

Uma Proposta de Arcabouço para Agentes Assistentes Pessoais

Saulo Popov Zambiasi e Ricardo J. Rabelo

Programa de Pós-Graduação em Engenharia de Automação e Sistemas (PPGEAS) – Universidade Federal de Santa Catarina (UFSC) – Florianópolis – SC – Brasil

E-mail: {popov, rabelo}@das.ufsc.br

Resumo-- A crescente evolução das novas tecnologias de informação e comunicação tem tornado o mercado cada vez mais competitivo. Se por um lado isso tem facilitado nos processos organizacionais, por outro a prática tem mostrado que essas mudanças têm levado a um aumento de responsabilidades e tarefas por parte das pessoas inseridas neste cenário. Este problema poderia ser amenizado com a disponibilidade de mais pessoas para tratar da crescente quantidade e complexidade dos processos, mas isso nem sempre é viável. Contudo, alguma ajuda no tratamento dos processos dessas pessoas já seria de grande valia. Atualmente já existem *softwares* assistentes pessoais, mas estes abordam apenas parte da problemática geral, tanto no artifício da conceituação, como na implementação. Ainda, não existe nenhuma padronização entre estes para interoperabilidade. Dessa forma, este artigo apresenta uma proposta de um Arcabouço para Agentes Assistentes Pessoais que deve servir como base para implementar tais agentes.

Palavras-chave: Arcabouço, Agentes, Assistentes Pessoais.

I. INTRODUÇÃO

Com a crescente evolução das tecnologias da informação e comunicação, o mercado tem se tornado cada vez mais competitivo. Isso tem obrigado as organizações a se aperfeiçoarem no âmbito de seus processos. Assim, a necessidade de utilização de novos métodos de trabalho tem se mostrado de vital importância para a sobrevivência dessas organizações [17]. Ainda, uma série de novas medidas vem sendo tomadas como meio de tornar os processos organizacionais mais ágeis, em conformidade com a conjectura de cada atividade específica e, na sua maioria, com enfoques em várias frentes distintas.

No entanto, a prática tem mostrado que essas mudanças, se por um lado têm facilitado o trabalho das pessoas, por outro as têm forçado a serem mais produtivas, levando a um aumento ainda maior de responsabilidades e tarefas, sendo algumas dessas consideradas repetitivas e desgastantes. Além disso, essas pessoas têm estado cada vez mais ocupadas e imersas em diversos problemas, muitas vezes não conseguindo cumprir com suas tarefas em tempo hábil e/ou com a qualidade esperada.

De uma maneira geral, a referida problemática poderia ser amenizada com a disponibilização de mais pessoas para as crescentes tarefas. Porém, na prática, o custo organizacional para isso torna esse caminho inviável. Seguindo outra linha de raciocínio, uma assistência fornecida através da utilização de

um *software* de computador, poderia munir as pessoas com um conjunto de ferramentas visando automatização de algumas de suas tarefas e dar certa assistência na gestão de atividades relacionadas ao seu trabalho, muitas vezes tal qual um secretário humano [1].

Estes *softwares* secretários, na forma de um assistente pessoal, se diferem de um programa personalizado de procura ou de filtragem e seleção de e-mails por ser baseado em rede, interativo, adaptativo, de propósito geral, de execução autônoma e que interagem com outros agentes. Estes percebem seu ambiente e reagem a ele tal como um agente. Tal autonomia permite ao agente agir sem interferência de usuários, pois este está apto a completar suas tarefas fazendo suas escolhas apropriadas a partir de um conjunto de estratégias possíveis [8].

Desse modo, este trabalho trata sobre a automatização de certas tarefas, sem ou com um baixo nível de intervenção humana, por meio de um subsídio para auxiliar as pessoas em suas tarefas organizacionais e/ou pessoais diárias. Essa assistência deve ser fornecida por meio de *softwares* assistentes pessoais em ambiente computacional em rede, baseado em padrões abertos, para representação e/ou substituição de pessoas na execução de certas tarefas, sem ou com algum grau de intervenção/supervisão humana.

Inspirado neste contexto, o presente artigo apresenta uma proposta de um Arcabouço para Agentes Assistentes Pessoais (AAPs) para auxiliar as pessoas em determinadas tarefas. Em um primeiro momento e considerando um cenário empresarial, seis classes de tarefas identificadas para serem gerenciadas por um AAP foram:

1. Substituição do usuário em determinadas tarefas (ex. leitura e resposta de alguns e-mails ou mensagens importantes; preenchimento de alguns formulários);
2. Gerenciamento da agenda e atividades do usuário;
3. Procura de informações em repositórios;
4. Auxílio ao usuário a compreender que tarefas precisam ser executadas e como;
5. Execução de algumas tarefas do usuário (ex. parte de uma negociação eletrônica, monitoramento dos processos de negócio);
6. Filtro e gerenciamento das informações providas de redes ubíquas (ex. filtro de conteúdo, adaptação da linguagem e formatação das informações no dispositivo móvel do usuário);

Este se trata de um trabalho exploratório aplicado, qualitativo, que visa identificar e conceber um arcabouço

aberto de AAP, que forneça níveis de ajuda ao usuário dentro de um conjunto de atividades ligadas ao dia-a-dia empresarial, com variados graus de autonomia e adaptabilidade. Desta forma, as pessoas podem direcionar seu tempo para atividades mais importantes como novos negócios, melhoramento de suas tarefas, inovações e auto-aprendizado.

Este artigo é organizado na seguinte estrutura: Na seção 2 é apresentado o estado da arte sobre a assistência pessoal via sistema computacional e certos requisitos funcionais. A seção 3 apresenta o modelo do Arcabouço para Agentes Assistentes Pessoais, além das tecnologias avaliadas e utilizadas para a implementação com base no modelo conceitual. A seção 4 descreve o protótipo parcial até agora implementado. Na seção 5 são apresentadas as considerações finais e os próximos passos do projeto.

II. ASSISTÊNCIA PESSOAL VIA SISTEMA COMPUTACIONAL

A utilização de uma ferramenta computacional para dar assistência aos usuários segue em conformidade com a evolução das tecnologias computacionais com equipamentos cada vez mais inteligentes, permitindo que a interação deixe de ser apenas entre parceiros humanos [16]. O conceito de um *software* para fornecer essa assistência ou substituição dos usuários na execução de determinadas atividades não é novo e já vêm sendo trabalhado em diversas frentes. Entretanto, a complexidade para representar uma pessoa em determinadas situações já é um grande problema do ponto de vista computacional.

O conceito desses assistentes surgiu como forma de auxiliar os usuários de sistemas e programas de computador na sua utilização [1]. Outros tipos de assistentes de *software* são programados para encontrar algum tipo de informação na Internet ou bases de dados, apenas provendo alguma ajuda para encontrar informações disponíveis e executar tarefas de forma mais efetiva [8]. De qualquer forma, ainda são as pessoas que devem executar as tarefas.

Em [8], é feito um paralelo de assistentes pessoais com agentes. Neste, os assistentes pessoais são vistos como agentes, os quais percebem o ambiente através da agenda do usuário, preferências, preferências de outros usuários, reputações, respondendo com ações e assistência ao usuário. Além disso, devido à comunicação deste com outros agentes, este passa a participar inevitavelmente de um sistema multiagente. Este paralelo é apresentado como uma justificativa natural da utilização da tecnologia de agentes como maneira de fornecer assistência às pessoas em suas tarefas.

Neste contexto, um agente de *software* pode substituir parcialmente ou totalmente uma pessoa para determinadas tarefas e em várias situações diferentes, assumindo o papel de um assistente. Para que estes possam executar suas tarefas, muitas vezes precisam se comunicar com outros sistemas, que também podem ser vistos como agentes. Estes agentes podem trocar informações automaticamente, sem a intervenção direta ou indireta de um humano, como no caso de negociações eletrônicas, mas enfocando apenas o processo de “negociação” em cenários de negócios bem definidos [6].

Além disso, para que possa haver uma interação mais eficiente entre os assistentes e as pessoas a qualquer momento e em qualquer lugar, deve ser possível que a comunicação entre os *softwares* e as pessoas seja quase que independente do local onde eles se encontram. Isso pode ser possível com a utilização de recursos computacionais móveis (telefone celular, PDA, etc.) que disponibilizam aos usuários um conjunto de serviços muito diversificados, além de conexão à Internet [16].

Ainda na trajetória de focar os assistentes de *software*, em [8] é apresentado um cenário de assistentes pessoais que representam indivíduos na web. Estes assistentes ajudam os usuários em atividades diárias como e-mails, chamadas de celulares, e podem coordenar e negociar com certo grau de autonomia, baseando-se em informações predefinidas pelo usuário. Para este tipo de assistente é previsto um cenário de computação ubíqua em que este verifica o contexto local para cada situação, além de ser possível a interação com outros usuários ou agentes por meio do dispositivo móvel do usuário. O enfoque tratado por [8] apresenta apenas um cenário conceitual, sem definição de qualquer modelo, arquitetura ou implementação, mas que já o torna bastante interessante do ponto de vista conceitual.

Seguindo a abordagem da utilização de dispositivos móveis para a integração e interação entre o usuário e seu assistente de *software*, é necessário evidenciar que cada dispositivo móvel possui suas próprias capacidades, às vezes muito limitadas. Dentre a combinação dos recursos desses dispositivos móveis, cada qual possui suas próprias especificidades em concordância com cada fabricante de *hardware* e *software* – Sistema Operacional – assim como capacidades limitadas de informações e comunicação. Neste cenário, [12] apresenta um arcabouço baseado em sistemas multiagentes que provê adaptações das informações dos usuários de acordo com as preferências e o histórico do sistema e em concordância com as capacidades limitadas de seus dispositivos móveis. O autor descreve um gerenciador de conhecimento, trocas de informações e adaptações para suportar certas capacidades de adaptação.

No panorama apresentado nos parágrafos anteriores, diversos requisitos funcionais podem ser identificados *a priori* com relação a um modelo de representação de uma pessoa por meio de *software*.

A. Requisitos Funcionais

Para que um sistema computacional possa “saber” o que fazer, quando determinada tarefa deve ser executada e como ela deve ser executada, são necessárias informações que contextualizem as preferências, necessidades e desejos do usuário deste sistema [13], [8], [4]. Porém um dos problemas envolvidos é definir eficientemente quais informações referentes aos usuários precisam ser armazenadas e como elas podem ser adequadamente alocadas em locais de eficiente acesso pelos sistemas computacionais. Além disso, quando e como utilizá-las é um fator de grande relevância para a execução da assistência aos usuários. Neste trabalho, devem ser apenas analisadas e utilizadas informações necessárias para a execução de cada tarefa do usuário, sem inicialmente se preocupar com o acoplamento entre elas.

Um *software* que fornece assistência às pessoas precisa se comunicar eficientemente com seu usuário de forma a fornecer os recursos necessários para sua assistência e disponibilidade de interatividade, independente de onde o usuário está localizado [1], [13], [14]. Contudo, quando o *software* deve se comunicar e interagir com o usuário, e em que situações específicas isso deve ser feito, não é necessariamente uma tarefa trivial. É necessário haver definições de quando, como e em que situações o usuário deve ser interrompido para que os processos que estão em operação possam ser tratados pelo assistente, quando existir a necessidade de interação com o usuário. Por meio disso, a comunicação se torna um fator importante para a realização do presente trabalho, não obstante de que ela é obrigatoriamente necessária para a eficiente interação entre usuário e assistente de *software*.

A interação entre o usuário e um *software* assistente, seguindo o trabalho de [14], deve acontecer quase que independente de onde o usuário esteja localizado com a utilização, em muitas das situações, de dispositivos móveis para este fim, ou seja, dispositivos de *hardware* com recursos muitas vezes limitados [14], [16]. Entretanto, é necessário que o assistente de *software* possa se adaptar a esses dispositivos de *hardware* e seu sistema operacional para que a interação com o usuário seja possível. Este trabalho não intenta abordar tal adaptabilidade para todos os dispositivos móveis, mas apenas apresentar uma forma de interação com o usuário para que o modelo possa ser validado.

Além de trabalhar com a execução de certas tarefas para auxiliar os usuários utilizando-se de uma configuração de informações adequadas, um assistente de *software* precisa ter a possibilidade de ampliar suas atividades e aprender com as situações já executadas, adaptando-se a novas situações por meio deste aprendizado [2], [15], [4]. Todavia, o aprendizado é um assunto muito abrangente e não deve ser abordado neste trabalho. No presente projeto, sugere-se uma forma de ampliação das atividades dos assistentes e adaptação a novas situações via acoplamento de novos comportamentos aos assistentes de *software* em uma estrutura modular.

Além disso, ainda no que tange aos processos, ou comportamentos, dos assistentes de *software*, podem existir conflitos na execução de determinadas tarefas. Um assistente de *software* deve ter a capacidade de gerenciar estes conflitos, identificando a ordem para a execução de cada tarefa por meio de configuração de prioridade e outras informações, cumprindo seus objetivos em conformidade com as necessidades dos usuários [1]. Entretanto, muitos dos conflitos podem ser resolvidos com a paralelização dessas tarefas, quando essas não estejam atreladas sequencialmente umas as outras. Assim, este trabalho procura contornar este problema com tecnologias de computação distribuída.

Porém, a distribuição dos comportamentos do assistente pessoal gera outros problemas, tais como a necessidade da utilização de padrões para comunicação entre os comportamentos e o assistente. Além disso, o agente deve estar preparado para agir em caso de indisponibilidade de comportamentos, quando armazenados em repositórios que em determinado momento não estão mais acessíveis. Dessa

forma surge a necessidade de um serviço que avalia e trata a disponibilidade de aplicações e serviços remotos [13].

Ainda, a escolha dos comportamentos a serem selecionados e executados em conformidade com cada situação específica e contexto local do usuário, é um requisito necessário para que o assistente pessoal possa agir corretamente para alcançar os objetivos dos usuários. Assim sendo, esta tarefa deve ser definida no presente trabalho, sendo prevista na arquitetura conceitual de forma eficiente e baseado no arquétipo de comportamentos distribuídos já citados.

Outros problemas que a assistência de pessoas via *software* acarretaram são falhas para encontrar novas oportunidades aos usuários, considerações inadequadas sobre as ações a serem tomadas e custos/benefícios com ações mal organizadas temporalmente [4]. Mesmo que seja de grande importância para o usuário a detecção de novas oportunidades de negócios e atividades, não é tarefa fácil determinar como isso pode ser feito automaticamente ou que nível de interação deve haver para cada tipo de oportunidade.

Contudo, os trabalhos estudados abordam apenas algumas características ou parte da problemática geral, tanto no artifício da conceituação e modelagem como na implementação. Além disso, não foi encontrado qualquer modelo ou arcabouço especificamente para *softwares* assistentes durante as pesquisas até agora efetuadas. Porém, todos apresentam um conjunto de idéias bastante atraentes e que podem servir de base para a modelagem e desenvolvimento de *softwares* que fornecem assistência pessoal.

III. PROPOSTA

No contexto desse trabalho, e conforme os trabalhos correlatos estudados, surge a necessidade de uma estrutura formal e padronizada para servir como base para suportar esses *softwares* assistentes pessoais. Sendo assim, este trabalho fomenta uma estrutura de base que possa ser utilizada, não para implementar, mas para sustentar todas as características identificadas.

Compilando as idéias já apresentadas, este trabalho apresenta a definição de um *software* designado para assistência pessoal – ora chamado de Agente Assistente Pessoal (AAP) – como um processo computacional virtual criado para representar um ou mais usuários na execução de certas tarefas, automaticamente ou com algum grau de intervenção/supervisão humana, e o arcabouço para estes agentes como o modelo de referência para a criação e suporte do ciclo de vida de tais elementos.

Dessa forma, as pessoas – auxiliadas pelos seus AAPs – podem se concentrar em tarefas mais essenciais, mais nobres e que realmente necessitam de sua participação direta. Em tempo, é importante salientar que não se deseja substituir as pessoas em toda a execução dos processos, mas ajudar a minimizar a carga de trabalho dessas. Assim, restando de forma efetiva apenas às tarefas essenciais e que agregam maior valor para elas. Fica então a cargo dos AAPs, executar para as pessoas suas tarefas repetitivas, enfadonhas e que consomem muito tempo [15].

Não é o propósito investigar de forma aprofundada todas as características apresentadas, suas ramificações, ou mesmo

seus subproblemas envolvidos para alcançar as subclasses de tarefas identificadas na primeira seção. Em verdade, busca-se uma estrutura geral para suportar os requisitos funcionais identificados de forma padronizada. Essa estrutura deve tornar possível a interoperabilidade entre os agentes, aplicações, pessoas e serviços que tratam da problemática abordada neste trabalho.

De fato, a definição de um arcabouço já caracteriza a inovação pretendida em si. Porém, para agregar maior valor ao trabalho e permitir que este possa ser visto como um padrão aberto, alguns requisitos são definidos conforme citados a seguir:

- O Arcabouço deve ser baseado em padrões abertos e atuais, independente de tecnologia e que possa ser facilmente adaptável.
- O AAP deve poder ser implementado em linguagens e tecnologias tanto comerciais quanto gratuitas ou *opensource*.
- O AAP deve permitir que novos comportamentos sejam agregados dinamicamente sem que seja necessário alterar sua implementação pronta.
- Deve ser possível substituir um comportamento por outro com atividade semelhante e mesmos parâmetros, fornecido por outra entidade, sem que sejam necessárias grandes alterações nas configurações do agente.
- Deve haver a possibilidade dos comportamentos do AAPs não estarem necessariamente localizados no mesmo computador do núcleo de execução do agente em si, ou seja, devem poder estar distribuídos em repositórios de comportamentos na Internet, podendo ser utilizados gratuitamente, alugados ou vendidos por terceiros.
- Um comportamento deve poder ser invocado pelo agente apenas quando este estiver habilitado para tal e todas as suas condições para ativação forem satisfeitas.

Para tanto, algumas características são necessárias para que o AAP possa ser executado com certa autonomia e independência. Essas formam o modelo do Arcabouço para AAPs, composto por sistemas e módulos, conforme Fig. 1.

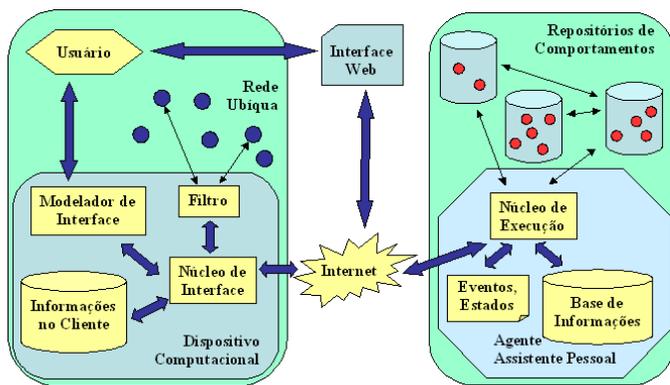


Fig. 1. Modelo do Arcabouço para AAPs.

Com base no modelo apresentado na acima, é necessário definir cada um dos módulos envolvidos. Dentre estes, pode-se separar os módulos que estão presentes no dispositivo

computacional do usuário, fazendo a comunicação e interface entre o AAP e o usuário e os módulos que estão ligados diretamente com a sua execução.

Os módulos necessários para a execução do AAP são:

- i. O “Núcleo de Execução” é a parte do agente responsável pela execução central, além de acessar as informações sobre objetivos, necessidades e preferências dos usuários. Além disso, é tarefa do núcleo de execução gerenciar os eventos e estados e acessar/invocar os comportamentos do agente.
- ii. A “Base de Informações” é uma base de dados que armazena as preferências do usuário e informações pertinentes à execução de cada um dos comportamentos que são utilizados pelo AAP.
- iii. O gerenciador de “Eventos e Estados” é o módulo que provê ao AAP informações extras em conformidade com os diversos contextos possíveis para a execução dos comportamentos. Alguns destes eventos podem, inclusive, ser a resposta da execução de outros comportamentos.
- iv. Ainda que não necessariamente localizados no mesmo dispositivo computacional do AAP, os “Repositórios de Comportamentos” contêm as estruturas em forma de algoritmos que devem ser executadas para a execução do AAP.
- v. Por fim a “Interface Web” é um sistema que se integra ao arcabouço para prover ao usuário um modo mais avançado para criação e configuração dos AAPs, não possível em dispositivos móveis com capacidades limitadas.

Do lado do usuário encontram-se:

- i. O “Núcleo de Interface”, módulo central de processamento do *software* responsável por fazer a comunicação do AAP com o usuário.
- ii. As “Informações no Cliente”, tais como preferências e outras informações relevantes. Estas informações são sincronizadas com a base de dados do AAP quando conectados.
- iii. O “Filtro”, responsável por selecionar as informações providas dos dispositivos de “redes ubíquas”, pegando apenas o que é de interesse para o usuário, com base nas suas preferências.
- iv. O “Modelador de Interface”, que ajusta o sistema para interagir com o usuário conforme as limitações do dispositivo computacional que o usuário está utilizando.

A execução do agente é composta por comportamentos distribuídos na Internet, vistos pelo AAP como serviços a serem utilizados conforme a necessidade. Ainda, cada comportamento pode ser composto da orquestração de vários serviços distribuídos na Internet (Fig. 2).

Seja A um conjunto não vazio, sendo que A representa o conjunto de n comportamentos distribuídos na Internet $\{a_1, a_2, \dots, a_n\}$. Define-se como B o comportamento resultante do AAP, sendo caracterizado pelo subconjunto de comportamentos de A , no qual $B \subseteq A$. Seja b_i um comportamento de B e $\{b_1, b_2, \dots, b_n\}$ o conjunto de

comportamentos de B , logo, o comportamento de B é caracterizado por $B = \sum_{i=1}^n b_i$.

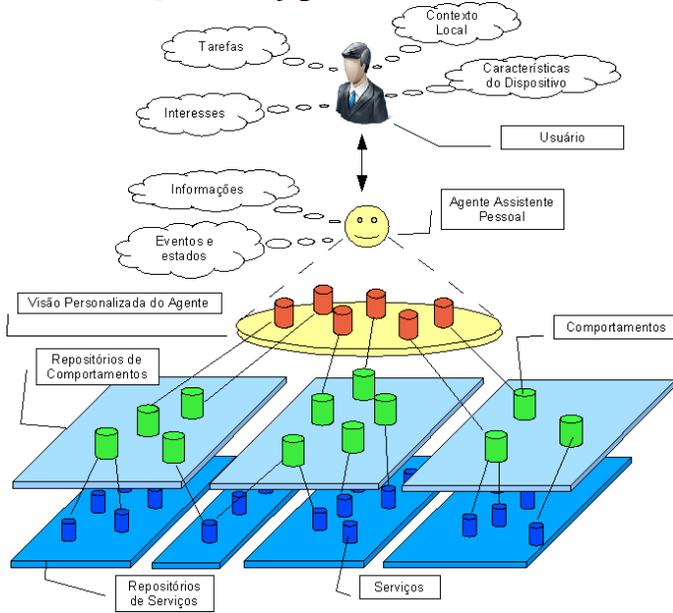


Fig. 2. Visão geral dos comportamentos do AAP.

Em suma, o comportamento do Agente Assistente Pessoal é o subconjunto de comportamentos selecionados e configurados para este, da totalidade de comportamentos disponíveis e distribuídos na Internet, em conformidade com os padrões abertos de conectividade e comunicação aceitos pelo mesmo.

A. Tecnologias da Informação Envolvidas na Implementação

Para o presente projeto foi realizado um estudo prévio de tecnologias necessárias para a criação e execução do AAP.

O acesso a serviços foi um dos pontos iniciais a ser analisado. Assim, a tecnologia dos serviços web, amplamente utilizada nos dias de hoje e já com um bom nível de maturidade e padronização, foi selecionado como padrão para os serviços acessados pelos comportamentos.

De fato, soluções distribuídas dessa tecnologia podem prover a padronização necessária para manter a compatibilidade entre serviços em uma arquitetura orientada a serviços. Os serviços web são compostos de pequenos módulos de softwares, acessados por outras aplicações por meio de um conjunto de protocolos web, por uma URL e baseado em um conjunto de padrões já consolidados. Através de um estudo prévio, podem-se citar alguns desses padrões tais como a linguagem XML (*Extended Markup Language*), uma interface para descrever os serviços - WSDL (*Web Services Description Language*), o protocolo SOAP (*Simple Object Access Protocol*) e a UDDI (*Universal Description, Discovery and Integration*). Através do uso desses padrões, é possível desenvolver aplicações modulares, abertas, baseadas na Internet e com interfaces padronizadas [7].

Como forma de padronização da implementação dos comportamentos a serem invocados pelo AAP, e a comunicação destes com outros serviços, foi escolhida a Linguagem de Execução de Processos de Negócios (*Business Process Execution Language – BPEL*). Esta pode prover uma

especificação formal para a execução e interação do AAP, fornecendo uma fácil interação entre organizações, assim como cenários *business-to-business* (B2B) [9]. Mais especificamente no nível de implementação, o WS-BPEL é uma especificação atual para a interação entre serviços web e estruturada com documentos XML [10].

A especificação BPEL define variáveis, interações como laços, condições e paralelismos. Isto permite aos desenvolvedores criarem processos de negócios que interagem com serviços web de organizações, assim como serviços web externos, de outras organizações. As primitivas do BPEL, e suas exceções são interpretadas e executadas por um mecanismo chamado orquestração [11]. Já para a interação entre o usuário e os processos, ou comportamentos do agente, sugere-se a especificação do BPEL4PEOPLE, que permite tal funcionalidade. Por meio da orquestração, o comportamento de um AAP pode ser baseado na composição de diversos serviços web disponíveis na própria organização, em outras organizações ou na Internet como um todo. Estes serviços web podem compor um comportamento de forma algorítmica, com estruturas de laços, serviços que executam em paralelo, e condicionais.

A tecnologia de Agentes surge como uma das tecnologias mais apropriadas para dar suporte a “objetos inteligentes”. Primeiramente, a tecnologia de agentes foi desenvolvida em ambientes próprios, linguagens próprias e protocolos de comunicação específicos. Porém, a tecnologia dos agentes tem migrado para o uso de tecnologias padronizadas como o XML e serviços web. Dessa forma, o comportamento dos agentes pode ser modelado como um serviço, interagindo com serviços web e redes *peer-to-peer*. Ainda, já existem muitas plataformas de sistemas multiagentes das quais muitas delas já possuem um bom nível de maturidade e robustez [6].

Para que uma pessoa possa estar em contato de todas as suas tarefas em andamento, resultados atingidos e demais informações, é necessário que essas informações possam ser acessadas a qualquer tempo e de qualquer lugar. Para isso, dispositivos computacionais distribuídos pelos ambientes em que esta pessoa está inserida podem trocar informações entre si, assim como os sistemas integrados podem fornecer informações contextuais de diversas fontes e podem possuir diversos tipos de estruturas [3].

Esse conjunto de dispositivos cria um ambiente de diversas fontes de informação, sensores, sistemas embarcados, altamente distribuídos. A tecnologia de “Computação Ubíqua” (*Ubiquitous Computing*) [5] consiste em ambientes/redes onde os dispositivos são integrados e, assim, se possa ter acesso às informações por eles capturadas.

Aplicando este conceito no contexto deste trabalho, um Agente Assistente Pessoal pode se comunicar com essas redes ubíquas a fim de complementar suas informações e, assim, tomar uma melhor decisão. O usuário pode possuir sistemas que fazem a comunicação com o agente, na forma de interfaces implantadas em dispositivos móveis (PDA, celular, etc.). Dessa forma, um AAP pode buscar e/ou filtrar as informações capturadas dessas redes e, além de tomar decisões, pode mostrá-las ao usuário por inferir (com base em uma lista de preferências e/ou necessidades associadas à tarefa em execução, previamente configuradas) que são importantes

de serem exibidas. Isso permite que usuários também possam executar tarefas.

O conceito de “Consciência de Contexto” (*Context-Awareness*) é parte integrante da computação ubíqua e grande impulsionador do desenvolvimento de novas interfaces com usuários: interfaces adaptativas e dinâmicas. Inicialmente motivado pela necessidade de se prover informações turísticas em aeroportos, museus e em automação de ambientes, passou para um estado da tecnologia e de necessidades dos usuários em que não bastava ter acesso às informações exatamente como elas eram produzidas; fazia-se necessário “adaptá-las” para que fossem efetivamente úteis e utilizadas. Tais adaptações – em tempo de execução – envolvem formatos e protocolos que o usuário tem em seu ambiente computacional vigente, o seu idioma, o contexto de um dado processo, filtragens de informações desnecessárias, restrições de acesso por questões de segurança, tipo de usuário, etc. [5]. Portanto, se devidamente configurados, os dispositivos móveis podem reagir de acordo com essas especificações [3].

Com as Tecnologias de Informação e Comunicação previamente avaliadas e aqui apresentadas, é possível propor um diagrama de implantação do Arcabouço para Agentes Assistentes Pessoais (Fig. 3):

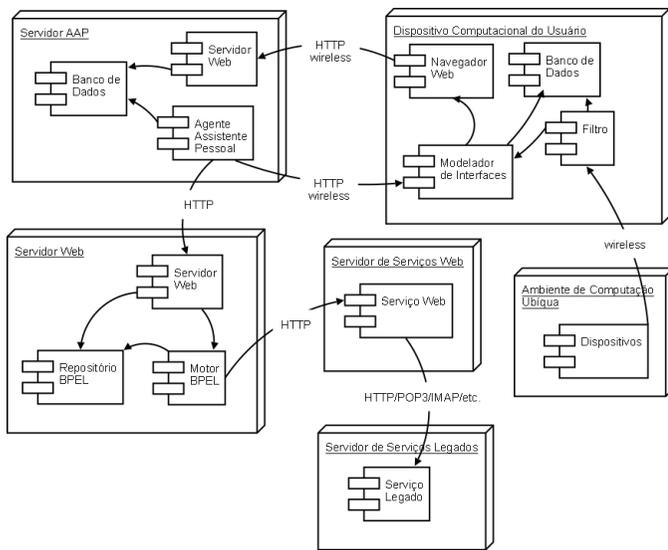


Fig. 3. Diagrama de Implantação do Arcabouço para Agentes Assistentes Pessoais.

O diagrama apresenta o Servidor de AAPs que pode ser caracterizado como um computador servidor onde esta em execução o banco de dados com as informações dos usuários, o núcleo de execução dos agentes e o aplicativo que deve rodar servidor web para a configuração do AAP. Este é acessado via internet pelo dispositivo computacional do usuário (Computador *desktop*, PDA, Telefone celular).

Do lado do usuário, em cada dispositivo móvel é necessária a instalação de um sistema que faz a comunicação entre o agente e o usuário. Os documentos BPEL podem ficar no mesmo computador servidor dos agentes ou em servidores separados e distribuídos na internet. Eles são acessados e invocados remotamente para a execução de cada um dos comportamentos do agente. Estes então acessam serviços *web*

que podem estar no mesmo computador ou em outros, distribuídos pela internet.

O ambiente de computação ubíqua se encaixa no arcabouço como um ambiente integrado onde dispositivos computacionais estão dispostos no ambiente, interagindo com os dispositivos móveis do usuário. Essas informações devem ser filtradas pelo dispositivo computacional do usuário de forma a obter apenas as informações que são de interesse do mesmo ou que possam ser importantes para alcançar os objetivos do agente e do usuário.

IV. PROTÓTIPO PARCIAL

Durante as pesquisas até agora efetuadas, um primeiro protótipo parcial foi desenvolvido para testes iniciais do modelo e para servir como embasamento prático para a continuidade do projeto. Neste foi desenvolvido um comportamento para a gerência de uma conta de e-mails de um usuário do AAP. Este comportamento representa uma das instâncias da primeira macro-ação mencionada na seção I. Para este protótipo, foi desenvolvida uma interface web com o propósito de criação e configuração dos AAPs em linguagem de programação PHP e acessando uma base de dados MySQL. Os comportamentos foram modelados em BPEL em um servidor Apache ODE. Os comportamentos, por sua vez, acessam serviços web em um servidor Apache Tomcat com Axis. Por fim, a implementação do núcleo de execução do agente foi desenvolvido em linguagem de programação Java.

Neste comportamento, o AAP verifica as informações do usuário, por exemplo, a situação no Gtalk (*Available, unavailable, away*), fazendo uma verificação de novas mensagens que chegam à conta de e-mail do usuário. Quando uma nova mensagem chega, o AAP analisa esta e toma uma atitude em conformidade com um conjunto de regras previamente definidas. A Fig. 4 apresenta, de forma geral, a execução dos passos neste caso:

1. O usuário configura suas preferências no Agente Assistente Pessoal.
2. Uma nova tarefa ou mensagem chega ao servidor de e-mails.
3. O comportamento responsável por ficar monitorando o servidor de e-mails verifica que há uma nova mensagem.
4. O comportamento de monitoramento de e-mails envia então uma mensagem para o AAP, informando-o da nova situação.
5. O núcleo de execução do agente entra em ação pelo estado desse evento e acessa seu banco de dados para verificar objetivos e necessidades do usuário.
6. O AAP seleciona o comportamento correspondente para tal tarefa, configura-o com as preferências do usuário e faz sua invocação, retornando o resultado de sua execução.

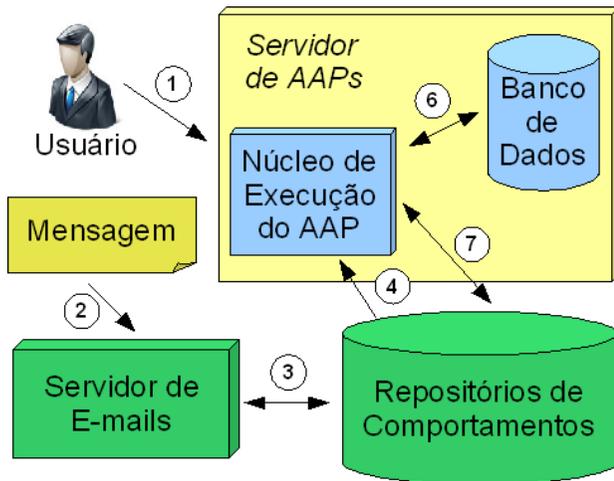


Fig. 4. Gerenciamento da conta de e-mails do usuário.

O AAP utiliza a execução dos documentos BPEL para descrever os comportamentos dos agentes com variáveis, estruturas, condicionais, assim como acessam funções locais ou remotas na forma de serviços. Graças a este recurso, é possível montar um comportamento baseado em uma macrovisão, reunindo diversas funções – ou serviços – em uma forma estruturada (Fig. 5), ou seja, fazendo uma orquestração de serviços – disponíveis na web.

```

<bpws:invoke inputVariable="WsJabberRequest"
  name="connect to jabber" operation="getPresence"
  outputVariable="WsJabberResponse"
  partnerLink="WsJabber"
  portType="ns0:WsJabber" />
<bpws:invoke inputVariable="WsMailMonitorRequest"
  name="connect to mail account" operation="open"
  outputVariable="WsMailMonitorResponse"
  partnerLink="WsMailMonitor"
  portType="nsl:WsMailMonitor" />
<bpws:while name="Jabber user unASVAILABLE">
  <bpws:condition>
  <![CDATA[ $WsJabberResponse.openReturn =
  'unASVAILABLE' ]]>
  </bpws:condition>
  <bpws:sequence name="Sequence">
  <bpws:invoke inputVariable="WsMailMonitorRequest"
  name="verify emails" operation="hasMessage"
  outputVariable="WsMailMonitorResponse"
  partnerLink="WsMailMonitor"
  portType="nsl:WsMailMonitor" />
  <bpws:if name="new email">

```

Fig. 5. Parte de um documento BPEL para o gerenciamento dos e-mails.

Neste exemplo, o AAP pode identificar a situação do usuário no GTalk e algumas palavras especiais no assunto das mensagens, por exemplo, URGENTE, TAREFA, TRABALHO, FAZER. Desta forma, ele pode responder aos e-mails ou executar essas atividades em conformidade com as informações registradas na base de dados do usuário. Na Fig. 5, o agente invoca a operação "getPresence" do Gtalk (serviço web WsJabber) e a operação "open" da conta de e-mails (serviço web WsMailMonitor). Caso seja verificado que o usuário está "unavailable" no Gtalk, o agente inicia o procedimento para o gerenciamento de novas mensagens nesta conta de e-mails.

V. CONSIDERAÇÕES E PRÓXIMOS PASSOS

Este artigo apresentou uma proposta de um Arcabouço para Agentes Assistentes. Estes agentes visam fornecer um suporte ao usuário com o intuito de minimizar o tempo gasto na execução de tarefas repetitivas e tediosas, ou quando uma pessoa não possa estar disponível ou encontram-se demasiado ocupada.

A arquitetura do AAP – concebido como um agente de software – e a implementação de um protótipo parcial para o caso de "leitura/resposta de mensagens de e-mail" são apresentadas de forma preliminar com a utilização de um conjunto de ferramentas de padrões abertos.

Foi apresentado um modelo representativo de um AAP na execução de certas tarefas como o monitoramento de e-mails e a gerência destes conforme preferências do usuário. A arquitetura foi mostrada em um nível conceitual com a indicação dos módulos relevantes para a execução dos processos dos AAPs. Foi apresentada uma análise preliminar e as potenciais tecnologias para um primeiro protótipo.

Os próximos passos do trabalho incluem refinamentos no modelo conceitual e na arquitetura genérica, apresentando de forma mais detalhada a utilização de cada uma das seis macroações que precisam ser tratadas, assim como a análise das tecnologias e padrões para a implementação final. Ainda, um novo protótipo deve ser modelado e implementado nas próximas fases do trabalho. A execução do protótipo parcial alcançou o que foi proposto nesta fase do projeto. Contudo, a principal motivação do trabalho é gerenciar não apenas uma tarefa, mas diversas tarefas, de forma dinâmica, detectando as informações que chegam e encontrando o comportamento que mais se adapta a cada situação. Além disso, o arcabouço deve permitir que o usuário agregue novos comportamentos dinamicamente.

VI. AGRADECIMENTOS

Este trabalho foi parcialmente financiado e desenvolvido no escopo do projeto brasileiro IFM-II (<http://www.ifm.org.br>) e do projeto europeu IST IP ECOLEAD (<http://www.ecolead.org>).

VII. REFERÊNCIAS

- [1] S. Bocionek, Software secretaries: learning and negotiating personal assistants for the daily office work. Systems, Man, and Cybernetics, 1994. 'Humans, Information and Technology', 1994 IEEE International Conference on, v. 1, 1994.
- [2] A. Angehrn, et al, K-InCA: Using Artificial Agents for Helping People to Learn New Behaviours. Proc. IEEE International Conference on Advanced Learning Technologies, p. 225-226, 2001.
- [3] G. D. Abowd, et al. Cyberguide: A mobile context-aware tour guide. Wireless Networks, v. 3, n. 5, p. 421-433, 1997.
- [4] S. Schiaffino; A. Amandi, Polite Personal Agents. IEEE INTELLIGENT SYSTEMS, p. 12-19, 2006.
- [5] K. Henriksen; J. Indulska, Developing context-aware pervasive computing applications: Models and approach. Pervasive and Mobile Computing, v. 2, n. 1, p. 37-64, 2006.
- [6] G. Weiss, Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence. MIT Press, 1999.
- [7] G. Alonso, Web Services: Concepts, Architectures and Applications. Springer, 2004.
- [8] M. N. Huhns; M. P. Singh, Personal Assistants. 1998.

- [9] D. H. Akehurst, Validating BPEL Specifications Using OCL. University of Kent at Canterbury, technical report, p. 15-04, 2004.
- [10] IBM. (2007) "Business Process Execution Language for Web Services". <http://www-128.ibm.com/developerworks/library/specification/ws-bpel/> (March).
- [11] Michael C. (2004) "What Is BPEL And Why Is It So Important To My Business?" SoftCare EC. http://www.softcare.com/whitepapers/wp_what_is_bpel.php. (December).
- [12] A. Carrilho-Ramos et AL, Knowledge Management for Adapted Information Retrieval in Ubiquitous Environments. Proceedings of the 2nd International Conference on Web Information Systems and Technologies (WEBIST 2006)(Setubal, Portugal, April 11-13, 2005), Insticc Press, Portugal, p. 21-29, 2006.
- [13] J. Bush; J. Irvine; J. Dunlop, Personal Assistant Agent and Content Manager for Ubiquitous Services. Wireless Communication Systems, 2006. ISWCS'06. 3rd International Symposium on, p. 169-173, 2006.
- [14] R. D. Franco, Et AL, Supporting mobile virtual team's coordination with soa-based active entities. International federation for information processing-publications-ifip, v. 243, p. 557, 2007.
- [15] M. A. Hoyle; C. Lueg, Open Sesame!: A Look at Personal Assistants. Proceedings of the International Conference on the Practical Application of Intelligent Agents (PAAM97), London, 1997, 51, v. 60
- [16] W. Vieira, Agentes Móveis Adaptáveis para Operação Remota. PhD, Universidade Nova de Lisboa, Lisboa, 2000.
- [17] L. Loss, Aprendizado de redes colaborativas de organizações: um arcabouço para dar suporte à criação e uso de conhecimento. 2007. Tese (Doutorado) - Curso de Engenharia Elétrica, UFSC, Florianópolis, 2007.

VIII. BIOGRAFIAS



Msc. Saulo Poppov Zambiasi possui graduação em Ciência da Computação pela Universidade do Oeste de Santa Catarina - Campus de Chapecó (1998), especialização em Ciência da Computação pela Universidade Federal de Santa Catarina (2000) e mestrado em Ciências da Computação pela Universidade Federal de Santa Catarina (2002). Atualmente é Pesquisador - Estudante do Programa de Pós-Graduação em Engenharia de Automação e Sistemas da Universidade Federal de Santa Catarina. É professor das Faculdades Barddal (em licença para estudos) e Revisor de periódico da Revista IEEE América Latina. Tem experiência na área de Ciência da Computação, com ênfase em Computação Aplicada. Atuando principalmente nos seguintes temas: Inteligência Artificial Distribuída, Sistemas Multiagentes, Agentes Inteligentes, Computação Gráfica, Jogos de Computador e Automação Residencial.



Dr. Ricardo José Rabelo concluiu o doutorado em Engenharia Elétrica pela Universidade Nova de Lisboa (UNL) em 1997. É Professor Associado do Departamento de Automação e Sistemas da Universidade Federal de Santa Catarina (UFSC) desde 2000, é coordenador do GSIGMA – Grupo de Sistemas Inteligentes de Manufatura. Suas atuais áreas de interesse incluem: redes colaborativas organizações, arquitetura orientada a serviços, gestão de conhecimento, sistemas multiagentes, e sistemas de apoio à decisão multi-critérios. Esteve envolvido em vários projetos de pesquisa europeus e brasileiros, tem mais de 90 publicações, incluindo conferências, periódicos e capítulos em livros. É também membro de diversas comissões do programa conferências nacionais e internacionais.