3
Gerência de Contexto

A informação contextual pode ser adquirida através de diferentes métodos. Por exemplo, algumas aplicações esperam que o usuário explicite uma informação relevante. Outras aplicações mais específicas fazem uso de sensores que podem captar localização, orientação e situação. Alguns agentes de aplicação pessoal inteligente também podem ser usados para capturar os hábitos potenciais do usuário.

Com o intuito de retirar a sobrecarga imposta pelas funcionalidades relacionadas à gerência de contexto, este trabalho propõe o projeto de uma estrutura independente e reutilizável por diferentes sistemas, em particular os sistemas hipermídia.

No capítulo anterior foi possível obter uma visão mais detalhada de conceitos diretamente relacionados à computação orientada a contexto. Também foi possível conhecer as principais questões em torno dos mecanismos de adaptação presentes em alguns sistemas hipermídia, bem como entender a importância do contexto para orientar as estratégias de adaptação.

Este capítulo tem como objetivo apresentar detalhadamente alguns aspectos importantes considerados de forma a permitir diversas implementações desta arquitetura. Um desses aspectos diz respeito à representação das estruturas de dados com um formato aberto e extensível que permita incorporações futuras de novos atributos ou, até mesmo, a descontinuação de algum deles, sem impactos para as aplicações que consomem essa informação contextual. Outro aspecto importante a ser considerado é o estabelecimento de um protocolo de comunicação aberto a ser utilizado tanto na coleta, como na disseminação da informação de contexto.

Essa arquitetura baseia-se na existência de um servidor de contexto, procurando promover uma reutilização da informação de contexto e reduzir o tamanho da mensagem de coleta da informação de contexto.

Os aspectos citados fazem parte da especificação de uma aplicação orientada ao contexto. Quando o foco é especificamente gerência de contexto, esses aspectos ganham maior relevância, pois não só podem contribuir para aumentar a eficácia dos resultados de consultas, como para aumentar a reutilização da estrutura por diferentes tipos de aplicações, em particular os sistemas hipermídia.

3.1. Representação da Informação de Contexto

3.1.1. RDF e CC/PP

A proposta de uma arquitetura para gerência de contexto envolve funções de coleta, armazenamento e disseminação da informação de contexto para aplicações interessadas. Então, a questão que surge é: como criar uma arquitetura que não esteja restrita a um formato de dados para representação da informação contextual? Ou em outras palavras, que forma de representação de dados utilizar para o intercâmbio e armazenamento da informação de contexto?

Ao analisar a informação de contexto em mais detalhes, percebe-se que ela não contém apenas dados estruturados. Outra característica importante é que ela pode ser obtida a partir de diferentes fontes de dados, dificultando sua recuperação. Por fim, um outro aspecto que merece destaque é o fato dessa informação trazer embutida consigo semântica dos dados, muito relevante aos mecanismos de adaptação dos sistemas hipermídia, especificamente no contexto deste trabalho.

Como arena de integração de diferentes sistemas, é possível estabelecer um paralelo entre a informação contextual e a segunda geração da *World Wide Web* (WWW, Web), denominada *Web Semântica* (Berners-Lee, 1999), uma vez que ambas envolvem não apenas o arranjo das idéias, mas das associações entre essas idéias. Para que a informação contextual também possa ser plenamente utilizada por diferentes aplicações, sua estruturação não pode estar limitada a hierarquias rígidas e restritivas, como por exemplo a um banco de dados, com colunas de informação relacionadas umas às outras.

Para representar, compartilhar e integrar a informação contextual, a instanciação deste framework optou por utilizar uma plataforma aberta e extensível composta pelo RDF (*Resource Description Framework*) e pelo CC/PP (*Composite Capability/Preference Profiles*), padrões recomendados pelo W3C (*World Wide Web Consortium*) que possuem esse propósito.

Com base nesses objetivos, a organização dessa informação pode ser vista em três níveis (Figura 3.1): o primeiro nível (**nível de dados**) provê um modelo de dados para descrever interrelacionamentos entre recursos, através de propriedades (atributos e relacionamentos) e valores a elas associados, muito semelhante a um modelo de entidades e relacionamentos. Entretanto, o nível de dados não provê mecanismos para a declaração dessas propriedades, nem para definir relacionamentos entre essas propriedades e outros recursos. Essa é a função do segundo nível, também conhecido como o **nível de esquema**.

O primeiro e segundo nível são constituídos pela tecnologia RDF, definida em detalhes nos documentos: Resource Description Framework (RDF) Model and Syntax Specification (RDF, 1999), que descreve o modelo de dados RDF, e Resource Description Framework (RDF) Schema Specification (RDF Schema, 2003), que trata das primitivas de modelagem utilizadas para a descrição de um domínio particular de interesse.

O **nível lógico** é a camada que fornece um entendimento comum e compartilhado de um domínio específico. É formada por linguagens denominadas ontologias – metadados ("dados que descrevem dados" mais complexos) que representam explicitamente a semântica dos dados, de forma processável por máquina. As ontologias se propõem a auxiliar pessoas e computadores a acessarem informações na forma em que precisam, assim como comunicarem-se entre si de forma mais efetiva.

Composite Capabilities/Preference Profiles (CC/PP) (CCPP, 1999b) é um *framework* proposto pelo W3C (*World Wide Web Consortium*) e utilizado neste trabalho para organizar o terceiro nível da informação contextual. Um perfil CC/PP é uma descrição das capacidades de um dispositivo, ou das preferências de um usuário, que podem ser usadas para adaptar a apresentação de conteúdo, de acordo com o dispositivo utilizado e com o usuário que navega através do documento.

Baseado em RDF (Resource Description Framework), um perfil CC/PP contém um determinado número de componentes (*Hardware*, Sistema Operacional ou Aplicação) e cada componente possui pelo menos um ou mais atributos (versão, nome, etc). O conjunto de atributos de componentes e seus valores constituem um vocabulário utilizado para validação do perfil CC/PP.

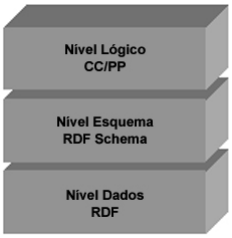

Figura 3.1 – Níveis de estruturação da informação contextual

Algumas informações presentes em um perfil podem ser consideradas sensíveis. O uso do CC/PP por si só não esgota essa questão, sendo importante a existência de mecanismos de segurança que possam proteger a privacidade do usuário.

O padrão da W3C, P3P (Projeto para Plataforma de Preferências de Privacidade ou *Platform for Privacy Preferences Project*) (P3P) permite que sites divulguem suas práticas de privacidade em formato padronizado, de modo que seja possível a interpretação destas normas por agentes de *softwares* heterogêneos bem como por seus usuários.

Classificada como uma recomendação pelo W3C em Abril de 2002, a versão 1.0 do padrão P3P tem como objetivo principal automatizar a tomada de decisão com relação às preferências dos usuários, facilitando a garantia de níveis de privacidade aceitáveis e permitindo que as aplicações troquem informações pessoais somente dentro do especificado por seus proprietários ou responsáveis. Como esse assunto diz respeito a uma aplicação de maior espectro, não faz parte do objetivo deste trabalho abordá-lo.

É importante ressaltar que o uso do CC/PP é estendido na instanciação desta arquitetura para gerência de contexto, descrevendo não só capacidades de dispositivos clientes e preferências dos usuários, mas também características da rede de comunicação e do(s) servidor(es) de conteúdo, como apresentam as Seções 3.1.2, 3.1.3 e 3.1.4.

Outro detalhe desta instanciação é o fato de que ela não se baseia na existência de perfis de referência, mas também explora o CC/PP como uma

estrutura para armazenamento de dados em um servidor de contexto centralizado e uma estrutura para intercâmbio da informação contextual, como será mencionado nas Seções 3.2 e 3.3, respectivamente.

3.1.2. Estrutura Básica do CC/PP

O perfil CC/PP está estruturado em dois níveis de hierarquia:

- um perfil com um conjunto de *componentes*, e
- cada componente com um conjunto de *atributos*.

Um componente pode ser usado para agrupar parâmetros relacionados, descrevendo, por exemplo, as características do *hardware* de um determinado dispositivo ou servidor. Um componente pode conter um ou mais atributos associados a valores atômicos, ou a uma lista de valores, ou ainda a valores estruturados de maneira mais complexa. Por exemplo, um componente que descreva as características de hardware de um dispositivo pode conter um atributo que indique se o dispositivo provê ou não suporte a áudio.

O CC/PP define sua estrutura usando o Esquema RDF (RDF Schema, 2003). O vocabulário CC/PP define um conjunto de componentes e atributos relacionados, muito embora ainda não tenha sido adotado nenhum padrão para um vocabulário em particular. Em função de determinadas necessidades, algumas aplicações e organizações têm definido um vocabulário próprio. Mais uma vez, para simplificação deste texto, maiores detalhes pertinentes à estrutura do CC/PP podem ser encontrados no Apêndice A.

Uma particularidade do CC/PP, que convém ser mencionada, é o fato de que ainda não há uma definição padrão de um protocolo de comunicação para transportar uma instância de um vocabulário CC/PP. Entretanto, como consta da especificação, o HTTP é o protocolo de comunicação recomendado. A Seção 3.3 discutirá algumas outras possibilidades para adoção de protocolos de comunicação similares ao HTTP, muitos deles compostos por uma estrutura básica compreendendo cabeçalho e corpo, sendo esse último ideal para transporte das descrições CC/PP, e implementando métodos que permitem um diálogo através de um pequeno número de mensagens.

A concepção do CC/PP baseado no RDF apresenta algumas vantagens. Uma delas refere-se ao fato de que um perfil codificado em CC/PP pode referenciar atributos de diferentes vocabulários. A base RDF evita a necessidade de definição de um super esquema para os dados contidos em um perfil. Outra vantagem provida pelo RDF como modelo de dados é a facilidade de mapear dados em qualquer novo formato.

A seção seguinte apresenta detalhes da vantagem quanto ao uso de vocabulários múltiplos. A possibilidade de utilização de espaços de nomes XML pode ajudar a evitar a proliferação de vocabulários, redefinindo atributos já definidos e padronizados por outras comunidades.

3.1.3. Extensibilidade e Espaços de Nomes

O CC/PP pode ser estendido através da criação de novos vocabulários. Qualquer aplicação ou ambiente operacional que use o CC/PP pode definir seu próprio vocabulário, mas para que haja uma interoperabilidade mais ampla, são necessárias definições de vocabulários de uso mais geral.

Por exemplo, pode haver um vocabulário estendido do padrão para descrição de dispositivos de imagem, para descrição de dispositivos de mensagem de voz, para descrição de dispositivos para acesso sem fio etc. Da mesma forma, a especificação (CCPP, 2003) define um pequeno vocabulário central de características que se aplicam a agentes de impressão e exibição, onde o uso, quando apropriado, é extremamente recomendável. Esse vocabulário central é baseado em uma especificação do IETF (Masinter et al.,1999) e exemplifica como vocabulários de atributos CC/PP podem ser definidos.

Qualquer expressão CC/PP pode usar termos provenientes de vocabulários diferentes, não havendo restrições quanto ao reuso desses termos a partir de vocabulários existentes, ao invés da criação de nomes novos para definir a mesma informação.

XML *Namespaces* são uma forma simples e direta de qualificar nomes de elementos e atributos usados em documentos XML associando-os através de seus prefixos a espaços de nomes identificados por alguma URI (*Uniform Resource Identifier*). Os espaços de nomes são freqüentemente utilizados para prover

vocabulários onde os elementos e os atributos têm um conceito associado e são usados por várias instâncias de documentos XML.

Apesar da suposição de que um espaço de nomes tem semântica associada, incluindo também a associação com esquema, não há nenhuma convenção que estabeleça que um espaço de nomes referenciado por uma URI tenha um esquema correspondente.

Outra consideração importante é com relação aos prefixos dos espaços de nomes: as aplicações não devem assumir, por exemplo, que o prefixo "rdf:" refere-se ao vocabulário RDF ou "ccpp:" refere-se ao vocabulário CC/PP e assim por diante. É a URI ao qual o prefixo está atrelado o que importa. No exemplo da Figura 3.2, os elementos prefixados com "xdc:" estão associados ao espaço de nomes cuja URI é http://www.books.com, enquanto que os que estão prefixados com "h:" estão associados ao espaço de nomes cuja URI é http://www.w3.org/TR/REC-html40. A declaração do espaço de nomes fornecida pela Figura 3.2 foi feita de forma explícita, onde os prefixos "h:" e "xdc:" são atalhos para os nomes completos de seus respectivos espaços de nomes.

```
<h:html xmlns:h="http://www.w3.org/TR/REC-html40"
        xmlns:xdc="http://www.books.com">
  <h:head><h:title>Book Review</h:title></h:head>
  <h:body>
     <xdc:bookreview>
     <xdc:title>XML: A Primer</xdc:title>
     <h:table>
       <h:tr align="center">
         <h:td>Author</h:td><h:td>Price</h:td>
         <h:td>Pages</h:td><h:td>Date</h:td></h:tr>
       <h:tr align="left">
         <h:td><xdc:author>Simon St. Laurent</xdc:author></h:td>
         <h:td><xdc:price>31.98</xdc:price></h:td>
         <h:td><xdc:pages>352</xdc:pages></h:td>
         <h:td><xdc:date>1998/01</xdc:date></h:td>
       </h:tr>
      </h:table>
     </xdc:bookreview>
  </h:body>
</h:html>
```

Figura 3.2 – Espaços de nomes com XML Namespaces

O escopo das declarações do espaço de nomes está relacionado ao escopo do elemento que contém essa declaração e a todos os seus descendentes, podendo ser redefinido em declarações aninhadas (XML Names, 1999).

Em resumo, embora não defina um mecanismo para intercâmbio de informações entre as aplicações, os espaços de nomes XML são uma forma de garantir que não haja colisões acidentais entre nomes de atributos. Através da

utilização de espaço de nomes, o perfil CC/PP consegue fazer uso de um ou mais vocabulários e, também, de versões múltiplas de um mesmo vocabulário.

3.1.4. Vocabulários

Como já mencionado, o CC/PP foi projetado para ser usado no âmbito de uma aplicação ou ambiente operacional mais amplo e, dessa forma, cabe ao projetista da aplicação especificar: o protocolo de intercâmbio, os modelos de segurança e privacidade, o vocabulário adotado e as regras para processamento do vocabulário.

Quanto ao vocabulário propriamente dito, na especificação (CCPP, 2003) são feitas algumas considerações mais detalhadas no que diz respeito a tipos de dados e possíveis representações de valor dos atributos.

De forma resumida, a especificação Estrutura e Vocabulários CC/PP (CCPP, 2003) estabelece que todo atributo CC/PP pode ser classificado como um atributo do tipo simples ou complexo. Os atributos do tipo simples são aqueles onde os valores são representados por literais, tais como valores descritos através de URIs, valores do tipo texto, números inteiros etc. Os atributos do tipo complexo são aqueles representados por uma coleção de propriedades RDF e valores associados, cuja ordem em que aparecem pode ser significativa ou não.

A especificação CC/PP não impõe limites quanto ao número de vocabulários que podem ser criados e utilizados em paralelo. Cada aplicação irá determinar o conjunto de atributos específicos do seu contexto. O RDF permite a extensibilidade e a interoperabilidade dos diversos vocabulários através dos espaços de nomes XML, como foi visto na Seção 3.1.3.

A arquitetura proposta para o gerente de contexto permite a incorporação de qualquer vocabulário como alguns existentes e analisados nas próximas seções. Entretanto, a instantiação do framework apresentada neste trabalho optou pelo uso de um vocabulário novo e particular, contendo atributos mais significativos para orientar os mecanismos de adaptação presentes em sistemas hipermídias, que consiga descrever não apenas dispositivos cliente, mas outras plataformas que compõem o conceito de contexto descrito por esta dissertação.

3.1.4.1.
Registro de característica de mídia (CONNEG) do IETF

O IETF (*Internet Engineering Task Force*) definiu duas publicações, (RFC2506) e (RFC2533), descrevendo características de mídia para cliente e servidor. Em seguida, definiu um pequeno vocabulário (Masinter et al.,1999) representado em CC/PP, formando uma base para a definição de características do dispositivo cliente.

A deficiência encontrada nesse vocabulário foi que apesar de toda característica de mídia e valores poderem ser representados em CC/PP, nem todo perfil CC/PP pode ser expresso através desse conjunto de propriedades e valores específicos.

A atual versão da especificação CC/PP (CCPP, 2003) não está adaptada para fazer comparação de valores. Comparações que utilizam operadores tais como >=, < etc., e combinações de expressões booleanas, como por exemplo, altura = 640 OR altura = 480 ainda não são previstas.

3.1.4.2.
WAP UAProf

O UAProf (WAP-UAProf) é uma especificação definida pelo grupo WAP Fórum, onde o objetivo é descrever as características de dispositivos móveis, sem fio. Sua representação também é baseada em RDF e, portanto, seu vocabulário utiliza o mesmo formato básico que o CC/PP.

Nesse vocabulário, cada propriedade do agente usuário pertence a um dos componentes que o formam, a saber: plataforma de hardware, software, características WAP, browser do agente usuário e características de rede.

As questões em torno da adoção desse vocabulário estão relacionadas, primeiramente, ao fato de que esse é um vocabulário muito especializado para um tipo de dispositivo cliente, isto é, dispositivos celulares móveis. Em outras palavras, esse vocabulário carece de algumas propriedades presentes em outros dispositivos clientes, como, por exemplo, tipos de memória, número de processadores etc. Além disso, o UAProf só representa características de cliente em detrimento de características de servidores, características da rede de acesso e preferências do usuário.

A segunda questão, ainda não resolvida pelo Esquema RDF, é a falta da representação do tipo da propriedade, isto é, se é numérica, decimal etc.

Uma terceira questão envolvida é quanto ao processo de resolução dos perfis onde o CC/PP não estabelece uma especificação formal para o tratamento dos perfis. A resolução dos perfis é o processo de montagem de um perfil final, resultante da junção do perfil de referência e do perfil que descreve apenas os atributos, onde os valores associados foram modificados. O UAProf define regras que dependem da seqüência como esses atributos estão descritos, embutindo algumas dessas regras no campo *comentário* de cada propriedade. Por exemplo, se o campo comentário contiver a palavra "*locked*", significa que o que vale é o primeiro valor assinalado para um determinado atributo; ou, se tiver a palavra "*override*", o que vale é o último valor atribuído.

Por fim, foi possível observar que o UAProf também não endereça o problema relacionado à definição de limites de valores, como por exemplo, permitir representar o atributo velocidade de conexão com valores entre14400 e 28800 bps.

O documento (WAP-UAProf) define detalhadamente a estrutura desse esquema no que diz respeito à definição de classe e semântica de atributos para dispositivos voltados para o padrão WAP.

3.1.4.3. FIPA

A FIPA (*Foundation For Intelligent Physical Agents*) também desenvolveu um vocabulário (uma linguagem de conteúdo) em RDF (FIPA, 2001) . A FIPA é uma organização internacional dedicada a promover a indústria de agentes inteligentes através do desenvolvimento de especificações que provêem suporte à interoperabilidade entre os agentes e aplicações baseadas em agentes. A ontologia FIPA pode ser usada pelos agentes para trocarem informação sobre o perfil do dispositivo de cada um deles. Em outras palavras, as características dos diferentes dispositivos são expressas através do uso de uma ontologia, através da qual os perfis são validados.

As restrições em relação à adoção desse vocabulário mais uma vez estão relacionadas ao fato de que a ontologia proposta diz respeito somente a

características de dispositivos, deixando de abranger outros atributos relevantes para adaptação em sistemas hipermídia. Além disso, o vocabulário proposto é constituído de um número reduzido de atributos e, portanto, não descreve todas as características dos dispositivos indispensáveis ao escopo deste trabalho.

3.1.4.4. Vocabulário adotado pelo Gerente de Contexto

Vários fatores que influenciaram este trabalho quanto à criação de um vocabulário particular (*Hypermedia Context Vocabulary*) são discutidos ao longo desta seção. A despeito da proposta de um novo vocabulário, o Capítulo 4 apresentará a arquitetura genérica modelada através de dois *frameworks* e inteiramente independente da escolha de um determinado vocabulário, específico ao domínio de uma única aplicação.

O primeiro fator que influenciou na proposição de um novo vocabulário foi o fato de que os mecanismos de adaptação necessários à abordagem de uma estratégia mista (adaptação no cliente e no servidor) precisavam de descrições que não estivessem restritas à plataforma cliente, em especial de um tipo em particular. Foi observado que para alimentar tais mecanismos, também são necessárias informações que descrevam as características dos múltiplos servidores de conteúdo, onde os objetos de um documento podem estar localizados, desde as características da rede de comunicação utilizada (ex: velocidade da conexão, perdas de pacotes etc.) até as preferências particulares de cada usuário (ex: em relação à utilização de um tipo de plataforma cliente ou uma aplicação, últimos elos (*links*) navegados em determinado documento etc.).

O segundo fator concentrou-se nos parâmetros mais relevantes aos sistemas hipermídia em geral, ou seja, os parâmetros foram selecionados com objetivo de orientar os mecanismos de adaptação localizados nos ambientes de armazenamento e de exibição, com demanda de informação contextual específicas ou não.

Outro fator que influenciou a proposta de um novo vocabulário diz respeito às novas entidades que suportam o requisito de adaptação encontrado no modelo NCM (Soares et al., 2003a). Essas entidades encontram-se refletidas na linguagem de autoria declarativa NCL (Muchaluat-Saade 2003), através de suas onze áreas

funcionais, em particular a *Presentation Control*, que permite especificar alternativas de conteúdo, de apresentação e de navegação através do elemento nó *switch*. A escolha de uma das alternativas de conteúdo se baseia em um conjunto de atributos, que também exerceram uma influência importante para a criação de um novo vocabulário.

Finalmente, o vocabulário adotado neste trabalho tomou como base a sintaxe XML descrita na especificação (RDF Primer, 2003), que introduz algumas potencialidades sintáticas adicionais como, a possibillidade de definição de atributos estruturados. Atributo estruturado é um atributo do tipo composto, formado pelo aninhamento de atributos simples e atributos compostos, recursivamente. Por exemplo, suponha o atributo memória disponível composto pelos seguintes atributos: tipo (ex: RAM), tamanho (ex: 256) e unidade (ex: MB).

O vocabulário proposto é divido em quatros perfis: perfil do dispositivo, perfil do usuário, perfil do servidor e perfil da rede, como apresentado pelo Capítulo 1. Para cada um destes perfis, foi construído um esquema que detalha o conjunto de atributos de cada um de seus componentes. Cada um dos esquemas é apresentado no Apêndice B deste trabalho.

Apesar do vocabulário proposto sugerir a adoção de quatro perfis principais, nada impede que, futuramente, outros perfis sejam incorporados ou que este vocabulário seja estendido, tornando-se ainda mais genérico para poder atender uma gama maior de aplicações.

3.2. O Gerente de Contexto – Uma arquitetura simples

O novo vocabulário proposto neste trabalho procura dar apoio à representação de uma informação contextual mais abrangente, isto é, que permita descrever características de mais de uma plataforma computacional e definir alguns atributos relevantes aos processos de adaptação de hiperdocumentos, levando em consideração as necessidades pertinentes de cada ambiente em que tais processos podem ser realizados (ambiente de exibição e/ou ambiente de armazenamento).

Assim sendo, se por um lado o uso do CC/PP oferece uma forma padrão para representação e intercâmbio da informação contextual, por outro lado o uso

desse formato tem a desvantagem de ser bastante extenso e verboso. Em um cenário onde a rede de acesso de dados utilizada pode apresentar uma velocidade de conexão lenta, o CC/PP pode vir a ser uma opção um tanto dispendiosa para negociação de metadados. Então, a grande questão que surge é: de que maneira seria possível adotar o CC/PP como um formato para representação da informação de contexto sem confiná-lo a determinados requisitos de infra-estrutura? A resposta para essa pergunta constitui um dos objetivos desta seção.

Uma série de propostas pode ser encontrada na literatura para ajudar a otimizar questões especificamente relacionadas a desempenho da rede de acesso como, por exemplo, o uso do XML em formato binário comprimido e o uso de URIs. Entretanto, como a questão vai além de aspectos relacionados ao desempenho da rede, um *repositório* é uma proposição mais abrangente e completa para viabilizar o uso do CC/PP e de outros padrões para representação de metadados em cenários mais heterogêneos, que não apenas o ambiente móvel.

Outro objetivo desta seção é apresentar detalhadamente a arquitetura para a gerência de contexto proposta por este trabalho e como o *repositório* está nela inserido. A seção introduz o papel das três principais entidades que participam dessa arquitetura: as aplicações orientadas ao contexto, o Gerente de Contexto e os mecanismos gerenciados.

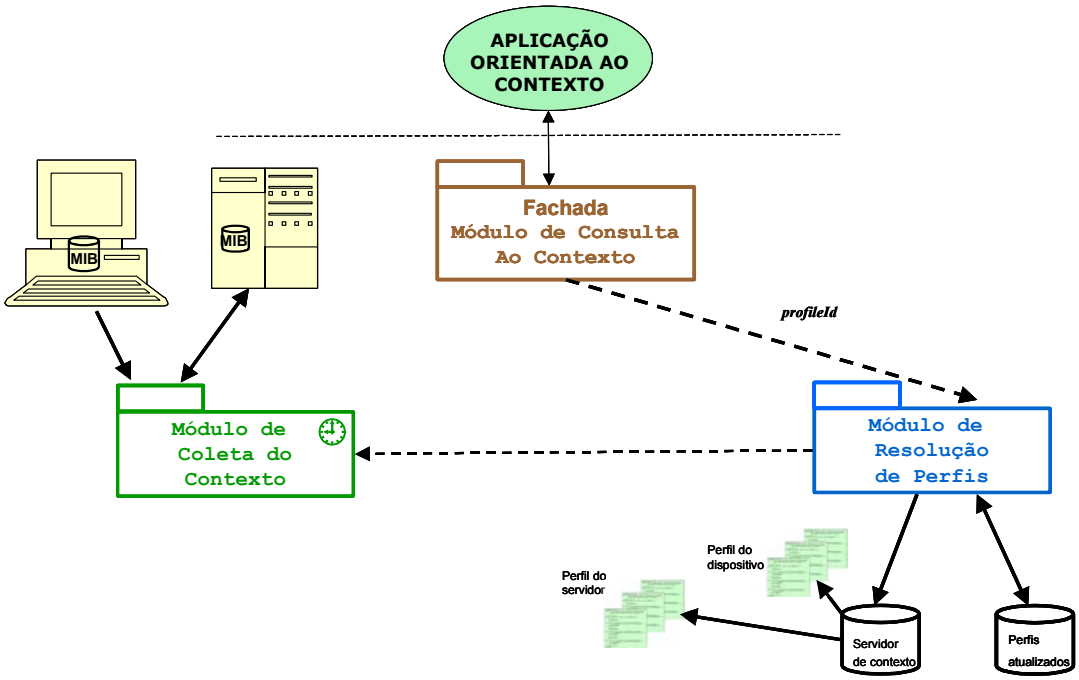

Figura 3.3 – Gerente de Contexto – Uma arquitetura simples

O gerente de contexto foi desenvolvido em linguagem Java, onde um dos padrões de projeto utilizado foi o padrão Fachada (Facade) (Gamma, 1995) com intuito de encapsular e esconder de seus usuários a complexidade do gerente. A arquitetura é composta por três módulos: coleta, resolução dos perfis e consulta ao contexto.

As aplicações orientadas ao contexto, em particular os sistemas hipermídia adaptativos, devem contar com um usuário da estrutura de gerência de contexto para poder enviar consultas ao gerente em um dado instante.

O primeiro módulo é o módulo de consulta ao contexto ou de disseminação do contexto. Ele é a fachada entre o gerente de contexto e as demais aplicações. Sua função é responder às requisições de consulta ao contexto enviadas pela diferentes aplicações. Um ponto de flexibilização da arquitetura que merece destaque é a possibilidade de uso de diferentes protocolos de comunicação para apoiar esse diálogo. Além disso, a estrutura de contexto projetada permite que sejam feitas consultas de acordo com a necessidade de cada aplicação, sem restrições ao número de consultas e com a possibilidade da aplicação especificar o tipo da consulta desejada (completa ou parcial). No caso da consulta parcial, a aplicação deve fazer referência a um perfil específico (cliente, servidor, rede ou usuário) ou a uma coleção de atributos.

O segundo módulo é o de resolução de perfis. Para melhor entendimento desse módulo, é importante salientar a existência de um servidor de contexto (*repositório* possivelmente externo ao gerente de contexto), que contém previamente armazenadas descrições contextuais padrões (*perfil de referência*). Essas descrições correspondem ao perfil do servidor e o perfil do dispositivo. Cada perfil desses é formado por alguns componentes (hardware, software etc.), onde cada componente é composto por um conjunto de atributos (número de processadores, capacidade de processamento local, vazão, resolução da tela, idiomas suportados, suporte a gráfico etc.)

Neste ponto é importante esclarecer que, embora a estrutura de gerência de contexto projetada por este trabalho utilize um *servidor de contexto centralizado*, não há restrições para que, futuramente, esse servidor de contexto seja distribuído. Além de contar com o servidor de contexto centralizado onde estão armazenados os perfis de referência, esse módulo também precisa reter as informações que

retratam o contexto corrente. O contexto corrente é formado pelo conjunto de *perfis atualizados* e cujos detalhes serão apresentados mais adiante.

Um ponto de flexibilização incorporado através do módulo de resolução de perfis à arquitetura proposta é a possibilidade de adoção de quaisquer vocabulários e estruturas de dados para representação da informação de contexto (esta instanciação usou o CC/PP, como já mencionado anteriormente).

O terceiro e último módulo é o módulo de coleta ao contexto, que é encarregado pela obtenção das informações de contexto junto aos agentes.

Os agentes correspondem a um tipo de *hardware* ou programa que residem próximos aos mecanismos gerenciados (dispositivo cliente, servidor de conteúdo, usuário e rede de acesso). Sua função principal é o atendimento das requisições enviadas pelo módulo de coleta do contexto e o retorno automático das informações de contexto, indicando a ocorrência de um evento previamente programado. Também é atribuição do agente efetuar a interface entre os diferentes mecanismos usados na instrumentação das funcionalidades de gerenciamento inseridas em um determinado mecanismo gerenciado. Contudo, os detalhes de funcionamento dessa interface não fazem parte do escopo deste trabalho.

O diálogo entre o módulo de coleta do contexto e os agentes objetiva a troca de dados existentes na base de gerenciamento de cada mecanismo gerenciado, conhecida na literatura como MIB (*Management Information Base*) (Pras et al., 1997). A MIB compreende um conjunto de atributos usados para representar informações estáticas ou dinâmicas.

Um ponto de flexibilização do módulo de coleta é o sentido da comunicação com os agentes: sob demanda – gerente solicita que o agente lhe envie seu contexto – ou através de notificação espontânea – periodicamente, o agente notifica o gerente das informações correspondentes ao seu contexto.

A integração entre os três módulos que compõem a arquitetura do Gerente de Contexto ocorre da seguinte forma: a aplicação (a aplicação orientada ao contexto) interessada em conhecer o contexto no qual está inserida requisita uma consulta ao módulo de consulta. Este por sua vez, repassa ao módulo de resolução de perfis, a identificação das entidades que compõe o contexto corrente. Com base nessa(s) identificação(ões), o módulo de resolução de perfis verifica se já dispõem dessa informação junto à sua base de perfis atualizados. Caso essa informação não

esteja presente, o módulo de resolução de perfis requisita a atualização dos perfis junto ao módulo de coleta do contexto.

Assumindo que o conjunto de atributos que compõe cada um dos perfis é conhecido, ao solicitar a coleta de contexto, os agentes precisam apenas informar as diferenças encontradas, isto é, o conjunto de atributos que sofreu alteração. Em outras palavras, ao invés de cada mecanismo gerenciado enumerar a coleção completa de atributos para descrever suas características e preferências a cada requisição, a abordagem adotada solicita que cada agente indique o local que contém a sua respectiva descrição completa ou perfil de referência (CCPP, 1999a). Essa indicação deve vir incorporada na própria resposta à requisição, através de uma URI adicional, referenciando o servidor de contexto no qual reside o repositório de perfis de referência. Juntamente com a URI, o agente deve apenas informar ao módulo de coleta aquele conjunto de atributos cujos valores sofreram alteração, denominado *perfil de diferença*.

Ao receber essas informações (*perfil de diferença*) o módulo de coleta repassa essas informações ao módulo de resolução de perfis encarregado de mesclar as diferenças com os *perfis de referência* correspondentes, originando o *perfil atualizado*. Em seguida, a estrutura de gerenciamento de contexto concebida possibilita que esses *perfis atualizados* sejam armazenados em memória *cache*, além de outro tipo qualquer de memória, de forma a não sobrepor a informação armazenada no *repositório* que contém os *perfis de referência*, constituindo outro ponto de flexiblização de destaque nesta arquitetura. A razão para que este repositório não seja sobreposto é para permitir a reutilização dos atributos e valores associados contidos no *perfil de referência* por diversos dispositivos ou servidores de característica idêntica.

Uma vez atualizados os perfis, o módulo de consulta retorna o resultado à aplicação responsável pela solicitação inicial.

Esta seção procurou mostrar uma peculiaridade concebida na arquitetura da estrutura para o gerenciamento de contexto que vai além da questão da melhoria do desempenho das redes de acesso. Como foi visto, o conceito de um *repositório de perfis* introduz algumas vantagens adicionais como, por exemplo: reutilização da descrição das informações de contexto (vários dispositivos com mesmas características podem fazer referência a um mesmo perfil) e redução no tamanho da descrição da informação de contexto referente a cada perfil.

A próxima seção esclarece os aspectos de intercâmbio da informação contextual e como ocorrem os diálogos entre sistemas hipermídia e o Gerente de Contexto e entre Gerente de Contexto e agentes sensores.

3.3. Protocolos para negociação de conteúdo

O CC/PP foi concebido para ser usado em um contexto bem amplo de aplicações e sistemas operacionais, e projetado para ser independente do protocolo de comunicação (CCPP, 1999a). Esta seção analisa alguns protocolos para intercâmbio da informação de contexto, além de descrever, de forma detalhada, qual é o formato de cada mensagem trocada entre aplicações-gerente e gerente-agentes.

3.3.1. Protocolos existentes para negociação de conteúdo

Alguns grupos de trabalho do W3C vêm conduzindo estudos sobre protocolos para intercâmbio de informações em um ambiente distribuído. Esta seção descreve algumas características de protocolos de comunicação que podem ser adotados pela estrutura de gerência de contexto proposta nesta dissertação.

Um desses estudos corresponde ao *CC/PP exchange protocol* (CCPP, 1999a) baseado no *HTTP Extension Framework* (Nielsen et al., 2000). O *HTTP Extension Framework* é um mecanismo de extensão genérico para o HTTP/1.1 (Fielding et al., 1999), projetado para interagir com aplicações HTTP existentes.

Segundo (HTTPext, 2000), o HTTP já teve várias iniciativas de extensão tanto no âmbito local como global. Como não existe um *framework* padrão que determine como essas extensões devem ser definidas, uma série de extensões vem sendo usada de forma *ad hoc*, sem promover o reuso e o intercâmbio dessas extensões. O *HTTP Extension Framework* é um mecanismo de extensão do HTTP poderoso que descreve *quais* extensões foram introduzidas, *quem* são os destinatários e *como* esses destinatários devem usar essas extensões.

O HTTP disponibiliza três principais métodos para manipulação e recuperação de dados: GET, PUT, POST. A grande vantagem das extensões

HTTP é prover uma forma de adicionar parâmetros a esses métodos, visíveis ao HTTP.

Para criar extensões utilizando o *HTTP Extension Framework,* em primeiro lugar é necessário criar uma declaração de extensão, onde o objetivo é indicar que uma mensagem foi estendida e, possivelmente, reservar parte do espaço de nome do cabeçalho, identificada por um prefixo do campo do cabeçalho. O *HTTP Extension Framework* introduz dois tipos de extensão: *mandatória* e *opcional*, além de dois tipos de escopo da declaração de extensão: "*hop-by-hop*" e "*end-to-end*" (Nielsen et al., 2000).

A declaração de extensão deve ser *mandatória,* caso o cliente precise receber uma notificação de erro quando houver incompatibilidade entre o servidor (ou um dos agentes intermediários) e o protocolo estendido (no caso, o protocolo *CC/PP Exchange Protocol*). A declaração deve ser de caráter *opcional* quando o cliente aceita receber o conteúdo adaptado ou não.

O escopo deve ser "*hop-by-hop*" quando o cliente sabe, a priori, que existem agentes intermediários compatíveis com o protocolo de intercâmbio da mensagem. Caso contrário, quando o cliente sabe que o primeiro agente intermediário não é compatível com o protocolo, o escopo deve ser "*end-to-end*".

O cabeçalho do *CC/PP Exchange Protocol* possui três campos (CCPP, 1999a):

- *Profile* é um campo de cabeçalho de uma requisição HTTP cujo valor é uma lista de referências que apontam para descrições *CC/PP*. A descrição abaixo representa a gramática do campo de cabeçalho *Profile*.

```
Profile                = profile-field-name ":" 1#reference
profile-field-name     = "Profile"
reference              = <"> (URIabsoluta | profile-diff-name ) <">
profile-diff-name      = profile-diff-number "-" profile-diff-digest
profile-diff-number    = 1#DIGIT
profile-diff-digest    = sp; < MD5 message digest codificada com base64 >
DIGIT                  = <qualquer dígito US-ASCII entre "0".."9">
```

  Cada referência indica uma descrição CC/PP correspondente. Quando a descrição CC/PP não pertence à requisição HTTP, ela é representada por uma URI absoluta. Quando a descrição CC/PP está contida na própria mensagem, ela é representada pelo *profile-diff-name*.

Quando o campo *Profile* tem associado um valor de *profile-diff-name*, o cabeçalho *Profile-Diff* correspondente tem que ser obrigatoriamente incorporado na mesma requisição HTTP. O *profile-diff-name* é composto de duas partes: *profile-diff-number* e o *profile-diff-digest*. Como uma mesma requisição HTTP pode conter mais de um cabeçalho *Profile-Diff*, o *profile-diff-number* indica o cabeçalho *Profile-Diff* correspondente. O *profile-diff-digest* é um valor randômico gerado pela aplicação dos algoritmos MD5 e Base64, e que tem por objetivo otimizar a operação de *lookup* nas tabelas *caches* dos *gateways*, *proxies* e do cliente.

- O campo de cabeçalho denominado *Profile-diff* é um campo de cabeçalho de uma requisição HTTP que contém uma descrição CC/PP. Este campo de cabeçalho é usado em conjunto com o campo de cabeçalho *Profile* contido na mesma requisição.

  A descrição abaixo representa a gramática do campo de cabeçalho *Profile-diff*.

```
Profile-Diff              = profile-diff-field-name ":" profile-desc
profile-diff-field-name   = "Profile-Diff-" profile-diff-number
profile-desc              = < Descrição CC/PP baseada em XML/RDF >
```

  Uma requisição HTTP pode conter vários cabeçalhos *Profile-diff* e todos eles devem estar listados no cabeçalho *Profile*, através do seu respectivo *profile-diff-name*. A correspondência entre *profile-diff-name* do cabeçalho *Profile* e o nome atribuído ao cabeçalho *Profile-diff* é dada através do *profile-diff-number*, que tem que ser o mesmo em ambos os cabeçalhos.

  O formato de uma descrição CC/PP é compatível com o da especificação HTTP/1.1, ou seja, os valores do campo de cabeçalho podem estar distribuídos por várias linhas.

- O campo de cabeçalho *Profile-warning* é um campo de cabeçalho de uma requisição HTTP que contém informações de aviso.

Para ilustrar o uso do *CC/PP Exchange Protocol* baseado no *HTTP Extension Framework*, dois exemplos são descritos abaixo.

O primeiro exemplo ilustra o uso do protocolo *CC/PP Exchange* para solicitação de conteúdo adaptado ao servidor de conteúdo, prevendo a

possibilidade de recebimento de uma mensagem de erro devido à incompatibilidade entre o servidor e o protocolo usado.

1. O cliente faz uma requisição com uma extensão de caráter obrigatório (M-GET) contendo os campos de cabeçalhos apropriados.

```
[1. Cliente --> Servidor de Conteudo]

M-GET /a-resource HTTP/1.1
Host: www.w3.org
Man: "http://www.w3.org/1999/06/24-CCPPexchange" ; ns=99
99-Profile: "http://www.aaa.com/hw","http://www.bbb.com/sw","1-uKhJE/AEeeMzFSejsYshHg=="
99-Profile-Diff-1: <?xml version="1.0"?>
      <RDF xmlns="http://www.w3.org/TR/1999/PR-rdf-syntax-19990105#"
          xmlns:PRF="http://www.w3.org/TR/WD-profile-vocabulary#">
          <Description ID="SoftwarePlatform" PRF:Sound="On" />
          </RDF>
```

2. O servidor verifica o cabeçalho da declaração de extensão e identifica se é compatível com o protocolo. Se não for compatível com o protocolo, ele envia uma resposta com código de erro.

```
[2. ServidordeConteudo --> Cliente (caso de falha)]

HTTP/1.1 510 Not Extended
```

3. Se o servidor verifica o cabeçalho da declaração de extensão e atesta sua compatibilidade com o protocolo, ele extrai o prefixo "M-" do nome do método M-GET e processa o resto da requisição de acordo com a semântica da declaração de extensão e do método GET do HTTP/ 1.1. Depois obtém a lista das URIs absolutas contidas no cabeçalho *Profile* que fazem referência aos repositórios CC/PPs. Para cada URI, ele emite uma requisição para recuperar a descrição CC/PP apropriada.

```
[3. Servidor de Conteudo  --> RepositoriosCCPP]

GET http://www.aaa.com/hw HTTP/1.1
Host: www.aaa.com
....

GET http://www.bbb.com/sw HTTP/1.1
Host: www.bbb.com
....
```

4. O servidor gera o conteúdo adaptado de acordo com as descrições CC/PP referenciadas e o envia encapsulado em uma mensagem de resposta com extensão de cabeçalho de caráter mandatório (Ext). No exemplo em questão, o campo do cabeçalho que descreve se o conteúdo deverá ser armazenado em memória "cache" está com indicação negativa.

```
[4. Servidor de Conteudo --> Cliente]

HTTP/1.1 200 OK
Ext:
Cache-control: no-cache
Content-Type: text/html
Content-Length: 1200
....
```

Vale observar que, se o conteúdo do campo de cabeçalho *Profile-diff* estiver muito extenso, alguns *proxies*, *gateways* ou servidores podem ter restrições para tamanho de cabeçalhos e não conseguir interpretar a requisição HTTP. A alternativa seria descrever essas mesmas informações no corpo da mensagem HTTP. Entretanto, não serão fornecidos mais detalhes, pois essa e outras abordagens estão fora do escopo deste trabalho.

O segundo exemplo procura ilustrar o uso do *CC/PP Exchange Protocol* para solicitação de conteúdo em um cenário onde exista pelo menos um agente intermediário. Além disso, esse cenário ilustra o procedimento que ocorre quando o *proxy* não é capaz de interpretar o perfil CC/PP .

1. O cliente faz uma requisição com uma extensão do tipo opcional (GET) contendo os campos de cabeçalhos Profile e Profile-Diff apropriados.

```
[1. Cliente --> Servidor de Conteudo]

GET /a-resource HTTP/1.1
Host: www.w3.org
C-Opt: "http://www.w3.org/1999/06/24-CCPPexchange" ; ns=80
80-Profile: "http://www.aaa.com/hw", "http://www.bbb.com/sw", "1-uKhJE/AEeeMzFSejsYshHg=="
80-Profile-Diff-1: <?xml version="1.0"?>
      <RDF xmlns="http://www.w3.org/TR/1999/PR-rdf-syntax-19990105#"
          xmlns:PRF="http://www.w3.org/TR/WD-profile-vocabulary#">
          <Description ID="SoftwarePlatform" PRF:Sound="On" />
          </RDF>
Connection: C-Opt, 80-Profile, 80-Profile-Diff-1
```

2. O *proxy* verifica o cabeçalho (C-Opt) da declaração de extensão e identifica que não é capaz de interpretar a mensagem. Assim, ele *elimina os cabeçalhos listados no campo "Connection" e encaminha a requisição ao servidor*.

```
[2. Proxy --> ServidordeConteudo (caso de falha)]

GET /a-resource HTTP/1.1
Host: www.w3.org
```

3. Caso ele possa interpretar a mensagem e sua extensão, ele emite uma solicitação aos repositórios CC/PP de acordo com as URIs absolutas informadas, para que lhes sejam enviadas as descrições CC/PP apropriadas.

```
[3. Proxy --> RepositoriosCCPP]

GET http://www.aaa.com/hw HTTP/1.1
Host: www.aaa.com
....

GET http://www.bbb.com/sw HTTP/1.1
Host: www.bbb.com
....
```

4. O *proxy* gera uma requisição HTTP para o servidor de conteúdo.

```
[4. Proxy --> Servidor de Conteudo]

GET /a-resource HTTP/1.1
Host: www.w3.org
Accept: text/xml, text/html, */*
Accept-Charset: UTF-16, iso-2022-JP
Accept-Encoding: compress, gzip
Accept-Language: fr, en
```

5. O servidor retorna o conteúdo solicitado ao *proxy* de acordo com as características que lhe foram providas através da requisição HTTP.

```
[5. ServidordeConteudo --> Proxy]

HTTP/1.1 200 OK
Cache-control: no-cache
Content-Type: text/html; charset=UTF-16
Content-Encoding: compress
Content-Language: fr
Content-Length: 1200
....
```

6. O *proxy* adapta o conteúdo enviado pelo servidor conforme as descrições CC/PP e envia esse conteúdo adaptado de volta para o cliente.

```
[6. Proxy --> Cliente]

HTTP/1.1 200 OK
Cache-control: no-cache
Content-Type: text/xml; charset=UTF-16
Content-Encoding: compress
Content-Language: fr
Content-Length: 900
....
```

Além da proposta do uso do *CC/PP Exchange Protocol* baseado no *HTTP Extension Framework,* existem outras alternativas de protocolos para o intercâmbio de mensagens que contenham descrições CC/PP, como por exemplo o SOAP (*Simple Object Accesss Protocol*) (SOAP, v. 1.2).

Antes de apresentar algumas das características do SOAP, é importante lembrar um conceito, que está intimamente vinculado ao entendimento desse protocolo, denominado *Web Services*.

Em meados dos anos 90, quando a Internet começou a se popularizar, as tecnologias existentes permitiam que os usuários se conectassem a um *site* e obtivessem o conteúdo disponível. Nos últimos anos, esse cenário vem se modificando e novas tecnologias e *frameworks* de desenvolvimento estão proliferando, permitindo uma maior integração entre os aplicativos e serviços disponíveis na internet. Esse novo cenário deve tratar tarefas complexas, como o gerenciamento de transações, através da disponibilização de serviços distribuídos que utilizem *interfaces* de acesso simples e bem definidas. Esses serviços e aplicações distribuídas são conhecidos como *Web Services*.

Web Services são identificados por uma URI (*Unique Resource Identifier*), e são descritos e definidos usando XML. Um dos motivos que tornam *Web Services* atrativos é o fato deste modelo ser baseado em tecnologias padrão, em particular XML e HTTP. WSDL (*Web Service Description Language*) é uma linguagem baseada em XML, utilizada para descrever um *Web Service*. Um *Web Service* deve, portanto, definir todas as suas interfaces, operações, esquemas de codificação, entre outros nesse documento. O SOAP está se tornando padrão para a troca de mensagens entre aplicações e *Web Services*, já que é uma tecnologia construída com base em XML e HTTP.

Definido pelo consórcio W3C, SOAP é um procolo projetado para invocar aplicações remotas através de RPC[4] (*Remote Procedure Calls* - Chamadas Remotas de Procedimento) ou trocas de mensagens, em um ambiente independente de plataforma e linguagem de programação. Com SOAP é possível criar aplicações para troca de dados entre empresas, o chamado B2B (*Bussiness to Bussiness*).

O processo de uma chamada RPC ocorre da seguinte forma: antes de serem enviadas pela rede, as chamadas RPC (emitidas pela aplicação cliente) são encapsuladas (ou serializadas) segundo o padrão SOAP. O serviço remoto, ao receber a mensagem faz o processo contrário, desencapsulando-a e extraindo as chamadas de método. A aplicação servidora então processa essa chamada, e envia uma resposta ao cliente. O processo então se repete: a resposta é também

[4] RPC são chamadas locais a métodos de objetos (ou serviços) remotos. É possível acessar os serviços de um objeto localizado em um outro ponto da rede, através de uma chamada local a esse objeto. Cada chamada ou requisição exige uma resposta.

serializada e enviada pela rede. Na máquina cliente, essa resposta é desencapsulada e é repassada para a aplicação cliente.

Uma mensagem SOAP (veja Figura 3.4) consiste basicamente dos seguintes elementos:

- ***Envelope***: Toda mensagem SOAP deve contê-lo. É o elemento raiz do documento XML. O Envelope pode conter declarações de *namespaces* e também atributos adicionais como o que define o estilo de codificação (*encoding style*). Um *encoding style* define como os dados são representados no documento XML.

- ***Cabeçalho***: É um cabeçalho opcional. Ele carrega informações adicionais, como por exemplo, se a mensagem deve ser processada por um determinado nó intermediário (É importante lembrar que, ao trafegar pela rede, a mensagem normalmente passa por diversos pontos intermediários, até alcançar o destino final). Quando utilizado, o Cabeçalho deve ser o primeiro elemento do Envelope.

- ***Corpo***: Este elemento é obrigatório e contém o *payload*, ou a informação a ser transportada para o seu destino final. O elemento Corpo (*Body)* pode conter um elemento opcional *Fault*, usado para carregar mensagens de status e erros retornadas pelos "nós" ao processarem a mensagem.

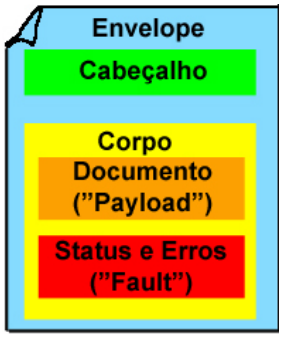

Figura 3.4 – Estrutura de uma mensagem SOAP

Os pedidos SOAP podem ser feitos através de três métodos: GET, POST e SOAP. Os métodos GET e POST são idênticos aos pedidos feitos por

navegadores Internet. O SOAP é um padrão semelhante ao POST, mas os pedidos são feitos em XML e permitem recursos mais sofisticados como passar estruturas e *arrays*. Independente de como seja feito o pedido, as respostas são sempre em XML. O XML descreve perfeitamente os dados em tempo de execução e evita problemas causados por inadvertidas mudanças nas funções, já que os objetos chamados têm a possibilidade de sempre validarem os argumentos das funções, tornando o protocolo muito robusto.

Outra alternativa é encontrada no protocolo ICAP (*Internet Content Adaptation Protocol*) (Elson & Cerpa, 2001), desenvolvido pelo Fórum ICAP (ICAP). Os objetivos do Fórum ICAP são: flexibilizar o conteúdo para os usuários finais; fornecer um padrão comum e aberto, para a comunicação de dispositivos baseados nas bordas da rede, para o manuseio desses serviços de valor agregado; transferir a carga das APIs (*Application Program Interface*), consumidoras de recurso, dos servidores web para servidores dedicados.

O ICAP é um protocolo projetado para execução de adaptação de alguns conteúdos, armazenados em servidores *Web* na Internet e em servidores dedicados (servidores ICAP). Cada servidor pode oferecer um serviço específico (ex: inserção de publicidade, tradução de conteúdo, filtragem de conteúdo etc.), procurando liberar os servidores *Web* e padronizando a forma através da qual esses serviços são implementados. A arquitetura envolvida compreende um Cliente ICAP (*switch box* ICAP), responsável por interceptar todas as transações entre um usuário final (dispositivo cliente) e um servidor de origem (servidor *Web*) e redirecionar essas transações para um servidor ICAP, que executará determinados tipos de adaptação.

Além do Cliente e Servidor ICAP, a arquitetura possui dois componentes adicionais: semântica das transações e política de controle. Entretanto, a especificação atual só define o segundo componente, isto é, como enviar uma mensagem HTTP do Cliente ICAP para o Servidor ICAP; como especificar a URI do recurso ICAP requisitado em conjunto com outros prâmetros relativos a esse recurso; como receber mensagens adaptadas.

O ICAP pode ser utilizado de duas maneiras denominadas modificação de requisição (*reqmode*) e modificação de resposta (*respmode*) (Souza et al., 2002).

No *reqmode*, o Cliente ICAP intercepta uma requisição enviada a um servidor de origem e redireciona para um Servidor ICAP. Esse servidor executa uma modificação e retorna a requisição ao Cliente ICAP, que tanto pode ser uma versão modificada da requisição contendo a URI original, como pode ser uma requisição modificada apontando para um página que contém uma mensagem de erro ao invés da URI original. Quando a requisição é retornada, o Cliente ICAP envia a resposta ao cliente. Essa resposta pode ter sido proveniente tanto do Servidor ICAP, como do servidor de origem.

O *respmode* visa o pós-processamento de respostas HTTP (ex: formatação HTML para apresentação em dispositivos especiais, tradução etc.) antes de entregá-la ao cliente. Nesse modo, a resposta do servidor de origem é interceptada e redirecionada a um Servidor ICAP que retorna uma versão modificada da resposta, ou retorna um erro.

Embora seja uma parte essencial do processo de adaptação do conteúdo, a semântica das transações especificada pelo Fórum ICAP precisa ser complementada com informações referentes ao momento em que uma solicitação de adaptação pode ser feita, o tipo de adaptação solicitada e para quem essa solicitação deve ser encaminhada. Em resumo, para cada requisição ou reposta interceptada pelo Cliente ICAP, é preciso definir regras de adaptação que determinem que adaptação, ou função de processamento, está sendo requerida ao Servidor ICAP.

3.3.2. Protocolo escolhido para Negociação de Conteúdo

Apesar de algumas particularidades, uma série de protocolos de comunicação conhecidos atualmente poderia ser adotada pelo Gerente de Contexto. Também existe a hipótese de se usar mais de um tipo de protocolo, de acordo com a existência de determinadas restrições do lado de alguns mecanismos gerenciados, como por exemplo, aquele que monitora as características da rede de acesso entre um determinado cliente e um servidor.

Apesar das várias alternativas de uso de protocolos apresentados na seção anterior, este trabalho optou por utilizar um protocolo bastante popular e bastante difundido atualmente, o HTTP/1.1 (Fielding et al., 1999). A justificativa para essa

escolha deve-se não só pela popularidade e simplicidade do protocolo, como pelas características da arquitetura de gerenciamento de contexto proposta por este trabalho. Essencialmente, essas características referem-se ao tipo de mensagens trocadas entre aplicação-gerente e gerente-agente, que diferentemente dos exemplos apresentados na seção anterior, só contém informação de contexto. Isto é, elas não são mensagens que contém solicitação de conteúdo e descrição de contexto representada em CC/PP. É importante salientar que, apesar da implementação efetuada neste trabalho usar o HTTP, a arquitetura proposta é flexível e permite a implementação de outros protocolos, como descrito em detalhes no próximo capítulo.

A seguir serão descritos os parâmetros que devem constar em cada uma das mensagens trocadas na arquitetura deste trabalho. É importante salientar que esses parâmetros devem constar do corpo da mensagem HTTP, respeitando a seqüência que é apresentada abaixo.

MENSAGEM	TRANSMISSOR	RECEPTOR	PARÂMETROS
HTTP_Request (GET)	Aplicação	Gerente	identificador da sessão, identificador da aplicação, (tipo de contexto desejado <todos os perfis, perfil específico ou atributo específico>)
HTTP_Response	Gerente	Aplicação	Descrição CC/PP com lista de atributos correspondente ao contexto desejado
HTTP_Request (GET)	Gerente	Agente	identificador da sessão, lista {FA, resultado}
HTTP_Response	Agente	Gerente	URI do perfil de referência, descrição CC/PP contendo somente os atributos alterados
HTTP_POST	Agente	Gerente	URI do perfil de referência, descrição CC/PP contendo somente os atributos alterados

Figura 3.5 – Parâmetros das mensagens trocadas pelo Gerente de Contexto

Este capítulo apresentou os principais aspectos considerados no projeto da arquitetura do gerente de contexto proposto por este trabalho, como uma estrutura independente e reutilizável por diferentes sistemas, em particular os sistemas hipermídia. A seguir, no Capítulo 4, serão apresentados detalhes específicos desta

arquitetura, desenvolvida através de dois frameworks: modelagem da informação de contexto e gerência do contexto.