5

Grade Computacional como Infra-Estrutura para a Computação Pervasiva/Ubíqua

Professores:

Iara Augustin[1]
(august@inf.ufsm.br)
Giuliano Pereira Ferreira[2]
(giuliano@inf.ufsm.br)
Adenauer Yamin[3]
(adenauer@ucpel.edu.br)

Resumo:

A área de Computação Pervasiva ou Ubíqua pode ser considerada ou a mais complexa versão da computação distribuída ou a sua evolução. Ela objetiva integrar o mundo físico ao mundo virtual e alterar o foco para as atividades diárias dos usuários, criando uma computação invisível aos olhos de não-especialistas. O enorme crescimento no uso de dispositivos móveis e embarcados e suas interações com seres humanos produz uma quantidade significativa de dados, serviços e aplicações que necessitam ser processados a qualquer momento (anytime), em qualquer lugar (anywhere) e adaptados a qualquer dispositivo (anydevice), qualquer rede (anynetwork), dentro de parâmetros que refletem alta disponibilidade (always-

1 Graduação em Licenciatura em Matemática pela Universidade Federal de Santa Maria (1983), graduação em Administração de Empresas pela Universidade Federal de Santa Maria (1985), mestrado em Pós Graduação em Ciência da Computação pela Universidade Federal do Rio Grande do Sul (1993) e doutorado em Ciências da Computação pela Universidade Federal do Rio Grande do Sul (2004). Atualmente é pesquisadora do Conselho Nacional de Desenvolvimento Científico e Tecnológico (DT II) , Coordenadora do Programa de Pós-Graduação em Informática, nível de mestrado, e professora adjunta II da Universidade Federal de Santa Maria. Tem experiência na área de Ciência da Computação, com ênfase em Linguagens de Programação e Sistemas Distribuídos, atuando principalmente nos seguintes temas: computação ubíqua e pervasiva, computação móvel, computação sensível ao contexto e programação para telefones celulares.

2 Possui graduação em Ciência da Computação - Bacharelado pela Universidade Federal de Santa Maria (2006). Acadêmico do Mestrado em Computação do Programa de Pós-Graduação em Informática (UFSM). Atualmente é técnico de laboratório industrial da Universidade Federal de Santa Maria.

3 Possui graduação em Engenharia Elétrica pela Universidade Católica de Pelotas (1981) , especialização em Pós Graduação em Informática na Educação pela Universidade Federal de Pelotas (1990) , mestrado em Ciências da Computação pela Universidade Federal do Rio Grande do Sul (1994) e doutorado em Ciências da Computação pela Universidade Federal do Rio Grande do Sul (2004) . Atualmente é Professor Adjunto da Universidade Católica de Pelotas, Técnico de Nível Superior da Universidade Federal de Pelotas e Revisor de periódico da Revista do CCEI. Tem experiência na área de Ciência da Computação , com ênfase em Sistemas de Computação. Atuando principalmente nos seguintes temas: Computação Pervasiva, Computação em Grade, Computação Móvel, Arquitetura de Software, Middleware adaptativo e Aplicações Conscientes do Contexto.

on). Essa computação é baseada em dois conceitos primordiais: onipresença (ubiquidade da computação) e centralização no usuário-final. O grau de administração de sistemas ubíquos, para implementar as características de pró-atividade e mobilidade total, é alto. Para gerenciar o ambiente pervasivo, propõe-se um middleware orientado a serviço que integra características de grade computacional, computação consciente do contexto e computação pervasiva de maneira uniforme, e forma um tripé onde contexto é o vértice de união das outras duas áreas. Nosso propósito é o de disponibilizar a semântica siga-me (mobilidade total) para o usuário pervasivo e construir um ambiente pervasivo em larga-escala, usando a infra-estrutura de grade. Este texto aborda as questões envolvidas na união de conceitos de grade e computação ubíqua para criar uma infra-estrutura para a computação ubíqua em larga-escala e aplicações que executam em um longo período de tempo.

5.1. Introdução

Computação Ubíqua (*Ubiquitous Computing*) ou Pervasiva (*Pervasive Computing*) tem emergido como um paradigma dominante para a sociedade futura onde o mundo físico é integrado ao mundo virtual (digital). Esta visão, inicialmente proposta por Mark Weiser [Weiser 1991], representa a visão de uma sociedade na qual a computação é onipresente e suporta as atividades diárias do usuário. O enorme crescimento observado no uso de dispositivos móveis e embarcados e suas interações com seres humanos produz uma quantidade significativa de dados, serviços e aplicações que necessitam ser processadas a qualquer momento (*anytime*), em qualquer lugar (*anywhere*) e adaptadas a qualquer dispositivo (*anydevice*), qualquer rede (*anynetwork*) dentro de parâmetros que refletem alta disponibilidade (*always-on*). Projetando essa tendência para o futuro, vê-se uma explosão de dispositivos, objetos (*anything*) e sistemas (*anysystems*) interconectados e atuando de forma inteligente (*smart*) que podem tornar as atividades diárias mais fáceis e produtivas.

Nas últimas décadas, pesquisas e esforços da indústria têm sido devotados à visão de Weiser, objetivando torná-la realidade em um futuro próximo, os quais têm contribuído para o desenvolvimento de infra-estruturas e uma variedade de aplicações focando habilitar rapidamente o mundo ubíquo [Augustin et al 2004],[Garlan et al 2002],[Roman et al 2003],[Yamin et al 2003]. Essas tecnologias têm o potencial de redefinir a natureza das aplicações e o modo como se interage com elas e se usa as informações.

A próxima era da conectividade – conectividade do produto – enfatiza a visão de 'objetos inteligentes' (*smart things*) onde, provavelmente, bilhões de dispositivos ligados à Internet produzirão inteligência e conectividade para produtos comerciais ou industriais, incluindo os eletro-eletrônicos, de forma a estender a Internet para atender muitos aspectos da vida humana.

No entanto, dispositivos ubíquos freqüentemente têm capacidades limitadas de armazenamento, processamento e recursos computacionais. Assim, existe a necessidade de transferir a computação para fontes computacionais mais poderosas. Este fato dirige-nos para ver a Computação em Grade como um candidato natural para prover a infra-estrutura de gerenciamento do ambiente computacional, que executa em background (invisível ao usuário-final) a fim de obter a orquestração de todos os processos.

Analisando as pesquisas em andamento, a integração de grade e ubiqüidade/pervasividade tem sido definida sob dois pontos de vista. Primeiro, uso de Grade Pervasiva (UbiGrid), onde a grade, que é freqüentemente vista como uma plataforma para uma rede de volumosos recursos computacionais, auxiliando a preencher a demanda de experimentos científicos, está-se movendo em direção a tornar-se uma plataforma genérica para compartilhamento de vários tipos de recursos na rede [de Roure et al 2003]. Assim, a Grade Pervasiva explora dispositivos ubíquos ou como fonte de dados ou como uma interface de saída. Nessa linha, não existe um real interesse da comunidade de grade em ativamente contribuir para a pesquisa em Computação Pervasiva. Segundo, uso da grade como infra-estrutura de gerenciamento do ambiente pervasivo (UbiSpace) [Augustin 2004]. Em nossa pesquisa, usa-se a integração de conceitos da computação consciente do contexto, computação móvel e computação em grade na modelagem de uma plataforma comum para o gerenciamento de serviços e

recursos em um ambiente pervasivo em escala global. Para tal, foi desenvolvido um protótipo de um middleware, chamado EXEHDA (disponível na versão beta em http://exehda.wkit.com.br) para embasar essas idéias. A chave do trabalho é a adaptação consciente do contexto e a semântica siga-me (*follow-me*) para modelagem do middleware e das aplicações.

Este curso identifica questões importantes de pesquisa e dos desafios na criação da infra-estrutura para o ambiente pervasivo através dos conceitos e mecanismos da Grade Computacional e descreve alguns trabalhos realizados para atingir tal objetivo. Sua estrutura é: a seção 5.2 sumariza a origem da Computação Ubíqua em uma visão evolutiva da mobilidade; a seção 5.3 detalha o cenário atual da Computação Ubíqua, enquanto que a seção 5.4 tenta identificar as principais propriedades intrínsecas a um sistema ubíquo que o torna diverso dos demais sistemas móveis; a seção 5.5 analisa a inserção dos conceitos da computação pervasiva nas Grades Computacionais; por outro lado, a seção 5.6 analisa as questões embutidas na inserção dos conceitos de grade nos sistemas pervasivos; a seção 5.7 detalha a proposta de um middleware baseado na integração de grade, contexto e pervasividade para o gerenciamento do cenário pervasivo, com seus recursos, serviços e aplicações; finalmente, as conclusões são colocadas na seção 5.8.

5.2. A Origem da Computação Ubíqua

A Computação Ubíqua/Pervasiva é um novo paradigma computacional oriundo das tecnologias de rede sem fio e sistemas distribuídos, em um processo evolutivo iniciado pela Computação Nômade e seguido pela Computação Móvel, estágio atual da tecnologia móvel [Satyanarayanan 2001]. A idéia deste novo paradigma é a criação de um ambiente físico onde o foco é o ser humano, especificamente a tarefa que ele deseja realizar, permitindo assim ao usuário dedicar-se às questões de maior interesse, deixando o ambiente pervasivo responsável pela execução das tarefas secundárias.

Por ser uma área emergente de pesquisa, termos como computação ubíqua, computação pervasiva, computação nômade, computação móvel e outros tantos têm sido usados muitas vezes como sinônimos, embora sejam diferentes conceitualmente e empreguem diferentes idéias de organização e gestão dos serviços computacionais. À medida que a área evolui, esses conceitos vão sendo melhor compreendidos e as suas definições tornam-se mais claras e amplamente utilizadas.

5.2.1. Da Mobilidade à Ubiquidade

Os sistemas de computação móvel não estão ainda bem definidos e este termo é usado pelos autores em um espectro de ambientes, que envolvem alguma forma de mobilidade. De forma geral, pode-se dizer que "'sistema de computação móvel é um sistema distribuído que envolve elementos (software, dados, hardware, usuário) cuja localização se altera no curso da execução" [Augustin 2004]. Esta definição torna evidente a amplitude de abrangência desta nova área da computação.

Dependendo dos elementos que possuem a propriedade de mobilidade, podem-se definir diferentes cenários. Entre eles:

- Computação Nômade (*Nomadic Computing ou Palm Computing*) – popularizada com o uso de dispositivos portáteis, tais como os palmtops (a Palm era líder desse mercado) e suas aplicações de gerenciamento pessoal. O usuário pode

utilizar os serviços que um computador oferece independentemente da sua localização. A mobilidade está mascarada através da `portabilidade` do hardware (PDA – *Personal Digital Assistant*) e não é transparente. No início dos anos 90, as facilidades de comunicação eram, basicamente, via acesso discado; a cada movimentação, uma nova conexão à rede era requerida;

- Computação via Redes Sem Fio (*Wireless Computing*) - usuário usando um equipamento portátil pode se mover dentro de uma área de acesso, enquanto mantém a conexão à rede fixa (infra-estruturada) ou à rede espontânea (ad-hoc) que se forma pelo encontro de dispositivos;
- Mobilidade de Código *(Mobile Computation)* - os componentes da aplicação podem se mover. Pode-se ter: a mobilidade de código; a mobilidade de dados; ou a mobilidade de todo o estado da execução da aplicação (por exemplo: agentes móveis);
- Computação Móvel (*Mobile Computing*) – a Computação Nômade, combinada com a capacidade de acesso permanente à rede sem fio, tem transformado a computação numa atividade que pode ser carregada para qualquer lugar. Observa-se que a crescente introdução de facilidades de comunicação tem deslocado as aplicações da computação móvel de uma perspectiva de uso pessoal para outras mais avançadas e de uso corporativo, como as aplicações móveis distribuídas;
- Computação Pervasiva (*Pervasive Computing*) - Nesta concepção, o computador tem a capacidade de obter informação do ambiente no qual ele está embutido e utilizá-la para dinamicamente construir modelos computacionais, ou seja, controlar, configurar e ajustar a aplicação parar melhor atender as necessidades do dispositivo ou utilizador (adaptação consciente do contexto). O ambiente também pode e deve ser capaz de detectar outros dispositivos que venham a fazer parte dele. Desta interação surge a capacidade dos sistemas agirem de forma "esperta" no ambiente no qual o usuário se move, um ambiente povoado por sensores e serviços computacionais;
- Computação Ubíqua *(Ubiquitous Computing)* – o ambiente é impregnado de dispositivos (móveis ou fixos) e equipamentos computacionais conectados entre si e invisíveis ao usuário final. O usuário dispõe de seu ambiente computacional independente de localização, tempo, dispositivo e rede subjacente. Surge dos avanços da computação móvel e da computação pervasiva, e da necessidade de integrá-las. Isto significa que, qualquer objeto computacional (presente no ambiente ou trazido pelo usuário) pode construir dinamicamente modelos computacionais dos ambientes entre os quais o usuário se move e configurar os seus serviços dependendo da necessidade e da tarefa que o usuário deseja realizar (*task-driven computing*).

Muitos pesquisadores consideram que Computação Pervasiva, termo cunhado pela IBM (2000), e Computação Ubíqua, proposto por Mark Weiser – XEROX Parc (*Ubiquitous Computing*) [Weiser 1991], como sinônimos [Satyanarayanan 2001]. Observando os trabalhos realizados por pesquisadores, a maioria destes usam os termos de forma indistinta. Desta forma, neste texto esses dois termos são usados de forma indistinta.

Sistemas distribuídos tradicionais são construídos com suposições sobre a infra-estrutura física de execução, como conectividade permanente à rede fixa e

disponibilidade dos recursos necessários. Porém, essas suposições não são válidas nos sistemas móveis. Isto impede o uso direto das soluções adotadas pelos sistemas distribuídos, as quais podem ser altamente ineficientes devido à variabilidade freqüente da conexão à rede e da disponibilidade de recursos e serviços.

Parece, portanto, ser necessário definir uma nova arquitetura para sistemas móveis, projetada com mobilidade, flexibilidade e adaptabilidade intrínsecas. A produção de software no ambiente móvel é, ainda, complexa. Seus componentes são variáveis no tempo e no espaço em termos de conectividade, portabilidade e mobilidade. O desafio que se apresenta é, portanto, projetar aplicações móveis distribuídas cujos níveis de serviço e disponibilidade de recursos são imprevisíveis. Emergem, portanto, novos requisitos para o desenvolvimento de aplicações, os quais geram uma nova classe de aplicações projetadas especificamente para este ambiente dinâmico.

Esta nova classe de aplicações tem sido referenciada de muitas formas: *environment-aware, network-aware, resource-aware, context-aware applications*. Porém, todas têm um conceito embutido: adaptação ao contexto. Isto significa que os sistemas devem ter consciência da localização e da situação onde estão inseridos, e devem tirar vantagem desta informação para (auto-configurar-se dinamicamente de um modo distribuído. A diferença entre as aplicações está no grau de adaptabilidade e nos recursos que são objetos dessa adaptação. O foco da complexidade na implementação dessas aplicações móveis, com comportamento adaptativo, está no fato de que os componentes distribuídos das mesmas podem sofrer influência dos diversos ambientes onde estão inseridos.

5.2.2. Computação Móvel: o momento atual

O atual momento da área de sistemas móveis é a Computação Móvel, enquanto que a Computação Pervasiva/Ubíqua é ainda uma promessa futura. A Computação Móvel pode ser caracterizada por três propriedades: portabilidade, mobilidade, e conectividade [Augustin et al 2002] que introduzem restrições no ambiente. Para ser portátil, um computador deve ser pequeno, leve e requer fontes pequenas de energia. Isto significa que um computador portátil tem restrições no tamanho de memória, na capacidade de armazenamento, no consumo de energia e na interface do usuário. Além disso, a portabilidade potencializa o risco de perda, queda ou roubo. Quando em movimento, o dispositivo móvel pode alterar sua localização e, possivelmente, seu ponto de contato com a rede fixa. Essa natureza dinâmica do deslocamento introduz questões relativas ao endereçamento dos nós, localização do usuário e informações dependentes da localização. Além da mobilidade física, a aplicação com seu código, dados e estado também pode se deslocar entre os nós da rede. A conexão à rede através do meio sem fio levanta outros obstáculos: comunicação intermitente (desconexões freqüentes, bloqueio no caminho do sinal, ruído), restrita (e altamente variável) largura da banda, alta latência e alta taxa de erros.

Outro requisito importante a considerar no projeto de aplicações móveis é o grau de conectividade à rede. Estudos, a partir dos sistemas de arquivos, têm demonstrado que existem basicamente três modos de operação de um sistema móvel [Satyanarayanan 2001]: fortemente conectado – conexão sobre uma rede fixa, rápida e confiável; fracamente conectado – conexão sobre um canal sem fio com largura de banda restrita; e desconectado – sem conexão à rede. O modo desconectado é eletivo[1] e utilizado, principalmente, para economia de recursos do *host* móvel, como o consumo de energia.

[1] Termo introduzido por Dan Duchamp, significando que o *host* pode informar antecipadamente ao sistema que ocorrerá a desconexão, sendo que este pode executar um "protocolo de desconexão".

Observa-se que o *host* móvel estará a maior parte do tempo desconectado da rede, sendo a desconexão um modo normal no ambiente móvel, enquanto que no ambiente distribuído é uma exceção. Desta forma, o sistema deve prover uma "ilusão" de conexão para o usuário móvel.

Três outros requisitos existentes em Sistemas Distribuídos assumem uma maior relevância quando (alta) mobilidade está presente. São eles: escalabilidade, tamanho do conjunto potencial de usuários atendidos em um determinado ponto; heterogeneidade, introduzida por diferentes equipamentos e redes móveis; e dinamismo introduzido pela alta variabilidade da disponibilidade de recursos e pela mobilidade do usuário. Esses aspectos não são isolados, e devem ser abordados na arquitetura do sistema móvel.

Como visto, restrições são da natureza da mobilidade e colocam novas demandas no projeto de aplicações, as quais devem ser mais flexíveis que as atuais aplicações distribuídas quando um recurso está indisponível/inacessível ou tem seu nível de disponibilidade/acessibilidade reduzido. Para serem efetivos e apresentarem um desempenho compatível com a expectativa do usuário, essas aplicações devem exibir a capacidade de **adaptação** às freqüentes e rápidas alterações no ambiente de execução durante o curso de evolução da aplicação.

As soluções para ambientes distribuídos tradicionais (redes fixas) não são apropriadas, pois não permitem uma adaptação dinâmica em face às alterações do contexto de execução. Além disso, a maioria das soluções adaptativas atuais satisfazem as necessidades de aplicações específicas, sendo insuficientes para serem reusadas em aplicações de propósito geral.

A Figura 1 ilustra a arquitetura de gerenciamento da mobilidade. A rede de acesso pode incluir redes de telefonia celular, redes de comunicação pessoal, redes sem fio, numa dimensão de macro a pico células. A rede *backbone* inclui várias redes de alta velocidade, como FDDI e ATM, que podem estar conectadas à Internet. O gerenciamento da mobilidade, inicialmente desenvolvido para suporte a mobilidade de usuário e terminal, é estendido para gerenciar mobilidade de serviços e recursos.

5.2.3. Futuro com a Computação Ubíqua

A Computação Ubíqua também é denominada Tecnologia Tranquila (*Calm Tecnhology*), Inteligência Ambiente (*Inteligence Ambient*), Computação Pró-ativa (*Proactive Computing*), Internet dos Objetos (*Internet of Things*) e Computação Invisível (*Invisible Computing*) entre outros nomes. Porém, os termos que têm predominado são Computação Pervasiva[2] e Computação Ubíqua.

Em um espaço pervasivo/ubíquo (também chamado *smart space*), computadores e outros (vários e variados) dispositivos digitais estão totalmente integrados ao ambiente do usuário e objetivam auxiliá-lo em suas tarefas diárias. Este é um ambiente altamente dinâmico e heterogêneo. Os recursos, incluindo serviços, dispositivos e aplicações, disponíveis podem alterar-se rapidamente. Diferentes espaços têm diferentes tipos de recursos disponíveis e diferentes políticas de uso dos recursos. Programas executando neste ambiente devem ser capazes de se adaptar àtroca do contexto e disponibilidade de recursos. Isto coloca um desafio para os desenvolvedores que devem especificar como o programa deve se comportar em diferentes contextos e quando diferentes tipos de recursos estão disponíveis. Além disso, diferentes espaços pervasivos

[2] O termo 'pervasivo' (espalhado, integrado, universal) não existe ainda na Língua Portuguesa. Alguns autores consideram que se deve usar somente o termo Computação Ubíqua, pois este existe na Língua Portuguesa e significa onipresente.

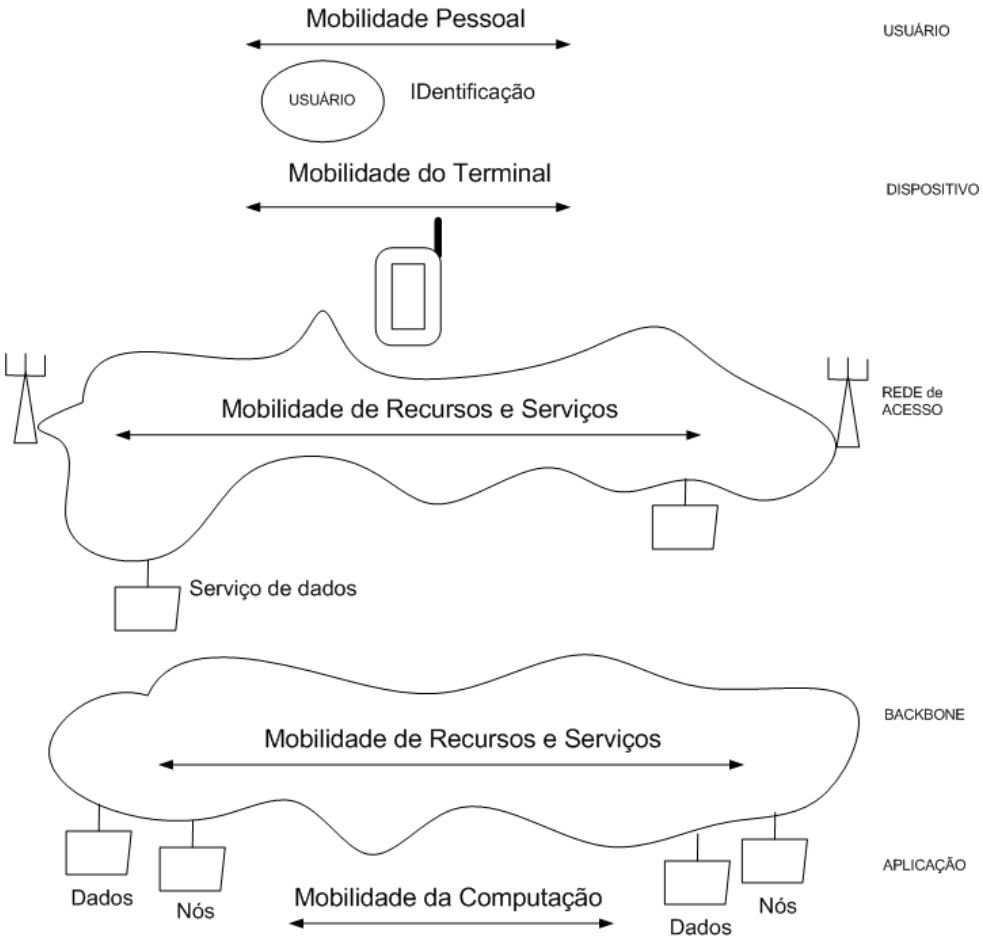

Figura 1. Tipos de Mobilidade

podem ter diferentes modos de executar a mesma tarefa uma vez que têm variados serviços, aplicações e recursos. O desenvolvedor não pode esperar saber antecipadamente como as várias tarefas serão executadas nos diferentes espaços pervasivos. Assim, programadores necessitam de abstrações de alto nível para programar aplicações no espaço pervasivo sem ter que ter consciência dos recursos disponíveis, contexto, políticas e preferência dos usuários [Augustin 2004].

Ambientes para a Computação Pervasiva têm sido explorados através de uma série de protótipos de pesquisa na academia, tais como o projeto Aura [Garlan and Stenskiste and Schmerl 2002], projeto Gaia [Roman et al 2002], projeto Oxigen do MIT (www.oxygen.lcs.mit.edu); a indústria também tem dado atenção a essa área, tais como IBM Pervasive Computing Unit (www.research.ibm.com/thinkresearch/ pervasive.shtml). Muitos enfatizam os requisitos de tecnologia e a previsibilidade da interação e comportamento do ambiente.

Pela integração de sensores, computadores, dispositivos e redes foi possível desenhar a **primeira geração de ambientes pervasivos**, referenciados como 'ambientes integrados'. Agora os esforços de pesquisa concentram-se em deslocar o paradigma de 'ambientes integrados' para 'espaços programáveis'. O grande desafio é que a

Computação Pervasiva afeta toda a Ciência da Computação em três distintas perpectivas: da experiência, da engenharia e teórica [Chalmers 2006].

Modelos de programação e middlewares baseados no modelo de sensores-atuadores-contexto foram os mais focados para a programação de aplicações da primeira geração e resultaram no conceito de Computação Orientada a Contexto (*context-aware programming*). Busca-se, agora, encontrar abstrações de alto nível que permitam programar aplicações que comporão um novo paradigma denominado Programação Orientada a Tarefas (*Task-Oriented Programming*).

5.3. O Cenário Ubíquo Atual

Dado o contínuo progresso técnico em comunicação e computação, parece que se está caminhando a uma total integração da computação nas atividades humanas. Previsões indicam que em poucos anos, microprocessadores se tornarão pequenos e baratos o suficiente para serem embutidos em quase tudo – não somente em dispositivos digitais, carros, eletroeletrônicos, brinquedos, ferramentas, mas também em objetos (lápis, por exemplo) e roupas. Todos esses artefatos devem estar interlaçados e conectados em uma rede sem fio.

De fato, a tecnologia espera uma revolução na qual bilhões de pequenos e móveis processadores estejam incorporados ao mundo físico, compondo objetos 'espertos' (smart[3]) – sabem onde estão, se adaptam ao ambiente e fornecem serviços úteis em adição ao seu propósito original, formam redes espontâneas (ad-hoc) e altamente distribuídas, numa ordem de magnitude muito maior que a de hoje.

Esse cenário está sendo considerado como o novo paradigma do século 21 [Saha and Mukherjee 2003],[Satyanarayanan 2001] ou a terceira onda da computação [Jansen et al 2005], o qual permite o acoplamento do mundo físico ao mundo da informação e fornece uma abundância de serviços e aplicações onipresentes visando que usuários, máquinas, dados, aplicações e objetos do espaço físico interajam uns com os outros de forma transparente (em background) [Ranganathan et al 2005].

É claro que se está movendo gradualmente em direção à visão de uma computação onipresente/ubíqua. Incrementalmente, está-se acostumando a usar uma coleção de heterogêneos dispositivos (*personal computer*) para suportar uma crescente faixa de atividades. A corrente geração de dispositivos interconectados é somente o ponto de partida em direção à computação ubíqua [Chalmers 2006].

Para construir-se o cenário visualizado pela Computação Pervasiva/Ubíqua é necessário uma pesquisa multidisciplinar envolvendo, praticamente, todas as áreas da computação: sistemas distribuídos, sistemas móveis, redes de sensores, banco de dados, inteligência artificial, interface homem-computador, segurança, rede, etc.

5.3.1. Requisitos e Desafios das Aplicações Pervasivas

Um exemplo de ambiente pervasivo é o hospitalar. Neste ambiente, algumas atividades são previsíveis e planejadas enquanto outras são randômicas, as atividades variam de simples a complexas, algumas atividades têm prioridade enquanto outras podem ser feitas quando houver tempo, algumas atividades são ligadas a determinadas salas e presença de certos artefatos. O trabalho dos clínicos é extremamente móvel e estes não podem carregar equipamentos pesados.

[3] dispositivo muitos em um (many-in-one device).

Logo, é interessante no ambiente o conceito de 'computador público' que não armazena atividades computacionais de ninguém, mas serve como um portal para acesso a elas. Este conceito requer uma infra-estrutura que gerencia, armazena e distribui atividades computacionais. Por exemplo, o usuário é identificado e autenticado no sistema por 'proximidade' – procedimento deve ser rápido, e o computador público recebe o ambiente virtual deste usuário para que ele possa desenvolver suas tarefas/atividades.

Outra propriedade necessária é a inferência pró-ativa das atividades baseada na localização da pessoa e artefatos ao redor. O trabalho dos clínicos é também altamente colaborativo por natureza – o atendimento a um paciente envolve, em geral, várias especialidades. Colaboração significa interromper a tarefa em execução para atender a solicitação por demanda (chaveamento de atividades).

O ambiente também requer o acesso a um conjunto de variadas e atualizadas informações, que podem ser requeridas por vários clínicos ao mesmo tempo. Uma organização de dados e acesso pervasivo a ele é requerida. Dispositivos móveis devem se comunicar com a infra-estrutura disponível, numa organização de rede infra-estruturada, ou descobrir novos dispositivos, numa organização de rede *ad-hoc* ou *mesh*.

Como o cenário pervasivo prevê uma mobilidade física (dos equipamentos e/ou dos usuários) e lógica (componentes da aplicação e serviços), potencialmente em escala global (larga-escala), deve fornecer transparência ao usuário, de forma que o usuário possa acessar seu ambiente computacional independente de localização, do meio de acesso e do tempo. O sistema de suporte para esse ambiente usa a metáfora de um ambiente virtual do usuário, onde as aplicações têm o estilo "siga-me" (*follow-me applications*) [Augustin et al 2005]. Soluções integradas (Figura 2) para disponibilizar este ambiente é o foco da primeira geração de sistemas pervasivos, dos quais pode-se citar o projeto ISAM (Infra-estrutura de Suporte às Aplicações Móveis Distribuídas – www.inf.ufrgs.br/~isam) .

5.3.2. Panorama Geral da Primeira Geração de Sistemas Pervasivos

Pesquisadores têm, recentemente, desenvolvido vários sistemas e protótipos de Computação Pervasiva para demonstrar como este novo paradigma pode beneficiar alguns domínios de aplicações, como segurança de residências (*home security*), educação (*pervasive learning*), saúde e emergências (*smart hospital*), casa virtual (*smart home*), e entretenimento.

Pode-se dizer que as estratégias adotadas dividem-se em:

- projeto de aplicações experimentais;
- desenvolvimento de sistemas genéricos experimentais;
- análise teórica, como o modelo Mobile Ambients [Cardelli and Gordon 1998] e a formalização do conceito de contexto [Jansen et al 2005].

Na maioria dos casos, a abordagem adotada é a integração de sistemas povoados com diversos dispositivos computacionais heterogêneos (*appliances*), incluindo sensores, atuadores, computadores, micro-controladores, etc. usando diversas redes e conectores. O objetivo desses sistemas é demonstrar a habilidade de envolver novas tecnologias emergentes e de entender os requisitos das aplicações. É uma estratégia com interesse experimental; não, necessariamente, prático. Dentre as diversas questões de

pesquisas em progresso, as seguintes têm sido mais abordadas: redes móveis, redes de sensores e middlewares.

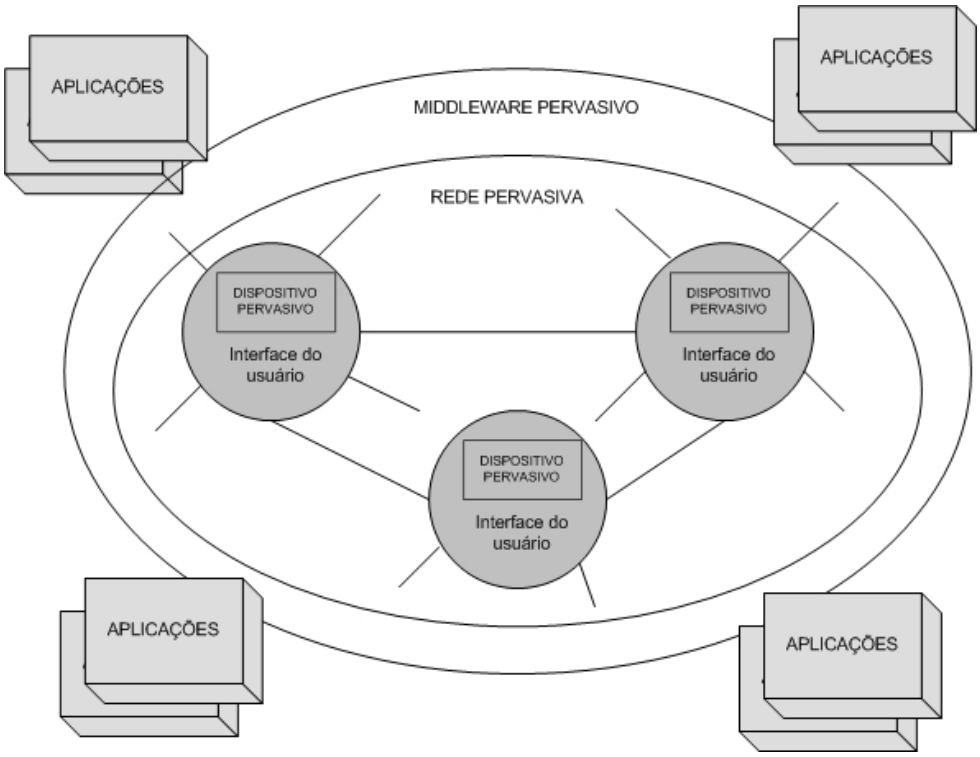

Figura 2. Espaço Pervasivo (versão, retirada de [Saha and Mukherjee 2003])

5.3.3. Tecnologias Disponíveis

Os principais componentes físicos da Computação Móvel envolvem os computadores portáteis, usados pelos usuários móveis, e as redes sem fio, com seus respectivos meios de transmissão de dados e padrões de comunicação de dados. Além disso, considerando que a Internet é o principal meio de comunicação usado hoje e que o uso de sua infra-estrutura é aplicável aos trabalhadores móveis, estão surgindo propostas de adequação da arquitetura TCP/IP à mobilidade, dentre as mais conhecidas está o protocolo WAP (*Wireless Application Protocol*).

5.3.3.1. Dispositivos Móveis e Convergência Tecnológica

Os dispositivos para o usuário final da Computação Móvel têm várias formas e configurações. As plataformas de *hardware*, o sistema operacional e as capacidades funcionais variam entre esses dispositivos. Entretanto, existem atributos que são compartilhados entre *notebooks*, *palmtops*, computadores *handhelds* e outros acessórios, tais como impressoras, fax e *scanners* móveis. Os **computadores portáteis** devem ser de tamanho reduzido e de pouco peso. Precisam ser resistentes, funcionais, práticos e portáteis. O desenvolvimento de computadores móveis deve conciliar objetivos que são conflitantes: oferecer recursos e desempenho semelhante aos do *desktop* com tamanho, peso e consumo de energia reduzidos. Também devem oferecer acesso remoto a outros computadores, servidores e *mainframes*, comunicação em rede e suporte a uma crescente variedade de mídias.

Apesar de os *notebooks* serem os pioneiros na Computação Móvel, a popularização de transportar um computador para qualquer lugar veio com o lançamento da primeira versão do Palm, da *Palm Computing* (www.palm.com), chamado de PalmPilot, ocorrido em 1996. Desde então uma crescente evolução na capacidade e funcionalidade destes dispositivos tem sido observada.

Outra linha de dispositivos usados é a de **telefones celulares**. A 2,5ª geração dos celulares, conhecida como *Personal Communciations Services* (PCS), dispõe de serviços que facilitam a comunicação e permitem ao usuário executar certas funções, como enviar mensagens, e o acesso à Internet móvel com o protocolo WAP (http://www.wapforum.org). A 3ª geração de telefones celulares permitirá aplicações de multimídia e videoconferência, embora ainda não contemple *roaming* global.

Na década de 1990, acompanhando o avanço das redes de telecomunicações, tornou-se mais comum falar em convergência tecnológica ou convergência digital. A popularização da Internet foi um passo fundamental para que o conceito se difundisse, principalmente fora dos meios corporativos. Os primeiros internautas não dispunham de recursos adequados para obter a qualidade esperada em serviços convergentes, pois a maioria dependia de conexões discadas por enlaces analógicos sobre par-de-cobre. O usuário doméstico comum só começou a beneficiar-se da convergência com a adoção em massa de conexões de banda larga (xDSL), que pela primeira vez forneceram, a um custo acessível, capacidade de transmissão suficiente para utilização de serviços, tais como Voz sobre IP (VoIP).

Com a consolidação da Internet como a mais importante rede de informações do mundo globalizado, também se estabeleceram os padrões tecnológicos que ela emprega, tais como o protocolo IP e a comutação de pacotes. Esses elementos, aliados ao barateamento e aprimoramento dos meios de transmissão em banda larga, crescente demanda por serviços multimídia, criação de novos protocolos como o SIP e de mecanismos como MPLS, estão dando forma à arquitetura de redes convergentes que vem sendo chamada de *Next Generation Networking*.

O ponto de partida para o fenômeno da convergência tecnológica é, evidentemente, a viabilidade de desenvolvimento e comercialização em grande escala de soluções de tecnologia convergentes, sejam redes, serviços ou terminais.

A convergência atual de tecnologias de distribuição de voz, dados, imagens e sons através da digitalização da informação passa por diversas instâncias, seja a convergência de equipamentos de comunicação, telecomunicações e informática; a convergência dos modelos de consumo de informação, entre comunicação de massa e comunicação interativa; a convergência dos produtos das indústrias culturais em um único produto multimídia; e a convergência da economia das comunicações que agrupa dois setores distintos – telecomunicações e comunicação eletrônica de massa – mediados pela informática.

Convergência de redes

É a unificação entre duas ou mais redes de comunicação distintas numa única rede capaz de prover os serviços antes prestados pelas diversas redes. Um dos primeiros exemplos é a convergência entre redes de voz e dados, inicialmente através de tecnologia RDSI e, mais recentemente, pela tecnologia xDSL.

Ultimamente, aos serviços de voz e dados tem-se incluído serviços de vídeo e/ou multimídia. Muitos desses serviços não existiam antes de se começar a falar em convergência de redes, por isso pode-se dizer que já "nasceram convergentes", como

IPTV (que é diferente de simplesmente enviar a transmissão da televisão analógica tradicional por protocolo IP). A oferta combinada de serviços de voz, Internet banda larga e televisão recebe o nome de *Triple play*, esse termo tem origem no Marketing e é um modelo de negócios para comercialização dos produtos.

Convergência fixo-móvel

Nos anos 1990 começou-se a falar na convergência entre telefonia fixa e móvel, mas sem resultados práticos. Uma década depois o assunto ressurge, ainda sem uma definição clara do que seria tal convergência, embora se possa dizer em linhas gerais que "tem como objetivo disponibilizar serviços convergentes pelos ambientes fixo, móvel e Internet".

Atualmente, as operadoras de telefonia enfrentam desafios para desenvolver estratégias para convergência fixo-móvel. As tecnologias que recebem mais atenção (*Unlicensed Mobile Access*, *IP Multimedia Subsystem*) são centradas na própria rede e estão em estágio imaturo, despendendo esforços que divergem da real necessidade da prestação efetiva de serviços para competir com outros provedores como Skype. Ainda falta demanda de mercado consistente, tanto de consumidores quanto empresas.

O Yankee Group (2004) publicou um estudo que identifica quatro estágios sucessivos na convergência fixo-móvel:

- Convergência por pacotes. Forma mais básica de convergência que consiste simplesmente na oferta comercial de telefonia fixa e móvel num único pacote de serviços. Não há integração entre tecnologias, mas unificação do atendimento ao consumidor e cobrança de faturas;
- Convergência de recursos. Integração de recursos que, anteriormente, existiam apenas para telefones fixos ou móveis. Podem-se citar funcionalidades de transferência automática de chamadas direcionadas para um telefone fixo (como na residência do cliente) para seu celular ou vice-versa, bem como caixa de mensagens de voz integrada;
- Convergência de produto. Convergência resultante da redundância entre produto fixo e móvel, fazendo com que efetivamente se tornem um só. É um amadurecimento da convergência de recursos, pois à medida que começam a ser oferecidos em um produto recursos que só eram disponíveis no outro;
- Convergência total. Quando a experiência do usuário no uso dos equipamentos ocorrerá de maneira transparente, coesa, contínua. Poder-se-á mudar de localização ou de terminal sem perceber, mantendo acesso às mesmas informações e serviços. A mesma agenda de contatos telefônicos, perfis e configurações ou arquivos multimídia estariam sempre disponíveis e sincronizados seja no telefone móvel, PDA ou computador desktop (PC).

Convergência de serviços

É a disponibilização de um mesmo serviço através de diferentes meios de comunicação. Essa modalidade de prestação de serviços tem sido utilizada por diversos segmentos, entre eles o segmento bancário. Há cada vez mais opções para o cliente consultar seu saldo: essa simples operação, que originalmente só podia ser realizada através do caixa humano ou pelo caixa eletrônico, já está disponível através da Internet, telefone fixo ou dispositivo móvel.

Convergência de terminais

É a utilização de um único terminal para acesso a múltiplas redes e serviços diversos. Exemplo inclui os *smartphones*, que combinam características de telefones celulares e PDAs. Desde os primeiros exemplares no início dos anos 2000, como o QCP da Kiocera ao lado, o uso desses aparelhos vem aumentando, principalmente no mercado corporativo onde há, ainda, um domínio da linha BlackBerry. Como as tarifas cobradas pelos serviços de transmissão de dados estão baixando, prevê-se que estes devem começar a substituir os telefones celulares (que estão saturados) também no mercado de massa [INFOEXAME 2007]. O iPhone, anunciado em janeiro de 2007, é um smartphone da Apple Inc. apresentado como um "telefone revolucionário", tanto que na mesma data do anúncio a empresa alterou sua razão social de "Apple Computer, Inc." para simplesmente "Apple Inc.". O objetivo das pesquisas que resultaram no iPhone foi a experimentação de telas sensíveis ao toque que, assim como o design e a facilidade de uso, é considerado um dos pontos fortes do aparelho. Por outro lado, além da comunicação por voz que se espera de qualquer telefone, o iPhone integra recursos multimídia, conexão à Internet por tecnologia EDGE com acesso à web e e-mails, e conectividade local por Wi-Fi e Bluetooth. Tais recursos conferem ao aparelho características de um terminal convergente.

Estas características também estão presentes em modelos novos de smartphones de outros fabricantes. O maior desafio para os desenvolvedores de aplicações é fazer com que essas executem satisfatoriamente em diferentes sistemas e modelos.

Convergência regulatória

O surgimento de serviços convergentes cria um ponto de contato entre dois mercados: o da telefonia, tradicionalmente regulamentado, e o mercado de serviços de dados, sujeito a pouca ou nenhuma regulamentação sobre a prestação dos serviços.

Dentre os desafios a serem enfrentados pelos órgãos reguladores, incluem-se a manutenção garantida de princípios, como a defesa da justa competição no setor de telecomunicações e radiodifusão. Por exemplo, através de ligações VoIP é possível enquadrar-se na lacuna não regulamentada dos serviços de transmissão de dados para evitar acordos internacionais e prover chamadas de voz mais baratas.

5.3.3.2. Redes Móveis **(*Mobile Networks*)**

Desde seu início na década de 70, as redes sem fio têm ganhado popularidade na indústria da computação, sendo utilizadas, principalmente, para permitir mobilidade. Atualmente, existem dois tipos principais de redes sem fio: redes infra-estruturadas e redes sem infra-estrutura (*ad-hoc*).

Nas redes infra-estruturadas, parte da rede é fixa e cabeada. Os nós móveis comunicam-se com a parte fixa através de estações-base, que fazem o enlace entre a rede cabeada e a rede sem fio. Quando um nó se move para fora do alcance de uma estação-base e entra no raio de cobertura de outra, as estações trocam informações sobre o nó, e este passa a se comunicar com a nova estação de forma transparente. Aplicações típicas desse tipo de rede incluem redes locais sem fio em escritórios ou empresas.

As redes sem infra-estrutura (redes *ad-hoc*) não possuem nenhum tipo de estação base ou roteador . Nela, todos os nós são móveis (comunicam-se por meio físico sem fio) e podem se conectar dinamicamente uns com os outros, formando a rede espontaneamente. Nós que não estão diretamente conectados se comunicam através do encaminhamento das mensagens através de nós intermediários. Cada nó da rede

funciona como um roteador que descobre e mantém rotas para outros nós. Por isso, muitas vezes, as redes sem fio espontâneas são chamadas de redes sem fio *multi-hop*.

Mais especificamente, uma rede sem fio espontânea é um conjunto de dispositivos sem fio que podem, dinamicamente, se auto-organizar em uma topologia aleatória e temporária, para formar uma rede sem usar nenhuma infra-estrutura pré-existente. Essas redes também são conhecidas como redes móveis espontâneas (*Mobile Ad-hoc Network* - MANET), e podem formar grupos de terminais sem fio autônomos.

Apesar de que (na prática) alguns terminais podem estar conectados a uma rede fixa, a característica principal das redes espontâneas é sua auto-configuração dinâmica, sem a intervenção de uma administração centralizada. As principais vantagens das redes espontâneas são flexibilidade, baixo custo e robustez. Aplicações típicas para essas redes incluem operações de resgate em desastres e encontros ou convenções em que pessoas precisam compartilhar informações rapidamente.

Dependendo do alcance da comunicação, as redes sem fio podem ser classificadas em: *Body Area Network* (BAN), *Personal Area Network* (PAN) e *Wireless Local Area Network* (WLAN). As BANs são formadas por um conjunto de dispositivos que têm alcance de comunicação em torno de dois metros. As PANs referem-se a comunicações entre diferentes BANs, tendo alcance de, aproximadamente, dez metros. E, finalmente, as WLANs têm alcance de comunicação na ordem de centenas de metros.

As BANs e PANs são implementadas, principalmente, com tecnologia *Bluetooth*, adotando o padrão IEEE 802.15.1. Já as WLANs, também conhecidas como Wi-Fi (*Wireless Fidelity*), são implementadas com tecnologia IEEE 802.11. A família 802.11 inclui vários padrões, como IEEE 802.11b e IEEE 802.11g, que diferem na camada física.

Os trabalhos de pesquisa atuais em redes móveis espontâneas (*ad-hoc*) estão direcionados, principalmente para os campos: (a) controle de acesso ao meio físico; (b) roteamento; (c) gerenciamento de recursos (*service discovery*); (d) gerenciamento de energia; (e) segurança.

5.3.3.3. Redes de Sensores

Na computação móvel deseja-se obter um acesso contínuo às informações e outros recursos computacionais através de uma comunicação sem fio. Um dos tipos de aplicações móveis mais usuais hoje é aquela que emprega redes de sensores.

Uma definição para rede de sensores é a de uma rede sem fio formada por um grande número de sensores pequenos e imóveis plantados numa base para detectar e transmitir alguma característica física do ambiente. A informação contida nos sensores é agregada numa base central de dados. Outro enfoque que se pode ter de redes de sensores é de um conjunto de nós individuais (sensores) que operam sozinhos, mas que podem formar uma rede com o objetivo de juntar as informações individuais de cada sensor para monitorar algum fenômeno. Estes nós podem se mover juntamente com o fenômeno observado. Por exemplo, sensores colocados em animais para observar seu comportamento. Ao observar o conjunto de sensores, estar-se-ia monitorando toda a manada [Pereira, Amorin e Castro 2007].

Sensores podem ser dispositivos eletrônicos (sensores físicos) ou componentes de software (sensores lógicos) que obtêm/medem algum tipo de informação. Sensores físicos hoje são utilizados para monitorar tanto ambientes de difícil acesso (oceano,

vulcões, áreas de desastre, etc.) quanto residências, áreas rurais (plantações e animais) e ruas (movimentação de pessoas e trânsito). Esses sensores, tipicamente, consistem de cinco componentes: detector de hardware, memória, bateria, processador embutido e transmissor-receptor.

Numa rede de sensores típica, os sensores individuais apresentam amostras de valores locais (medidas) e disseminam informação, quando necessário, para outros sensores e eventualmente para o observador [Hac 2003]. O acesso a essas informações pode ser sob demanda (reativo) ou pré-determinado pela aplicação (pró-ativo).

Para formar-se uma rede de sensores, têm-se 3 componentes: (i) sensores, (ii) protocolo de rede, para a comunicação entre sensores (propagação de dados) e com a aplicação, e (iii) aplicação/observador.

Redes de sensores são redes móveis e podem ser estáticas (infra-estruturadas) ou dinâmicas (ad-hoc). Os protocolos de roteamento *ad hoc* podem ser usados como protocolos para redes de sensores, porém esses apresentam desvantagens devido às restrições de capacidade de processamento e energia dos sensores físicos. Pesquisas em andamento procuram gerar protocolos e soluções mais adequadas às redes de sensores.

Essa nova tecnologia de sensores cria um conjunto diferente de desafios provenientes dos seguintes fatores [Pereira, Amorin e Castro 2007]: (i) os nós encontram-se embutidos numa área geográfica e interagem com um ambiente físico; (ii) são menores e menos confiáveis que roteadores de redes tradicionais; (iii) geram (e possivelmente armazenam) dados detectados ao contrário de roteadores de rede e (iv) podem ser móveis.

Para o ambiente ubíquo, as redes de sensores sem fio desempenham um papel importante para detectar, coletar e disseminar informações dos objetos físicos/lógicos monitorados. Aplicações de sensores representam um novo paradigma para operação de rede, que têm objetivos diferentes das redes sem fio tradicionais.

5.3.3.4. Middlewares

Middlewares implementam as camadas Sessão e Apresentação do Modelo de Referência ISO/OSI e objetivam habilitar a comunicação entre componentes distribuídos. Para tal, fornecem aos programadores de aplicações abstrações de alto nível, construídas usando primitivas do sistema operacional de rede, que escondem a complexidade introduzida pela distribuição. Tecnologias de middlewares existentes têm sido construídas com a metáfora de caixa preta, onde a distribuição torna-se transparente ao usuário e ao projetista de software.

Essas tecnologias têm sido projetadas com sucesso para sistemas distribuídos estacionários, executando sob redes fixas. Porém, a simples adoção da tecnologia atual de middlewares não é adequada ao ambiente móvel devido, principalmente, aos seguintes motivos: (i) as primitivas de interação, tais como transações distribuídas, requisições a objetos ou chamada remota de procedimento, assumem uma alta largura de banda e conexão permanente e disponível, o que contrasta com as características inerentes ao ambiente móvel; (ii) middlewares orientados a objetos, como Java-RMI (*Remote Method Invocation*), suportam principalmente comunicação ponto-a-ponto síncrona, a qual requer que o cliente solicite um serviço e o servidor o atenda, ambos executando simultaneamente, o que, novamente, contrasta com o ambiente móvel que requer anônima e assíncrona comunicação; (iii) ambientes distribuídos assumem que o ambiente de execução é estacionário, com confiável e alta largura de banda, localização

fixa de cada *host* e serviços bem conhecidos, contrastando com o cenário altamente dinâmico proposto pela computação móvel onde a localização dos hosts altera-se no tempo, e novos serviços podem ser descobertos dinamicamente enquanto este se move; (iv) a carga computacional para execução de middlewares é, geralmente, alta para ser carregada e executada em host móveis.

Outro aspecto a ser ressaltado é relativo à questão transparência x consciência. Middlewares são construídos em abordagens que enfatizam a transparência, onde programadores não necessitam conhecer detalhes sobre o objeto ao qual o serviço é requerido. Enquanto que em sistemas distribuídos estacionários é possível (e desejável) esconder completamente as informações de contexto (por exemplo, localização) e detalhes de implementação da aplicação, em ambientes móveis isto se torna mais difícil e, muitas vezes, inadequado.

Para oferecer transparência, os middlewares tomam decisões em nome das aplicações, sacrificando a flexibilidade. No entanto, considerando as novas demandas das aplicações móveis e ubíquas é mais eficiente que as decisões sobre utilização dos recursos levem em conta informações específicas de cada aplicação, o host móvel e o ambiente de execução corrente. Em sistemas móveis é essencial que o middleware seja adaptativo, leve e (auto)reconfigurável.

Desta forma, é necessário o desenvolvimento de novas plataformas de middleware para atender aos requisitos impostos pela mobilidade. Por exemplo, middlewares que utilizam o paradigma de comunicação assíncrona através dos mecanismos de subscrição (*publish/subscribe*), são mais adequados às redes móveis.

5.4. Em Direção a uma Definição de Sistema Ubíquo

Uma separação simplificada entre aplicações com mobilidade e conectividade que poderiam ser classificadas como pertencentes à Computação Móvel, Computação Pervasiva e Computação Ubíqua é ilustrada na Tabela 1.

Tabela 1. Propriedades e Domínios da Mobilidade

DOMÍNIO	MOBILIDADE	CONTEXTO	ATIVIDADES HUMANAS
COMPUTAÇAO MÓVEL	PRESENTE		
COMPUTACAO PERVASIVA	PRESENTE	PRESENTE	
COMPUTACAO UBÍQUA	PRESENTE	PRESENTE	PRESENTE

Em nosso ponto de vista, a Grade atua como o gerenciador do ambiente pervasivo. Para projetar um suporte em background para as aplicações pervasivas é necessário entender quais conceitos e propriedades são relevantes para descrever um sistema ubíquo. O conhecimento sobre projeto e construção de sistemas ubíquos é descrito em muitas pesquisas e artigos visionários. Existe uma dificuldade em produzir um livro-texto que claramente descreve os princípios que governam os sistemas ubíquos. Muitos desses princípios estão evoluindo conforme os pesquisadores avançam

em sistemas experimentais e prototipados. Um rol de conceitos e propriedades, listados na Figura 3, ilustra a complexidade de projeto presente em sistemas ubíquos, os quais envolvem esforços de pesquisa multidisciplinar. A figura mostra as características básicas que são consideradas relevantes para sistemas pervasivos de uma forma categorizada, auxiliando projetistas a reconhecer e tratar os desafios. Esta foi estabelecida após alguns anos de experiência na pesquisa de sistemas pervasivos e análise de muitos projetos e aplicações.

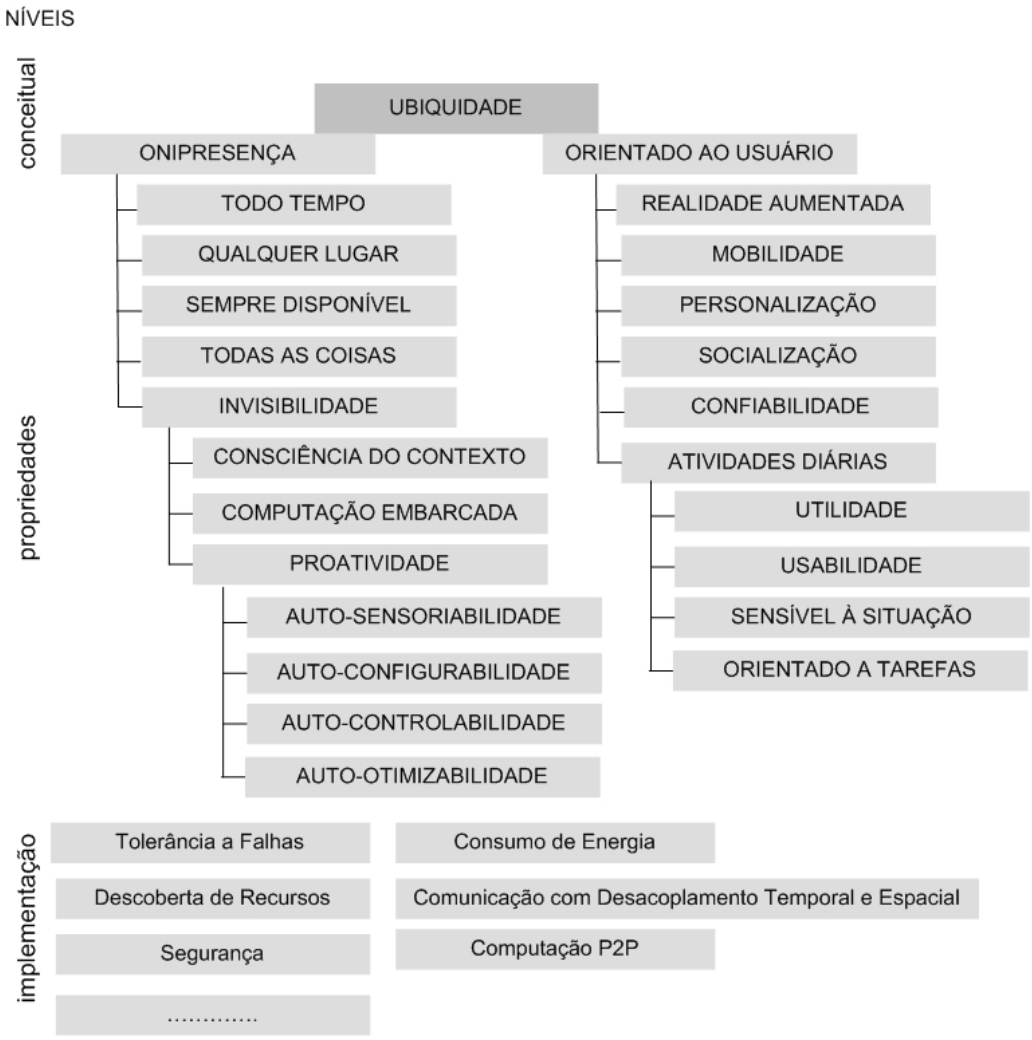

Figura 3. Conceitos e Propriedades dos Sistemas Ubíquos

O Espaço Ubíquo (UbiSpace), ou ambiente pervasivo, é definido de diversas formas as quais refletem níveis espaciais, tais como local – casa, hospital, campus, escritório, etc., urbano – ruas, distritos, cidades, e global. As necessidades computacionais alteram-se de acordo com a extensão geográfica do ambiente pervasivo.

Para definir um sistema pervasivo começa-se com o mais alto nível de abstração que é consecutivamente refinado, desde o nível conceitual até alcançar o nível mais concreto que permite implementação. Considerando a definição mais empregada de computação pervasiva têm-se o primeiro nível de abstração (conceitual): **onipresença**

da computação e **orientado ao usuário**, o qual traduz a mais alta prioridade: essencial para definir um sistema ubíquo. O segundo nível de abstração é derivado da questão "como realizar tais conceitos?". Para o primeiro, a resposta considera o que está embutido no conceito de onipresença: invisibilidade, em todo lugar, a qualquer tempo, sempre disponível, em todas as coisas. Para o segundo conceito, a base é a orientação às atividades diárias do usuário e os relacionamentos entre usuários: orientado a tarefas, mobilidade, realidade aumentada, confiabilidade, personalização e socialização.

No próximo nível, tenta-se responder a questão "como alcançar tais conceitos refinados?". Para isso, identificam-se as propriedades dos sistemas ubíquos relativas a cada conceito refinado, sendo que cada propriedade (ou conjunto de propriedades) pode ser materializada em diferentes mecanismos no nível de implementação. Por exemplo, muitos pesquisadores investigam questões sobre consumo de energia, tolerância a falhas, comunicação com desacoplamento espacial e temporal para disponibilizar o conceito 'sempre disponível'. Nota-se que propriedades e suas combinações podem conter diferentes e diversas possibilidades de realização, isto é evidente quando se consideram as diversas soluções encontradas em muitos projetos exploratórios da Computação Pervasiva. Com o objetivo de simplificar a ilustração, algumas caixas (correspondentes às propriedades) estão abertas, outras fechadas.

Devido à complexidade do sistema ubíquo, muitas aplicações – principalmente as exploratórias (de pesquisas) – tratam somente com uma ou poucas dessas propriedades. Entretanto, alguns projetos [Augustin et al 2004],[Garlan et al 2002],[Roman et al 2003] se propõem a abordar uma infra-estrutura mais geral para colocar junto muitas dessas propriedades (essenciais). A maioria dos projetos exploratórios foram inspirados na modelagem baseada em cenário, a qual auxilia a entender a situação de projeto e a criar sistemas computacionais e aplicações como artefatos das atividades humanas.

5.5 Inserindo Pervasividade na Grade Computacional

Os campos de Grade Computacional e Computação Ubíqua estão sendo integrados de duas formas distintas considerando a comunidade de origem: (a) a comunidade de Computação em Grade adota o conceito de Grade Pervasiva ou Grade Ubíqua (UbiGrid) ou usa conceitos da mobilidade como agentes móveis para introduzir flexibilidade no acesso ao sistema altamente distribuído; já (b) a comunidade de Computação Ubíqua usa a infra-estrutura de grade como suporte para o gerenciamento e a disponibilização do ambiente pervasivo (UbiSpace) e da semântica 'siga-me'. Nesta seção ver-se-á questões embutidas em ambas visões.

5.5.1. Gerações das Grades Computacionais

Desde sua criação, a computação em grade tem atravessado diferentes fases ou gerações [de Roure et al 2003]. Na década de 90, a primeira geração permitiu a interconexão de grandes centros de supercomputação para obter poder computacional agregado não disponível em nenhum dos participantes da grade, ou para compor e coordenar computação distribuída em centenas de estações de trabalho dos usuários. Esta geração originou projetos como SETI@HOME (http://setiathome.ssl.berkeley.edu), Globus [Foster et al 2001],[Foster et al 2002] e Condor [Thain et al 2003] e visava aplicações de dados intensivos (data grids) ou computação intensiva (*computation grids*). Nesta época, a definição de grade era "uma grade computacional é uma infra-estrutura de software e hardware que fornece confiável, consistente, e pervasivo acesso a capacidades computacionais".

A segunda geração é caracterizada pela capacidade de ligar centros regionais, nacionais e internacionais e pela adoção de padrões que permitem o emprego da infra-estrutura computacional em escala global. Do ponto de vista arquitetural, a segunda geração usa middlewares de grade como 'cola' entre sistemas distribuídos heterogêneos, usuários, recursos e políticas locais. Os middlewares têm desafios técnicos nas áreas de comunicação, escalonamento, segurança, informação, acesso a dados e tolerância a falhas. A grade é definida como 'compartilhamento coordenado de recursos e resolução de problemas distribuídos em dinâmicas e multi-institucionais Organizações Virtuais' [Foster et al 2001]. Um representante desta geração é a evolução do Globus: Globus Toolkit 2 [Foster et al 2002].

O Global Grid Forum (www.gridforum.org) é uma organização que discute padrões e questões envolvidas na construção da computação em grade, e tem empreendido esforço para definir a Open Grid Services Architecture (OGSA) a qual moderniza e estende o Globus Toolkit para tratar novos requisitos e adota Web Services como tecnologia referencial. O Globus Toolkit 3 define *grid services* que estendem o conceito de *Web Services* devido ao fato de que: (a) serviços em grade devem ser dinâmicos e transientes, pois nem todos os serviços participam da computação no nodo; (b) serviços em grade são globalmente distribuídos, sem um controle central; (c) aplicações podem requerer centenas de serviços, que devem ser coordenados de modo eficiente; (d) aplicações podem ter vida longa (*long-lived*), impactando os requisitos dos serviços.

A próxima geração de grade é dirigida pelo uso eficiente e efetivo de dados armazenados e sua transformação em informação (*information grids*) e conhecimento (*knowledge grids*). Estudos recentes definem a Grade com Semântica (*Semantic Grid*) como uma extensão da grade corrente na qual informação e serviços adquirem um significado bem-definido para permitir a pessoas e computadores trabalharem em cooperação. Esses estudos propõem o emprego de tecnologias da Web Semântica na grade [Goble and de Roure 2004] e despertam o interesse de pesquisadores das Ciências Sociais e Humanas e da comunidade de Inteligência Artificial.

Por outro lado, o advento de dispositivos sem fio, que estão aumentando em número e em poder de processamento, cria oportunidades de aumentar a diversidade de usuários e tecnologias que utilizam a grade. Estudos recentes defendem o uso de recursos móveis na computação em grade a qual abrirá oportunidade de desenvolver novos tipos de aplicações [Storz and Friday and Davies 2003],[Davies and Friday and Storz 2004],[IEEE 2004],[McKnight and Gaynor 2003]. Um levantamento dos desafios e questões envolvidas está relatado na referência [Davies and Friday and Storz 2004].

5.5.2. Grade Pervasiva

A computação em grade tem sido o paradigma dominante para computação distribuída geograficamente. Originalmente o objetivo das grades era combinar recursos através de muitas organizações para representar uma organização virtual, normalmente maior e mais efetiva do que cada umas das organizações reais isoladas, permitindo a resolução de problemas que antes não poderiam ser resolvidos por elas isoladamente. A categoria de grades pervasivas une as duas tecnologias aproveitando o melhor das duas e permitindo que os usuários tenham mobilidade com alto desempenho e alto desempenho com uma comunicação direta com o ambiente físico de cada aplicação, através das redes de sensores conectadas às grades. Nota-se que ainda não há um consenso sobre a real utilidade das grades pervasivas. Para alguns eles são apenas uma tendência passageira, para outros, apenas uma outra forma de enxergar as grades atuais.

A integração entre mobilidade e computação em grade enfatiza uma série de problemas a serem vencidos para diminuir o impacto que as constantes variações no grau de disponibilidade de recursos e serviços causam nas aplicações. Entre estes, citam-se: restrições dos recursos dos nodos móveis, restrições impostas pelo ambiente sem fio como desconexões freqüentes e baixa largura de banda, desconexão voluntária para economia de energia do dispositivo móvel, escalabilidade devido ao movimento dos usuários.

Acesso a Dados Independente de Localização

Uma das premissas inerentes à noção de uma grade pervasiva (disponível todo tempo e acessível de qualquer lugar) é a capacidade que as aplicações, e, por conseqüência, os usuários, devem ter de acessar dados relevantes aos processos sendo executados, especialmente os arquivos pessoais de cada usuário. Especificamente, a mobilidade do usuário e das aplicações ficaria bastante comprometida se estas aplicações e usuários perdessem acesso aos arquivos quando modificassem sua localização.

Descoberta Automática de Recursos e Serviços

Estratégias para descoberta de recursos permitem a localização automática de dispositivos ou serviços em rede. A pesquisa nessa área motiva-se no crescente enriquecimento computacional dos ambientes com os quais interagimos, devido ao surgimento de avançados dispositivos pessoais móveis e da popularização de infra-estruturas de comunicação sem-fio. Adicionalmente a recursos computacionais típicos (poder de processamento e armazenamento, por exemplo), dispositivos sem-fio disponibilizam novas funcionalidades, como câmeras digitais, microfones e receptores GPS, entre outros. O modelo PerDiS (Pervasive Discovery Service) [Filho et al 2005], desenvolvido como dissertação de mestrado na Universidade Federal do Rio Grande de Sul, foi projetado com o intuito de ser uma estratégia para descoberta de recursos voltada para uso em um ambiente pervasivo. Sua concepção baseou-se na identificação dos principais requisitos de uma solução apropriada para utilização em um cenário de computação pervasiva. A proliferação de pesquisas na área de descoberta de recursos confirma que as soluções existentes atualmente não tratam suficientemente bem as necessidades impostas pelos dispositivos móveis.

Sensibilidade ao Contexto

Na base de uma arquitetura de grade pervasiva está o conceito de **consciência do contexto**, o qual permite a identificação de alterações no estado dos recursos e o disparo automático de uma adaptação de código na aplicação para ajustar-se ao novo consumo de recursos/serviços. Simulações iniciais foram realizadas e atestam a viabilidade da arquitetura proposta.

5.5.3. Projetos com Integração Grade e Mobilidade

A Grade Européia permitirá a pesquisadores e outros usuários compartilhar recursos através do continente (http://www.cordis.lu/ist/grids/ building_grids_for_europe.htm). Em 15 de setembro de 2004, 12 projetos de grade foram oficialmente iniciados. Os quatro maiores projetos são: Akogrimo, CoreGRID, NextGRID and SIMDAT. Destes,

o projeto **Akogrimo** (www.mobilegrids.org), da Telefonica, integra os conceitos de grade e comunicação móvel e prevê três cenários de validação: (i) e-learning, focaliza construir estudo de caso para novos modos de aprender utilizando a infraestrutura Akogrimo; (ii) e-health, explora a tecnologia de grade e mobilidade no domínio da saúde; (iii) atendimento a desastres, onde Akogrimo servirá como plataforma de colaboração.

Uma nova iniciativa da Universidade de Lancaster é o Grupo de Pesquisa **UbiGrid** (http://ubigrid.lancs.ac.uk/) para investigar o uso de tecnologias de grade no suporte a experimentos de sistemas de computação ubíqua. As idéias do grupo foram apresentadas no Ubicomp 2003 em um 'position paper'. A grade promete o acesso a recursos computacionais além dos limites institucionais de forma padronizada, uniforme e confiável, sendo essencial para alcançar a Computação Ubíqua em escala global. Os pontos de sinergia entre computação em grade e computação ubíqua relacionados pelo grupo são: heterogeneidade, interoperabilidade, pagamento pelo uso e interação de grande número de recursos. Entretanto, o grupo reconhece que hoje há pouco reconhecimento na comunidade de grade computacional da necessidade da computação ubíqua. Acreditam que esta situação deverá se alterar nos próximos anos.

No Brasil, o **projeto ISAM** (www.inf.ufrgs.br/~isam), pioneiro no país, iniciou suas atividades em 2000 e propõe uma arquitetura de software para criar e gerenciar um ambiente pervasivo em larga escala. Os conceitos-base da arquitetura são derivados da integração de conceitos definidos em áreas distintas: computação móvel e pervasiva, computação em grade e computação consciente do contexto.

O **projeto MAG** (*Mobile Agents for Grid Computing Environments*), em desenvolvimento na UFMA, explora a tecnologia de agentes móveis como uma forma de superar os desafios de construção de grades de computadores. O MAG executa aplicações carregando dinamicamente seus códigos nos agentes moveis. O agente do MAG pode ser realocado dinamicamente entre nos da grade através de um mecanismo de migração transparente chamado MAG/Brakes, como uma forma de prover suporte a nós não dedicados. O MAG inclui mecanismos de tolerância a falhas de aplicações, de grade pervasiva e grade de dados. O paradigma de agentes foi extensivamente utilizado para projetar e implementar os componentes do MAG, formando uma infra-estrutura multiagente para grades computacionais.

5.6. Inserindo Grade em Sistemas Ubíquos

Nas próximas subseções, detalham-se as características mais importantes da Computação Ubíqua e cujos conceitos da Grade Computacional podem auxiliar a implementar.

5.6.1. Mobilidade Total

Mobilidade é um conceito amplo que pode se referir à propriedade de troca de lugar no espaço de qualquer elemento de um sistema computacional: dados, código e processos, rede e usuário. Os três primeiros foram já amplamente abordados, suporte para o último é mais recente. Juntar todas as possibilidades de mobilidade em um sistema único é um dos grandes desafios da computação pervasiva.

Mobilidade de Dados

Gerenciamento de dados ubíquos tem sido estudado no campo de banco de dados [Chemiak and Franklin and Zdonik 2001]. As funcionalidades requeridas para

novas bases de dados incluem suporte à mobilidade, sensibilidade ao contexto e suporte à colaboração. Uma propriedade está presente em todos eles: adaptabilidade.

Mobilidade de Código

Mobilidade de Código (Mobile Computation) foi tema de muitas pesquisas desenvolvidas há décadas atrás, especialmente no campo de sistemas operacionais. [Jul et al 1988],[Chemiak and Franklin and Zdonik 2001],[Smith 1988], mas elas não produziram um produto comercial de sucesso. O esforço para introduzir mobilidade forte (mobilidade do processo em execução) é alto em relação às vantagens obtidas. Hoje, a abordagem comum de mobilidade de código é referenciada como mobilidade fraca – mecanismo de instanciação remota [Vigna 1998]. O sucesso da linguagem Java no domínio da Internet pode ser atribuído ao fato desta estar baseada em mobilidade fraca de código. Uma discussão sobre o tema é encontrada na referência [Fugetta and Picco and Vigna 1998].

Implementar mobilidade de interface do usuário entre diversos dispositivos e plataformas é uma tarefa difícil. A tese do projeto MDAT [Banavar et al 2004] é uma adaptação híbrida em tempo de projeto e em tempo de execução. A adaptação em tempo de projeto converte a aplicação genérica em múltiplas versões adaptadas a dispositivos específicos antes de armazenar a aplicação no servidor. A adaptação em tempo de execução converte a aplicação genérica em uma versão específica para o dispositivo em uma aplicação Web em resposta a uma requisição do cliente. MDAT trata do domínio Internet e, correntemente, suporta a tradução entre XHTML, XHTML Mobile profile e WML.

Mobilidade de sessão pode ser obtida com a Computação Fluida a qual denota a replicação e sincronização em tempo-real dos estados da aplicação em muitos dispositivos. Tais estados da aplicação fluem entre os dispositivos [Graf 2003]. Nesta solução, cada dispositivo tem uma réplica do estado da aplicação, o qual permite-lhe operar autonomamente. A consistência da réplica é fraca, dependendo da qualidade da conectividade da rede. O middleware Computação Fluida é uma biblioteca Java que executa em PDAs, tais como Sharp Zaurus and PocketPC devices.

Mobilidade da Rede

Rede espontânea (ad-hoc network) é uma rede local ou uma pequena rede especificamente com conexões temporárias e sem fio, na qual alguns dispositivos de rede fazem parte dela somente durante a sessão de comunicação ou, no caso de dispositivos móveis e portáteis, enquanto estes mantém proximidade do resto da rede. Esta rede autoconfigurada de roteadores móveis (e hosts associados) e conectados por ligações sem fio – a união forma uma topologia arbitrária da rede – é um desafio. Roteadores são livres para moverem-se randomicamente e organizarem-se arbitrariamente; assim, a topologia da rede sem fio pode alterar-se rapidamente e de forma imprevisível. Questões relevantes incluem aumentar a eficiência da informação transferida e habilitar sua usabilidade para suportar futuras aplicações comerciais.

Mobilidade do Usuário

A mobilidade do usuário é constantemente associada à portabilidade do dispositivo móvel (mobilidade do terminal). Entretanto, este é um conceito mais amplo: o usuário pode mover-se no globo com ou sem seu equipamento, porém mantendo o

acesso a seu ambiente computacional pessoal. No ambiente pervasivo, o usuário-final deseja ter acesso usando os diversos dispositivos disponíveis para basicamente a mesma informação e funcionalidade. Conteúdo e serviços podem ser entregues ao usuário aonde ele se encontra e no dispositivo que ele está usando no momento ou que está disponível no local onde ele está. Enquanto que este cenário representa um ambiente computacional atraente, ele contém três grandes desafios na forma de heterogeneidade, dinamismo e alterações no contexto de execução, potencializados pela mobilidade do usuário.

5.6.2. Sistema Pervasivo em Escala Global

Muitos projetos de computação ubíqua focalizam um único ambiente, tais como casa, escritório, carro, hospital, sala-de-aula [Storz and Friday and Davies 2003]. Entretanto, esses ambientes devem ser capazes de operarem juntos, a fim de customizar todas as interações dos indivíduos com o mundo externo. Alguns pesquisadores consideram que o sistema pervasivo em escala global é muito difícil de ser realizado devido a este pressupor a adoção de padrões. As primeiras pesquisas em sistemas pervasivos enfatizaram o escopo local. Recentemente, este tem se deslocado para o escopo urbano. Entretanto, o escopo global tende a se tornar importante à medida que aumenta o uso das tecnologias ubíquas. As redes celulares podem ser o caminho tecnológico através do qual a computação pervasiva em larga escala acontecerá.

Alguns projetos começaram a abordar este tema [Al-Myhtadi 2004],[Augustin 2005]. Super Spaces [Al-Myhtadi 2004] estende o conceito de Espaços Ativos (Active Spaces), definido no projeto Gaia, para habilitar o gerenciamento, operação e manutenção em larga escala através do agrupamento dos espaços em espaços maiores. O framework permite dois modelos de serviços e aplicações: modelo recursivo hierárquico e modelo par-a-par (P2P). Entretanto, essas soluções têm um problema clássico: o desempenho. SuperSpaces tem um conjunto de serviços básicos – nomeação, descoberta, repositório, gerenciamento de eventos, sistema de arquivos e serviços de contexto – que permitem executar operações em múltiplos subespaços, dados e atividades sincronizados através de diferentes subespaços. O projeto está em andamento.

Aplicações ubíquas tenderão a ser aplicações de longo tempo de execução. Para gerenciá-las é necessário uma infra-estrutura capaz de manter uma execução quase-contínua das tarefas dos usuários e de permitir ao usuário (móvel) o acesso a elas de qualquer lugar, a qualquer tempo, com o dispositivo disponível, enquanto ele libera o usuário do gerenciamento da aplicação e do controle do processo de adaptação. Ao longo da execução, a aplicação é exposta a alterações do contexto, originado da oscilação na disponibilidade de recursos e mobilidade do usuário. Assim, cresce em importância o papel que a adaptação consciente do contexto representa.

5.7. Arquitetura que Integra Grade, Contexto e Pervasividade

O projeto ISAM – Infra-estrutura de Suporte às aplicações Móveis Distribuídas – [Augustin et al 2002] vem criando uma infra-estrutura de suporte para projeto, implementação e execução de softwares pervasivos. O ISAM provê um *ambiente virtual do usuário*, onde as aplicações têm o estilo *siga-me* (*follow-me*), permitindo ao usuário ter acesso ao seu ambiente computacional independentemente de localização e de tempo. Assim, ele atende às exigências desse novo cenário da Computação Pervasiva, onde coexistem a mobilidade física (equipamentos e/ou usuários) e a mobilidade do software (aplicações e serviços).

Integrando o projeto ISAM, foi proposto e implementado um *middleware* para suporte à Computação Pervasiva que visa criar e gerenciar um ambiente pervasivo, bem como promover a execução de aplicações que expressam a semântica *siga-me* sobre esse ambiente. Esse *middleware*, denominado EXEHDA – *Execution Environment for Highly Distributed Applications* – [Yamin 2004], é adaptativo ao contexto e baseado em serviços, sendo chamado ISAMpe o ambiente por ele disponibilizado.

O EXEHDA é estruturado em um núcleo mínimo com serviços carregados sob demanda, os quais estão organizados em subsistemas que gerenciam: (a) execução distribuída; (b) comunicação; (c) reconhecimento de contexto; (d) adaptação; (e) acesso pervasivo aos recursos e serviços; (f) descoberta de recursos; (g) gerenciamento de recursos. O contexto é monitorado proativamente, e o *middleware* permite que tanto a aplicação como ele próprio utilizem as informações de contexto para gerenciar a adaptação de seus aspectos funcionais e não-funcionais. Dessa forma, o mecanismo de adaptação do EXEHDA utiliza uma estratégia colaborativa entre a aplicação e o ambiente de execução para gerenciar o comportamento de cada componente da aplicação [Yamin 2004]. As políticas que irão reger os mecanismos de adaptação são especificadas no ambiente de desenvolvimento provido pelo ISAMadapt [Augustin 2004].

As aplicações-alvo são distribuídas, adaptativas ao contexto em que executam e compreendem a mobilidade lógica e a física, sendo baseadas no modelo de programação ISAMadapt [Augustin 2004].

Na perspectiva do ISAM, entende-se por mobilidade lógica a reorganização dos componentes de software da aplicação, disparando ou não a migração de software. Por sua vez, a migração de software é quando algum componente da aplicação troca de equipamento hospedeiro. Já por mobilidade física, entende-se o deslocamento do usuário portando ou não um dispositivo móvel.

5.7.1. A Perspectiva da Arquitetura ISAM

A Figura 4 apresenta uma visão geral da arquitetura ISAM. A representação da consciência do contexto como um módulo virtual tem por objetivo ressaltar sua importância na arquitetura e caracterizar sua presença na concepção de todos os outros componentes. Como se pode observar na figura, a arquitetura ISAM é dividida em três camadas: camada de aplicação, camada de *middleware*, e camada de sistemas básicos.

Na camada de aplicação encontra-se a linguagem de programação ISAMadapt, a qual disponibiliza abstrações para o desenvolvimento de aplicações pervasivas conscientes do contexto [Augustin 2004].

A camada de middleware, na qual estão os mecanismos de suporte à execução da aplicação pervasiva e os mecanismos de adaptação [Yamin, et al 2003], é formada por dois níveis. O primeiro nível é composto por três módulos de serviço à aplicação: Acesso Pervasivo a Código e Dados, Ambiente de Execução da Linguagem e Reconhecimento de Contexto. O segundo nível, composto pelos serviços básicos do EXEHDA, provê as funcionalidades necessárias ao primeiro nível, bem como, disponibiliza serviços de migração, persistência, descoberta de recursos, comunicação, escalonamento e monitoração.

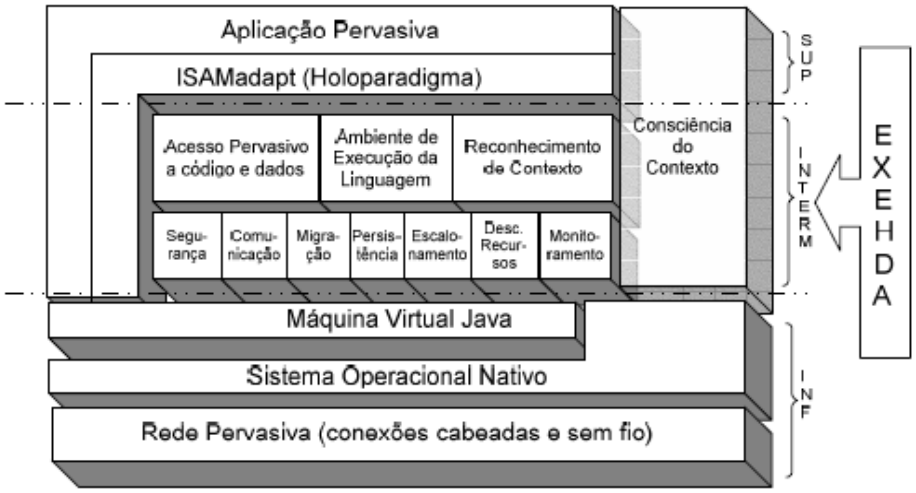

Figura 4. Arquitetura ISAM

O módulo de Acesso Pervasivo a Código e Dados é responsável por disponibilizar o Ambiente Pervasivo (ISAMpe), formado pelo Ambiente Virtual do Usuário (AVU), Ambiente Virtual da Aplicação (AVA) e Base de Dados Pervasiva das Aplicações (BDA). Já, o Ambiente de Execução da Linguagem é encarregado do gerenciamento da aplicação durante seu tempo de vida. E, por fim, o módulo de Reconhecimento de Contexto é responsável por informar o estado dos elementos de contexto de interesse da aplicação e do próprio ambiente de execução.

A camada de sistemas básicos é composta pela Máquina Virtual Java (JVM), podendo ser utilizada tanto a J2SE (Java Standard Edition) como a J2ME (Java Micro Edition), pelo sistema operacional sobre o qual a JVM é executada, e pela camada de rede pervasiva que deve integrar uma rede sem fio a uma rede cabeada infra-estruturada.

5.7.2. Adaptação Colaborativa e Sensível ao Contexto

A modelagem do processo de adaptação no ISAM, esquematizada na Figura 5, foi dividida em etapas:

- detecção de alterações, que engloba monitoração, interpretação e notificação, visa observar o ambiente para detectar e notificar alterações significativas. Esta pode ser realizada de duas formas: *pull*, a informação é solicitada por uma requisição; *push*, a informação é enviada ao cliente que se inscreveu no serviço;
- escolha da ação, engloba a seleção da ação adaptativa entre alternativas pré-definidas pelo programador. A seleção pode ser realizada periodicamente, ou quando ocorre o evento de notificação de alteração;
- ativação da ação, refere-se à execução da ação adaptativa selecionada.

A adaptação se refere à alteração no comportamento, na estrutura ou na interface da aplicação em resposta a trocas arbitrárias no estado do elemento de contexto. Os tipos de adaptações que podem ser empregadas pela aplicação dependem da natureza desta e dos recursos que ela requer. Alguns exemplos de comportamento adaptativo incluem: (i) variedade de filtros (compressão, omissão, conversão de formato) inseridos entre o servidor e o cliente; (ii) alteração da fidelidade de saída; (iii) interface reduzida; (iv)

migração de componentes para máquinas mais poderosas; (v) fornecer a funcionalidade conectada em face à desconexão utilizando *buffering* e *prefetching*.

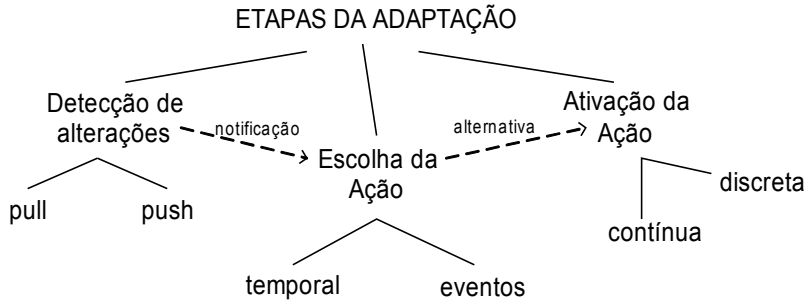

Figura 5. Etapas da Adaptação ISAM

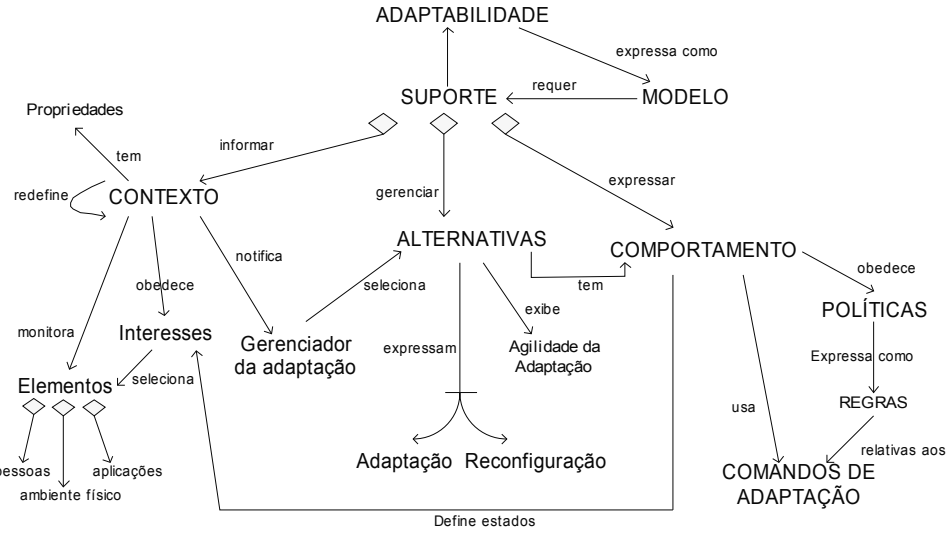

Figura 6. Modelando a Adaptação ISAM

O conjunto de conceitos usados para tratar adaptação no ISAM é mostrado na Figura 6. Os conceitos são derivados do suporte necessário para tratar a adaptação sob três aspectos: as informações para adaptação derivadas do ambiente; o gerenciamento da execução da adaptação; e a expressão do comportamento adaptativo.

Considera-se que as abstrações de sistemas operacionais existentes não são suficientes nem apropriadas para tratar as necessidades das aplicações *pervasivas*. Defende-se um estilo de programação que força a aplicação a exibir explicitamente um comportamento adaptativo que será gerenciado e executado por um ambiente de suporte (*middleware*).

A análise dos vários sistemas móveis adaptativos e aplicações-protótipo, com suas variadas noções de adaptação permitiu identificar os requisitos de propósito-geral

para expressar adaptabilidade ao contexto no nível de linguagem de forma genérica. Estes requisitos incluem:

- descrição do elemento de contexto de interesse – identificação dos elementos que compõem o ambiente e que modelam uma visão particular de contexto;
- descrição do comportamento adaptativo – coleções de ações e conjunto de restrições onde estas podem ocorrer. Adaptação pode incluir alterações internas (parâmetros, algoritmos ou representação de dados) ou alterações externas (reconfiguração, como migração de componentes);
- descrição de políticas – regras de propósito geral ou particular que orientam as decisões de adaptação realizadas pelo *middleware*.

Estes requisitos tornam-se operacionais através de um conjunto mínimo de primitivas para atender a expressividade, relativos a contexto, adaptadores e mecanismos de adaptação, implementadas na linguagem ISAMadapt. A programação é facilitada pelas interfaces de programação do ambiente de desenvolvimento e o suporte dado pelo *middleware*. Detalhes são discutidos no capítulo 4 e na referência [Yamin 2003], respectivamente.

Na perspectiva do EXEHDA, a adaptação não é uma propriedade funcional da aplicação. Assim, o tratamento da adaptação é realizado pela gerência de execução da aplicação de modo autônomo. Para isso, o código da aplicação é implementado de forma adaptativa e reflete as possibilidades de comportamento da aplicação frente às alterações do estado do elemento de contexto para o qual a aplicação é sensível [Augustin 2004].

O *middleware*, além de cooperar na adaptação da aplicação, também se adapta ao contexto. Atualmente, os principais elementos de contexto considerados pelo EXEHDA para adaptação são: o tipo de equipamento, o estado de ocupação dos seus recursos e a situação de sua conectividade no momento [Yamin 2004]. É importante salientar que a adaptação ocorrerá se o sistema dispuser dos recursos necessários para sua efetivação, descartando-se situações onde o nível de disponibilidade de recursos inviabiliza o processo de adaptação.

5.7.3. Ênfase no Contexto

No ambiente *pervasivo*, a mobilidade física introduz a possibilidade do movimento do usuário durante a execução da aplicação. Enquanto se movimenta, os recursos podem se alterar, não só em função da área de cobertura e heterogeneidade das redes, como em função da sua disponibilidade ser variável no tempo, devido à alta escalabilidade dos sistemas móveis distribuídos. Em conseqüência, a localização corrente do usuário determina o contexto de execução da aplicação. A Figura 7 esquematiza essa situação.

A abstração de contexto permite tanto focalizar em alguns aspectos que são relevantes em uma situação particular quanto ignorar outros. No ISAM, contexto é definido como "toda informação relevante para a aplicação e que pode ser obtida por esta". O programador explicitamente identifica as entidades[4] e define seus atributos, os quais integram o contexto da aplicação [Augustin 2004].

[4] Entidades, como pessoas e nodos, não têm uma representação explícita no modelo de contexto ISAM, somente seus atributos (localização, atividades, preferências, capacidades,...) formam os elementos de contexto presentes na modelagem.

Figura 7. Contexto Determinado pela Mobilidade

Por exemplo, a entidade nodo poderá ter como elemento de contexto: ociosidade, carga computacional, poder computacional, número de nodos. Alterações no estado dos atributos das entidades disparam o processo de adaptação na aplicação. Assim, pode-se refinar a definição de contexto para "todo atributo de uma entidade cuja alteração em seu estado dispara um processo de adaptação na aplicação ISAMadapt".

O contexto pode referir-se a informações ambientais (recursos físicos), funcionais (recursos lógicos) ou comportamentais (usuário). Estas informações exibem características temporais (variáveis no tempo) que podem ser caracterizadas como estáticas – descrevem aspectos invariantes, como tipo de dispositivo, ou dinâmicas – caracterizam aspectos com alterações freqüentes, como a localização do usuário. As informações estáticas podem ser obtidas diretamente com o usuário ou através de arquivos de configuração dos sistemas. As informações dinâmicas devem ser obtidas instantaneamente ou periodicamente do ambiente. Para tal é necessário um serviço de monitoramento para obter os dados de sensores tanto de hardware quanto de software.

Informações de contexto podem refletir o estado corrente ou serem derivadas de informações históricas (passado) com vistas a predizer o futuro. Para obter tal funcionalidade, o modelo deve armazenar as informações de contexto coletadas para análise/acesso futuro.

Informações de contexto podem ser imprecisas ou ficarem desatualizadas por várias razões: produtores e consumidores de informações de contexto podem estar distribuídos e distantes uns dos outros, conduzindo a um atraso entre a geração da informação e o uso desta; sensores e algoritmos para monitoração são propensos a falhas; desconexões ou falhas de conexão entre produtor e consumidor podem levar a um desconhecimento parcial ou total do contexto (*unknow* é sempre um resultado possível). Esta questão reflete a necessidade de se conhecer a qualidade da informação obtida.

Muitas das informações úteis à aplicação são derivadas de sensores. Porém, existe um *gap* entre a saída do sensor e o nível de informação requerido pela aplicação. Este *gap* pode ser quebrado com uma série de processamento sobre a informação sensorada (filtragem, agregação, refinamento) até transformá-la em informação de contexto útil à aplicação. Além disso, cada aplicação pode requerer uma interpretação diferente para o mesmo dado sensorado com diferentes níveis de abstração. Desta forma, a informação de contexto deve ter representações alternativas que permitam atender a este requisito.

Informações de contexto podem ser derivadas ou compostas de outras informações mais simples. Por exemplo, a atividade em execução pode ser derivada da localização corrente do usuário e de informações de atividades passadas nesta localização. Isto reflete a necessidade de estabelecer relacionamentos entre os elementos de contexto.

Resumindo, o modelo de contexto deve ser projetado para capturar as informações relevantes (elementos de contexto) para o projeto e construção de sistemas e aplicações que devem se ajustar ao contexto corrente ou previsto. Este deve prover informações de vários tipos. As informações de contexto podem ser sensoradas, derivadas ou fornecidas pelo usuário, como suas preferências. Podem ser correntes ou históricas. As informações podem ser simples – obtidas de uma única fonte, ou compostas – obtidas de várias fontes através do conceito de coleção (média, máximo, etc.), alternativas (ou, e). As informações também podem ser relativas ao tempo (temporal), a localização geográfica (espacial), ao usuário (pessoal) ou grupos de usuários (social).

O modelo ISAM considera contexto como o conjunto de dados externos que o programador decide ser o(s) elemento(s) que influencia(m) o comportamento da aplicação, e que podem ser obtidos. Logo, para uma aplicação específica somente um subconjunto do contexto disponível é capturado e usado. Logo, necessita-se de um esquema de propósito geral e extensível para descrever a informação de contexto. Esta descrição será analisada e usada para guiar o processo de adaptação. O esquema de descrição é estabelecido por uma relação de tradução que será definida e implementada pelo projetista do sistema, orientado por um modelo (*template)*. O protocolo de entrega de informações de contexto para a aplicação é baseado em tupla com natureza reativa.

5.7.4. Suporte à semântica siga-me

No EXEHDA, o suporte à semântica *siga-me* é construído pela agregação de serviços relativos ao reconhecimento de contexto, ao acesso pervasivo e à comunicação. Como estratégia para o tratamento da complexidade associada ao suporte da semântica *siga-me*, no EXEHDA é adotada a decomposição das funcionalidades de mais alto nível, recursivamente, em funcionalidades mais básicas [Yamin 2004].

Dessa forma, o *middleware* possui dois mecanismos principais que dão suporte a semântica *siga-me*: (*i*) um mecanismo de monitoramento que permite inferir sobre o estado dos recursos e das aplicações; (*ii*) e outro que promove adaptações funcionais e não-funcionais de acordo com o contexto monitorado.

Entende-se por adaptação não-funcional a capacidade do *middleware* de modificar a localização física dos componentes das aplicações, seja pela migração do mesmo, ou pela instanciação remota dele. Por sua vez, a adaptação funcional é a capacidade do *middleware* de selecionar a implementação do componente, dentre as disponibilizadas no desenvolvimento, a ser utilizada em um determinado contexto de execução.

A instanciação remota do componente implica que o código a ser instanciado deve estar disponível através do acesso pervasivo a um repositório de código. Enquanto que o acesso ao ambiente computacional do usuário, independentemente de localização e de dispositivo empregado, implica o acesso pervasivo a seus dados e configurações pessoais.

5.7.5. O Cenário Pervasivo Alvo

Do ponto de vista das aplicações pervasivas, o EXEHDA é o provedor dos serviços que dão suporte às abstrações definidas no desenvolvimento. A interação da aplicação com o ambiente computacional, através dos serviços disponibilizados pelo *middleware*, proporciona à aplicação a visão do ambiente pervasivo ISAMpe.

O ISAMpe é composto por três abstrações básicas: EXEHDAcel, EXEHDAbase e EXEHDAnodo. A EXEHDAcel denota a área de atuação de uma EXEHDAbase, sendo composta por essa e por EXEHDAnodos. A EXEHDAbase é responsável pelos serviços básicos do ISAMpe e, embora constitua uma única referência lógica, seus serviços podem estar distribuídos entre vários equipamentos. Por fim, os EXEHDAnodos são os equipamentos de processamento disponíveis no ISAMpe, sendo responsáveis pela execução das aplicações pervasivas. A Figura 8 representa o ambiente pervasivo disponibilizado pelo EXEHDA.

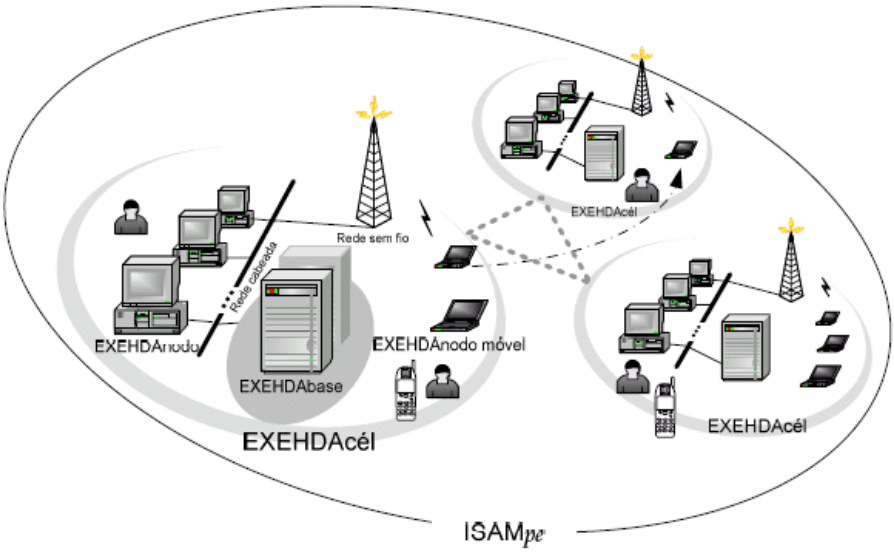

Figura 8. ISAM Pervasive Environment

5.7.5.1. O Núcleo Mínimo

A organização do EXEHDA é baseada em serviços, sendo que um núcleo mínimo do *middleware* tem suas funcionalidades estendidas por serviços carregados sob demanda. Essa organização é baseada em um padrão de projeto referenciado na literatura como *micro-kernel*. Dessa forma, os serviços do EXEHDA estão organizados em um núcleo mínimo e quatro grandes subsistemas: execução distribuída, adaptação, comunicação e acesso pervasivo. Além disso, os serviços carregados sob demanda têm perfil adaptativo, podendo ser utilizada a versão do serviço melhor sintonizada às características do dispositivo.

Um perfil de execução define um conjunto de serviços a ser ativado em um EXEHDAnodo, associando a cada serviço uma implementação específica dentre as disponíveis, bem como definindo parâmetros para sua execução. Adicionalmente, o perfil de execução também controla a política de carga a ser utilizada para um determinado serviço, a qual se traduz em duas opções: (i) quando da ativação do nodo (*bootstrap* do *middleware*); (ii) sob demanda.

Desta maneira, a informação definida nos perfis de execução é também consultada quando da carga de serviços sob demanda, assim, a estratégia adaptativa para carga dos serviços acontece tanto na inicialização do nodo, quanto após este já estar em operação e precisar instalar um novo serviço.

Esta política para carga dos serviços é disponibilizada por um núcleo mínimo (Figura 9), o qual deve ser instalado em todo EXEHDAnodo que for integrado ao ISAMpe, é formado por dois componentes:

- **Profile Manager** – interpreta as informações disponíveis no perfil de execução, que pode ser individualizado para cada EXEHDAnodo, e disponibiliza-as aos outros serviços do *middleware*.
- **Service Manager** – realiza a ativação dos serviços no EXEHDAnodo a partir da informações disponibilizadas pelo *Profile Manager*, carregando sob demanda o código dos serviços a partir de um repositório de serviços, que pode ser local ou remoto.

Assim, as funcionalidades providas pelo EXEHDA são personalizáveis em nível de nodo, sendo determinada pelo perfil de execução. O perfil de execução de cada EXEHDAnodo define o conjunto de serviços a ser ativado e os parâmetros para sua execução, associando a cada serviço uma implementação específica. Adicionalmente, o perfil de execução controla a política de carga de cada serviço, determinando se ele será ativado juntamente com a ativação do EXEHDAnodo (inicialização do *middleware*) ou sob demanda.

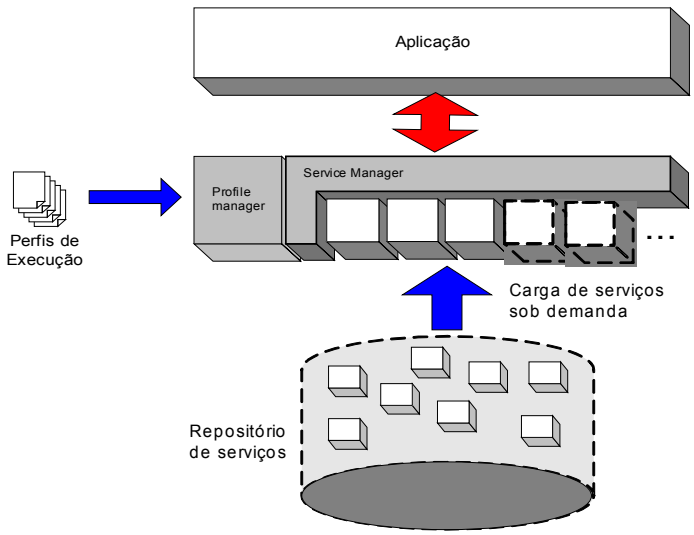

Figura 9. Organização do núcleo do EXEHDA

O perfil de execução de cada EXEHDAnodo é especificado através de um documento XML. Esse documento associa nomes de serviços a componentes que implementam a interface definida para eles. Ainda é possível definir propriedades para determinados serviços, as quais serão recuperadas durante o processamento. No documento XML, um bloco <PROFILE> define um perfil de execução, que tem seu nome definido pelo atributo *name*. Dentro desse bloco, são utilizados blocos <SERVICE> para definir os serviços de farão parte do perfil de execução. Nesse caso, o

atributo *name* define o nome canônico do serviço, enquanto o atributo *loadPolicy* determina o momento de ativação do serviço (*boot* ou *demand*). Por fim, dentro dos blocos <SERVICE> podem ser usados elementos <PROP>, os quais possuem os atributos *name* e *value* que servem para definir propriedades dos serviços, personalizando sua execução.

5.7.5.2. Administração das células

O administrador de uma EXEHDAcel é responsável por: (*i*) manter os recursos de uso compartilhado da célula, definindo suas políticas de acesso; (*ii*) manter a EXEHDAbase; (*iii*) adicionar e remover usuários da EXEHDAcel.

Com o objetivo de facilitar as tarefas do administrador da célula, foi prototipada uma ferramenta para manutenção das células e de seus serviços. Esta ferramenta foi denominada EXEHDA-AMI (EXEHDA *Architecture Management Interface*). A EXEHDA-AMI é constituída por um módulo-base, ao qual podem ser agregados, dinamicamente, outros componentes para ampliar suas funcionalidades.

Entre as funcionalidades oferecidas pela EXEHDA-AMI (Figura 10) estão o suporte à navegação pelas células atualmente ativas no ISAMpe, e o serviço de adição e remoção de usuários. Os usuários podem ser comuns, apenas executam aplicações, ou desenvolvedores, os quais podem instalar e remover aplicações no serviço BDA da EXEHDAcel.

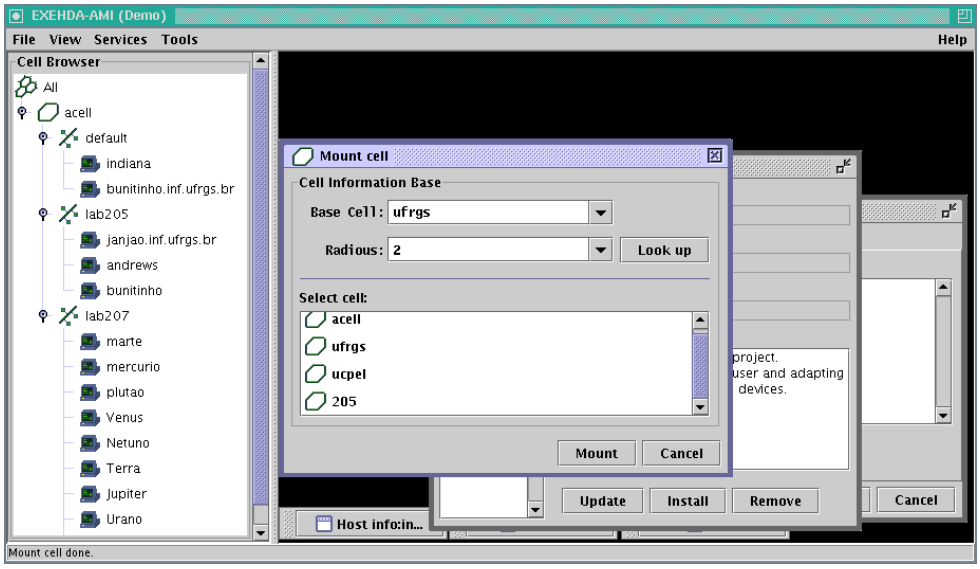

Figura 10. Visão geral da ferramenta EXEHDA-AMI

5.7.6. Organização Baseada em Serviços

Os serviços do EXEHDA estão organizados em quatro grandes subsistemas: execução distribuída, adaptação, comunicação e acesso pervasivo (Figura 11) [Yamin 2004].

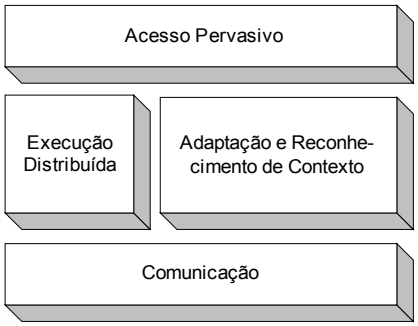

Figura 11. Organização dos subsistemas do EXEHDA

5.7.6.1. Subsistema Execução Distribuída

O Subsistema de Execução Distribuída é responsável pelo suporte ao processamento distribuído no EXEHDA. Esse subsistema interage com outros subsistemas, como o de reconhecimento do contexto e o de adaptação, para promover uma execução efetivamente pervasiva. Os serviços que compõem esse subsistema são listados abaixo.

- **Executor** – possui as funções de disparo de aplicações, e de criação e migração de seus objetos. Na implementação desse serviço é empregada a instalação de código sob demanda, e sua interface define métodos para controle do ciclo de vida das aplicações e dos objetos.
- **Cell Information Base (CIB) –** implementa a base de informação da célula, e sua principal funcionalidade está relacionada à manutenção da infra-estrutura que forma o ISAMpe. Esse serviço mantém os dados estruturais da EXEHDAcel, como informações sobre recursos, vizinhança e atributos que descrevem as aplicações em execução. Esses atributos são registrados pelo serviço *Executor* quando ele realiza o disparo da aplicação, e incluem identificação do proprietário, referência para o código da aplicação, descritor de disparo da aplicação. O CIB ainda é responsável pela manutenção das informações dos usuários registrados na célula, como um certificado assinado para autenticação.
- **OXManager** – sua atribuição é a gerência e a manutenção da meta-informação associada a um OX, conferindo às operações de consulta e atualização dos atributos de um OX o caráter pervasivo necessário para que se possa acessá-los a partir de qualquer nodo do ISAMpe. A abstração OX (Objeto eXehda) é uma instância de objeto criada pelo *Executor*, a qual pode ser associada meta-informação em tempo de execução. Essa abstração provida pelo EXEHDA é útil no mapeamento entre o modelo de computação do ISAMadapt e as funcionalidades oferecidas pelo *middleware*.
- **Discoverer** – é o servido de descoberta de recursos do EXEHDA, sendo responsável pela localização de recursos no ISAMpe a partir de especificações abstratas dos mesmo, as quais caracterizam o recurso a ser descoberto por meio de atributos e seus respectivos valores. Para isso, a interface desse serviço disponibiliza métodos para registro e remoção de recursos, e para pesquisa de recursos que atendam a um determinado critério. Os recursos registrados são catalogados no CIB, ficando visíveis no ISAMpe. Além disso, o registro do recurso deve ser periodicamente

renovado, caso contrário, será automaticamente removido do conjunto de recursos registrados no CIB. Deve-se observar que a localização de um recurso não implica a alocação ou reserva do mesmo. O serviço responsável pela gerência dos recursos é o *ResourceBroker*, abordado a seguir.

- **ResourceBroker** – é responsável pelo controle da alocação de recursos para as aplicações no EXEHDA, atendendo tanto requisições originárias da própria célula quanto oriundas de outras células do ISAMpe. No tratamento das requisições de localização e alocação de recursos, o *ResourceBroker* interage com os serviços *Discoverer* e *Scheduler* para, respectivamente, localização de recursos especializados e alocação de nodos de processamento.
- **Gateway** – é responsável pela intermediação da comunicação entre os nodos externos à célula e os recursos internos a ela. A interação desse serviço com o *ResourceBroker* promove o controle de acesso aos recursos de uma EXEHDAcel. A interface do serviço *Gateway*, o qual mantém uma tabela que registra quais recursos são visíveis por cada aplicação, define três métodos direcionados a concessão, revogação e renovação de permissões de acesso.
- **StdStreams** – provê suporte ao redirecionamento dos *streams* padrões de entrada, saída e erro. Esse serviço associa a cada aplicação um *Console* que agrupa os três *streams* padrões. Assim, podem-se redirecionar esses *streams* sem a necessidade de modificar a aplicação, bastando apenas configurar os atributos de execução do serviço.
- **Logger** – provê a funcionalidade do registro de rastro de execução, e pode ser usado na depuração de programas (fase de desenvolvimento) ou no registro de operações críticas, facilitando o controle da segurança do sistema. A interface desse serviço disponibiliza métodos para registro de mensagens, as quais podem ser classificadas segundo níveis de prioridade.
- **Dynamic Configurator (DC)** – tem a função de realizar a configuração do perfil de execução do *middleware* em um EXEHDAnodo de forma automatizada. Para isso, o nodo deve ser inicializado com um perfil de ativação base que contempla referência ao serviço DC, informado ao *ServiceManager* que deve ser gerado um profile para o nodo. Dessa forma, o serviço DC executa um processo de detecção básico para identificar as características do nodo, as quais incluem informações sobre a JVM e sobre o sistema operacional, configurações de rede, e capacidade de memória. Além disso, o DC pode ser configurado para interagir com o serviço *Collector*, responsável pela aquisição de dados de sensores, para criar um perfil mais específico. Após isso, as informações detectadas pela instância local do DC são enviadas para a instância celular do mesmo, onde são utilizadas para a geração dinâmica do perfil adequado àquele nodo. O perfil gerado é retornado ao *ServiceManager*, que dispara uma nova inicialização do *middleware* considerando a configuração gerada automaticamente.

5.7.6.2. Subsistema Reconhecimento de Contexto e Adaptação

O Subsistema de Reconhecimento de Contexto e Adaptação inclui serviços de aquisição de informações sobre o ISAMpe, de identificação em alto nível dos elementos de

contexto e de disparo das ações de adaptação. Integram esse subsistema os serviços listados abaixo.

- **Collector** – é responsável pela aquisição da informação bruta que, posteriormente, formará os elementos de contexto. Para isso, o *Collector* aglutina informações oriundas de vários monitores (*Monitor*) e as repassa aos consumidores registrados (*MonitoringConsumer*). Esse repasse pode ser feito via *callback* ou via canais de multicast providos pelo serviço *Deflector*. Um *Monitor* gerencia um conjunto de sensores parametrizáveis. Cada sensor contribui com a aquisição de um valor que descreve um aspecto específico, que pode ser estático ou dinâmico, do recurso monitorado. O conjunto de sensores de um EXEHDAnodo, bem como os parâmetros suportados por eles, integra a informação de descrição daquele nodo, disponibilizada no serviço CIB.
- **Deflector** – tem o objetivo de disponibilizar a abstração de canais de multicast para uso na disseminação das informações monitoradas. A interface desse serviço disponibiliza métodos para criação de um canal multicast, inscrição e desinscrição dos consumidores no canal, e para disparar a disseminação de determinada informação no canal.
- **ContextManager** – realiza o tratamento da informação bruta produzida pela monitoração, produzindo as informações abstratas referentes aos elementos de contexto. A definição dos elementos de contexto (objeto *Context*) é parametrizada por uma descrição XML que descreve como o dado referente àquele elemento de contexto deve ser produzido a partir da informação proveniente da monitoração. O método que cria o objeto *Context* também define todos os estados possíveis para o elemento de contexto criado, também com base nas informações da descrição XML. Após a definição de um elemento de contexto, os interessados no recebimento de informações sobre esse contexto devem registrar-se junto ao *ContextManager*. O registro pode ser cancelado quando não houver mais interesse num contexto específico.
- **AdaptEngine** – é responsável pelo controle das adaptações funcionais, provendo facilidades para definição e gerência de comportamentos adaptativos por parte das aplicações. Dessa forma, libera o programador de gerenciar aspectos de mais baixo nível de gerenciamento de contexto. Esse serviço faz a parametrização do *ContextManager*, a partir de definições em XML, para criação dos elementos de contexto de interesse de cada aplicação, e registra seu interesse nas notificações de alteração de estado desses elementos. Em face de uma notificação, o *AdaptEngine* é responsável pela gerência e notificação dos componentes registrados como adaptativos/sensível àquele elemento de contexto cujo estado foi alterado. Outra função desse serviço é prover o mecanismo de carga de código contextualizado, ou seja, selecionar e carregar o código da aplicação que melhor se adapta ao estado do contexto corrente.
- **Scheduler** – é o serviço responsável pelas adaptações não-funcionais, isto é, que não implicam alteração de código. Para isso, o *Scheduler* emprega informações de monitoração do serviço *Collector* para orientar operações de instanciação remota, migração, ou re-escalonamento de código, de acordo com estado dos recursos de processamento.

5.7.6.3. Subsistema Comunicação

O Subsistema de Comunicação disponibiliza mecanismos para atender, principalmente, aspectos relacionados às desconexões, muito comuns em ambiente pervasivos devido tanto à existência de enlaces sem fio como às estratégias de economia de energia dos dispositivos móveis. Fazem parte desse subsistema os serviços relacionados abaixo,

- **Dispatcher** – disponibiliza um modelo de comunicação por troca de mensagens ponto-a-ponto com garantia de entrega e ordenamento, o qual é especializado para operação no ambiente pervasivo. Quando inicializado, o *Dispatcher* atualiza as informações do EXEHDAnodo na CIB, provendo um conjunto de protocolos e endereços que podem ser usados para alcançar o nodo. Durante os períodos de desconexões planejadas, objetivando manter a consistência da comunicação, esse serviço emprega um mecanismo de *checkpointing* do estado dos canais para fazer o tratamento transparente dessas desconexões, procedendo à entrega das mensagens assim que ocorre a reconexão.
- **WORB** – tem o objetivo de simplificar a construção de serviços distribuídos, permitindo ao programador abstrair aspectos de baixo nível relativos ao tratamento das comunicações em rede. Para tanto, o *WORB* oferece um modelo de comunicação baseado em invocações remotas de métodos. Esse serviço é construído sobre o serviço *Dispatcher*, tornando a invocação remota de métodos também sintonizada à premissa de desconexão do ambiente pervasivo.
- **CCManager** – disponibiliza um mecanismo de comunicação baseado na abstração espaço de tuplas, o qual não precisa da coexistência temporal de emissor e receptor. Esse mecanismo atende a demanda de desacoplamento espacial e temporal, causada pela premissa da mobilidade lógica dos componentes que constituem as aplicações pervasivas.

5.7.6.4. Subsistema Acesso Pervasivo

O Subsistema de Acesso Pervasivo tem por finalidade dar suporte à premissa da Computação Pervasiva de acesso em qualquer lugar e todo o tempo a dados e código. Compõem esse subsistema os serviços listados abaixo.

- **BDA** – o serviço Base de Dados pervasiva das Aplicações provê a capacidade de instalação de código sob demanda, que é uma necessidade inerente à execução de aplicações pervasivas. Para isso, esse serviço mantém um repositório de código que fornece a mesma visão de software disponibilizado a partir de qualquer dispositivo do ISAMpe, mesmo após migrações. Além disso, o BDA também suporta o controle de versões, que é importante na manutenção da operacionalidade das aplicações. Assim, as requisições geradas aos EXEHDAnodos de uma determinada célula são sempre direcionadas à instância celular do serviço BDA da mesma EXEHDAcel. Essa, se necessário, realiza o acesso a instâncias remotas (BDAs de outras células).
- **AVU** – é atribuição do serviço Ambiente Virtual do Usuário à manutenção do acesso pervasivo ao ambiente computacional do usuário, o qual compreende aplicações em execução, informações de personalização das aplicações, conjunto de aplicações instaladas e seus arquivos privados. Com

esse objetivo, o AVU adota uma estratégia de utilização análoga a do BDA, ou seja, as requisições geradas pela instância local do EXEHDAnodo são direcionadas à instância celular do serviço. A instância celular, por sua vez, consulta a célula *home* do usuário de forma a descobrir a última localização física de seu ambiente virtual, e acessá-lo para conclusão da requisição. O AVU ainda inclui operações *prefetch* e *release* para otimização do acesso aos dados armazenados face aos aspectos de mobilidade e desconexão planejada.

- **SessionManager** – é responsável pela gerência da sessão de trabalho do usuário, que é o conjunto de aplicações correntemente em execução para aquele usuário. A informação que descreve o estado da sessão de trabalho é armazenada no AVU, estando disponível de forma pervasiva. Esse serviço trabalha com objetos *SessionDescriptor*, o qual representa uma sessão e define métodos para inclusão e remoção de aplicações. Além disso, o *SessionManager* disponibiliza métodos para salvamento e recuperação de sessões. Normalmente, esse serviço é ativado através do serviço *GateKeeper*, após o procedimento de autenticação do usuário.
- **GateKeeper** – é responsável por intermediar o acesso entre entidades externas à plataforma ISAM e os serviços do *middleware*, realizando os procedimentos de autenticação necessários. O protocolo de autenticação baseia-se em um mecanismo de chave pública/privada, sendo a chave pública disponibilizada na CIB e a privada armazenada em um meio portátil do usuário. Em caso de sucesso na autenticação, um identificador de sessão é retornado, pelo qual o usuário poderá disparar as aplicações. O método *logout* permite invalidar um identificador de sessão.

5.7.7. Gerenciamento da execução de aplicações no EXEHDA

Como descrito anteriormente, uma característica presente na Computação Pervasiva é o deslocamento do usuário portando ou não seu dispositivo móvel, mantendo o acesso ao seu ambiente computacional. A seguir serão apresentados os mecanismos do EXEHDA que permitem a semântica *siga-me*.

Cada usuário com uma sessão de trabalho ativa está apto a executar as aplicações de seu interesse. Assim, os comandos de controle de sessão são necessários para viabilizar a implementação da semântica *siga-me* das aplicações. Os comandos de controle de sessão podem ser manipulados através da aplicação ISAM Desktop, e os principais são descritos abaixo.

- **Login/Logout** – o comando *login* é responsável pela autenticação do usuário junto ao serviço *Gatekeeper*. Esse procedimento é baseado num mecanismo de chave pública/privada, na qual a chave privada é mantida pelo usuário em um meio de armazenamento portátil, enquanto que a chave pública é armazenada na forma de um certificado no serviço CIB da célula onde o usuário foi cadastrado. No caso de o usuário efetuar *login* em outra EXEHDAcel, o CIB fará a busca transparente do certificado do usuário. O resultado de uma operação *login* realizada com sucesso é a ativação da sessão padrão do usuário, recolocando em operação as aplicações anteriormente interrompidas, caso existam. Geralmente, a sessão padrão inclui a aplicação ISAM Desktop, a qual permite acesso às demais aplicações instaladas no ambiente virtual do usuário. Por sua vez, o comando l*ogout* libera os recursos empregados na gerência dos contextos de execução das aplicações que integram a sessão ativa do usuário. Esse comando implica

operação dos serviços *ContextManager* e *AdaptEngine*, além de armazenar o estado da sessão no AVU para recuperação na próxima vez que o usuário entrar no sistema.

- **Save/Restore Session** – permite ao usuário possuir várias sessões além da sessão padrão. O comando *save session* permite mover a sessão padrão (atual) para uma sessão alternativa, armazenando seu estado. Uma sessão alternativa armazenada no AVU pode ser restaurada com o comando *restore session*. Esses comandos estão relacionados ao serviço *SessionManager*.
- **Disconnect/Reconnect** – correspondem a chamadas ao serviço *AdaptEngine*, atuando sobre o estado de conectividade do dispositivo do usuário, através do disparo do procedimentos associados à desconexão planejada.
- No EXEHDA, toda a aplicação é caracterizada pelo seu descritor de disparo, que é um documento XML gerado durante o desenvolvimento da aplicação. Esse descritor agrupa metadados que habilitam a execução da aplicação a partir de EXEHDAnodos em qualquer EXEHDAcel que integra o ISAMpe. Entre os metadados estão uma descrição da funcionalidade da aplicação, o desenvolvedor, os parâmetros fixos utilizados pela aplicação, e uma referência, independente de localização (BDA), ao arquivo JAR que contém as classes da aplicação.

Considerando que o EXEHDAnodo no qual está sendo disparada a aplicação não pertence à EXEHDAcel onde essa está armazenada, será necessário um acesso intercelular, que é realizado de forma transparente pelo serviço BDA do *middleware*. Esse comportamento operacional viabiliza, no que se refere a acesso a código, a utilização da semântica *siga-me*.

O disparo de aplicações pode ser feito de maneira manual, através da aplicação *isam-run*, a qual não é uma aplicação pervasiva, portanto, não possui as características definidas para esse tipo de aplicação. Assim, o *isam-run* deve ser usado para disparar uma aplicação pervasiva que habilite ao usuário manipular seu AVU.

Normalmente, a primeira aplicação efetivamente pervasiva a ser disparada é o utilitário ISAM Desktop, cuja interface gráfica é adaptada ao tipo de dispositivo em uso. O ISAM Desktop provê acesso às aplicações instaladas pelo usuário em seu ambiente virtual. A instalação dessas aplicações é feita adicionando-se o respectivo descritor de disparo no AVU, e inserindo-se uma entrada para a aplicação no arquivo *contents.xml*, o qual informa ao ISAM

Desktop sobre as aplicação instaladas.

5.8. Conclusões

Os principais objetivos da computação pervasiva são tornar o uso do computador transparente ao usuário, diferente de como é feito hoje, onde o homem tem que ligar, operar e desligar as máquinas. Na computação pervasiva, o homem seria inundado por tantos dispositivos (*appliances*) que ele estaria interagindo mesmo sem perceber. Para isso é necessário que a Computação crie um ambiente amigável de tal forma que se torne invisível ao usuário – não está em seu foco.

As restrições naturais do ambiente móvel colocam novos desafios para os projetistas de aplicações e exigem novas tecnologias para que as aplicações sejam úteis em sistemas com recursos limitados. Sente-se a necessidade de sistemas mais flexíveis que dividam a responsabilidade entre o projetista da aplicação e o sistema de suporte

(*middleware*) para fornecer o comportamento dinâmico e adaptativo que a aplicação requer.

Em nossa visão, os conceitos e tecnologias da Computação em Grade podem contribuir para construir e gerenciar o ambiente pervasivo, especialmente para promover a mobilidade total e implementar a semântica siga-me para as aplicações.

Referências

Al-Myhtadi, J. et al (2004) "Super Spaces: a Middleware for Large-Scale Pervasive Computing Environments". *2nd IEEE Annual Conference on Pervasive Computing and Communications Workshops* (PERCOMW'04), New York.

Augustin, I. et al (2005) "Managing the Follow-me Semantics to Build Large-scale Pervasive Applications". In *Middleware Conference 2005 - Workshop on Middleware for Ad-hoc and Pervasive Computing*, Grenoble, France.

Augustin, I. (2004) "Abstrações para uma Linguagem de Programação visando Aplicações Móveis em um Ambiente de Pervasive Computing". Tese de Doutorado, II/UFRGS, Porto Alegre, janeiro.

Augustin, I. et al. (2004a) "ISAM, Joing Context-awareness and Mobility to Building Pervasive Applications", *Mobile Computing Handbook*. Mahgoub, I. and Ilyas, M. Editors, CRC Press, New York.

Augustin, I et al. (2002) "Towards Taxonomy for Mobile Applications with Adaptive Behavior". In *International Symposium on Parallel and Distributed Computing and Networks* (PDCN 2002), Innsbruck, Austria. feb.

Banavar, G. et al. (2004) "An Authoring Technology for Multidevice Web Applications". *IEEE Pervasive Computing: Special Report on Emerging Technologies for Pervasive and Mobile Computing* . New York, dec.

Cardelli, L. and Gordon, A. D. (1998) "Mobile Ambients". In First International Conference on Foundations of Software Science and Computation Structure, mar. p.140-155.

Chalmers, D. et al. (2006) "Ubiquitous Computing: Experience, Design and Science", http://www-dse.doc.ic.ac.uk/Projects/UbiNet/GC/index.html, abril.

Chemiack, M. and Franklin, M. J. and Zdonik, S. B. (2001) "Data Management for Pervasive Computing". *VLDB*, available DBLP, http://dblp.uni-trier.de.

Davies, N. and Friday, A. and Storz, O. (2004) "Exploring the Grid's Potential for Ubiquitous Computing", IEEE Pervasive Computing, vol. 3(2), pp. 74 – 75.

de Roure, D. and Jenings, N.R. and Shadbolt, N. (2003) "The Evolution of the Grid". In BERMNA, F. & FOX, G. & HET, T. (Eds.) Grid Computing: Making the Global Infrastructure a Reality. New York:Wiley & Sons.

Filho, A. E. S. et al. (2005) "Serviço Adaptativo para Descoberta de Recursos em Larga Escala na Arquitetura ISAM." Simpósio de Redes de Computadores. Fortaleza, CE, maio,.

Foster, I. et al (2002) "The Physiology of the Grid: an Open Grid Services Architecture for Distributed Systems Integration". Open Grid Service Infrastructure WG, Global Grid Forum, june.

Foster, I. and Kesselman, C. (2001) "The Anatomy of the Grid: Enabling Scalable Virtual Organizations". International Journal and Supercomputing Applications, 15(3).

Fugetta, A. and Picco, G. P. and Vigna, G. (1998) "Understanding Code Mobility". *IEEE Transactions on Software Engineering* , v.24, n.5, may.

Garlan, D. and Steenkiste, P. and Schmerl, B. (2002) "Project Aura: Toward Distraction-free Pervasive Computing". In *IEEE Pervasive Computing*, New York, v.1, n.3, sept.

Goble, C. and De Roure, D. (2004) "The Semantic Grid: Myth Busting and Bridge Building".

Graf, M. (2003) "Fluid Computing". *ERCIM News - Special Theme: Applications and Service Platforms for the Mobile User* . New York, jul.

Hac, A. (2003) "Wireless Sensor Network Designs", John Wiley & Sons, dec., 391 p.

IEEE Internet Computing (2004) Special Issue on Wireless Grid, v. 8, n.4, jul-aug.

INFOEXAME (2007) "Smartphones – porque é a hora de comprar um e aposentar seu celular". n. 257, agosto.

Jansen, E. et al (2005) "A Programming Model for Pervasive Spaces", In *International Conference on Service-Oriented Computing*, Netherlands, dec.

Jul, E. et al. (1988) "Fine-Grained Mobility in the Emerald Systems". *ACM Transactions on Computer Systems* , v.6, n.2, feb.

McKnight, L. W.; Gaynor, M. (2003) "Wireless Grid Issues: . 8th Global Grid Forum, 2003. Disponível em http://www.wirelessgrids.net/docs.

Pereira, M. R; Amorim, C. L e Castro, M. C. S. "Tutorial sobre Redes de Sensores".

Ranganathan, A. et al (2005) "Towards a Pervasive Computing Benchmark". In *3rd International Conference on Pervasive Computing and Communications Workshops* (PerCom), New York, IEEE Computer Society.

Roman, M. et al. (2002) "Gaia: a Middleware Infrastructure to Enable Active Spaces", *IEEE Pervasive Computing*, New York, v.1, n. 4, dec.

Saha, D. and Mukherjee, A. (2003) "Pervasive Computing: a Paradigm for the 21st Century", *IEEE Computer*, v.36, n.3, p.25-31, mar.

Satyanarayanan, M. (2001) "Pervasive Computing: Vision and Challenges", *IEEE Personal Communications*, New York.

Smith, J. M. (1988) "A Survey of Process Migration Mechanisms". *ACM SIGOP Operating Systems Review*, v.22, n.3, jul.

Storz, O and Friday, A. and Davies, N. (2003) "Towards 'Ubiquitous' Ubiquitous Computing: an alliance with the Grid". System Support for Ubiquitous Computing Workshop at the Fifth Annual Conference on Ubiquitous Computing (UbiComp 2003), Seattle, October 2003. CSEG/2/03, April 2003.

Thain, D. ; Tannembaum, T.; Livny, M. (2003) "" Condor and the Grid". In F. Berman, G. Fox, and T. Hey, editors, Grid Computing: Making The Global Infrastructure a Reality. John Wiley, 2003.

The Yankee Group (2004) “Divergent Approach to Fixed/Mobile Convergence”. November.

Vigna, G. (1998) “Mobile Code, Technologies, Paradigms, and Applications”. Tesi di Dottorato, Politecnico di Milano, Italy.

Weiser, M. (1991) “The Computer of the 21st Century”. *Scientific American*, New York, v.265, n.9, sep.

Wikipedia - Mobile Agent. (2007). http://en.wikipedia.org/wiki/Mobile_agent, agosto.

Wikipedia – Convergência Tecnológica (2007). agosto.

Yamin, A. et al. (2003) “Towards Merging Context-aware, Mobile and Grid Computing”. Journal of High Performance Computing Applications. London: Sage Publications.

Yamin, A. C. (2004) “Arquitetura para um Ambiente de Grade Computacional direcionado às Aplicações Distribuídas, Móveis e Conscientes do Contexto da Computação Pervasiva”, Tese de Doutorado, II/UFRGS, Porto Alegre, agosto.