

Capítulo

1

Modelos e Processos para o Desenvolvimento de Sistemas Sensíveis ao Contexto

Vaninha Vieira, Patricia Tedesco e Ana Carolina Salgado

Abstract

The massive amount of available information, together with users' needs to accomplish increasingly complex tasks in short periods of time, present computer systems with a new challenge: how to lessen users' need to explicitly interact with the system to obtain what they want? Context is the knowledge that allows identifying what is or is not relevant in a given situation. Context-sensitive systems (CSS) are those that use context to provide more relevant information and to adapt services to users' needs. Developing CSS entails implementing additional and complex tasks related to context specification, management and usage. This text presents an overview of the concept of context, discusses models for context representation and management and processes to aid CSS developers. An example of CSS design is detailed to show the applicability of the presented concepts. Challenges and open research avenues are also indicated.

Resumo

A necessidade dos usuários de realizar tarefas complexas e de processar crescente volume de informação em pouco tempo apresenta um novo desafio aos sistemas computacionais: como diminuir a necessidade da interação explícita do usuário com o sistema para obter o que deseja? Contexto é o conhecimento que permite definir o que é ou não relevante em uma dada situação. Sistemas sensíveis ao contexto (CSS) são aqueles que usam o contexto para filtrar e disseminar informações mais úteis e adaptar seus serviços às necessidades dos usuários. Ao desenvolver CSS deve-se considerar tarefas não triviais relacionadas à especificação, gerenciamento e uso do contexto. Este texto apresenta uma visão geral do conceito de contexto, discute modelos para representação e gerenciamento de contexto e processos de apoio a desenvolvedores de CSS. Um exemplo de projeto de um CSS é detalhado para mostrar a aplicabilidade dos conceitos apresentados. Desafios e direções de pesquisa são, também, indicados.

1 Introdução

O desenvolvimento de aplicações computacionais cada vez mais complexas demanda a criação de mecanismos que aprimorem seus serviços, enriqueçam sua semântica e provejam funcionalidades que aproximem mais o homem da máquina, de maneira transparente. Atualmente, os usuários ainda precisam se esforçar muito informando suas preferências de uso, indicando o que precisam, configurando parâmetros sobre como um serviço deve ser apresentado, respondendo perguntas, enfim, transmitindo idéias acerca da atividade que estão realizando ou que pretendem realizar. O ideal seria que os sistemas computacionais percebessem a intenção do usuário e, com isso, minimizassem a complexidade e esforço destes na execução das suas tarefas.

Projetistas de sistemas computacionais têm percebido uma demanda na necessidade dos usuários de que os sistemas sejam mais flexíveis, adaptáveis, interativos e fáceis de usar. Sistemas que exijam demasiada intervenção do usuário para identificar informações e serviços relevantes tendem a perder espaço para aqueles capazes de oferecer serviços mais adaptáveis às necessidades e intenções dos usuários. Com isso, os desenvolvedores têm-se confrontado com dois novos desafios: (1) como tornar os sistemas mais adaptáveis aos usuários oferecendo-lhes as informações e serviços relevantes às suas necessidades no tempo desejado? e (2) como diminuir a necessidade do usuário interagir explicitamente com o sistema para obter o que deseja?

Contexto pode ser visto como um conjunto de condições e influências relevantes que tornam uma situação única e compreensível [Brézillon 1999]. Essa situação pode referir-se a uma pessoa, um grupo de pessoas, um objeto físico, uma entidade computacional, entre outros. Em outras palavras, contexto é o conhecimento que está por trás da habilidade de discriminar o que é ou não importante em um dado momento, apoiando indivíduos a melhorar a qualidade da conversação e a compreender certas situações, ações ou eventos. O termo *elemento contextual* (CE, do inglês *Contextual Element*) refere-se a dados, informações ou conhecimento, que podem ser utilizados para definir um contexto [Vieira et al. 2007].

Em sistemas computacionais, contexto é um instrumento de apoio à comunicação entre os sistemas e seus usuários. A partir da compreensão do contexto, o sistema pode, em circunstâncias diversas, mudar sua seqüência de ações, o estilo das interações e o tipo de informação fornecida aos usuários de modo a adaptar-se às necessidades atuais destes. Sistemas que utilizam contexto para direcionar ações e comportamentos recebem o nome de *Sistemas Sensíveis ao Contexto* (CSS, do inglês *Context-Sensitive Systems*).

As áreas da Computação Ubíqua e Inteligência Artificial foram as pioneiras nos estudos e utilização do conceito de contexto (e.g. [Brézillon 1999; McCarthy e Buvac 1997; Schilit et al. 1994]) e, com isso, foram as que demonstraram o potencial da aplicação desse conceito nos sistemas computacionais. Pesquisas recentes vêm utilizando o conceito de contexto para beneficiar sistemas ligados a outras áreas como, por exemplo: *Sistemas Colaborativos*, visando melhorar os serviços de percepção e prover assistência aos grupos; *Hipermídia Adaptativa*, possibilitando a personalização e adaptação do conteúdo de sites web; *Integração de Dados*, facilitando a resolução de conflitos semânticos; e *Interface Humano-Computador*, com o uso do contexto para

adaptar as interfaces dos sistemas tornando mais intuitiva a sua interação com os usuários.

Atualmente, vem crescendo o interesse no desenvolvimento de aplicações sensíveis ao contexto. No entanto, compreender e identificar o contexto, e executar ações automaticamente de acordo com esse contexto é uma tarefa não trivial em sistemas computacionais. É preciso definir o que exatamente considerar como contexto, em que cenários este se aplica e que informações são necessárias para descrevê-lo. É preciso viabilizar formas de adquirir o contexto o mais automaticamente possível, sem que o usuário tenha que ser questionado insistentemente. Mecanismos de raciocínio e inferência devem ser implementados, pois nem todo tipo de informação contextual pode ser obtida, automaticamente, via monitoramento. Por exemplo, uma informação contextual relacionada a um sistema de guia turístico indica se o clima está quente ou frio no local onde o turista pretende ir. No entanto, os conceitos *frio* e *quente* não podem ser inferidos apenas mediante a temperatura. Outros fatores, como a localidade ou a estação do ano, devem ser levados em consideração para derivar essa informação.

Para tratar a inferência é necessário considerar questões associadas ao tratamento da incerteza [Ranganathan et al. 2004] e da qualidade do contexto processado [Zimmer 2006]. O monitoramento dos usuários traz a necessidade de considerar questões relativas à sua segurança e privacidade [Hong e Landay 2004]. Pesquisadores de diversas áreas da Computação estudam as melhores técnicas e métodos para suplantam os desafios inerentes ao desenvolvimento de sistemas sensíveis ao contexto, porém ainda não há consenso sobre este assunto. O presente texto tem por objetivo discutir questões relacionadas ao uso de contexto em sistemas computacionais e como CSS podem ser projetados e desenvolvidos.

O texto está estruturado como segue. A Seção 2 apresenta os conceitos básicos sobre contexto e sistemas sensíveis ao contexto, indica alguns exemplos de aplicações e discute como diferentes áreas da Computação interpretam e utilizam esse conceito. A Seção 3 discute os requisitos necessários para o desenvolvimento de CSS. Um desses requisitos está relacionado à modelagem das informações contextuais, e é discutido em maiores detalhes na Seção 4. Essa seção compara diferentes técnicas de representação de contexto e descreve um metamodelo de contexto. Visando formalizar as atividades relacionadas ao desenvolvimento de CSS, a Seção 5 discute alguns processos de software propostos na literatura. O uso de um desses processos na construção de um CSS específico é descrito na Seção 6. Finalmente, a Seção 7 encerra o documento com considerações finais e indicações de desafios e oportunidades de pesquisa na área.

2 O Conceito de Contexto

As atividades cotidianas dos seres humanos demandam algum tipo de comunicação. Quando duas pessoas se comunicam, naturalmente, observam o contexto sobre o qual a comunicação ocorre. Para isso, elas levam em consideração a riqueza da linguagem que compartilham, o entendimento comum de como o mundo funciona e o entendimento implícito das situações do dia-a-dia. Embora seja um conceito presente e conhecido nas interações entre pessoas, contexto ainda é um tópico pouco explorado nos sistemas computacionais. Esta seção discute esse conceito, apresentando definições para o termo e como este é tratado em diferentes áreas da Computação.

2.1 Definições

A definição do que é contexto e o que ele engloba, no caso particular de sistemas computacionais, ainda não é um consenso entre os pesquisadores. Essa seção discute algumas propostas relativas aos conceitos de contexto (Seção 2.1.1) e sistemas sensíveis ao contexto (Seção 2.1.2), enfatizando a nossa definição para ambos.

2.1.1 Contexto

Existem várias definições propostas para o conceito de contexto, algumas complementares, outras limitadas à área da Computação que buscam apoiar. Em [Bazire e Brézillon 2005], os autores coletaram um conjunto de, aproximadamente, 150 definições vindas de diferentes domínios, e chegaram a duas conclusões principais: (1) o contexto atua como um conjunto de restrições que influenciam o comportamento de um sistema, embutido em uma dada tarefa; e (2) a definição de contexto depende da área de conhecimento à qual pertence.

Uma definição clássica e bastante referenciada é proposta em [Dey e Abowd 2001] em que contexto é *qualquer informação que caracteriza a situação de uma entidade, sendo que uma entidade pode ser uma pessoa, um lugar ou um objeto considerados relevantes para a interação entre um usuário e uma aplicação, incluindo o próprio usuário e a aplicação. O contexto é tipicamente a localização, a identidade e o estado das pessoas, grupos ou objetos físicos e computacionais*. Zimmermann e colegas [Zimmermann et al. 2007], em busca de uma definição operacional para contexto, estendem a definição de Dey e Abowd por meio da separação dos elementos que caracterizam a situação de uma entidade em cinco categorias: *individualidade* (propriedades e atributos que definem a entidade em si); *atividade* (todas as tarefas que a entidade pode estar envolvida); *localização e tempo* (coordenadas espaço-temporais da entidade); e *relações* (informações sobre qualquer relação que a entidade possa estabelecer com outras entidades).

Na área da Inteligência Artificial, Brézillon define contexto como “o que restringe a solução de um problema sem, entretanto, interferir nele explicitamente” [Brézillon 1999]. Gross e Prinz, na área de Sistemas Colaborativos, definem contexto como “as condições (i.e., circunstâncias como tempo e localização) inter-relacionadas (i.e., algum tipo de continuidade no sentido mais amplo) na qual alguma coisa (e.g. um usuário, um grupo ou um artefato) existe (e.g. presença) ou ocorre (e.g. uma ação executada por um ator)” [Gross e Prinz 2003]. Já Alarcón e colegas argumentam que a pesquisa sobre contexto concorda em dois aspectos: primeiro, o contexto é relativo a tudo que cerca alguma coisa, em que essa “coisa” é uma situação, uma atividade, uma idéia, porém não a “coisa” em si; segundo, o contexto compreende um conjunto de elementos interrelacionados onde cada relacionamento traz um significado específico para a “coisa” [Alarcón et al. 2005].

Em [McCarthy e Buvac 1997] é apresentada uma tentativa de formalização lógica do conceito de contexto e os autores elaboram as seguintes observações: (i) um contexto é sempre relativo a um outro contexto; (ii) o contexto possui uma dimensão infinita; (iii) o contexto não pode ser descrito completamente; e (iv) quando diversos contextos ocorrem em uma discussão, existe um contexto comum, acima de todos os contextos existentes, para os quais todos os termos e predicados podem ser mapeados. Kokinov desenvolveu uma teoria dinâmica do contexto [Kokinov 1999] que define

contexto como o conjunto de todas as entidades que influenciam o comportamento humano (ou de um sistema) em uma ocasião específica. Essa teoria é centrada em quatro princípios: (1) contexto é um estado mental; (2) contexto não possui limites claros; (3) contexto consiste de todos os elementos associativamente relevantes; e (4) contexto é dinâmico.

Embora existam diversas definições para contexto, pode-se verificar que: um contexto existe apenas quando associado a alguma outra entidade (e.g. tarefa, agente ou interação); contexto é um conjunto de itens (e.g. conceitos, regras e proposições) associado a uma entidade; e um item é considerado como parte de um contexto apenas se esse item for útil para dar suporte à resolução de um determinado problema. Por exemplo, a proposição “está chovendo” é considerada como parte do contexto em um *sistema de apoio ao tráfego viário*, uma vez que a chuva tem influência direta na visibilidade, velocidade e, conseqüentemente, no tráfego dos veículos. No entanto, essa mesma proposição não deve ser considerada como parte do contexto em um *sistema de visita guiada de um museu*.

Na nossa visão sobre o que é contexto, nós fazemos uma distinção clara entre dois conceitos: *contexto* e *elemento contextual*, como descrito a seguir.

Um elemento contextual (CE, do inglês Contextual Element) é qualquer dado, informação ou conhecimento que permite caracterizar uma entidade em um domínio.

O contexto da interação entre um agente e uma aplicação, para executar alguma tarefa, é o conjunto de elementos contextuais instanciados que são necessários para apoiar a tarefa atual.

Por essa definição, pode-se verificar que consideramos o contexto como aplicado à interação entre um agente e uma aplicação. Um *agente* pode ser um agente humano ou um agente de software. Além disso, os elementos que compõem o contexto devem possuir um relacionamento de relevância com a tarefa que o agente está executando. Um CE é um tipo de informação que pode ser conhecida, codificada e representada antecipadamente; ela é estável e pode ser definida em tempo de projeto. O contexto é dinâmico, depende da tarefa atual do agente e deve ser construído em tempo de execução, quando uma interação ocorre.

2.1.2 Sistemas Sensíveis ao Contexto

Um sistema é considerado sensível ao contexto se ele utiliza contexto para fornecer informações ou serviços relevantes para o usuário, sendo que a relevância depende da tarefa do usuário [Dey e Abowd 2001]. O termo, em inglês, *context-aware system*, é o mais utilizado na literatura para referenciar esses sistemas. Outros termos considerados sinônimos são *sistemas baseados em contexto*, *aplicações adaptativas* e *aplicações dirigidas à resposta*. Neste texto, utilizaremos o termo *sistema sensível ao contexto* (*context-sensitive system*). Essa escolha deve-se ao fato de considerarmos que esse termo traduz melhor a semântica de um sistema que *percebe* mudanças no contexto, *adapta-se* e *reage* a essas mudanças. Dessa maneira, nós utilizaremos a definição:

Sistemas Sensíveis ao Contexto (CSS, do inglês Context-Sensitive Systems) são aqueles que gerenciam CEs relacionados a uma aplicação em um domínio e usam esses elementos para apoiar um

agente na execução de alguma tarefa. Esse apoio pode ser alcançado pelo aumento da percepção do agente em relação à tarefa sendo executada ou pelo provimento de adaptações que facilitem a execução da tarefa.

De forma similar à nossa definição de contexto, consideramos o uso do contexto em CSS com a finalidade de apoiar um agente na execução de alguma tarefa. O elemento básico manipulado pelo CSS são elementos contextuais. O apoio fornecido pelo CSS pode estar associado tanto a possibilitar adaptações no comportamento do sistema de acordo com mudanças no contexto, quanto a ampliar a percepção do agente com informações relevantes relacionadas à tarefa em execução.

2.2 Aplicações Tradicionais x Aplicações Sensíveis ao Contexto

O que chamamos neste texto de “aplicações tradicionais”, como ilustrado na Figura 1a, são sistemas computacionais que agem levando em consideração apenas as solicitações e informações fornecidas explicitamente pelos usuários. As aplicações sensíveis ao contexto (Figura 1b) consideram as informações explícitas fornecidas pelos usuários, aquelas armazenadas em bases de conhecimento contextuais, as inferidas por meio de raciocínio e, ainda, aquelas percebidas a partir do monitoramento do ambiente. Essas informações obtidas de forma não-explicita é o que chamamos de informações contextuais.

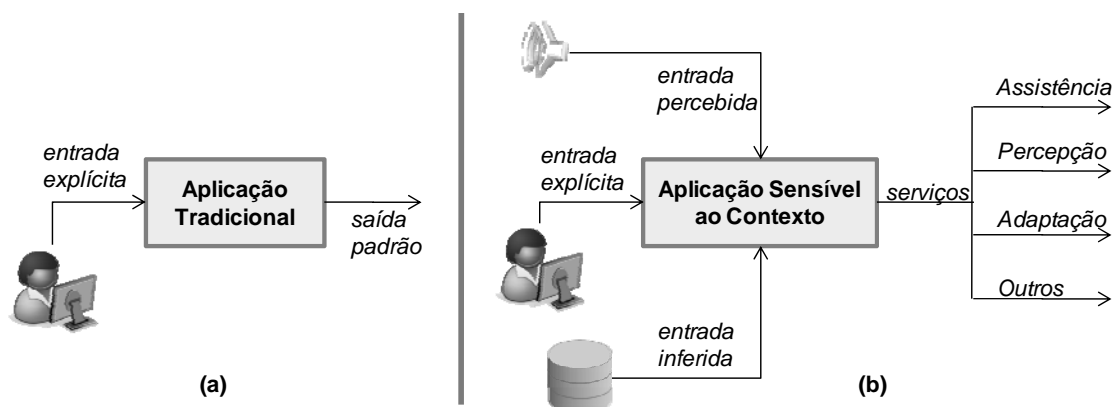


Figura 1. Visão geral de uma aplicação tradicional (a) e de uma aplicação sensível ao contexto (b)

Com base nas informações contextuais, a aplicação pode enriquecer semanticamente a solicitação explícita do usuário e, com isso, executar serviços mais próximos às suas necessidades. Dentre estes serviços estão: (1) *assistência* na execução da tarefa sendo realizada como, por exemplo, alertar o usuário sobre ações que ele deve executar para alcançar seus objetivos, ou recomendar recursos existentes relacionados à tarefa; (2) *percepção* do contexto, que se refere a notificar o usuário sobre o contexto associado a pessoas e interações do seu interesse, relativos à tarefa em execução, apoiando-o a coordenar suas próprias ações; (3) *adaptação*, ou variação do comportamento do sistema, respondendo de forma oportuna às mudanças ocorridas no ambiente e às ações e definições dos usuários (e.g. personalização de interfaces e conteúdo); e (4) *outros serviços*, como o uso do contexto para enriquecer semanticamente o conhecimento gerenciado pela aplicação.

2.3 Exemplos de Aplicações Sensíveis ao Contexto

Alguns exemplos de CSS citados na literatura, pertencentes a diferentes domínios de utilização, são discutidos a seguir.

2.3.1 *Dynamic Tour Guide*

Turistas, muitas vezes, procuram serviços que facilitem seus passeios. Quando conseguem informações antecipadas sobre horários especiais ou restaurantes lotados, eles podem planejar novas atividades. Pensando em atender a essa demanda, foi criado o *Dynamic Tour Guide* (DTG), uma aplicação composta por agentes móveis cujo objetivo é prover rotas personalizadas, um guia navegacional e um conjunto de informações ambientais ao usuário-turista, considerando diferentes tipos de turista ou necessidades de passeio [Kramer et al. 2005]. A indicação personalizada de roteiros e pontos turísticos é realizada por meio da análise da informação contextual disponível para uma dada localização, em um determinado momento. Assim, informações como o *período de tempo* disponível para o passeio, a *localização* corrente (e.g. pontos turísticos próximos), a *data* (e.g. dia da semana ou estação do ano), os interesses pessoais do usuário e a velocidade possível para a caminhada, são levadas em consideração para que o sistema reaja de forma compatível com o contexto. Quando um turista começa a caminhar, o DTG verifica sua velocidade e as condições das ruas por onde ele deve passar. O turista será guiado até o próximo ponto turístico que tenha selecionado e receberá informações sobre ele.

2.3.2 *CO2DE e POLITeam*

Collaborate to Design (CO2DE) é um editor síncrono/assíncrono de diagramas UML, originalmente projetado para permitir que participantes compartilhem o mesmo espaço de edição e controlem versões dos documentos sendo construídos [Borges et al. 2004]. Como ferramenta colaborativa, ele provê funcionalidades para ajudar membros individuais a estar “cientes” das informações contextuais relacionadas à tarefa corrente. Por exemplo: informação sobre a composição do grupo, o papel de cada membro do grupo, cada novo documento sendo simultaneamente discutido, entre outras.

Visando resolver os problemas de sobrecarga inerentes aos mecanismos de percepção em sistemas colaborativos, o sistema *POLITeam* [Mark et al. 1997] utiliza o contexto de trabalho do usuário (e.g. *trabalhando em um documento*, *acessando uma pasta*, *executando um passo em um processo*) para permitir que usuários especifiquem perfis de percepção. Esses perfis serão usados para filtragem seletiva das informações de percepção. Para isso, determinados artefatos compartilhados (e.g. um documento) são associados a um conjunto de contextos. Cada contexto indica ao sistema o momento e a forma como ele deverá prover a informação de percepção. O usuário pode configurar preferências de percepção em seu perfil que será anexado a um ou mais artefatos.

2.3.3 *GeoDialogue/DAVE_G*

Sistemas de informação geográfica carecem de uma interface mais próxima do mundo real do usuário e, com isso, a utilização de linguagem natural vem sendo incentivada como um meio de facilitar a interação. Contudo, a comunicação humana é inerentemente contextualizada pelo conhecimento existente entre os participantes. Esse

conhecimento contextual não é facilmente adquirido pelos sistemas computacionais e a linguagem se torna ambígua, incompleta ou vaga [Cai et al. 2003]. DAVE-G (*Dialogue-Assisted Visual Environment for Geoinformation*) é um ambiente genérico no qual o usuário pode interagir com sistemas de informação geográfica por meio de uma interface multimodal. Integrado a este ambiente encontra-se o *GeoDialogue* – um agente que implementa uma abordagem colaborativa de diálogos que considera fatores contextuais para a interpretação de uma solicitação, como: o espaço geográfico, o contexto pessoal do usuário, sua trajetória de interação e o contexto da tarefa.

2.3.4 Outros exemplos

O contexto vem sendo largamente utilizado em aplicações de hipermídia adaptativa para promover personalizações dos sites, de acordo com o contexto do usuário, e adaptações do conteúdo das páginas. Por exemplo, o GMail¹, sistema de gerenciamento de emails da Google, utiliza recomendação de links e propaganda adaptada ao conteúdo do email sendo lido pelo usuário. Para isso, ele extrai e utiliza informações contextuais, sob a forma de palavras-chave, contidas no email corrente (em leitura) e, também no histórico de emails. Essa área de propaganda é periférica e embutida na tela de visualização dos emails. Além disso, o GMail utiliza uma abordagem contextual para manter as mensagens, usando o conceito de discussão, ou seja, ele agrupa uma mensagem e as respostas a ela associadas como se fossem uma única mensagem, simulando, assim, uma conversa.

Outro exemplo de uso do contexto para promover adaptação de sistemas web é observado no portal Amazon.com². Esse portal mantém o histórico de utilização do site pelos usuários (e.g. materiais que visualizou, materiais que comprou e pesquisas que realizou), bem como avaliações dos materiais feitas por outros usuários, e o tipo de material que o usuário está consultando (e.g. livro ou DVD). A partir do cruzamento dessas informações contextuais o sistema descobre padrões de associação entre o perfil do cliente e o tipo de material que ele poderia ter interesse e, com isso, apresenta recomendações de materiais que, potencialmente, o usuário poderia estar interessado. Além disso, o sistema sugere guias de navegação pelo site seguindo orientações elaboradas por outros usuários relacionadas ao assunto do material que o usuário está visualizando no momento.

2.4 Diferentes Visões sobre Contexto

As pesquisas sobre o conceito de contexto em Computação podem ser divididas em duas categorias principais: de um lado, há os pesquisadores interessados em aplicar o conceito a suas aplicações para melhorar os serviços e funcionalidades oferecidos (e.g. [Siebra et al. 2007; Park et al. 2007]); do outro lado, encontram-se os pesquisadores interessados em contexto como um conceito, em busca de maneiras de tratá-lo computacionalmente por meio de formalizações, modelos, *frameworks* e metodologias (e.g. [Vieira et al. 2008; Gonzalez e Brézillon 2008; Hirschfeld et al. 2008]). Resultados do segundo grupo são aplicados nas pesquisas conduzidas pelo primeiro. A seguir

¹ <http://mail.google.com>

² <http://www.amazon.com>

analisamos como diferentes áreas da Computação conduzem suas pesquisas sobre o conceito de contexto.

A *Inteligência Artificial* (IA) realiza pesquisa sobre contexto desde os anos 80, sendo considerada a primeira grande área do conhecimento, dentro da Computação, a estudar esse conceito. Em [McCarthy e Buvac 1997] é proposta uma formalização da teoria de contexto por meio da definição de uma “verdade relativizada dentro de um contexto”. A idéia é definir um axioma como um predicado especial *ist(c,p)* que significa que uma proposição *p* é verdadeira (*is true*) apenas dentro de um contexto *c*. Além da formalização do conceito, pesquisadores da IA investigam o uso de técnicas apropriadas de modelagem e raciocínio (e.g. [Franklin 2003; Bouquet et al. 2003]), e o uso de contexto em aplicações inteligentes (e.g. [Petry et al. 2008; Park et al. 2007]).

O termo Computação Ciente de Contexto (*Context-Aware Computing*) foi empregado pela primeira vez em [Schilit et al. 1994] para designar os sistemas capazes de examinar o ambiente computacional e reagir a mudanças nesse ambiente. Essa visão de contexto é especialmente associada à área da Computação Ubíqua, descrita por Weiser como a área que torna possível disponibilizar informações e recursos computacionais “a qualquer hora, em qualquer lugar e em qualquer dispositivo” [Weiser 1991]. Avanços tecnológicos, como a expansão no uso dos dispositivos móveis, comunicação sem fio e sensores mais sofisticados, embutidos em dispositivos cada vez mais poderosos e portáteis, abrem possibilidades para a concretização do sonho de Weiser.

Em áreas que visam apoiar as *interações sociais*, como a *Interação Humano-Computador* (IHC) e a área de Trabalho Colaborativo Apoiado por Computador (CSCW, do inglês *Computer Supported Cooperative Work*), o uso de contexto aparece cada vez mais como um fator relevante. Em IHC, contexto é visto como uma informação implícita que pode ser utilizada para interpretar ações explícitas e melhorar a comunicação entre usuários e aplicações [Moran e Dourish 2001]. Nessa área, contexto está fortemente associado ao problema de personalização em tópicos como *computação centrada no humano* [Bellotti e Edwards 2001] e *adaptação de interface* [de Bra et al. 2004].

Na área de CSCW, contexto tem uma forte associação ao conceito de percepção (*awareness*) [Vieira et al. 2004; Gross e Prinz 2003]. O contexto pode ser relativo ao grupo (e.g. porque o grupo foi constituído), aos membros individuais do grupo (e.g. seu perfil técnico) e ao projeto em si (e.g. seu objetivo) [Borges et al. 2004]. Contexto ajuda, ainda, a identificar relações sociais, uma vez que o conhecimento do contexto é uma forma de estabelecer e reforçar relações de confiança e de reputação entre os membros do grupo. Contexto facilita o entendimento de eventos ocorridos, a comunicação e, em consequência, a execução do trabalho da equipe. Uma característica do uso de contexto em CSCW é que não apenas o contexto de um indivíduo deve ser considerado, mas também o contexto do grupo, por meio do conceito de *contexto compartilhado* [Brézillon e Araújo 2005].

A pesquisa em *Hipermídia Adaptativa* (HA) situa-se na fronteira dos estudos em hipermídia e modelagem do usuário, sendo estes os dois pilares básicos que sustentam o desenvolvimento de aplicações nesta área [Palazzo 2006]. Sistemas baseados na web que fazem uso de HA podem se beneficiar do uso de contexto como forma de otimizar

tanto a apresentação quanto a navegação adaptativa. O contexto do usuário permite que tais sistemas adaptem seus conteúdos, habilitando, incluindo ou removendo *hyperlinks* das páginas de acordo com o que é mais relevante naquela interação.

Contexto aplica-se, ainda, a problemas associados ao gerenciamento de conteúdo e recuperação de informações. A comunidade de Banco de Dados investiga o uso de técnicas de representação de informação para a modelagem do contexto [Vieira et al. 2008; Stefanidis et al. 2005]. O contexto pode ser empregado no apoio à resolução de conflitos semânticos em sistemas de integração de informação [Belian 2008] e na otimização do desempenho de consultas [Bunningen 2004], reduzindo espaços de busca e otimizando padrões de identificação em minerações de dados [Vajirkar et al. 2003].

Uma vez que o uso do contexto implica a inclusão de novos requisitos no desenvolvimento de sistemas computacionais, pesquisadores da área de Engenharia de Software percebem a necessidade de prover metodologias [Vieira et al. 2009; Ayed et al. 2007; Bulcão Neto 2006] e ferramentas arquiteturais que possam apoiar o desenvolvimento de CSS [Hirschfeld et al. 2008; Costa 2007; Henricksen e Indulska 2006; Bardram 2005; Dey e Abowd 2001].

3 Requisitos para o Desenvolvimento de Sistemas Sensíveis ao Contexto

Muitos são os desafios quando se projeta uma aplicação sensível ao contexto. Dentre eles, podemos destacar: (i) a caracterização dos elementos contextuais para uso na aplicação e a sua representação em um modelo semântico; (ii) a aquisição dos elementos contextuais a partir de fontes heterogêneas (e.g. sensores físicos, bases de dados, agentes e aplicações); (iii) o processamento e interpretação das informações adquiridas; (iv) a disseminação e compartilhamento dos elementos contextuais entre diferentes aplicações; (v) a adaptação da aplicação a variações no contexto processado. Além desses, outros requisitos também devem ser considerados como o tratamento da qualidade da informação contextual, o tratamento de questões como segurança, privacidade, e desempenho do sistema.

Pesquisas vêm sendo desenvolvidas com o propósito de gerar ferramentas e métodos específicos para apoiar o tratamento desses desafios. As soluções propostas categorizam-se em *toolkits* [Dey e Abowd 2001], *frameworks* [Vieira 2008; Henricksen e Indulska 2006], *middlewares* [Gu et al. 2005] e *engines* de contexto [Belotti 2004]. Essas abordagens objetivam suprir funcionalidades básicas relativas ao gerenciamento da informação contextual de forma que aplicações possam fazer uso desses serviços, simplificando o desenvolvimento desses sistemas.

Em [Vieira et al. 2008] esses requisitos são sumarizados em três categorias:

- (i) *Especificação do Contexto*: cuida do levantamento dos requisitos de contexto e da modelagem das informações contextuais necessárias para a aplicação;
- (ii) *Gerenciamento do Contexto*: indica como o contexto será manipulado pelo sistema, em termos das seguintes tarefas: aquisição, armazenamento, processamento e disseminação (de CEs);
- (iii) *Uso do Contexto*: define como o contexto influencia o comportamento do CSS e como será efetivamente utilizado.

O desenvolvimento de um CSS deve ser visto de acordo com duas perspectivas: uma parte *que é dependente* de um domínio e aplicação específicos (i.e. especificação e uso do contexto) e outra parte *que é independente* de domínio ou aplicação (i.e. gerenciamento do contexto). Diferentes domínios ou aplicações demandam distintos conjuntos de CEs e de tratamentos do contexto para seu uso. Por outro lado, o gerenciamento de contexto pode ser modularizado e tratado de maneira independente, uma vez que envolve as funcionalidades de adquirir, processar, armazenar e disseminar CEs. Essas categorias de requisitos são detalhadas nas próximas seções [Vieira 2008].

3.1 Especificação do Contexto

Um grande desafio ao desenvolver um sistema sensível ao contexto é delimitar as ações dependentes de contexto nestes sistemas e identificar os elementos contextuais que caracterizam a situação em que essas ações são executadas. Por sua natureza inerentemente dinâmica, na maioria dos casos é muito difícil, ou mesmo impossível, para o projetista de um CSS, enumerar o conjunto de situações que podem existir na aplicação, identificar quais elementos contextuais podem determinar com precisão uma situação desse conjunto e definir que ações devem ser executadas em uma situação particular [Greenberg 2001]. Em [Korpipää et al. 2003] são propostos os seguintes critérios para identificar o que escolher como elemento contextual: (i) habilidade para descrever propriedades úteis do mundo real; (ii) possibilidade de inferência de contextos complexos; e (iii) facilidade ou viabilidade de ser medido ou reconhecido, automaticamente, da forma mais precisa possível.

Visando facilitar a identificação do que considerar como contexto foram propostas diversas classificações para as informações contextuais. Essas classificações separam as informações contextuais segundo critérios diversos, como: as *dimensões* dos elementos contextuais; a *granularidade* da informação; sua *periodicidade de atualização* e sua *relevância em relação ao foco* de atenção do usuário.

Diversos autores (e.g. [Nunes et al. 2007; Bulcão Neto 2006; Vieira et al. 2004]) indicam que as informações contextuais podem ser identificadas a partir da análise de seis *dimensões* básicas, referenciadas como 5W+1H [Bulcão Neto 2006; Vieira et al. 2004; Morse et al. 2000]. Essas dimensões visam responder às questões quem (*who*) está fazendo o que (*what*), em que local (*where*), em que momento (*when*), com qual motivação (*why*) e de que maneira (*how*). Essas questões são consideradas básicas para contextualizar uma situação. Who (*identificação*) indica informações contextuais relacionadas à identidade das entidades (ex. nome, email, impressão digital). Where (*localização*) determina informações contextuais que indicam a localização da entidade (ex. longitude, latitude, cidade, país). What (*atividade*) identifica as atividades em que uma entidade (ex. pessoa) está envolvida (ex. caminhando, ministrando uma palestra). When (*tempo*) indica o contexto temporal relacionado a uma interação (ex. data corrente, estação do ano). Why (*motivação*) relata a motivação por trás das ações do usuário ao executar uma tarefa em uma dada interação. How (*meio*) define a forma como os elementos contextuais são adquiridos (ex. sensor, base de conhecimento).

Quanto à sua *granularidade*, cada elemento contextual pode ser classificado como *básico* ou *complexo* [Wang et al. 2004]. Um *CE básico* é um elemento de granularidade mais baixa e que pode ser identificado, geralmente, de forma automática (e.g. identidade de atores ou dispositivos, coordenadas geográficas, informações

temporais). Já um *CE complexo* é formado a partir da composição e/ou inferência sobre vários outros CEs. Por exemplo, a partir de CEs básicos como a sala onde um usuário está e a indicação de outras pessoas presentes na sala pode-se inferir o CE complexo *atividade atual*, indicando que o usuário está *em reunião*.

Cada elemento contextual, básico ou complexo, pode ser identificado quanto à sua *periodicidade de atualização* e classificado como *estático* ou *dinâmico* [Hong e Landay 2004]. *CE estático* indica uma informação que, em geral, é fixa ou não muda muito frequentemente (e.g. dados pessoais de um usuário – data de nascimento, cidade natal, país de origem, idiomas que fala – e localização de monumentos em uma cidade). *CE dinâmico* é um elemento que muda quase que instantaneamente e necessita ser constantemente monitorado e atualizado (e.g. localização física de um indivíduo, sua atividade atual).

Uma outra classificação do contexto trata do problema da *relevância do contexto em relação ao foco de atenção* atual do usuário [Brézillon 2007]. Essa classificação trata do aspecto dinâmico do contexto e indica que o contexto deve sempre ser visto como relativo a um foco. Um *foco* pode ser um passo na execução de uma tarefa ou em uma tomada de decisão, e é o que permite determinar quais CEs devem ser instanciados e utilizados para compor um contexto. Como ilustra a Figura 2, a partir de um dado foco, o contexto pode ser classificado em *conhecimento contextual* (CK), ou *conhecimento externo* (EK).

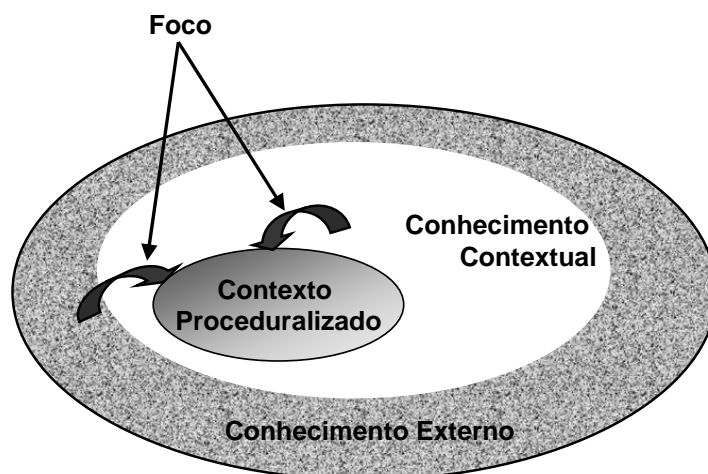


Figura 2. Classificação do contexto quanto à relevância em relação ao foco de atenção (Adaptado de [Brézillon e Araújo 2005])

CK indica os CEs que estão diretamente relacionados com o foco, enquanto EK são os CEs que não possuem qualquer relação com o foco. Por exemplo, se o foco do usuário é encontrar especialistas em desenvolvimento de software, EK inclui elementos como *peso*, *idade* e *estado civil* do especialista, e o CK inclui *localização* do especialista, sua *presença*, *disponibilidade* e *habilidades* – informações que podem apoiar a identificação do especialista adequado. Quando os CEs que compõem o CK são instanciados, estes passam a constituir o que os autores chamam de *contexto proceduralizado*, e que nós chamamos de *contexto ativo*. Um contexto ativo é o conjunto dos CEs que serão efetivamente utilizados em um foco, com seus valores instanciados de acordo com a situação atual no foco. No exemplo, um contexto ativo

inclui a informação que o especialista *denominado* Carlos está *presente e disponível* e possui habilidades na linguagem Java.

3.2 Gerenciamento do Contexto

A Figura 3 ilustra os elementos envolvidos no gerenciamento do contexto, conforme proposto em [Vieira 2008]. As fontes de contexto podem estar associadas ao ambiente físico, ao ambiente virtual de trabalho, a bases de dados existentes, a perfis do usuário, ou ainda ao próprio usuário, que pode informar diretamente ao sistema o seu contexto atual. A aquisição do contexto refere-se ao processo de monitorar, capturar e extrair elementos contextuais de diferentes fontes de contexto [Vieira et al. 2006]. A qualidade dos serviços sensíveis ao contexto está diretamente relacionada com a qualidade da informação coletada. Diferentes fontes de contexto trazem diferentes requisitos para o mecanismo de aquisição. Por exemplo, informações do ambiente físico podem ser obtidas a partir do monitoramento do espaço físico usando sensores enquanto informações provenientes de bases de dados podem ser obtidas por meio de linguagens de consulta ou pela aplicação de técnicas de mineração de dados.



Figura 3. Visão Geral dos Elementos Envolvidos no Gerenciamento do Contexto (Adaptado de [Vieira 2008])

Nem toda informação contextual pode ser adquirida diretamente via monitoramento. Pode ser necessário derivar essa informação a partir da composição de outros elementos contextuais. Por exemplo, a informação se um usuário está “em casa” ou “no trabalho” pode ser derivada a partir de regras de inferência associadas a coordenadas geográficas (que podem ser monitoradas por um sensor de presença). Dessa maneira, o processamento do contexto pode ser entendido como o conjunto de métodos e processos que realizam raciocínio, transformação, combinação e resolução de conflitos das informações contextuais adquiridas de modo a produzir outras informações mais refinadas, relevantes e coerentes com o que é necessário em um

determinado momento [Vieira et al. 2006]. Bases de conhecimento permitem o armazenamento do contexto histórico e juntamente com motores de inferência apóiam o processamento do contexto. O mecanismo de processamento do contexto deve considerar, ainda, o tratamento de incertezas, assumindo que a informação contextual pode conter inconsistências, ser ambígua ou incompleta. Por exemplo, um sensor de presença pode detectar o celular do usuário em casa e inferir que “o usuário está em casa”, enquanto outro sensor de presença baseado em câmeras percebe esse usuário em seu escritório. Essas informações são conflitantes e devem ser resolvidas. O módulo de processamento pode indicar que a informação recebida pela câmera tem maior veracidade e deve, pois, ser a considerada.

Algumas informações contextuais podem ser solicitadas posteriormente para apoiar processos de inferência ou resolução de conflitos. O histórico de determinadas informações contextuais (ex. trajetória de interação de um usuário) demandam armazenamento apropriado. Históricos são de suma importância ao se tentar prever necessidades e ações dos usuários, assim como estabelecer tendências nos processos comuns da aplicação. Ferramentas de recuperação de informação e linguagens de consulta podem ser empregadas para manipular o contexto histórico armazenado.

Uma característica relevante de CSS é que não apenas a interação entre um usuário e uma aplicação é considerada, mas a interação entre dispositivos e aplicações também é fortemente recomendada. Uma forma de obter implicitamente alguma informação sobre um usuário é perguntar a outras aplicações que o usuário utiliza. Com isso, informações contextuais produzidas e armazenadas por uma aplicação podem e devem ser compartilhadas com outras aplicações. Dessa forma, há a necessidade de estabelecer políticas de compartilhamento dessas informações direcionando a sua *disseminação*, considerando a privacidade e a segurança dos usuários. Uma política de privacidade consiste de um conjunto de regras que servem para controlar a execução de ações pelos sistemas. Uma política de segurança identifica os tipos de informações que precisam de proteção e cria mecanismos que identifiquem os riscos e provejam padrões de proteção. Após o sistema aceitar tais políticas, ele concorda em reforçar essas regras e mecanismos quando executar suas ações, de maneira a restringir o acesso a informações de contexto por agentes humanos ou de *software* [Chen e Finin 2004].

3.3 Uso do Contexto

Dentre os tipos de serviços que podem ser afetados pelo conhecimento do contexto, destacamos o suporte à *percepção*, *assistência* e *adaptação*. O conceito de *percepção* é referente à compreensão do estado de um sistema, incluindo suas atividades passadas, estado atual e opções futuras [Gutwin e Greenberg 2002]. A percepção permite que um usuário conheça e compreenda contextos associados a outros usuários, segundo várias dimensões: contextos de localização, contextos organizacionais, contextos sociais, contextos de tarefas, contextos de tecnologias, contextos de interação, entre outros [Gutwin e Greenberg 2002]. Um desafio nos mecanismos de percepção é evitar que os usuários sejam sobrecarregados com informações sobre situações e pessoas dos quais eles não tenham interesse em tomar conhecimento. O contexto é um importante instrumento para apoiar a determinação de que tipo de informação deve ser apresentada aos usuários e de que forma, do modo menos intrusivo possível. O uso do contexto para

melhorar serviços de percepção é denominado, na literatura, percepção contextual ou percepção situacional [Gauvin et al. 2004].

A *assistência* diz respeito a auxiliar o usuário na execução de suas tarefas, por exemplo, com dicas e apresentação de roteiros. Também pode ocorrer na forma de recomendações de artefatos (e.g. objetos, materiais ou pessoas). Um ambiente de ensino a distância, por exemplo, envolve indivíduos geograficamente distantes, com diferentes habilidades, idiomas e interesses. Nesses ambientes pode ser difícil para os aprendizes encontrar materiais de estudo que se adequem às suas necessidades. Com isso, é interessante prover um sistema de recomendação de materiais didáticos, que conheça o contexto individual de cada estudante (e.g. assunto que está estudando, idiomas que domina, nível de conhecimento e dificuldades de aprendizagem), bem como o contexto dos materiais didáticos disponíveis (e.g. finalidade, idioma em que foi escrito, nível de recomendação por outros participantes, grau de impacto na área de estudo e tipo de estudante ao qual se destina). O sistema de recomendação sensível ao contexto pode, de posse dessas informações, apoiar o aprendiz identificando, de forma pró-ativa, materiais que se adequem às suas necessidades atuais. Além disso, pode recomendar o contato com outros estudantes que estejam realizando estudos sobre o mesmo assunto, formando comunidades virtuais de aprendizagem.

A *adaptação* contextual, por sua vez, ocorre quando o contexto é usado para modificar ou adaptar serviços. A adaptação pode ser classificada em: (i) ações disparadas pelo contexto (uma ação é disparada quando um determinado conjunto de informações contextuais atinge valores específicos); (ii) mediação contextual (serviços e dados são adaptados de acordo com os limites e preferências impostos pelo contexto). Projetos de TV Digital Interativa, por exemplo, vêm utilizando informações contextuais relacionadas aos interesses dos usuários com o objetivo de prover adaptação e personalização.

Projetar um sistema que se adapte de acordo com mudanças no contexto é uma tarefa desafiadora uma vez que interpretações errôneas podem acarretar em comportamentos indesejados, que tornariam o sistema irritante e inoportuno ao invés de útil. Para garantir a usabilidade, o CSS não pode ser projetado para agir no lugar dos seus usuários. O usuário deve sentir que possui o controle sobre o sistema podendo autorizar ou recusar ações de uma maneira não intrusiva. Além disso, o usuário deve poder regular as adaptações e regras definidas no sistema de acordo com suas preferências. Para isso, ele deve ser capaz de entender as ações do sistema, especialmente quando estas forem executadas automaticamente de acordo com mudanças no contexto. Em [Bellotti e Edwards 2001], os autores apontam a necessidade de prover mecanismos de *accountability* (prestação de contas) e *intelligibility* (clareza). Ou seja, a aplicação deve prover explicações sobre o raciocínio envolvido na escolha da ação a ser executada e deve permitir que o usuário proveja *feedback* sobre sua satisfação com a ação escolhida.

A adaptação ao contexto deve ocorrer de maneira periférica, não intrusiva, de modo a não perturbar o usuário da execução da sua tarefa atual. Alguns pesquisadores (e.g. [Greenberg 2001; Bellotti e Edwards 2001]) argumentam que CSS devem ser projetados de maneira pessimista, partindo da premissa de que eles possuem uma alta probabilidade de inferir erroneamente o contexto e o comportamento sensível ao contexto. Alguns aspectos inerentes ao ser humano, como intenções, emoções,

interpretações, medos e outras motivações não podem ser percebidas com precisão por um sistema computacional. Dessa maneira, esses sistemas devem ser conservadores nas ações a serem tomadas, tornando visíveis essas ações com possibilidade de serem desfeitas, deixando ações de risco sob o controle do usuário [Bellotti e Edwards 2001].

Um exemplo bem sucedido de CSS que considera estas questões é provido pela Google³. A Google inovou com seu mecanismo de propaganda sensível ao contexto, o AdSense, destinado a prover anúncios mais relevantes e úteis aos usuários, tomando como base as informações exibidas na página que o usuário está visitando (incluindo o email que o usuário está lendo no sistema GMail). Um contra-exemplo é a ferramenta de assistência do Microsoft Office, chamada Clippy. Clippy foi projetado como um pequeno clip de papel que educadamente oferece dicas aos usuários para utilizar os produtos da linha Office. Em [Swarts 2003], os autores analisam porque tantas pessoas não gostam de utilizar o Clippy. Uma razão identificada é que as pessoas, em geral, não gostam que lhes digam o que elas devem fazer, especialmente se elas já sabem como devem proceder. Além disso, torna-se incômodo ser constantemente interrompido da sua tarefa atual com sugestões e questões, muitas vezes, irrelevantes. Em versões mais recentes do Office, o Clippy foi substituído por outra funcionalidade, chamada legendas inteligentes (*smart tags*). Uma *smart tag* provê ações relacionadas a um texto sublinhado as quais podem ser ativadas ou não pelo usuário. Essa funcionalidade é menos intrusiva e deixa o usuário no comando da aplicação.

4 Modelagem de Contexto

Modelos de contexto representam que informações contextuais devem ser consideradas em um domínio ou aplicação e como essas informações se relacionam ao comportamento do sistema. Em geral, modelos de contexto enumeram os conceitos de um domínio que devem ser, eventualmente, considerados como contexto (e.g. contexto do usuário, contexto de localização). Eles estruturam entidades de um domínio e indicam características dessas entidades que são gerenciadas pelo sistema. Entretanto, apenas essa enumeração de elementos não fornece a noção de dinâmica do contexto. Regras de produção são comumente utilizadas para esse fim.

Modelos genéricos de contexto visam descrever a informação que deve ser considerada como contexto de um modo genérico. Esses modelos provêm uma classificação para um conjunto inicial de elementos que compõe o contexto em um domínio. Diferentes aplicações podem reusar a informação modelada estendendo o modelo para tratar particularidades da aplicação. Modelos genéricos de contexto foram propostos em áreas como sistemas pervasivos [Chaari et al. 2007], sistemas colaborativos [Vieira et al. 2005], integração de dados [Souza et al. 2008] e ambientes inteligentes [Gu et al. 2005].

Um sistema sensível ao contexto requer que informações contextuais sejam trocadas e utilizadas por diferentes entidades, como agentes humanos e de software, dispositivos e serviços, com uma mesma compreensão semântica. Para isso, um modelo de contexto apropriado deve dar suporte à interoperabilidade semântica e permitir que esquemas comuns sejam compartilhados entre diferentes entidades. Uma característica

³ <http://www.google.com/corporate/>

de CSS é que os elementos contextuais podem (e devem) ser procedentes de fontes diversas e heterogêneas, muitas vezes externas ao CSS. Assim, a representação das informações contextuais provavelmente não irá considerar um modelo único, padrão e globalizado de representação [Power 2003]. Formalizar todas as possíveis informações contextuais em ambientes restritos e em situações controladas já é bastante difícil. Em cenários inter-organizacionais ou considerando diferentes sistemas se torna praticamente impossível.

Pesquisas relacionadas à modelagem de contexto focam em: (1) identificar técnicas de representação que melhor se adaptem às características do contexto [Strang e Linnhoff-Popien 2004]; (2) enumerar os elementos que devem ser considerados como contexto em um domínio ou conjunto de aplicações [Souza et al. 2008; Siebra et al. 2007]; e (3) guiar a modelagem do contexto por meio de *modelos genéricos* [Bulcão Neto 2006] e metamodelos [Vieira et al. 2008; Sheng e Benatallah 2005; Fuchs et al. 2005]. A Seção 4.1 aborda algumas técnicas propostas para representação de contexto e a Seção 4.2 apresenta uma proposta de metamodelo de contexto.

4.1 Técnicas para Representação de Contexto

Pesquisadores vêm investigando o uso de diferentes técnicas de representação de informações e conhecimento para a representação do contexto [Strang e Linnhoff-Popien 2004] já há algum tempo. Algumas dessas técnicas, sumarizadas no Quadro 1, são discutidas na Seção 4.1.1. Uma técnica, em particular, os grafos contextuais, é apresentada na Seção 4.1.2.

4.1.1 Análise Comparativa das Principais Técnicas

Como descrito no Quadro 1, existem diferentes técnicas propostas para apoiar a modelagem de elementos contextuais. Pares de chave-valor e linguagens baseadas em marcações são as técnicas mais simples e fáceis de usar e implementar. Entretanto, nenhuma delas é adequada para sistemas complexos, pois não provêem suporte para visualização do modelo e são difíceis de manter e evoluir.

Embora a abordagem de mapa de tópicos proveja suporte para a visualização, navegação e integração de elementos contextuais provenientes de fontes heterogêneas, essa técnica ainda está em estágio imaturo em termos de definições de padrões, tecnologias e ferramentas. Outra característica é que o seu foco é na representação de indivíduos e não de conceitos (categoria de indivíduos). Com isso, o apoio ao processamento do contexto e o uso de regras contextuais se torna mais complicado, uma vez que não permite generalizar o tratamento do contexto para categorias de indivíduos.

O suporte à representação e processamento de regras e à representação semântica de conceitos é o ponto mais relevante da abordagem baseada em ontologias. Ontologias permitem formalizar a semântica dos elementos contextuais, associando conceitos semânticos (e.g. sinônimo, parônimo, meronímia, e outros) aos elementos contextuais, além de permitir a ligação de regras de contexto aos elementos contextuais. Por essas razões, essa técnica tem sido a mais utilizada para modelar informações contextuais. Dentre as desvantagens observadas nessa abordagem citamos a imaturidade das tecnologias e ferramentas associadas que tornam custosa a criação de uma ontologia e impactam no desempenho dos sistemas. Além disso, ontologias permitem modelar

apenas a estrutura de um modelo de contexto (i.e. os conceitos), porém não oferecem apoio para a modelagem do comportamento do CSS.

Quadro 1. Resumo das Técnicas de Representação de Contexto

Técnica	Vantagens	Desvantagens	Processamento e Recuperação
Par Chave-Valor	Estrutura simples, de fácil implementação e uso.	Não considera hierarquia. Inadequado para aplicações com estruturas complexas	Busca linear com casamento exato de nomes
Linguagem de Marcação	Prevê hierarquia. Esquema de marcação implementa o próprio modelo.	Modelo não resolve incompletude e ambigüidade Inadequado para representar estruturas complexas	Linguagem de consulta baseada em marcação
Mapas de Tópicos	Facilita a navegação entre os elementos contextuais e a leitura por humanos.	Técnica imatura com baixo suporte de ferramentas.	Navegação por redes semânticas
Ontologias	Agrega regras, conceitos e fatos em um só modelo. Padrões facilitam reuso e compartilhamento. Viabiliza compreensão semântica entre humanos e máquinas.	Não permite modelar o comportamento do CSS. Tecnologia ainda imatura e com poucas ferramentas. Impacto no desempenho.	Motor de Inferência, Linguagens de consulta baseadas em OWL ou frames.
Modelos Gráficos	Facilita a especificação dos conceitos e definição do comportamento do CSS	Não permite processar os conceitos: mapeamento para estruturas de dados	Pode ser traduzido para XML e usa de processamento em XML

O comportamento do CSS pode ser modelado por meio de outras técnicas de representação, como modelos gráficos (e.g. UML). A grande vantagem de modelos gráficos é apoiar a modelagem conceitual do CSS. Modelagem conceitual, como definido em [Mylopoulos 1992] é a atividade de descrever formalmente alguns aspectos do mundo físico e social que nos cerca para propósitos de compreensão e comunicação. Extensões à UML vêm sendo propostas como uma forma de adaptar os modelos da UML às particularidades relacionadas a contexto [Simons e Wirtz 2007; Bauer 2003].

A abordagem de grafos contextuais [Brézillon 2007] é um formalismo voltado para a representação da dinâmica do contexto, identificando a influência dos elementos contextuais sobre o comportamento do CSS. O problema com a abordagem de grafos contextuais é que ela não provê um formalismo que permita manipular os conceitos modelados, sendo necessário um mapeamento para outra estrutura de representação para fins de implementação.

Cada técnica de representação de contexto possui vantagens e deficiências. Com isso, não há uma técnica que seja considerada unanimemente a ideal para todos os sistemas sensíveis ao contexto, uma vez que diferentes sistemas impõem diferentes restrições. Dessa maneira, uma abordagem híbrida, com a conjunção de diferentes técnicas em cenários específicos, é o mais apropriado para um modelo de contexto. Um exemplo de modelo híbrido, combinando ontologias com um modelo gráfico baseado em ORM (*Object-Role Modeling*) é proposto em [Henricksen et al. 2004]. Em [Vieira

2008] é proposto um modelo híbrido de representação de contexto com o uso de ontologias para representar a estrutura da informação contextual e a associação desse modelo de estrutura a grafos contextuais [Brézillon 2007] para representar o comportamento sensível ao contexto. Devido à sua relevância para o metamodelo de contexto, descrito na Seção 4.2, a próxima seção detalha o formalismo de grafos contextuais.

4.1.2 Grafos Contextuais

Grafo contextual (CxG)⁴ [Brézillon 2007] é uma técnica de representação que objetiva modelar as atividades e raciocínio envolvidos na resolução de um problema, considerando condições específicas (o contexto) em que as atividades são executadas. Um CxG (ver exemplo na Figura 4) é um grafo acíclico direcionado com uma única entrada, uma única saída e uma organização serial-paralela de nós, conectados por arcos orientados, indicando um caminho de um problema até uma solução. Existem três tipos de nós: *nós contextuais* (círculos claros na figura), indicam pontos de decisão onde o valor do elemento contextual é avaliado; *nós de recombinação* (círculos escuros), que indicam que o valor testado para o elemento contextual não é mais válido; e *nós ação*, que representam uma atividade a ser executada. Os arcos indicam caminhos dependentes do contexto que avaliam se um conjunto de atividades deve ou não ser executado na resolução daquele problema.

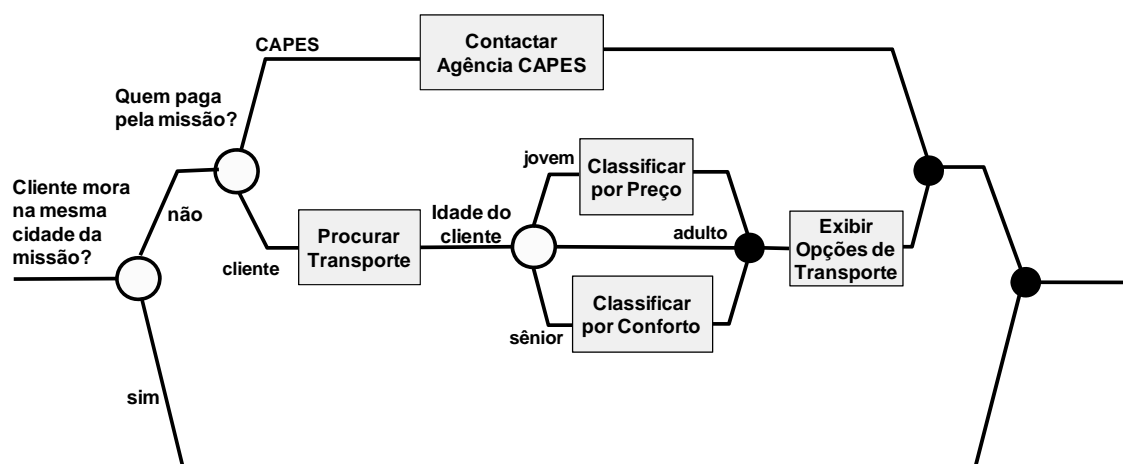


Figura 4. Exemplo de grafo contextual para a tarefa *Reservar Transporte*

O exemplo da Figura 4 trata do problema de reservar um transporte para uma pessoa que deseja participar de um evento acadêmico (missão). Um caminho possível é não fazer nada, caso a pessoa more no mesmo local de realização do evento. Um outro caminho é contatar a agência da CAPES caso a pessoa more em um local diferente do evento e a missão seja paga pela CAPES. Grafos contextuais surgem como uma abordagem interessante para modelar o *comportamento* sensível ao contexto, pois descrevem as ações a serem executadas pela aplicação e indicam, explicitamente, como o contexto afeta essas ações.

⁴<http://www.cxg.fr>

4.2 Um Metamodelo de Contexto

Modelagem e metamodelagem são atividades similares. A diferença está na interpretação. Metamodelos de contexto definem a terminologia e semântica dos principais conceitos que devem ser considerados ao construir um modelo de contexto. Eles provêm novos elementos de modelagem relacionados ao contexto, fornecendo, assim, uma infra-estrutura conceitual para apoiar a construção de modelos de contexto. Para isso, os conceitos relacionados a contexto são considerados em uma camada mais alta de abstração. Uma arquitetura de metamodelagem em quatro níveis, proposta em [OMG 2007], é ilustrada no Quadro 2. Cada camada superior oferece linguagens que permitem a criação da camada imediatamente inferior. O metamodelo de contexto está na camada M2. Modelos de contexto são criados a partir da instanciação do metamodelo de contexto. O nível M0 indica os elementos reais do domínio modelado.

Abordagens para modelagem de contexto em geral estão associadas à camada M1 e os modelos de contexto são construídos como extensões de linguagens de modelagem existentes (e.g. OWL, UML). Uma vez que essas linguagens foram concebidas para uso geral, e não especificamente para sistemas sensíveis ao contexto, elas oferecem pouco suporte para a criação de modelos de contexto. Um metamodelo de contexto permite abstrair as particularidades associadas à manipulação das informações contextuais, oferecendo um suporte aos projetistas desses modelos, que podem usar o metamodelo como base para estruturar os seus modelos de contexto.

Quadro 2. Metamodelo de contexto em uma arquitetura em quatro camadas (Adaptado de [Vieira 2008])

Camada	Exemplo
(M3) Meta-metamodelo	Modelo MOF (e.g. Metaclass)
(M2) Metamodelo	Metamodelo UML (e.g. Class, Attribute) Metamodelo de Contexto (e.g. Entity, CE, Focus)
(M1) Modelo	Modelo de Hotel (Hotel, nome, Hotel tem nome)
(M0) Instâncias	Banco de Dados de Hotel (<h1>,<“Ritz”>, <h1, “Ritz”>)

Em [Vieira 2008] é proposto um metamodelo de contexto que abstrai os conceitos relativos à manipulação de contexto como uma forma de tornar explícito o tratamento do contexto pela aplicação e apoiar a criação de modelos de contexto. Esse metamodelo inclui suporte para a modelagem da estrutura das informações contextuais e seu uso para prover o comportamento sensível ao contexto.

Como indicado no Quadro 2, o metamodelo de contexto está associado ao Metamodelo da UML disponível na especificação 2.0 da UML [OMG 2007]. O metamodelo de contexto é dividido em dois pacotes principais que organizam os conceitos em duas categorias de modelagem: estrutura dos elementos contextuais (Seção 4.2.2) e comportamento do CSS (Seção 0). Para ilustrar a apresentação dos conceitos do metamodelo, a Seção 4.2.1 apresenta um cenário de uso.

4.2.1 Cenário de Exemplo: Missões Acadêmicas

Considere um sistema que apóia pesquisadores no planejamento das suas missões acadêmicas. Uma missão acadêmica é qualquer evento científico ou acadêmico do qual

podem participar pesquisadores, professores e alunos (e.g. conferência, estágio, reunião ou workshop). Missões podem possuir características particulares como período de realização, duração, localização e tarefas que devem ser realizadas para viabilizar a ida do participante ao local da missão (e.g. registrar no evento, buscar apoio financeiro, reservar passagem e hotel, prestar contas). Diferentes participantes podem necessitar executar diferentes passos e preencher diferentes requisitos ao planejar a ida a uma missão. Por exemplo, em uma universidade, os recursos disponíveis (e.g. apoio financeiro) e os requisitos a serem preenchidos são bastante distintos para um professor e para um aluno. Alunos de pós-graduação, por sua vez, podem ter acesso a recursos não disponíveis para alunos da graduação.

Um sistema de apoio ao planejamento de missões acadêmicas sensível ao contexto poderia considerar as seguintes funcionalidades: (i) identificar os passos a serem cumpridos e os requisitos a serem preenchidos de acordo com o tipo da missão e o perfil do participante; (ii) apoiar o participante na reserva de transporte e acomodação (e.g. pela busca automática de informações disponíveis em serviços web de reservas); ou (iii) identificar outras questões relacionadas à missão que podem ser úteis para o participante (e.g. atividades acadêmicas que ele poderia participar, informações sobre conversão de moeda, ou dicas para planos turísticos).

4.2.2 *Metamodelo de Contexto: Estrutura dos elementos contextuais*

Para modelagem da estrutura dos elementos contextuais, os seguintes conceitos principais foram definidos (Figura 5): `ContextualEntity`, `ContextualElement`, `Focus` e `ContextSource`.

- `ContextualEntity` (*entidade contextual*): uma entidade é uma representação de um objeto concreto do mundo real que pode ser identificado distintamente e que é relevante para descrever o domínio. Ela pode ser utilizada para classificar indivíduos com características similares e contém descrições desses indivíduos por meio de atributos embutidos. Exemplos de entidades no domínio de missões acadêmicas são: *Pessoa*, *Professor*, *Aluno*, *Missão* e *Hotel*. No metamodelo de contexto, uma entidade contextual representa as entidades do modelo da aplicação que devem ser consideradas para propósitos de manipulação das informações contextuais. Uma entidade contextual é caracterizada por, pelo menos, um elemento contextual.
- `ContextualElement` (*elemento contextual*): Um elemento contextual (CE) representa uma propriedade usada para caracterizar uma entidade contextual. propriedades são relações binárias que ligam dois indivíduos (ou um indivíduo com ele mesmo) ou um indivíduo a um valor de dado. O primeiro tipo de propriedade é chamado de relacionamento e o segundo tipo de atributo. Exemplos de CEs no cenário de missões acadêmicas associados à entidade *Aluno* incluem: *idade*, *sexo*, *nivelAcademico*, *orientador*, *localOndeMora*. Para a entidade *Missão*, exemplos de CEs são: *localRealizacao*, *duracao*, *tipo*, *dataInicio*, *dataFim*. Todo CE deve estar associado a uma entidade contextual, porém nem todas as propriedades dessa entidade são, necessariamente, elementos contextuais. O critério para indicar se uma propriedade é um CE é subjetivo e depende dos requisitos de contexto definidos para o CSS.

percebido, derivado, informado pelo usuário ou consultado), a frequência de atualização do CE (i.e. *frequentemente, raramente, nunca*).

Para apoiar a modelagem da estrutura dos elementos contextuais, o metamodelo propõe o uso de extensões ao modelo conceitual de classes da UML. A Figura 6 ilustra o modelo conceitual de contexto para o domínio de missões acadêmicas. Os estereótipos nas classes e relacionamentos representam os conceitos definidos no metamodelo.

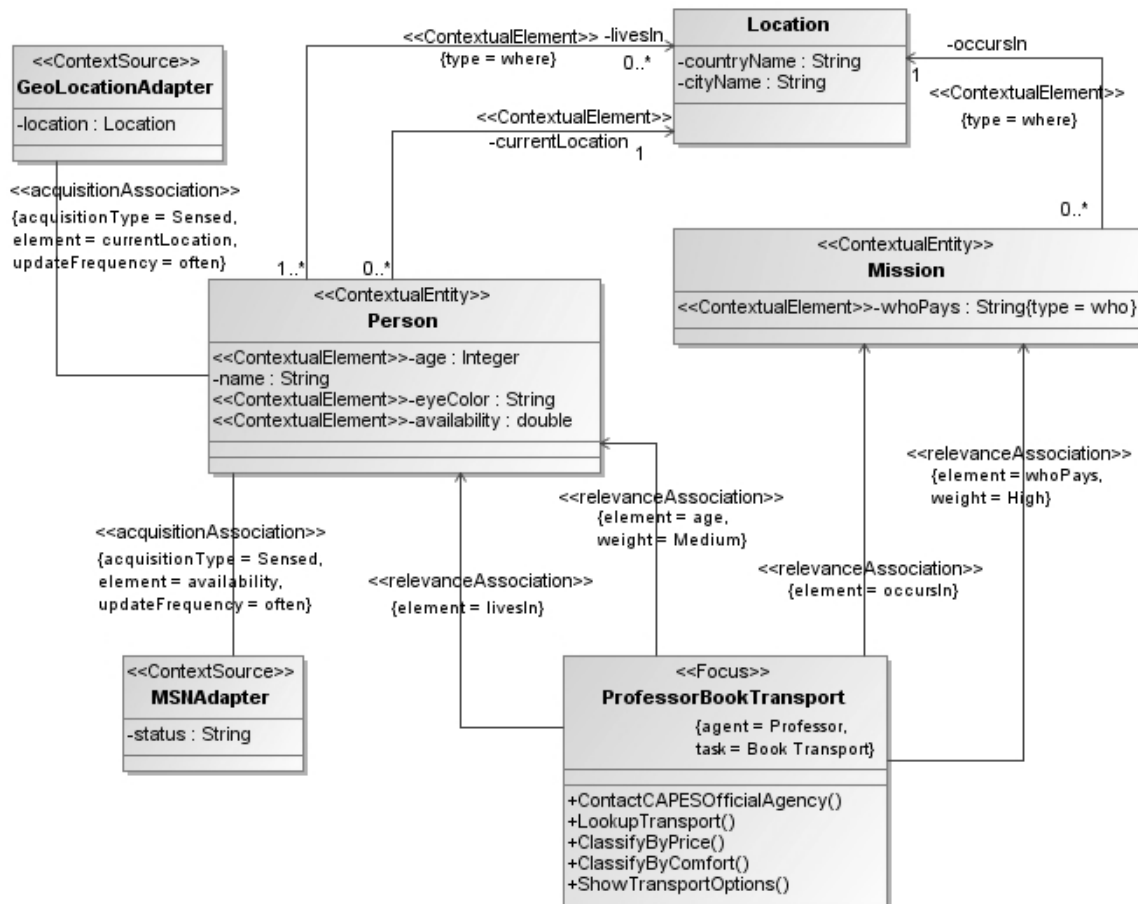


Figura 6. Modelo de contexto para o domínio de missões acadêmicas [Vieira 2008]

4.2.3 Metamodelo de Contexto: Comportamento do CSS

Para modelar o comportamento do CSS o metamodelo de contexto usa os seguintes conceitos principais: *CERelevance* e *Rule* (ver Figura 5).

- *CERelevance* (*relevância do CE*): uma questão importante em um modelo de contexto é a associação entre foco e os CEs relevantes para apoiar a tarefa em execução naquele foco. Esse nível de relevância pode mudar de acordo com diferentes tarefas sendo executadas e também quando diferentes agentes são considerados. Por exemplo, ao executar a tarefa *Reservar Hotel*, um agente *Aluno* pode indicar que o CE *precoHotel* deve ser considerado com uma relevância mais alta em relação ao CE *nivelConfortoHotel*. Um outro agente pode ter uma percepção diferenciada sobre a relevância desses CEs na reserva de um hotel. O metamodelo de contexto contempla essa associação de

caminhos são identificados de acordo com o valor associado a esse CE. Por exemplo, se (value="<26"), indicando que o participante é bastante jovem, o grafo indica a execução da ação “*Classificar por Preço*”, segue com a execução de mais uma ação “*Exibir Opções de Transporte*” e, por fim, termina a execução da tarefa. O grafo contextual possui ainda o conceito de *nó de recombinação*, que indica a convergência dos diferentes caminhos abertos em um nó contextual. Cada nó contextual tem um nó de recombinação associado. Por exemplo, a condição verificada no nó contextual CE3 se encerra no nó de recombinação R3.

Cada caminho no grafo contextual contém a definição de uma regra de produção que indica a influência do contexto sobre o comportamento da aplicação. Assim, considerando o grafo contextual da Figura 7, exemplos de regras incluem:

Regra1:

Condições

not (Missao.ocorreEm=Pessoa.moraEm)
Missao.quemPaga="CAPES"

Ações

CallBehavior("Contactar Agência CAPES")

Regra2:

Condições

not (Missao.ocorreEm=Pessoa.moraEm)
Missao.quemPaga="participante"
Pessoa.idade < 26

Ações

CallBehavior("Buscar Transporte")
CallBehavior("Classificar por Preço")
CallBehavior("Exibir Opções de Transporte")

5 Processos para o Desenvolvimento de Sistemas Sensíveis ao Contexto

Projetistas de sistemas têm dificuldade em especificar o que exatamente considerar como contexto e como projetar a sua representação, gerenciamento e uso. Na literatura relacionada a contexto, poucos trabalhos abordam processos de apoio ao projeto e desenvolvimento de CSS. A seguir, apresentamos e discutimos três processos propostos.

5.1 Processo POCAp

O processo POCAp (*Process for Ontological Context-aware Applications*) [Bulcão Neto 2006] é um processo de software proposto como um conjunto de atividades para o desenvolvimento de sistemas sensíveis ao contexto baseados em ontologias. Ele utiliza a notação SPEM (*Software Process Engineering Metamodel*) [OMG 2008]. POCAp considera o desenvolvimento de um CSS de acordo com quatro atividades principais: *análise e especificação*, *projeto*, *desenvolvimento*, e *verificação e validação*.

A atividade de *análise e especificação* visa estabelecer o que a aplicação sensível a contexto deve fazer (requisitos funcionais), as restrições na operação e desenvolvimento da aplicação (requisitos não-funcionais), e o conjunto de informações de contexto necessário. Ela subdivide-se em quatro atividades de *análise e especificação de* (Figura 8): *requisitos*, *informação contextual*, *reuso do modelo* e *extensão do modelo*. Um elemento importante dessa atividade é o modelo ontológico de informações de contexto que fornece um vocabulário de termos a ser utilizado pela aplicação.

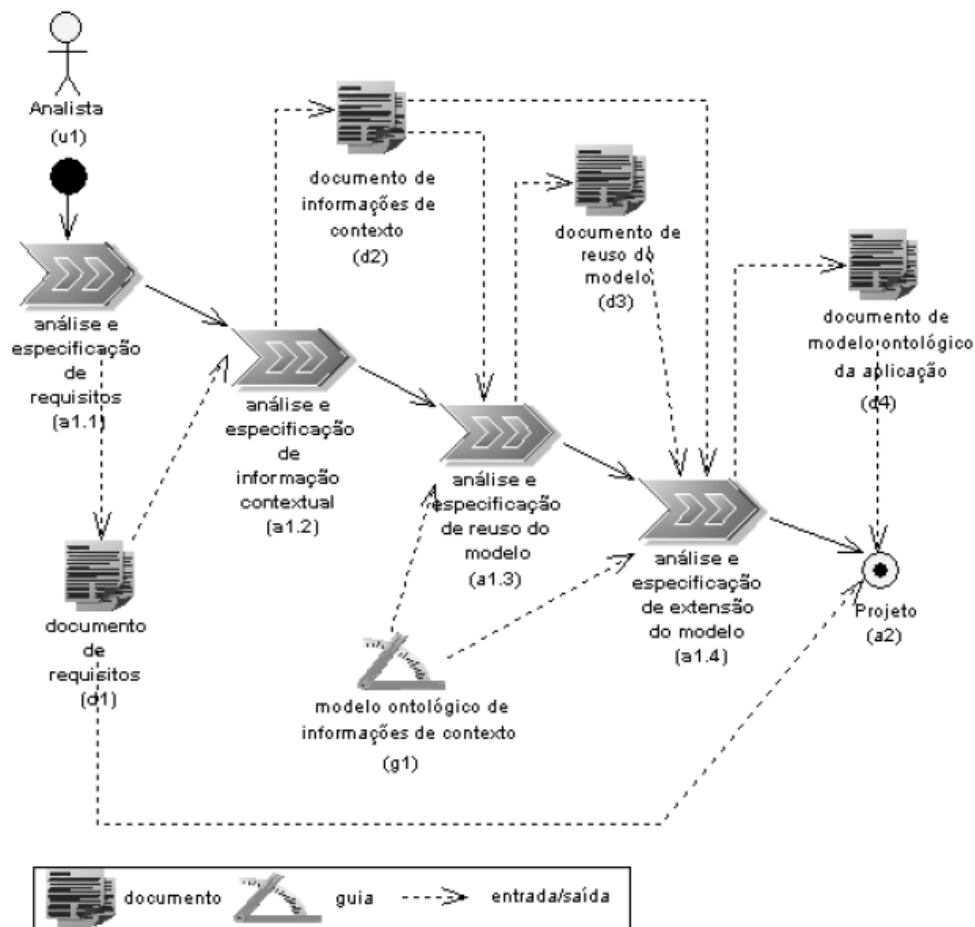


Figura 8. A atividade de análise e especificação do processo POCAp [Bulcão Neto 2006]

A atividade de *Projeto* visa produzir um projeto arquitetural de um software que satisfaça à especificação da aplicação sensível a contexto definida na fase de análise. Ela compreende três tarefas relativas ao *projeto de*: *reuso de serviços*, *extensão de serviços* e *novos serviços*. A atividade de *Desenvolvimento* visa traduzir a estrutura de software produzida na atividade de projeto para um programa executável. Por fim, a atividade de *Verificação e Validação* corresponde às atividades de checar, revisar e testar se a aplicação sensível a contexto está em conformidade com sua especificação e se atende aos requisitos estabelecidos.

O processo POCAp oferece como artefatos para apoiar as tarefas de análise e especificação e projeto, o modelo SeCoM (*Semantic Context Model*), um conjunto de ontologias relacionadas às diferentes dimensões das informações contextuais e a arquitetura SCK (*Semantic Context Kernel*), uma infraestrutura para manipular as ontologias definidas no modelo de contexto [Bulcão Neto 2006].

A abordagem proposta nesse processo é centralizada nos problemas relacionados à modelagem e manipulação de ontologias, assumindo que o sistema sensível ao contexto necessariamente fará uso de ontologias como mecanismo de representação. O processo organiza as diferentes atividades relacionadas ao desenvolvimento de aplicações sensíveis ao contexto, em relação à manipulação dessas ontologias. A notação SPEN provê elementos que permitem detalhar as atividades

especificando produtos de entrada e saída, bem como guias que podem apoiar a execução de cada atividade.

5.2 Uma Proposta de Metodologia de Engenharia de Software

Em [Henricksen e Indulska 2006] é proposta uma metodologia de Engenharia de Software, que indica o fluxo de atividades que deve ser executado ao desenvolver uma aplicação sensível ao contexto. Essa metodologia envolve as fases de *análise*, *projeto*, *implementação*, *customização da infra-estrutura* e *teste* (Figura 9).

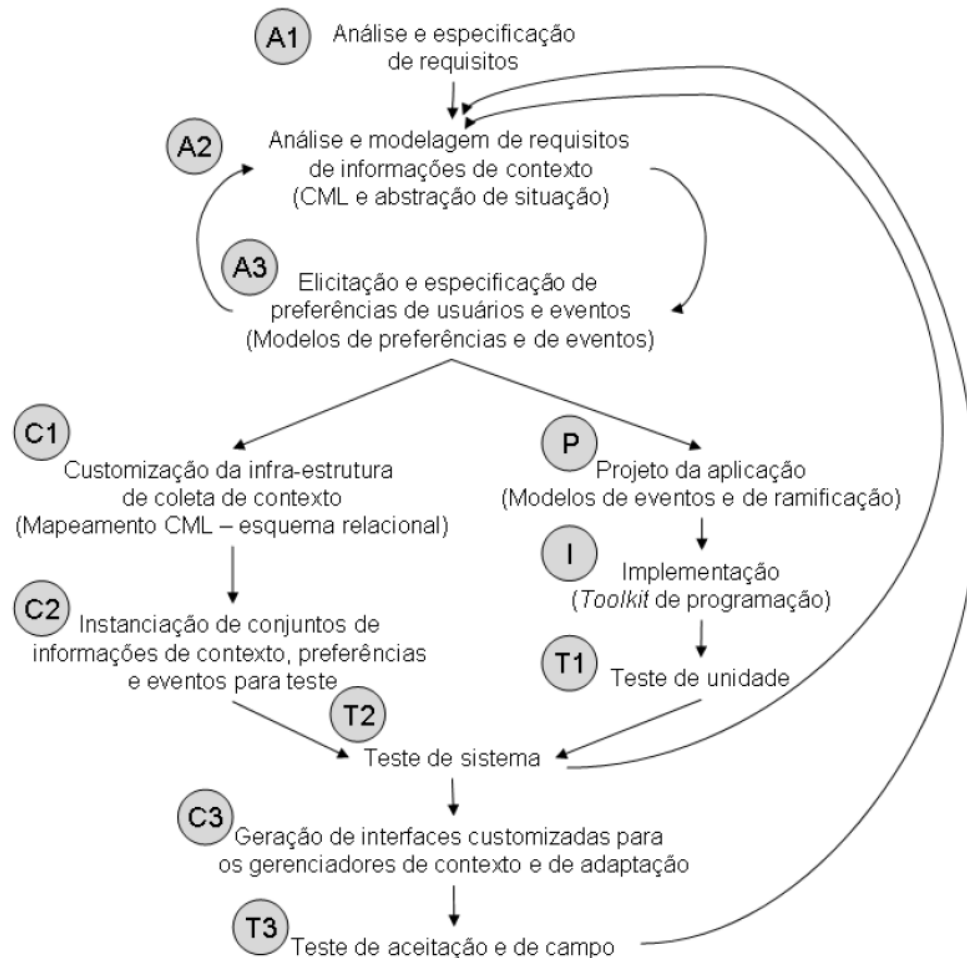


Figura 9. Visão geral do processo de engenharia de software sensível ao contexto (Adaptado de [Henricksen e Indulska 2006])

A fase de análise inicia com a documentação dos requisitos da aplicação (passo A1). Em seguida, há dois passos específicos para aplicações sensíveis a contexto: o passo A2, que busca identificar os tipos de informação de contexto (fatos e situações) a partir do documento definido no passo A1; e o passo A3, que refina requisitos levantados no passo A1 que são dependentes de contexto, tais como a definição de eventos e de preferências de usuários. Após a fase de análise, dois conjuntos de tarefas podem ser executados em paralelo: um que envolve projeto (P) e implementação (I), e outro que abrange a customização (C) da infra-estrutura de software.

Tarefas de projeto e de implementação visam incorporar o comportamento sensível a contexto às aplicações usando metodologias e linguagens de programação tradicionais. A fase de customização ocorre quando é necessário definir novos tipos de fatos ou situações que demandem a implementação de novos receptores, agregadores e interpretadores para as camadas de coleta e recepção de contexto (passo C1). Para o teste da aplicação, devem ser geradas amostras de informações de contexto, de gatilhos associados a eventos e de preferências de usuários: tudo será utilizado pelos gerenciadores de contexto e de adaptação (passo C2). Antes da implantação da aplicação, deve-se verificar a conformidade (passo C3) entre as interfaces de customização com o modelo de informação de contexto e os conjuntos de preferências e de eventos definidos nos passos A2, A3, C2 e T2. As fases de teste incluem teste de unidade (T1), teste do sistema (T2) e teste final (T3), no qual a aplicação é avaliada usando um ambiente real de hardware, com dados reais de sensores e usuários finais

5.3 CSS Design Process

O *CSS Design Process* [Vieira et al. 2009] detalha atividades relacionadas à especificação de informações de contexto e ao projeto de sistemas sensíveis ao contexto, provendo uma maneira sistemática de executar essas atividades. Assim como o POCap, explicado na seção 5.1, o CSS Design Process também utiliza a notação SPEM [OMG 2008]. As atividades do processo são descritas em detalhes com indicação dos artefatos que são usados como entrada para a atividade, os artefatos que são produzidos a partir da execução da atividade e recursos que podem ser utilizados como guias para apoiar a execução da atividade.

Uma visão geral do processo é ilustrada na Figura 10. O processo considera na equipe de desenvolvimento de software, além dos papéis tradicionais de analista e projetista do sistema, a inclusão de um novo papel, o *projetista de contexto*. Este deve ser desempenhado por uma pessoa ou equipe especializada na identificação de requisitos de contexto e no projeto de soluções sensíveis ao contexto. Dentre as habilidades indicadas para o projetista de contexto está a especialidade multidisciplinar em assuntos relacionado à cognição humana, tecnologias de aquisição automática de informações (e.g. sensores), técnicas de Inteligência Artificial, desenvolvimento de software e usabilidade de software. O processo prevê as seguintes atividades principais: *Especificação do Contexto*, *Projeto do Gerenciamento do Contexto*, *Projeto do Uso do Contexto*, *Geração de Código*, *Teste do CSS* e *Avaliação do CSS*. A versão atual do processo detalha a execução das três primeiras atividades.

A *Especificação do Contexto* (Figura 11) tem por objetivo identificar os requisitos de contexto com base nos requisitos do negócio da aplicação e criar o modelo conceitual de contexto. Para isso, quatro tarefas devem ser executadas: *Identificar o Foco* (S1), *Identificar as Variações de Comportamento* (S2), *Identificar as Entidades e CEs* (S3) e *Verificar a Relevância dos CEs* (S4). Caso sejam identificados diferentes focos (ver definição de foco e elemento contextual na Seção 3.4.1) este ciclo de tarefas deve ser executado para cada foco.

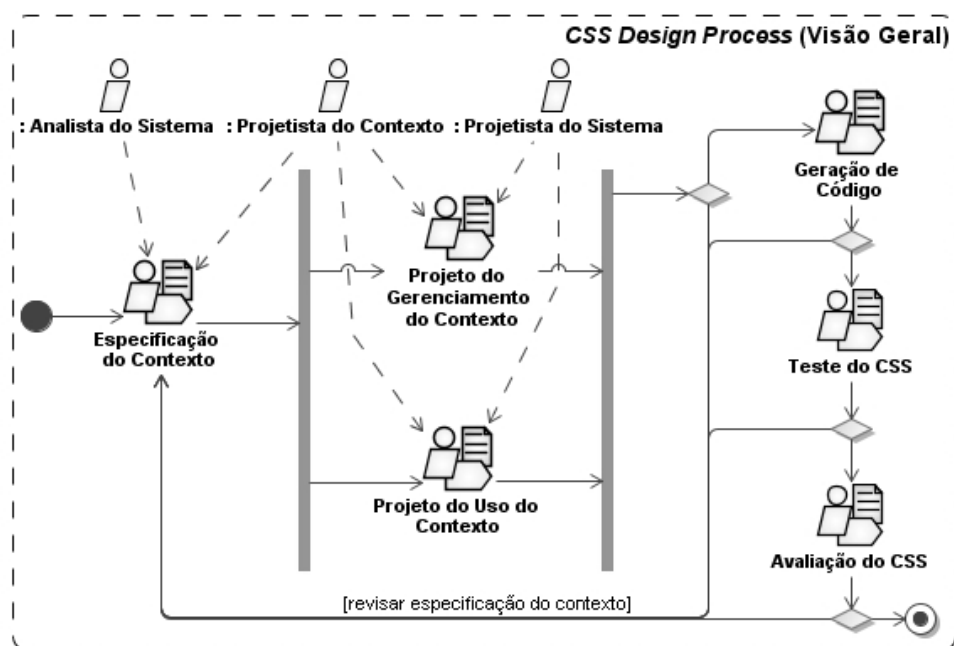


Figura 10. Visão Geral do CSS Design Process (Adaptado de [Vieira 2008])

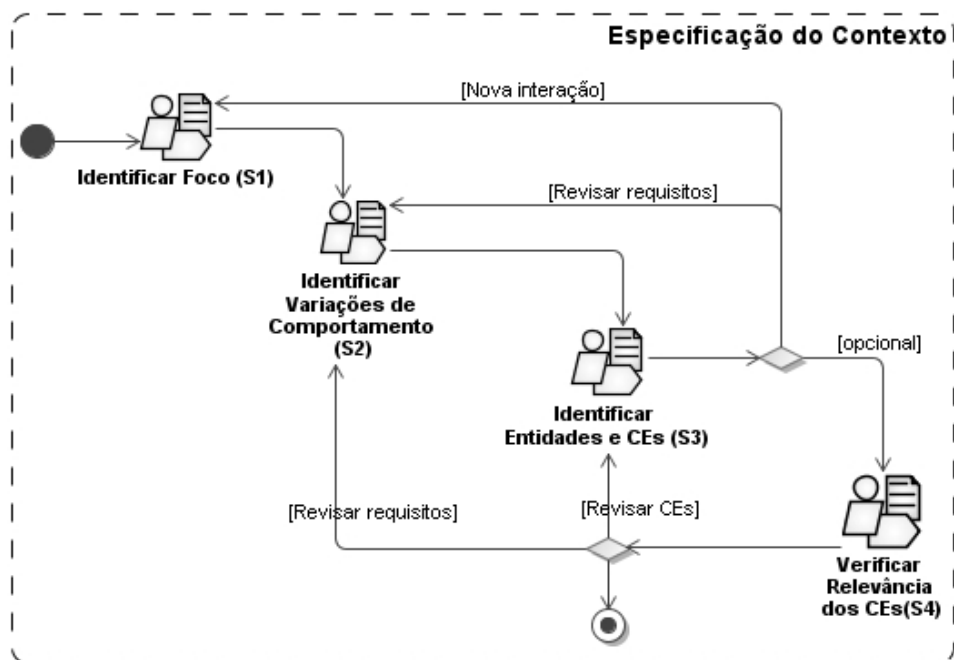


Figura 11. Atividades relacionadas à fase de Especificação do Contexto (Adaptado de [Vieira 2008])

Após a identificação dos requisitos de contexto e da elaboração do modelo conceitual de contexto, o projetista deve definir como será o gerenciamento dos CEs, ou seja, a aquisição, processamento e disseminação das informações contextuais definidas na fase de especificação do contexto. Dessa forma, a atividade *Projeto do Gerenciamento de Contexto* prevê as seguintes tarefas: *Especificação da Aquisição do*

Contexto (M1), *Projeto do Módulo de Aquisição* (M2), *Projeto do Módulo de Processamento* (M3) e *Projeto do Módulo de Disseminação* (M4).

A terceira atividade principal definida no CSS Design Process é o *Projeto do Uso do Contexto* que tem por objetivo definir como o contexto será efetivamente utilizado na aplicação. O processo considera que existem dois usos principais do contexto em uma aplicação: apoiar uma adaptação do comportamento da aplicação ou enriquecer a percepção dos usuários com informações contextuais gerenciadas pelo sistema. Com isso, essa atividade subdivide-se em: *Projetar o Modelo de Comportamento* (U1), *Projetar a Adaptação ao Contexto* (U2) e *Projetar a Apresentação do Contexto* (U3).

5.4 Considerações sobre os Processos

Existe uma demanda não suprida por ferramentas de Engenharia de Software que apóiem a construção de sistemas sensíveis ao contexto. Nesta seção revisamos três propostas de processos voltados para apoiar desenvolvedores desses sistemas. O processo POCap (Seção 5.1) descreve, em alto nível, as atividades relacionadas à construção de um CSS como uma instanciação de um processo de software tradicional. A principal desvantagem dessa abordagem é que ela foca basicamente no problema de desenvolver um sistema que considere o uso de ontologias, assumindo que essa é a técnica de representação de contexto adotada. O processo não detalha aspectos inerentes a sistemas sensíveis ao contexto apontando o que diferencia essas aplicações das convencionais (que não consideram contexto). O processo proposto em [Henricksen e Indulska 2006] contém apenas um fluxo de alto nível de atividades que devem ser seguidas para usar a linguagem de modelagem de contexto e as abstrações de programação propostas em seu trabalho. O processo não detalha as atividades, e nem provê guias ou indicação de artefatos de entrada e saída consumidos e produzidos pelas atividades. O *CSS Design Process* visa tornar explícito o uso do contexto em uma aplicação, por meio da separação clara dos requisitos de contexto dos requisitos convencionais da aplicação. Esse processo, no entanto cobre apenas a parte de projeto do CSS não explorando questões relativas à implementação da aplicação. Na próxima seção usaremos o *CSS Design Process* para exemplificar o projeto de um CSS.

6 Exemplo de Construção de um Sistema Sensível ao Contexto

Para exemplificar a execução das atividades descritas no *CSS Design Process*, esta seção descreve a construção de um sistema de recomendação de especialistas (ERS, do inglês *Experts Recommender System*) denominado ICARE (*Intelligent Context Awareness for Recommending Experts*) [Petry et al. 2008]. ICARE é um ERS que usa o contexto do usuário e dos especialistas para melhorar a qualidade das recomendações. Detalhes sobre a construção desse CSS podem ser encontrados em [Vieira 2008].

6.1 Visão Geral do ICARE

Um ERS é um sistema que retorna referências a indivíduos identificados como especialistas em um domínio de conhecimento solicitado e que pode ser utilizado para conectar atores humanos [Reichling et al. 2005]. Ao executar uma tarefa, resolver um problema ou tomar uma decisão, as pessoas podem economizar tempo e esforço se interagirem com outros com o conhecimento e experiência necessários para a tarefa.

Contexto pode apoiar ERS a prover melhores recomendações uma vez que ele pode informar quem é o usuário, o papel que ele desempenha na organização, as áreas de conhecimento em que ele trabalha, os assuntos que possui especialidade, e outros. Pode ser importante, por exemplo, saber se o especialista está disponível para interagir, se possui uma boa reputação, se o usuário já conhece o especialista recomendado ou se eles possuem amigos/colegas em comum.

A Figura 12 exibe uma visão geral do ICARE. O ICARE mantém uma *base de especialistas*. Usuários acessam o ICARE por meio de uma *interface de recomendação*, provêm um conjunto de palavras-chave e recebem como resposta uma lista de especialistas, classificada pela sua relevância para o problema que o usuário está enfrentando. Essa é a funcionalidade básica do ICARE (sem considerar contexto). Para melhorar as recomendações provendo especialistas que casem melhor com as necessidades do usuário, o ICARE utiliza uma *interface de contexto*. Contextos relacionados ao usuário e aos especialistas recomendados são considerados como complemento às palavras-chave.

Para modularizar a manipulação do contexto no ICARE, seguimos as atividades definidas no *CSS Design Process* (apresentado na seção 5.3), como descrito nas próximas seções.

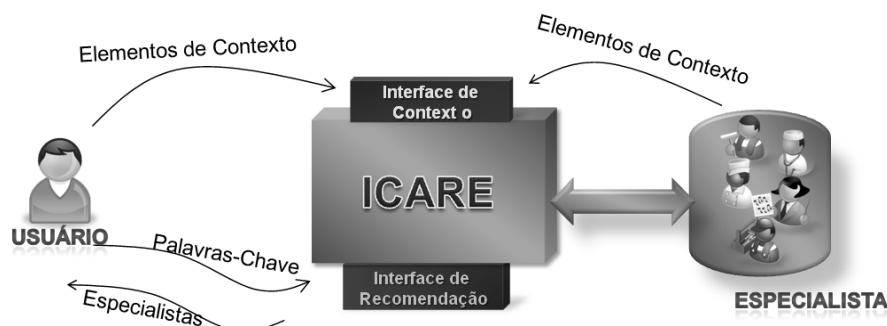


Figura 12. Visão Geral do ICARE

6.2 Fase 1: Especificação do Contexto

Esta seção descreve os passos realizados para especificar os elementos contextuais a serem usados no ICARE e os produtos gerados em cada passo.

6.2.1 S1: Identificar o Foco

Ao usar o ICARE, o usuário possui um único foco: a associação entre um agente *Usuário* e a tarefa *Buscar Especialistas*. Outros requisitos do ICARE (e.g. registrar perfil, gerar lista de especialistas, exibir lista) são complementares e visam apoiar a tarefa principal do foco. Um outro foco que poderia ser considerado seria a associação entre um agente *Sistema Externo* e a tarefa *Buscar Especialistas*, no caso do ICARE ser usado como um *plugin* associado a uma outra aplicação, a qual seria responsável por solicitar a recomendação. Entretanto, consideraremos apenas o primeiro foco.

6.2.2 S2: Identificar as Variações de Comportamento

As variações de comportamento no ICARE são relacionadas a adaptações na classificação dos especialistas de acordo com mudanças nos estados dos CEs. Dessa maneira, os seguintes requisitos relacionados a contexto foram identificados:

- R1: identificar os CEs associados ao usuário que está executando a busca;
- R2: utilizar os CEs do usuário para melhorar o casamento entre palavras-chave e as especialidades procuradas;
- R3: identificar os CEs para categorizar os especialistas;
- R4: identificar, para cada especialista recuperado, o valor dos seus CEs e incluí-los na lista gerada como resultado da busca (percepção do contexto);
- R5: comparar os CEs associados aos especialistas com os CEs identificados para o usuário, visando melhorar a classificação da lista de especialistas.

Essa lista de requisitos compõe o documento de *Requisitos de Contexto* especificado no processo como produto de saída para essa atividade.

6.2.3 S3: Identificar as Entidades Contextuais e CEs

As entidades contextuais consideradas no ICARE são: *Usuário* e *Especialista*. Os CEs associados a estas entidades são descritos a seguir. A Figura 13 ilustra o diagrama com o modelo conceitual de contexto do ICARE. O idioma inglês foi mantido no diagrama por compatibilidade com os componentes internos do ICARE.

- *disponibilidade* (Usuário, Especialista): indica o quão ocupado está o usuário ou o especialista. Um especialista disponível tem mais chances de aceitar participar de uma interação do que um especialista ocupado;
- *relacionamento* e *distanciaSocial* (Usuário, Especialista): indicam, respectivamente, um relacionamento social entre duas pessoas e o número de pessoas que socialmente as separam. De acordo com estudos existentes [Galegher et al. 1990], colegas e amigos possuem maior probabilidade de ajudar e interagir uns com os outros do que estranhos;
- *localizacaoAtual* (Usuário, Especialista): refere-se à localização física do usuário e do especialista. Sabendo que o especialista encontra-se no mesmo local que ele, o usuário pode escolher contactá-lo em uma interação face-a-face;
- *infoContato* (Especialista): informa como o especialista pode ser contactado. Sabendo quais são os meios de entrar em contato com o especialista, o usuário pode decidir se vai tentar uma interação com o especialista imediatamente (comunicação síncrona) ou em um momento mais oportuno (comunicação assíncrona);
- *trabalhaEm* e *nivelOrganizacional* (Usuário, Especialista): identificam, respectivamente, a relação profissional entre a pessoa e uma organização e a posição dessa pessoa na organização. Usuários em alto nível organizacional podem exigir o apoio do seu melhor especialista, independente do mesmo estar ocupado no momento. Por outro lado, usuários com baixa posição na

organização enfrentarão dificuldades em estabelecer interações de ajuda na execução de tarefas de especialistas que ocupem altos cargos na organização;

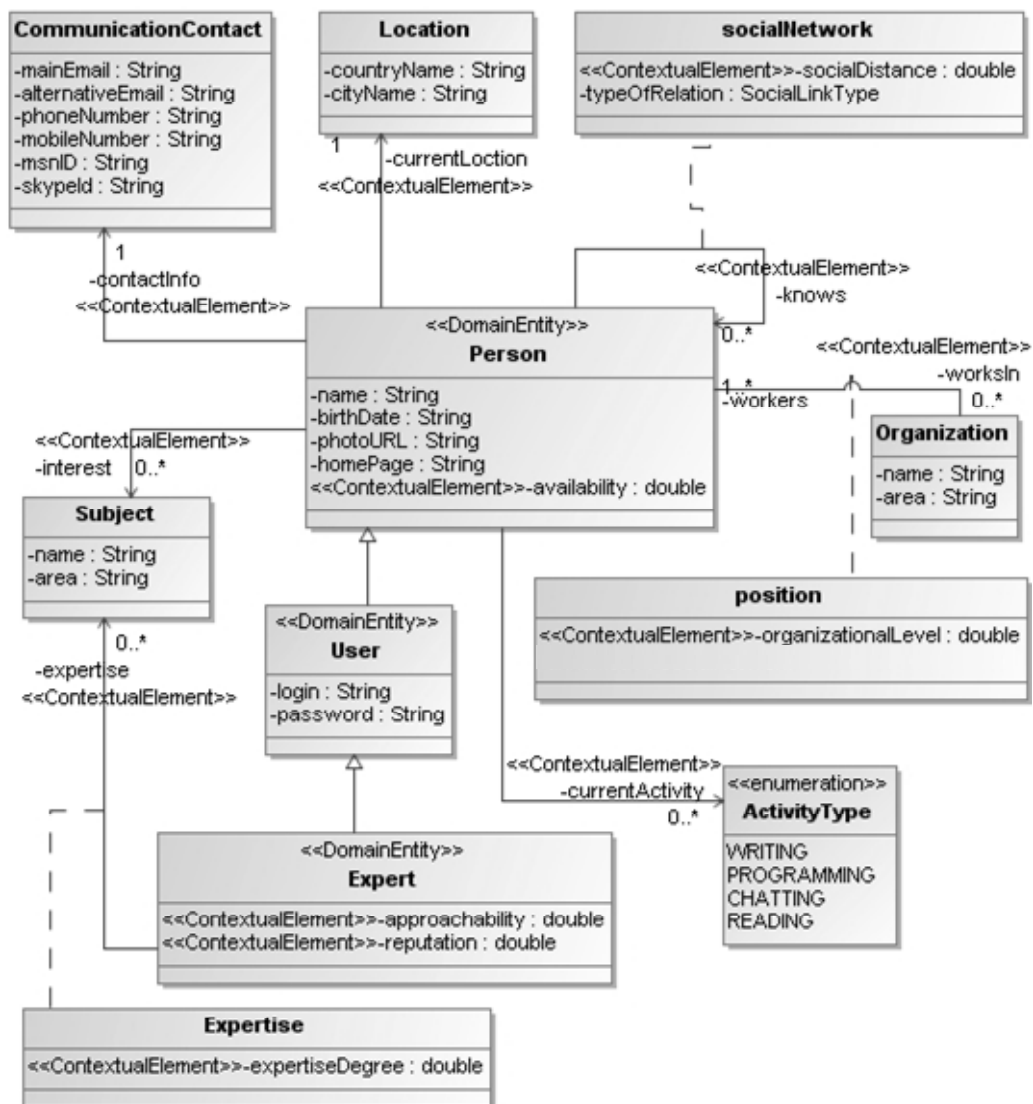


Figura 13. Modelo Conceitual de Contexto do ICARE

- *atividadeAtual* (Usuário, Especialista): representa a atividade que a pessoa está atualmente executando. Esse conhecimento pode ser utilizado para apoiar o mapeamento das palavras-chave informadas para a especialidade mais adequada e resolver ambiguidades semânticas entre palavras-chave informadas e especialidades conhecidas do sistema;
- *assuntosInteresse* (Usuário, Especialista): identifica assuntos em que a pessoa possui interesse. Pode auxiliar a identificação de afinidades mútuas entre usuários e especialistas;
- *especialidade* e *grauEspecialidade* (Especialista): indicam, respectivamente, uma relação de especialidade de uma pessoa em um dado assunto e o grau dessa especialidade;

- *acessibilidade* (Especialista): denota quão fácil é contatar o especialista. Por exemplo, um especialista que está frequentemente presente no ambiente, mesmo que esteja ocupado, é mais acessível do que um outro que nunca está presente;
- *reputação* (Especialista): indica o quão bem avaliado foi o especialista de acordo com julgamentos realizados por usuários que o contataram.

Essa identificação de Entidades e CEs é utilizada para enriquecer o modelo conceitual do ICARE, produzindo como artefato de saída da atividade o Modelo Conceitual de Contexto (Figura 13).

6.2.4 S4: Verificar a Relevância dos CEs

Para avaliar se os CEs identificados para o ICARE eram de fato adequados para classificar especialistas, foi realizado um experimento conduzido com 50 participantes de organizações de pesquisa e desenvolvimento de software. Um questionário foi aplicado apresentando os CEs identificados em S3 e as pessoas foram questionadas se elas considerariam aqueles CEs ao filtrar e classificar especialistas. Foi solicitado, ainda, que elas atribuísem uma classificação dos CEs por ordem de relevância. Os resultados desta investigação foram, de certa maneira, surpreendentes, mostrando que o que o projetista tinha considerado como relevante nem sempre era compatível com a opinião dos usuários. Esses resultados são apresentados e discutidos em [Petry et al. 2008].

6.3 Fase 2: Projeto do Gerenciamento de Contexto

A próxima atividade no processo é projetar os elementos arquiteturais de gerenciamento do contexto relacionados às atividades de *aquisição*, *processamento* e *disseminação* dos CEs. A atividade de disseminação do contexto não foi considerada, pois havia apenas um consumidor de contexto. As outras atividades são descritas nas próximas seções.

6.3.1 M1: Especificar a Aquisição do Contexto

Para a aquisição dos CEs no ICARE, foram identificadas três fontes de contexto externas e duas internas, como descritas a seguir:

- (1) *Plataforma Lattes*⁵, uma base de currículos e instituições das áreas de Ciência e Tecnologia no Brasil. Os CEs adquiridos a partir dessa fonte são: *especialidade*, *grauEspecialidade* e *assuntosInteresse*. O tipo de aquisição adotado é a *extração* dos CEs a partir de currículos exportados no formato XML, usando linguagens de consulta. A frequência de atualização desses CEs é *ocasional*, podendo ser atualizado em horários pré-determinados;
- (2) *GeoLite City*⁶, uma base de dados aberta que apóia a identificação de informações de localização (e.g. país, estado/região, cidade, latitude e longitude) a partir de endereços IP, em todo o mundo. Essa fonte informa o valor para o CE *localizacaoAtual*, sua periodicidade de atualização é *frequente* e o tipo de aquisição adotado é o *monitoramento* via sensor de IP;

⁵ <http://lattes.cnpq.br/english/index.htm>

⁶ <http://www.maxmind.com/app/geolitecity>

- (3) *Windows Live Messenger (MSN)*⁷, uma aplicação de mensagens instantâneas que permite que pessoas se comuniquem com outras, em tempo real. Provê valores para os CEs: *disponibilidade*, *atividadeAtual*, *relacionamento* e *distanciaSocial*. Os CEs *disponibilidade* e *atividadeAtual* são percebidos via monitoramento da interface de uso do MSN e mudam *frequentemente*. Os CEs *relacionamento* e *distanciaSocial* mudam apenas *ocasionalmente* e são atualizados por meio de consulta à base do servidor do MSN;
- (4) *Perfil do Usuário*, um formulário preenchido pelos usuários ao se registrar ao ICARE. Os CEs *infoContato*, *assuntosInteresse*, *trabalhaEm* e *nivelOrganizacional* são obtidos usando essa fonte de contexto. Sua frequência de atualização é *ocasional*.
- (5) *Histórico de Casos*, contém histórico de recomendações passadas e permite derivar os CEs *acessibilidade* e *reputação*. Cada caso de recomendação contém as palavras-chave informadas, os CEs identificados, a lista de especialistas retornada e um *feedback* do usuário indicando quão útil foi a recomendação recebida. Sua atualização é *frequente*.

A lista de fontes de contexto, com informações sobre os CEs que cada fonte é capaz de prover bem como detalhes sobre o tipo de aquisição e a frequência de atualização dos CEs constitui o documento de *Configuração da Aquisição* descrito no processo como artefato de saída dessa atividade.

6.3.2 M2: Projeto do Módulo de Aquisição

O próximo passo é projetar a arquitetura de aquisição e as interfaces de interação entre o ICARE e as fontes de contexto. A Figura 14 ilustra o Diagrama de Classes UML para o Módulo de Aquisição do ICARE.

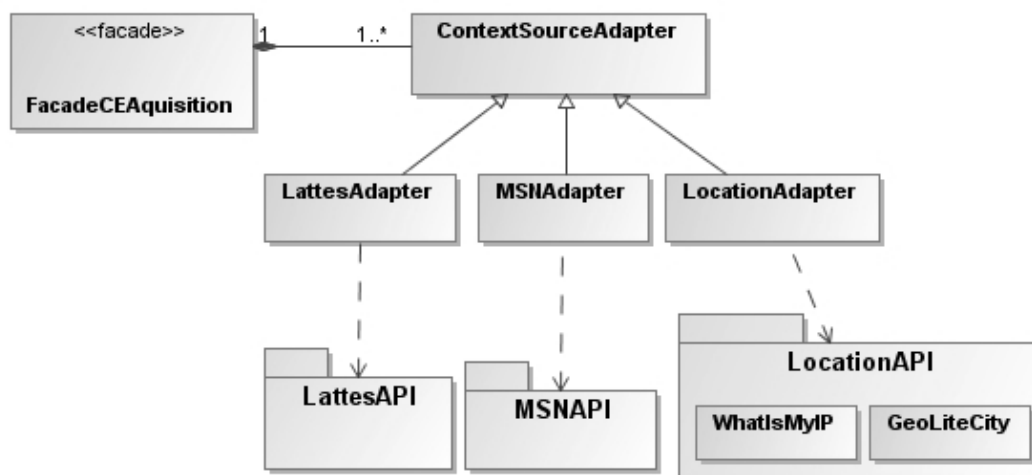


Figura 14. Diagrama de Classes UML para o Módulo de Aquisição do ICARE

De modo a isolar as funcionalidades internas do ICARE dos requisitos de aquisição do contexto foi utilizado o padrão de projeto Façade [Gamma et al. 1995].

⁷ <http://get.live.com/messenger/>

Com isso, mudanças nas fontes de contexto não impactam no seu uso pelo ICARE. A comunicação entre as fontes de contexto e o ICARE é feita com o *FacadeCEAcquisition*. Um supertipo *ContextSourceAdapter* foi definido para uniformizar o projeto de cada adaptador de fonte de contexto. Assim, para cada fonte de contexto externa foi criado um adaptador como instância da classe *ContextSourceAdapter* (*LattesAdapter*, *MSNAdapter*, e *LocationAdapter*).

6.3.3 M3: Projeto do Módulo de Processamento

Para o processamento dos elementos contextuais e das regras contextuais definidas no ICARE foi utilizada a máquina de inferência JEOPS [Figueira Filho e Ramalho 2000]. Para identificar as regras contextuais, foi utilizado o Weka [Waikato 2008]. O Weka recebeu como entrada os dados do questionário coletado na atividade S4, sendo identificados padrões de associação que permitiram atribuir pesos aos CEs. Um exemplo de regra contextual definido a partir do Weka é:

```
<condições>
    Usuario.nivelOrganizacional < 0.5
    Usuario.acessibilidade < 0.3
<ações>
    Especialista.pesoNivelEspecialidade = 0.8
    Especialista.pesoDistanciaSocial = 0.2
```

Essa regra indica que se o usuário ocupa uma baixa posição na organização e não é muito acessível, então a recomendação deve favorecer especialistas com alto grau de especialidade e dar menor importância à distância social entre usuário e especialista.

6.4 Fase 3: Projeto do Uso do Contexto

A fase seguinte é modelar como o contexto afeta o comportamento da aplicação.

6.4.1 U1: Projeto do Modelo de Comportamento

No ICARE, o contexto é utilizado para modificar o peso de relevância associado aos CEs usados para classificação dos especialistas. De acordo com as regras identificadas em M3, as condições associadas aos CEs são *disponibilidade* e *nivelOrganizacional*. O formalismo de grafos contextuais foi utilizado para modelar os diferentes caminhos associados a esses CEs. Cada caminho denota uma regra contextual, como por exemplo:

```
Regra1:
Condições
    Usuario.acessibilidade >= 0.7
    Usuario.nivelOrganizacional > 0.5
Ações
    CallBehavior ("Resolver Palavras-Chave")
    CallBehavior ("Procurar Especialistas")
    CallBehavior ("Atribuir peso Acessibilidade ALTO")
    CallBehavior ("Atribuir peso Especialidade ALTO")
    CallBehavior ("Calcular Fitness")
    CallBehavior ("Classificar por Fitness")
    CallBehavior ("Exibir Especialistas")
```

6.4.2 U2: Projeto de Adaptação ao Contexto

A classificação dos especialistas de acordo com o contexto é feita no ICARE por meio de uma função de aptidão (*Fitness*), como descrita abaixo. Essa função separa os elementos diretamente proporcionais daqueles inversamente proporcionais à adequação do especialista à busca do usuário. Por exemplo, quanto mais alta a reputação do especialista maior sua adequação à recomendação. Por outro lado, quanto menor a distância social entre o especialista e o usuário, mais alta a adequação do especialista. Para melhor adequar a recomendação às expectativas do usuário, cada CE é associado a um peso de relevância correspondente. Os pesos de relevância são ajustados usando as regras contextuais definidas em M3. A aptidão é computada em tempo de execução sobre CEs associados a cada especialista e ao usuário que solicita a recomendação.

$$Fitness(e,u) = \frac{\alpha_1 \times ed_e + \alpha_2 \times (ap_e + av_e) + \alpha_3 \times rep_e}{\alpha_4 \times socialDist(e,u) + \alpha_5 |OL_u - OL_e|}$$

Onde:

α_i = peso de relevância para cada CE;

ed_e = grau de especialidade;

ap_e = acessibilidade do especialista;

av_e = disponibilidade do especialista;

rep_e = reputação do especialista;

$socialDist(e, u)$ = distância social (especialista e usuário);

$|OL_u - OL_e|$ = diferença do nível organizacional do usuário (OL_u) e especialista (OL_e).

6.4.3 U3: Projeto de Apresentação do Contexto

Contexto também é utilizado no ICARE para ampliar a percepção do usuário que solicita a recomendação sobre a situação atual dos especialistas recomendados. A classificação dos especialistas de acordo com a heurística definida no ICARE pode ser muito restritiva. Assim, o usuário conhecendo o contexto atual do especialista pode, ele mesmo, definir que especialista é mais adequado para a sua consulta. Dessa forma, os seguintes CEs foram selecionados para serem apresentados aos usuários como resultado da recomendação: *infoContato*, *assuntosInteresse*, *reputação*, *disponibilidade*, *nivelOrganizacional*, *distanciaSocial*, *atividadeAtual* e *localizacaoAtual*.

7 Conclusões e Oportunidades para Pesquisa

Projetistas de sistemas computacionais têm percebido uma demanda na necessidade dos usuários de que os sistemas sejam mais flexíveis, adaptáveis, interativos e fáceis de usar. Sistemas que exijam demasiada intervenção do usuário para identificar informações e serviços relevantes tendem a perder espaço para aqueles capazes de identificar o usuário e seu contexto atual e oferecer serviços mais adaptáveis às suas necessidades e intenções. O uso do contexto não é primordial, muitas vezes, para o funcionamento de um sistema computacional. Em geral, os sistemas poderiam funcionar, normalmente, sem conhecer o contexto do usuário. No entanto, aplicações que acrescentem características de adaptação entre suas funcionalidades tendem a ser mais utilizáveis.

Atualmente, a utilização de contexto é um requisito fundamental em aplicações móveis e ubíquas, nas quais informações contextuais como a identificação do usuário, suas preferências, localização física (em constante alteração) e o dispositivo utilizado (que possuem diferentes limitações de recursos) são fundamentais para que essas

aplicações funcionem adequadamente. A Web, com seu dinamismo inerente e o excesso de informações publicadas, também exige que os sistemas sejam mais amigáveis e adaptáveis aos usuários. Além desses, diversos outros domínios da Computação percebem, também, a necessidade de utilizar o contexto para melhorar seus serviços aos usuários, como a Inteligência Artificial, Sistemas Colaborativos, Sistemas de Hipermissão Adaptativa e Interação Humano-Computador.

No entanto, compreender e identificar o contexto, e executar ações automaticamente de acordo com ele é uma tarefa não trivial em sistemas computacionais. Muitos são os desafios, relacionados à identificação do que considerar como contexto, sua aquisição, processamento e uso. Apesar da Computação Sensível ao Contexto ser uma área de pesquisa que vem crescendo bastante e atraindo a atenção e interesse de um número cada vez maior de pesquisadores, ainda existem muitos desafios que estão em aberto. Pesquisadores de diversas áreas estudam as melhores técnicas e métodos para suplantam as dificuldades inerentes a esses sistemas, porém ainda não há consenso sobre a melhor forma de desenvolver essas aplicações. Dentre esses desafios e perspectivas de pesquisa destacamos os seguintes⁸:

- *Representação do contexto*: um desafio a ser enfrentado no desenvolvimento de sistemas sensíveis ao contexto diz respeito à representação das informações contextuais. Neste quesito, três grandes problemas se apresentam: (i) a identificação das informações contextuais a serem consideradas; (ii) a determinação da técnica de representação a ser utilizada, levando em consideração fatores como interoperabilidade, extensibilidade, reuso e compartilhamento; e (iii) a interpretação semântica do contexto, para permitir que informações contextuais sejam trocadas e utilizadas por diferentes entidades, como agentes humanos e de software, dispositivos e serviços com uma mesma compreensão semântica.
- *Mecanismos de raciocínio*: informações contextuais monitoradas e adquiridas automaticamente demandam que mecanismos de raciocínio sejam utilizados para inferir informações de contexto relevantes a partir de um conjunto de dados contextuais básicos. Diversos trabalhos propõem diferentes métodos e abordagens para adicionar inteligência aos sistemas sensíveis ao contexto, como o raciocínio baseado em casos [Kofod-Petersen e Aamodt 2003]. Os desafios aqui estão relacionados à avaliação do método de raciocínio mais adequado para cada tipo de informação contextual, como tratar a complexidade e baixa maturidade dos motores de inferência existentes, e como tratar a incerteza e inconsistências nos elementos contextuais.
- *Mecanismos de aquisição automática*: com o avanço tecnológico dos dispositivos portáteis e seus sensores embutidos, sensores físicos vêm sendo cada vez mais adotados para adquirir, automaticamente, informações contextuais. Além disso, interfaces mais intuitivas, como assistentes inteligentes, podem estimular o usuário a prover informações de contexto como parte da sua tarefa atual. Assim, a implementação de mecanismos de aquisição automática de

⁸ A ordem de apresentação desses itens foi escolhida de forma aleatória e não por critério de relevância.

informações contextuais é um tópico desafiador e de extrema relevância para sistemas sensíveis ao contexto.

- *Mecanismos de privacidade e segurança*: sistemas sensíveis ao contexto, em geral, se baseiam em informações pessoais do usuário, tornando-as visíveis para outros usuários e sistemas. Com isso, é de vital importância que esses sistemas disponibilizem mecanismos que possam garantir ao usuário privacidade e, também, segurança sobre os seus dados. Se o usuário se sentir invadido ou ameaçado pelo sistema poderá se recusar a fornecer informações sobre si mesmo ou desabilitar mecanismos de monitoramento. É importante que o usuário tenha controle sobre a divulgação das suas informações, como estas serão utilizadas, quem terá acesso, e o que será feito delas.
- *Tratamento da Qualidade do Contexto*: uma vez que as informações contextuais podem vir de diversas fontes, muitas das quais podem ficar indisponíveis, elas possuem uma alta probabilidade de imperfeição ou inconsistência. Com isso, é necessário considerar parâmetros de qualidade do contexto (e.g. precisão, correteza, consistência, confiança, validade). Cada parâmetro deve ser descrito por uma ou mais métricas de qualidade apropriadas que definam como medir ou computar a qualidade do contexto em relação ao parâmetro. Os parâmetros de qualidade do contexto auxiliam a detecção e resolução de conflitos de contexto. Por exemplo, o contexto definido pelo usuário é mais confiável do que aquele percebido por um sensor ou deduzido por uma máquina de inferência.
- *Gerenciamento do histórico do contexto*: o aprendizado é um aspecto que deve ser considerado em sistemas sensíveis ao contexto. Para isso, o contexto histórico é um elemento essencial. É necessário armazenar, de forma eficiente, um grande volume de informação contextual e considerar o seu dinamismo. Por outro lado, a maioria das fontes de contexto é baseada em eventos e as aplicações podem estar interessadas em ser notificadas desses eventos. Adicionalmente, o acesso a essas informações deve ser simples e eficiente e deve permitir saber se o dado é atual ou não. Um desafio neste tipo de cenário é identificar se as demandas de armazenamento e acesso em sistemas sensíveis ao contexto são similares aos sistemas convencionais, se as mesmas abordagens podem ser utilizadas, quais as diferenças mais significativas e que estruturas de armazenamento e recuperação são mais eficientes.
- *Suporte para a provisão do contexto a dispositivos móveis*: informações contextuais relevantes devem ser fornecidas a aplicações executando em dispositivos com recursos limitados. O processamento da informação contextual pode demandar muitos recursos computacionais e, assim, não é ideal que seja feito no próprio dispositivo móvel, mas deve ser um serviço provido por um servidor externo. Com isso, a informação contextual pode ser compartilhada entre múltiplos dispositivos, reduzindo a sobrecarga do processamento do contexto. Como os dispositivos móveis possuem recursos limitados é importante investigar a melhor maneira de prover as informações de contexto a esses dispositivos, considerando essas limitações.
- *Tratamento de Desempenho*: gerenciar e manipular informações contextuais introduz um processamento extra às aplicações. É importante que esse

processamento não impacte no desempenho da aplicação, especialmente em domínios em que os recursos são limitados, como é o caso das aplicações móveis. Com isso, estudos que busquem melhorias no desempenho dos sistemas sensíveis ao contexto podem trazer grandes contribuições.

Como é possível perceber, muitos são os tópicos ainda em aberto nessa área, que demandam mais pesquisas, estudos e experimentos práticos. Este texto apresentou uma introdução ao conceito de contexto, sua aplicação a algumas áreas da Computação, bem como aspectos específicos de modelagem de contexto e de processos de desenvolvimento de sistemas sensíveis ao contexto. Esperamos que esse material sirva como referência básica àqueles interessados em iniciar seus estudos sobre Computação Sensível ao Contexto, que sua leitura motive e incentive o surgimento de novos pesquisadores, e que ele auxilie desenvolvedores de software a perceber as potencialidades do uso do contexto para tornar suas aplicações mais ricas, expressivas e adaptáveis.

Agradecimentos

Os autores agradecem ao CNPq e à CAPES pelo apoio financeiro e a primeira autora agradece à UFBA pelo apoio.

Referências Bibliográficas

- Alarcón, R., Guerrero, L. A., Ochoa, S. F., Pino, J. A. (2005) "Context in Collaborative Mobile Scenarios", In: Proc. of the Workshop on Context and Groupware, CONTEXT-05, Paris, France.
- Ayed, D., Delanote, D., Berbers, Y. (2007) "MDD Approach for the Development of Context-Aware Applications", In: Proc. of the 6th International and Interdisciplinary Conference on Modeling and Using Context (CONTEXT'07), LNAI 4635, pp. 15-28, Roskilde, Denmark.
- Bardram, J. E. (2005) "The Java Context Awareness Framework (JCAF) - A Service Infrastructure and Programming Framework for Context-Aware Applications", In: Proc. of the 3rd International Conference on Pervasive Computing, LNCS 3468, pp. 98-115, Munich, Germany.
- Bauer, J. (2003) "Identification and Modeling of Contexts for Different Information Scenarios in Air Traffic", *Master thesis*, Technische Universität Berlin, Fakultät IV - Elektrotechnik und Informatik, Institut für Computergestützte Informationssysteme, Berlin.
- Bazire, M., Brézillon, P. (2005) "Understanding Context Before Using It", In: Proc. of the 5th International and Interdisciplinary Conference on Modeling and Using Context (CONTEXT'05), LNAI 3554, pp. 29-40, Paris, France.
- Belian, R. B. (2008) "A Context-based Name Resolution Approach for Semantic Schema Integration", *Tese de Doutorado*, Centro de Informática, Universidade Federal de Pernambuco.
- Bellotti, V., Edwards, K. (2001) "Intelligibility and Accountability: Human Considerations in Context-Aware Systems", *Human Computer Interaction*, v. 16, n. 2-4, pp. 193-212.
- Belotti, R. (2004) "Sophie - Context Modelling and Control", *Diploma thesis*, Swiss Federal Institute of Technology Zurich.
- Borges, M. R. S., Brézillon, P., Pino, J. A., Pomerol, J.-C. (2004) "Bringing context to CSCW", In: Proc. of the 8th International Conference on Computer Supported Cooperative Work in Design (CSCWD'04), pp. 161-166, Xiamen, China.
- Bouquet, P., Ghidini, C., Giunchiglia, F., Blanzieri, E. (2003) "Theories and Uses of Context in Knowledge Representation and Reasoning", *Journal of Pragmatics*, v. 35, n. 3, pp. 455-484.

- Brézillon, P. (2007) "Context modeling: Task model and model of practices", In: Proc. of the 6th International and Interdisciplinary Conference on Modeling and Using Context (CONTEXT'07), LNAI 4635, pp. 122-135, Roskilde, Denmark.
- Brézillon, P. (1999) "Context in Artificial Intelligence: IA Survey of the Literature", *Computer&Artificial Intelligence*, v. 18, pp. 321-340.
- Brézillon, P., Araújo, R. M. (2005) "Reinforcing Shared Context to Improve Collaborative Work", *Revue d'Intelligence Artificielle*, v. 19, n. 3, pp. 537-556.
- Bulcão Neto, R. F. (2006) "Um processo de software e um modelo ontológico para apoio ao desenvolvimento de aplicações sensíveis a contexto", *Tese de Doutorado*, Instituto de Ciências Matemáticas e de Computação – ICMC-USP.
- Bunningen, A. (2004) "Context Aware Querying - Challenges for data management in ambient intelligence", *Internal Report*, University of Twente, In: <http://doc.utwente.nl/50087/> (Access in 04/2009).
- Cai, G., Wang, H., MacEachren, A. (2003) "Communicating Vague Spatial Concepts in Human-GIS Interactions: A Collaborative Dialogue Approach", In: Proc. of the Conference on Spatial Information Theory, LNCS 2825, pp. 304-319, Kartause Ittingen, Switzerland.
- Chaari, T., Ejigu, D., Laforest, F., Scuturici, V. (2007) "A comprehensive approach to model and use context for adapting applications in pervasive environments", *The Journal of Systems and Software*, v. 80, pp. 1973-1992.
- Chen, H., Finin, T. (2004) "An Ontology for Context-Aware Pervasive Computing Environments", *Knowledge Engineering Review*, v. 18, n. 3, pp. 197-207.
- Costa, P. D. (2007) "Architectural Support for Context-Aware Applications - From Context Models to Services Platforms", *CTIT Ph.D.-Thesis Series*, v. 07-108, n.021. Telematica Instituut Fundamental Research Series, Enschede, The Netherlands.
- de Bra, P., Aroyo, L., Chepegin, V. (2004) "The Next Big Thing: Adaptive Web-Based Systems", *Journal of Digital Information*, v.5, n.1, In:<http://jodi.tamu.edu/?vol=5&iss=1> (Access in 04/2009).
- Dey, A. K., Abowd, G. D. (2001) "A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications", *Human-Computer Interaction*, v. 16, n. 2-4, pp. 97-166.
- Figueira Filho, C., Ramalho, G. (2000) "JEOPS – Java Embedded Object Production System", In: Proc. of the 15th Brazilian Symposium on AI (SBIA'00), LNCS 1952, pp. 53-62, Atibaia, Brasil.
- Franklin, D. (2003) "The Representation of Context: Ideas from Artificial Intelligence", *Law, Probability and Risk*, v. 2, pp. 191-199.
- Fuchs, F., Hochstatter, I., Krause, M., Berger, M. (2005) "A Metamodel Approach to Context Information", In: Proc. of the PerCom Workshops 2005, pp. 8-14, Kauai Island, HI.
- Galegher, J., Kraut, R., Egido, C. (1990) "Intellectual Teamwork: Social and Technological Bases for Cooperative Work", Hillsdale, NJ: Lawrence Erlbaum Associates.
- Gamma, E., Helm, R., Johnson, R., Vlissides, J. (1995) "Design Patterns: Elements of Reusable Object-Oriented Software", Addison-Wesley, USA.
- Gauvin, M., Bourry-Brisset, A. C., Auger, A. (2004) "Context, Ontology and Portfolio: Key Concepts for a Situational Awareness Knowledge Portal", In: Proc. of the 37th Hawaii International Conference on System Sciences - Track 4, pp. 401-411, Hawaii, USA.
- Gonzalez, A. J., Brézillon, P. (2008) "Integrating two context-based formalisms for improved representation of human tactical behavior", *Knowledge Engineering Review*, v. 23, n. 2, pp. 1-21.
- Greenberg, S. (2001) "Context as a Dynamic Construct", *Human Computer Interaction*, v. 16, n. 2-4, pp. 257-268.

- Gross, T., Prinz, W. (2003) "Awareness in Context: A Light-Weight Approach", In: Proc. of the 8th European Conference on Computer-Supported Cooperative Work (ECSCW'03), pp. 295-314, Helsinki, Finland.
- Gu, T., Pung, H. K., Zhang, D. Q. (2005) "A Service-Oriented Middleware for Building Context-Aware Services", *Journal of Network and Computer Applications*, v. 28, n. 1, pp. 1-18.
- Gutwin, C., Greenberg, S. (2002) "A Descriptive Framework of Workspace Awareness for Real-Time Groupware", *Computer Supported Cooperative Work Journal*, v. 11, n. 3-4, pp. 411-446.
- Henricksen, K., Indulska, J. (2006) "Developing Context-Aware Pervasive Computing Applications: Models and Approach", *Pervasive and Mobile Computing Journal*, v. 2, n. 1, pp. 37-64.
- Henricksen, K., Livingstone, S., Indulska, J. (2004) "Towards a Hybrid Approach to Context Modelling, Reasoning and Interoperation", In: Proc. of the 1st International Workshop on Advanced Context Modelling, Reasoning and Management, pp. 54-61, Nottingham, UK.
- Hirschfeld, R., Costanza, P., Nierstrasz, O. (2008) "Context-Oriented Programming", *Journal of Object Technology*, v. 7, n. 3, pp. 125-151.
- Hong, J. I., Landay, J. A. (2004) "An Architecture for Privacy-Sensitive Ubiquitous Computing", In: Proc. of the 2nd International Conference on Mobile Systems, Applications, and Services, pp. 177-189, Boston, USA.
- Kofod-Petersen, A., Aamodt, A. (2003) "Case-based situation assessment in a mobile context-aware system", In: Proc. of the Workshop on Artificial Intelligence in Mobile Systems, pp. 41-49, Seattle, USA.
- Kokinov, B. (1999) "Dynamics and Automaticity of Context: A Cognitive Modeling Approach", In: Proc. of the 2nd International and Interdisciplinary Conference on Modeling and Using Context (CONTEXT'99), LNCS 1688, pp. 200-213, Trento, Italy.
- Korpiopää, P., Mäntyjärvi, J., Kela, J., Keränen, H., Malm, E. (2003) "Managing Context Information in Mobile Devices", *IEEE Pervasive Computing*, v. 2, n. 3, pp. 42-51.
- Kramer, R., Modsching, M., Schulze, J., Hagen, K. (2005) "Context-Aware Adaptation in a Mobile Tour Guide", In: Proc. of the 5th International and Interdisciplinary Conference on Modeling and Using Context (CONTEXT'05), LNCS 3554, pp. 210-224, Paris, France.
- Mark, G., Fuchs, L., Sohlenkamp, M. (1997) "Supporting Groupware Conventions through Contextual Awareness", In: Proc. of the 5th European Conference on Computer-Supported Cooperative Work (ECSCW'97), pp. 253-268, Lancaster, England.
- McCarthy, J., Buvac, S. (1997) "Formalizing Context (Expanded Notes)", *Computing Natural Language*, pp. 13-50, Stanford University.
- Moran, T. P., Dourish, P. (2001) "Introduction to the Special Issue on Context-Aware Computing", *Human Computer Interaction*, v. 16, n. 2, 3 & 4, pp. 87-95.
- Morse, D. R., Armstrong, S., Dey, A. K. (2000) "The What, Who, Where, When and How of Context-Awareness", In: <http://www-static.cc.gatech.edu/fce/contexttoolkit/pubs/CHI2000-workshop.pdf> (Access in 04/2009).
- Mylopoulos, J. (1992) "Conceptual modeling and Telos", *Conceptual modeling, databases, and CASE*, chapter 2, pp. 49-68, Wiley.
- Nunes, V. T., Santoro, F. M., Borges, M. R. S. (2007) "Capturing Context about Group Design Processes", In: Proc. of the 11th International Conference on Computer Supported Cooperative Work in Design (CSCWD'07), pp. 18-23, Melbourne, Australia.
- OMG (2007) "Unified Modeling Language: Superstructure, Version 2.1.2", In: <http://www.omg.org/spec/UML/2.1.2/> (Access in 04/2009).

- OMG (2008) "Software & Systems Process Engineering Meta-Model (SPEM) Specification v.2.0", In: <http://www.omg.org/technology/documents/formal/spem.htm> (Access in 04/2009).
- Palazzo, L. (2006) "Sistemas de Hipermídia Adaptativa", <http://ia.ucpel.tche.br/~lpalazzo/sha/sha.htm> (Acesso em 04/2009).
- Park, D., Hwang, S., Kim, A., Chang, B. (2007) "A Context-Aware Smart Tourist Guide Application for an Old Palace", In: Proc. of the 3rd International Conference on Convergence Information Technology, pp. 89-94, Gyeongju, Republic of Korea.
- Petry, H., Tedesco, P., Vieira, V., Salgado, A. C. (2008) "ICARE: A Context-Sensitive Expert Recommendation System", In: Proc. of the ECAI'08 Workshop on Recommender Systems, pp. 53-58, Patras, Greece.
- Power, R. (2003) "Topic Maps for Context Management", In: Proc. of the Workshop on Adaptive Systems for Ubiquitous Computing, pp. 199-204, Dublin, Ireland.
- Ranganathan, A., Al-Muhtadi, J., Campbell, R. H. (2004) "Reasoning about Uncertain Contexts in Pervasive Computing Environments", *IEEE Pervasive Computing*, v. 3, n. 2, pp. 62-70.
- Reichling, T., Schubert, K., Wulf, V. (2005) "Matching Human Actors Based on their Texts: Design and Evaluation of an Instance of the Expert Finding Framework", In: Proc. of the ACM SIGGROUP Conference on Supporting Group Work, pp. 61-70, Florida, USA.
- Schilit, B., Adams, N., Want, R. (1994) "Context-Aware Computing Applications", In: Proc. of the Workshop on Mobile Computing Systems and Applications, pp. 90-99, Santa Cruz, CA.
- Sheng, Q. Z., Benatallah, B. (2005) "ContextUML: A UML-Based Modeling Language for Model-Driven Development of Context-Aware Web Services", In: Proc. of the International Conference on Mobile Business, pp. 206-212, Sydney, Australia.
- Siebra, S., Salgado, A. C., Tedesco, P. (2007) "A Contextualised Learning Interaction Memory", *Journal of the Brazilian Computer Society*, v. 13, pp. 51-66.
- Simons, C., Wirtz, G. (2007) "Modeling context in mobile distributed systems with the UML", *Journal of Visual Languages and Computing*, v. 18, pp. 420-439.
- Souza, D., Belian, R. B., Salgado, A. C., Tedesco, P. (2008) "CODI - A Contextual Ontology for Data Integration", In: Proc. of the 4th Workshop on Ontologies-based Techniques for Databases (in VLDB'08), Auckland, New Zealand.
- Stefanidis, K., Pitoura, E., Vassiliadis, P. (2005) "On Supporting Context-Aware Preferences in Relational Database Systems", In: Proc. of the International Workshop on Managing Context Information in Mobile and Pervasive Environments, Ayia Napa, Cyprus.
- Strang, T., Linnhoff-Popien, C. (2004) "A Context Modeling Survey", In: Proc. of the Workshop on Advanced Context Modelling, Reasoning and Management, in 6th International Conference on Ubiquitous Computing, Nottingham/England.
- Swarts, L. (2003) "Why People Hate the Paperclip: Labels, Appearance, Behavior, and Social Responses to User Interface Agents", *B.Sc. Thesis*, Symbolic Systems Program, Stanford University, In: <http://xenon.stanford.edu/~lswartz/paperclip/> (Access in 04/2009).
- Vajirkar, P., Singh, S., Lee, Y. (2003) "Context-Aware Data Mining Framework for Wireless Medical Application", DEXA'2003 - Lecture Notes in Computer Science (LNCS), Volume 2736, Springer-Verlag, pp. 381-391, Prague, Tcheque, Republique.
- Vieira, V. (2008) "CEManTIKA: A Domain-Independent Framework for Designing Context-Sensitive Systems", *Tese de Doutorado*, Centro de Informática – UFPE, Brasil, Disponível em http://www.cin.ufpe.br/~vvs/cemantika/docs/2008_Thesis_VieiraV_CEManTIKA.pdf (Acesso em 04/2009).
- Vieira, V., Brézillon, P., Salgado, A. C., Tedesco, P. (2008) "A Context-Oriented Model for Domain-Independent Context Management", *Revue d'Intelligence Artificiel*, v. 22, n. 5, pp. 609-627.

- Vieira, V., Mangan, M. A. S., Werner, C. M. L., Mattoso, M. L. Q. (2004) "Ariane: An Awareness Mechanism for Shared Databases", In: Proc of the X International Workshop on Groupware (CRIWG'04), pp. 92-104, San Carlos, Costa Rica.
- Vieira, V., Souza, D., Salgado, A. C., Tedesco, P. (2006) "Uso e Representação de Contexto em Sistemas Computacionais", *Tópicos em Sistemas Interativos e Colaborativos*, Cap. 4, pp. 127-166, São Carlos: UFSCAR.
- Vieira, V., Tedesco, P., Salgado, A. C. (2005) "Towards an Ontology for Context Representation in Groupware", In: Proc. of the 11th International Workshop on Groupware (CRIWG'05), pp. 367-375, Porto de Galinhas, Brasil.
- Vieira, V., Tedesco, P., Salgado, A. C. (2009) "A Process for the Design of Context-Sensitive Systems", In: Proc. of the 13th International Conference on Computer Supported Cooperative Work in Design (CSCWD'09), pp. 143-148, Santiago, Chile.
- Vieira, V., Tedesco, P., Salgado, A. C., Brézillon, P. (2007) "Investigating the Specificities of Contextual Elements Management: The CEManTIKA Approach", In: Proc. of the 6th International and Interdisciplinary Conference on Modeling and Using Context (CONTEXT'07), LNAI 4635, pp. 493-506, Roskilde, Denmark.
- Waikato (2008) "WEKA - Waikato Environment for Knowledge Analysis", In: <http://www.cs.waikato.ac.nz/ml/weka/> (Access in 04/2009).
- Wang, X. H., Gu, T., Zhang, D. Q., Pung, H. K. (2004) "Ontology based context modeling and reasoning using OWL", In: Proc. of the Workshop on Context Modeling and Reasoning, Orlando, Florida.
- Weiser, M. (1991) "The computer for the 21st century", *Scientific American*, v. 265, n. 3, pp. 66-75.
- Zimmer, T. (2006) "QoC: Quality of Context – Improving the Performance of Context-Aware Applications", In: Proc. of the 4th International Conference on Pervasive Computing, pp. 209-214, Dublin, Ireland.
- Zimmermann, A., Lorenz, A., Oppermann, R. (2007) "An Operational Definition of Context", In: Proc. of the 6th International and Interdisciplinary Conference on Modeling and Using Context (CONTEXT'07), LNAI 4635, pp. 558-571, Roskilde, Denmark.