware EXEHDA* [YAMIN et al, 2005], que integra a arquitetura.

A arquitetura para a programação e gerenciamento personalizado das tarefas foi organizada em níveis que refletem as visões do sistema, conforme ilustra a figura 1: (i) nível superior, é composto pelo usuário-final (médico) que interage com a ferramenta para (re)definir suas tarefas que executarão num ambiente pervasivo; (ii) nível intermediário, é composto pelo mapeamento entre tarefas (definidas pelo usuário) e subtarefas (aplicações pervasivas) e pelo gerenciamento de ambas; (iii) nível inferior, é

composto pelo conjunto de serviços do *middleware* de gerenciamento do ambiente pervasivo e de suporte à execução das aplicações pervasivas: EXEHDA [FERREIRA, 2009].

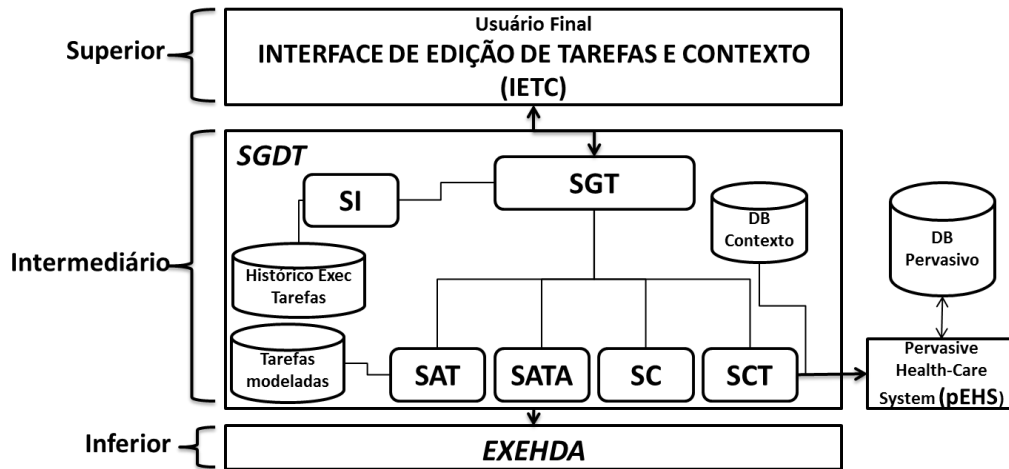


Figura 1: Arquitetura ClinicSpace

A arquitetura ClinicSpace é composto pelos seguintes componentes: (i) Interface de Edição de Tarefas e Contexto (IETC); (ii) Subsistema Gerenciador Distribuído de Tarefas (SGDT) [FERREIRA et al, 2009]; (iii) bases de suporte: banco de dados de contexto, banco de dados pervasivo de informações do paciente; (iv) pEHS – sistema pervasivo de informações em saúde [VICENTINI et al, 2010]; (v) *Middleware* EXEHDA de gerenciamento do ambiente pervasivo [YAMIN et al, 2005]. O SGDT [FERREIRA et al, 2010] faz a mediação entre o *middleware* de controle pervasivo de ambientes (EXEHDA), o Sistema Eletrônico de Saúde Pervasivo (pEHS), as bases de dados, e a Interface de Edição, provendo o acesso às informações de forma transparente, de acordo com as tarefas programadas pelo usuário clínico.

5. Contexto na arquitetura ClinicSpace

O contexto clínico concebido na arquitetura, até o momento, está vinculado aos elementos de contexto *tempo*, *localização*, *paciente*, *dispositivos*, *sensores*, *recursos* e *perfil*, que geram informações temporais, atemporais e deduzidas. Como entrada de dados leva-se em conta as informações coletadas pelos sensores, informações disponibilizadas pelos sistemas eletrônicos de saúde (EHR – *Electronic Health Record*) e informações deduzidas através da composição das duas informações conhecidas [ALENCAR and AUGUSTIN, 2010][ALENCAR et al, 2010].

Na modelagem clínica proposta, os elementos de contexto podem se relacionar uns com outros, na forma de associação composta. Por exemplo, num *paciente* são colocados *sensores* que monitoram suas funções vitais; um *médico* utiliza um *dispositivo* para a realização de alguma *tarefa* em um determinado *tempo* utilizando *recursos* (como rede e impressoras) do ambiente, as informações providas através do *dispositivo* são filtradas de acordo com seu *perfil*.

Na concepção da arquitetura ClinicSpace, uma tarefa tem um conjunto de elementos (contexto) que ela deve ser ciente e se adaptar em tempo de execução. Um médico é *dono* de um conjunto de tarefas, podendo manipulá-las conforme sua preferência. As características das informações de contexto estão vinculadas a

informações dinâmicas (sensoradas e/ou informadas), informações estáticas (históricas) e informações deduzidas (extraída da composição das duas primeiras) e são baseadas na pesquisa de Henriksen et al. (2002).

As *informações dinâmicas* variam periodicamente de acordo com a relação existente entre as entidades, pois tem um período de validade (vida). Por exemplo, um médico está na sala de pronto atendimento. Esta informação só é útil enquanto a informação “médico saiu da sala” não for verdadeira. A principal fonte de dados dinâmicos são os sensores monitorados pelo *Middleware* EXEHDA.

Já as *informações estáticas* são obtidas através de Sistemas Eletrônicos de Informações de Saúde (como o pEHS), as quais disponibilizam as informações dos elementos de contexto, como os dados pessoais de um paciente, descrição de um dispositivo entre outras. São informações estáticas, pois possivelmente não mudam de valor durante um período considerável de tempo.

As *informações deduzidas* são criadas a partir da interpretação do conjunto de informações dinâmicas e estáticas. Isso é útil para os sistemas que necessitam de informações geradas a partir de uma dedução lógica das informações, oriundas do contexto corrente (inferência). Essas informações de contexto podem ser levadas em consideração para geração de módulos pró-ativos, vindo a inferir futuros contextos. Atualmente, a arquitetura ClinicSpace está utilizando algumas informações deduzidas no Serviço de Inferência[SOUZA and AUGUSTIN, 2010].

5.1. Modelo de Contexto para Ambientes Clínicos

A modelagem gráfica do contexto proposta neste artigo foi baseada nos estudos de Simons (2007) e Henriksen and Indulska (2006), os quais propuseram artefatos que contemplam a modelagem de elementos de contexto gráfica, ontológica e orientados a objeto. Adotou-se esses trabalhos buscando uma melhor representação e processamento do contexto associado às tarefas clínicas, sempre levando em consideração a personalização do usuário.

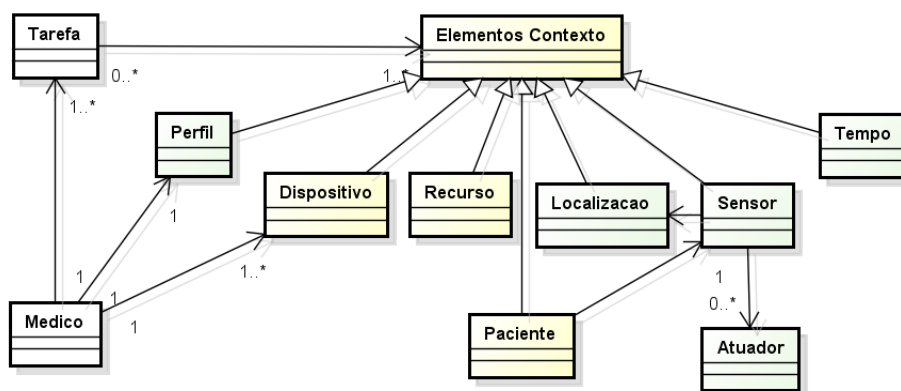


Figura 2: Modelagem do contexto clínico

A representação genérica do contexto clínico é ilustrada na figura 2, percebem-se as relações entre os elementos de contexto, onde paciente está sendo monitorado pelos sensores de sinais vitais, por exemplo. O médico tem uma ou várias atividades a realizar (Tarefa) e, para isso, utiliza algum dispositivo (móvel ou fixo) com recursos obtidos no ambiente, os quais são disponibilizados de acordo com o seu perfil. Deste modo, todos os elementos de contexto derivam de uma classe genérica, a qual encapsula

os comportamentos pertinentes a todos os elementos de contexto.

A modelagem do contexto onde a aplicação pervasiva irá executar é necessária para que a mesma conheça os elementos de contextos existentes em um espaço pervasivo que são relevantes para ela. A proposta de um modelo genérico de contexto clínico (i) é inovadora pois, até o momento, não foram encontrados trabalhos com esta abordagem (modelagem de contexto clínico), (ii) evita um grande *overhead de controlar todo o contexto de um ambiente*, pois possibilita que a aplicação conheça quais elementos de um ambiente ela deverá levar em conta, quando sua execução distribuída estiver sensorando uma atividade em um ambiente específico, assim evitando processamentos desnecessários.

5.2. Associação do Contexto às Tarefas

Na arquitetura ClinicSpace, as atividades são decompostas em tarefas e subtarefas, seguindo a forma particular de cada indivíduo realizá-la (personalização). As tarefas tem descrição ontológica e são implementadas diretamente como objetos Java, gerenciados pelo ambiente pervasivo definido pela arquitetura.

Na ontologia que descreve cada tarefa é especificado o seu contexto, o qual é responsável por manter os contextos associados às tarefas e o vínculo com seu dono (médico). Esta ontologia de tarefas e contexto foi modelada na ferramenta Protégé (<http://protege.stanford.edu>) gerando o arquivo OWL-DL, com a estrutura semântica original dos conceitos [ALENCAR et al, 2010].

A figura 3 apresenta a taxonomia ontológica que representa os conceitos Médico, Tarefa e Contexto. Este gráfico foi gerado através do plugin OntoGraf no Protégé e algumas relações entre as classes foram omitidas para melhor compreensão da ontologia.

A Tarefa é sensível a um Contexto, sendo a propriedade *eSensivel*, uma propriedade funcional (determina que somente existe *um* indivíduo Contexto vinculado ao indivíduo Tarefa), entre Tarefa e Contexto. Uma Tarefa contém a propriedade *temSubTarefas* que realiza uma relação com outros indivíduos Tarefa, criando o conceito que uma tarefa em determinado momento pode ser subtarefa de outra tarefa. E, por fim, uma Tarefa tem uma relação com Médico determinando seu dono, deste modo médico é dono (propriedade *eDono*) de uma ou várias tarefas, e uma tarefa tem um dono (propriedade inversa *temDono*).

A classe *Contexto* representa todos os elementos de contexto descritos anteriormente, cada qual com relações específicas entre eles, tais relações são: (i) propriedade *eMonitorado* que representa a relação entre Paciente, Médico e os Sensores que monitoram seus sinais vitais; (ii) propriedade *monitorado* que realizada a relação inversa com *eMonitorado*, descrevendo quais pacientes e médicos o sensor está coletando os dados; (iii) propriedade *temLocalizacao* que descreve a relação entre Localização e as classe Recurso, Paciente e Médico, fornecendo as coordenadas da localização de cada indivíduo relacionado.

A geração de indivíduos do tipo Tarefa e Contexto utilizando a propriedade *eSensivel* faz a associação destes indivíduos. Para o médico, basta ele selecionar quais indivíduos de um contexto é de seu interesse e associar a uma tarefa específica. Sendo assim, no momento da execução da tarefa pelo seu *dono*, tem-se a informação a quais

elementos de contexto ela deve ser sensível.

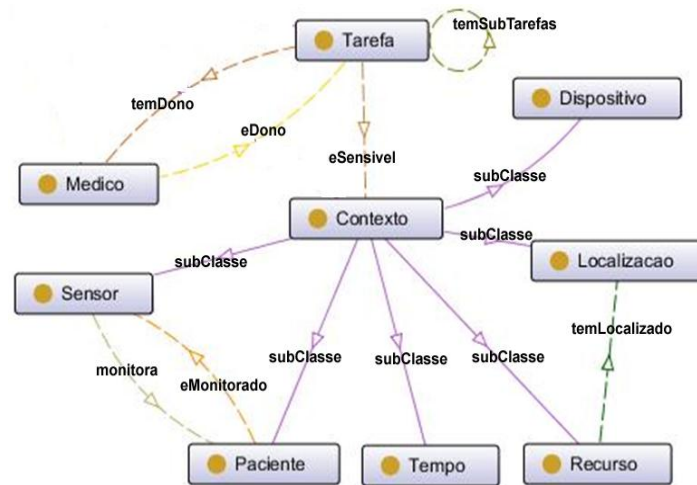


Figura 3: Representação gráfica do modelo ontológico de tarefa e contexto

Desta forma, é possível realizar o processamento dos elementos de contexto existentes no ambiente de forma robusta. A associação do contexto a uma atividade clínica pelo médico é exemplificada no estudo de caso.

6. Estudo de Caso

O estudo de caso realizado neste trabalho partiu da descrição de uma situação-problema que relata um caso de uso da IETC. Desta forma procurou-se avaliar a solução desenvolvida.

Considerou-se a seguinte situação: um médico trabalha em um hospital que tem um sistema de gerenciamento computacional atuando de forma pervasiva e ubíqua. O médico chega ao hospital e tem agendado uma rotina diária de atividades para realizar, como o atendimento a pacientes em seu consultório, visita (ronda) de rotina a seus pacientes baixados para acompanhamento. Cada atividade tem uma rotina específica para ser realizada, a qual pode ser decomposta em um fluxo de subatividades. A atividade de atendimento a pacientes pode ser decomposta em subatividades, como: (i) verificação dos sinais vitais, tais como pressão, batimentos cardíacos e temperatura; (ii) verificação do histórico de saúde, identificando alergias e enfermidades anteriores; (iii) análise de exames laboratoriais; (iv) diagnóstico; (v) prescrição de receita.

6.1. Associação do Contexto às Tarefas

Para modelar a atividade da situação-problema, o médico inicia o uso do ClinicSpace, via IETC, nela é montada a interface com base nas suas tarefas já programadas.

Como apresenta a figura 4, na IETC foram modeladas as tarefas da situação-problema contendo as cinco subtarefas descritas, nesta, para a atividade de Atendimento ao Paciente, o médico modela o fluxo de tarefas necessárias, conforme a sua forma particular de realizá-la. Para programar a funcionalidade desejada, o médico X seleciona o ícone correspondente à subtarefa desejada, arrasta-o para a área de edição, e associa-lhe as entidades de contexto, por exemplo *paciente* e *dispositivo*. Para programar os

demais procedimentos, a metodologia é a mesma. O médico informa também qual é o fluxo de execução, criando associações entre as tarefas, ou seja, ligando uma à outra.

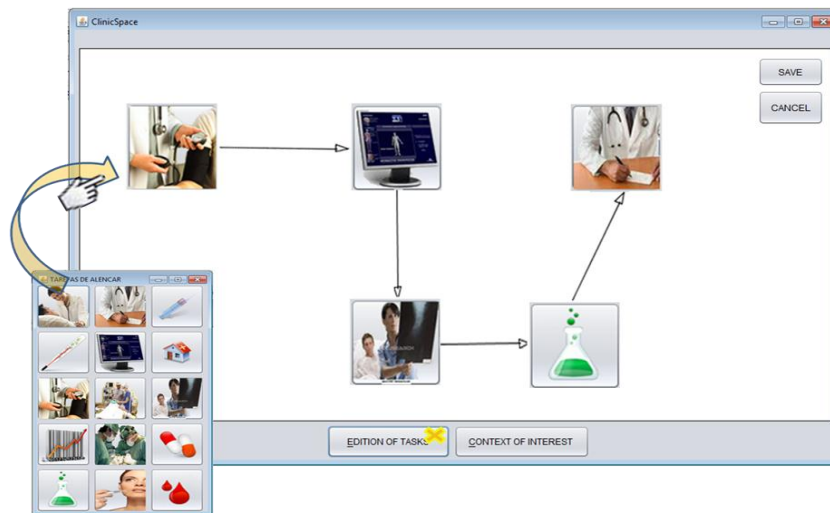


Figura 4: Modelagem da Tarefa

Para associação do contexto às tarefas, a IETC disponibiliza para o médico, a opção de associação de contexto de interesse a uma tarefa.

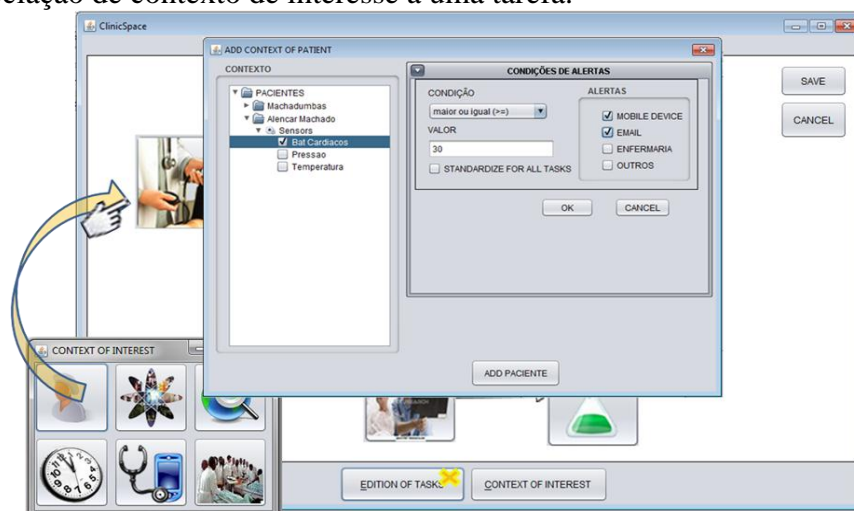


Figura 5: Modelagem do Contexto de Interesse

Na situação-problema, conforme ilustrado na Figura 5, o médico associa a entidade de contexto *paciente* à tarefa de *Atendimento ao Paciente*. A IETC, então, apresenta os pacientes vinculados ao perfil do médico. Para cada paciente é apresentado um elemento de contexto específico, modelado anteriormente, com os sensores que estão realizando o monitoramento de funções vitais.

Após salvar as alterações da ontologia, a IETC armazena a nova estrutura ontológica do contexto para posterior execução.

6.2. Execução das Tarefas Programadas (Suporte Computacional à Realização das Tarefas)

O médico interage com o Clinicspace sempre através da IETC e precisa modelar

uma única vez uma atividade específica. Após sua modelagem, esta será ativada explicitamente pelo médico ou quando um contexto associado a ela for detectado pelo sistema, pois a IETC fica sempre em execução. Quando uma tarefa é iniciada, por vontade do médico ou por gatilho de um elemento de contexto, são obtidas as informações de perfil do médico e sua localização. Na realização da tarefa *Atendimento ao Paciente*, telas específicas vinculadas com cada subtarefa são apresentadas de forma pró-ativas para simplificar a execução, exigindo menos interferência e atenção do médico em termos de controle do sistema computacional. Assim, uma definição ontológica da tarefa é carregada e as subtarefas são executadas.

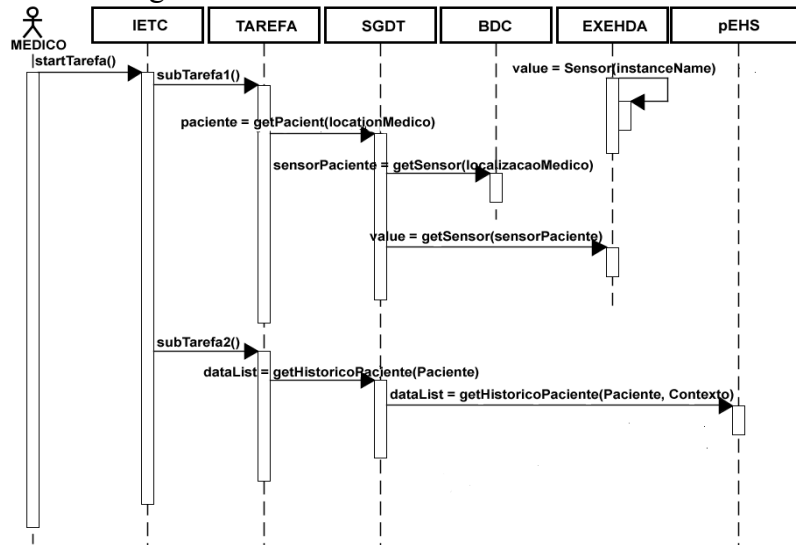


Figura 6: Diagrama de Sequência da execução da tarefa de Atendimento ao Paciente.

A Figura 6 ilustra o diagrama de sequência para a subtarefa de *verificação do histórico de saúde do paciente*. Quando esta é iniciada através da IETC, é chamada a subtarefa de *identificação do paciente (subTarefa1 da figura)*. Baseando-se na localização do médico, é apresentada uma tela de escolha de qual paciente o médico irá atender, caso mais de um de seus pacientes seja detectado no raio de ação do médico, no momento da realização da tarefa. Conhecendo qual paciente está sendo atendido, a tarefa inicia a subtarefa *busca de dados do paciente (subTarefa2 da figura)*. Está realiza uma chamada a uma aplicação específica do pEHS que retorna dados contextualizados do histórico do paciente. Por exemplo, o médico associou a essa subtarefa os elementos de contexto *especialidade* e *dispositivo*. Assim, as informações retornadas são somente as de seu interesse e cuja visualização está de acordo com a capacidade do dispositivo, como formato (imagem, texto, último dado, dados mais relevantes...), em uso pelo médico, no momento. As demais subtarefas do cenário proposto se relacionam com os mesmos subsistemas e serviços da arquitetura, que, por motivos de espaço, não serão detalhados.

7. Trabalhos Relacionados

O conceito de Computação Baseada em Tarefas foi introduzido pelo Projeto Aura [SOUSA and GARLAN, 2002]. Esse *middleware* é pró-ativo, ou seja, não há a programação de aplicações envolvidas por parte do usuário, o que leva ao aumento da interferência do sistema no ambiente. Já no projeto Gaia [ROMAN, 2002] visualiza-se

um futuro onde o espaço habitado pelas pessoas é interativo e programável. Assim, os usuários podem interagir com seus escritórios, casas e carros, para requisitar informações, beneficiarem-se dos recursos disponíveis, e configurar o comportamento de seu habitat. Na área de Saúde, o projeto *Activity-Based Computing* [BARDRAM and CHRISTENSEN, 2007] apresenta uma proposta para utilização de Computação baseada em Tarefas destinada ao Ambientes de Saúde.

Como se observa, nenhum desses projetos apresenta a visão centrada no usuário final (médico) defendida pelo projeto ClinicSpace. Este, além de permitir a modelagem das atividades diárias realizadas pelo próprio usuário e a possibilidade de associação do contexto de interesse do médico, também utiliza sensibilidade ao contexto de uma forma robusta, utilizando informações estáticas (pEHS) e dinâmicas (sensores) para melhor detectar as mudanças de contextos durante a realização de uma atividade.

8. Conclusão e Agenda de Pesquisa

A Computação Ubíqua/Pervasiva aplicada a hospitais torna esses ambientes mais inteligentes e centrados no usuário final. Para tanto, é necessário pesquisas em diferentes áreas, como modelagem de contexto, sensibilidade ao contexto e programação orientada a atividades. O projeto ClinicSpace constrói uma infra-estrutura para que o médico possa tornar o sistema o mais próximo possível da sua realidade e necessidade (personalização) e, assim, espera-se diminuir a rejeição desses profissionais em relação à utilização de sistemas computacionais para o auxílio nas suas tarefas diárias. Visando a construção de uma arquitetura para utilização em um hospital do futuro, são adicionados os conceitos de Computação Ubíqua procurando reduzir a interferência direta da computação nas atividades diárias dos médicos. Espera-se, desta forma, reduzir a rejeição dos sistemas computacionais em atividades clínicas hospitalares.

Os próximos passos do projeto prevêem a realização de testes com profissionais de saúde, em um ambiente controlado, para verificar com usuários alvo-projeto se a solução é útil e efetiva para eles bem como melhorias na interface de edição de tarefas e contexto.

9. References

- Alencar, M; Augustin, I; Ferramenta para Definição de Contexto pelo Usuário-Final na Programação de Tarefas Clínicas em um Sistema de Saúde Pervasivo. In: SBC - SBCUP, (2010).
- Alencar, M; Librelotto, G; Augustin, I. Ciência do Contexto para Tarefas Clínicas em um Sistema de Saúde Pervasivo. In: Conferência Latino-Americana de Informática, CLEI, (2010).
- Augustin, I., Yamin, A., Silva, L., Real, R., Frainer, G., Geyer, C., “ISAMadapt: Abstractions and Tools for Designing General-Purpose Pervasive Applications. Software” - Practice & Experience. (2006)
- Bardram, Jakob E. and Christensen, Henrik B.. Pervasive Computing Support for Hospitals: An overview of the Activity-Based Computing Project. IEEE Pervasive Computing, vol. 6, issue 1, p. 44-51, (2007).
- Bettinia, C., Brdiczka, O., Henriksen, K., Indulskad, J., Nicklase, D., Ranaganathan,

- A., Riboni, D.: A survey of Context Modelling and Reasoning Techniques. *Pervasive and Mobile Computing*, 6(2), 161-180 (2010)
- Dey, A., and Abowd, G., “The Context Toolkit: Aiding the Development of Context-Aware Applications”, In *Proceedings of Human Factors in Computing Systems: CHI 99*, Pittsburgh, PA: ACM Press, pp.434-441. (2006)
- Ferreira G, Augustin I, Librelotto G, Silva L, Alencar M, Yamin, A; *Extending a Middleware for Pervasive Computing to Programmable Task Management in an Environment of Personalized Clinical*; In: *UBICOMM*, (2010).
- Ferreira, G., Librelotto, G., Silva, L., Yamin, A., “Middleware for management of end-user programming of clinical activities in a pervasive environment” In *Workshop on Middleware for Ubiquitous and Pervasive Systems*, Vol.389, pp.07–12. (2009)
- Henricksen, K., Indulska, J.: Developing context-aware pervasive computing applications: Models and approach. *Journal of Pervasive and Mobile Computing* (2006)
- Henricksen, K., Indulska, J., and Rakotonirainy, A., “Modeling context information in pervasive computing systems” In: *Proceedings of 1st International Conference on Pervasive Computing*, Springer, pp.167-180. (2002)
- Roman M, Hess C., Cerqueira R., Campbell and Nahrstedt K. Gaia: a Middleware Infrastructure to Enable Active Spaces. In *IEEE Pervasive Computing*. 74-83. (2003)
- Simons C. CMP: A UML Context Modeling Profile for Mobile Distributed Systems. In *Proceedings of the 40th Hawaii International Conference on System Sciences (HICSS'07)*, page 289b, (2007).
- Souza, M, V, B., Augustin, I. Inferência de Atividades Clínicas na Arquitetura ClinicSpace a partir de Propriedades do Contexto. In: *SBC - WIM*, (2010).
- Sousa, J. and Garlan, D. Aura: An architectural framework for user mobility in ubiquitous computing environments, *The Working IEEE/IFIP Conference on Software Architecture (WICSA)* (2002).
- Strang, T., and Popien, C., “A context modeling survey” In: *Workshop on Advanced Context Modeling, Reasoning and Management as Part of UbiComp*, (2005)
- Vicentini C, F., Alencar M., Ferreira G., Silva, L., Augustin, I. PEHS – Arquitetura de um Sistema de Informação Pervasivo para Auxílio às Atividades Clínicas. *Revista Brasileira de Computação Aplicada RBCA*, Vol 2 nº 2 (2010)
- Yamin, A., and Augustin, I., “EXEHDA: adaptive middleware for building a pervasive grid environment” *Frontiers in Artificial Intelligence and Applications - Self-Organization and Autonomic Informatics (I)*, volume 135. s.l. : IOS Press. (2005)
- Weiser, Mark. “The Computer of the 21st Century. *Scientific American*” vol.265, (1991)
- W3C Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies 2.0 W3C Working Draft 30 April 2007; Acesso em 18/01/2011: <http://www.w3.org/TR/2007/WD-CCPP-struct-vocab2-20070430/>