

Uma Arquitetura de Referência para Assistentes Pessoais: Uma Abordagem Baseada em Agentes e Serviços de Software

Saulo Popov Zambiasi¹ & Ricardo J. Rabelo¹

¹Programa de Pós-Graduação em Engenharia de Automação e Sistemas (PPGEAS)
Universidade Federal de Santa Catarina (UFSC)
Bairro Trindade, Caixa Postal 476
CEP 88040-900 – Florianópolis – SC – Brasil
Telefone/Fax: +55 (48) 3721-7793
{popov, rabelo}@das.ufsc.br

Resumo

A crescente evolução das tecnologias da computação e o mercado cada vez mais competitivo tem obrigando as empresas a se aperfeiçoarem no âmbito de seus processos. Novas ferramentas têm surgido para fornecer o devido suporte com respostas mais rápidas e confiáveis aos clientes e fornecedores. Porém, se por um lado têm facilitado o trabalho das pessoas, por outro as têm levando a um aumento das suas tarefas e responsabilidades. Uma maneira alternativa de resolver essa problemática é prover às pessoas uma assistência via software de computador, munindo-as de um conjunto de ferramentas para automatizar algumas tarefas e para gestão das suas atividades, assim como um secretário humano especializado. Vários esforços vêm sendo feito para a criação de sistemas computacionais com esse objetivo. Porém, esses geralmente seguem por caminhos distintos que não necessariamente mantêm a compatibilidade e a interoperabilidade uns com os outros, além de não se preocuparem em estarem conectados com os sistemas empresariais. Neste sentido, este trabalho visa propor uma arquitetura de referência para softwares assistentes pessoais a partir da qual instâncias

possam ser feitas consoante as características das pessoas e das empresas onde trabalham.

Palavras-chave: Assistentes Pessoais, Agentes, Serviços de Software, Arquitetura.

1. INTRODUÇÃO

Com a crescente evolução das Tecnologias de Informação e Comunicação (TIC) e o aperfeiçoamento das ferramentas computacionais disponíveis para pessoas e organizações, o mercado tem se tornado cada vez mais competitivo. Isso tem obrigado as empresas e organizações a se aperfeiçoarem no âmbito de seus processos em vários níveis de abstração, conceitos e tarefas [1, 2].

Não obstante a isso, a necessidade de antecipar novas tendências, e responder a elas adequadamente, tais como a implantação e utilização de novos métodos de trabalho, tem se mostrado de vital importância para a sobrevivência, continuidade e ampliação dos negócios das organizações inseridas nesse novo contexto e de, conseqüentemente, tornar os processos organizacionais mais ágeis em conformidade com a conjectura de cada atividade específica [2, 3].

À luz de questões referentes às atividades organizacionais e seus procedimentos de supervisão, as novas ferramentas disponíveis passam a fornecer um grande suporte à inserção de inovações, que vão ao encontro de respostas mais rápidas e confiáveis aos clientes e fornecedores.

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 das suas tarefas e responsabilidades [4]. Além disso, observa-se empiricamente que elas têm estado cada vez mais ocupadas e imersas em diversos problemas, alguns desses conseqüentes da própria implantação dessas novas tecnologias, e nem sempre conseguindo cumprir com suas tarefas em tempo hábil e/ou com a qualidade esperada.

Nesse sentido, o principal aspecto que se deseja focar neste trabalho é o que permeia o tempo das pessoas empregado no seu trabalho e outras formas onde as TICs podem contribuir para ajudá-las.

A premissa disso é que as pessoas são o mais nobre dos recursos organizacionais. Portanto, a sua participação deveria ser direcionada para a realização de atividades tão essenciais hoje em dia quanto à operação dos processos em si, ou seja, para atividades que agregam maior valor pessoal, interpessoal e para a própria entidade da qual participam, tais como a realização de negócios, o exercício da criatividade para a inovação, a aprendizagem e o aperfeiçoamento de suas funções.

Contudo, a rotina diária das organizações apresenta uma ruptura nas idéias acima ilustradas, obrigando as pessoas a se dedicarem a tarefas repetitivas, desgastantes, pouco atrativas e muitas vezes sem grande valor [5]. Mesmo nas tarefas que agregam maior valor, as pessoas encontram-se tão ocupadas que não conseguem executá-las a tempo e/ou com a qualidade esperada. Isso tem efeito direto na agilidade e na eficiência das organizações como um todo, principalmente ao se considerar que as organizações funcionam de forma cada vez mais integrada, com suas várias atividades dependentes cada vez mais umas das outras.

Toda essa situação acaba por culminar em uma problemática ainda maior à medida que se observa que as pessoas geralmente estão envolvidas em diversas tarefas simultaneamente, por vezes de diferentes setores de atuação, com terminologias próprias e com seus próprios contextos de execução e de negócio [6].

2. SOFTWARES ASSISTENTES PESSOAIS

De uma maneira geral, a referida problemática poderia ser amenizada com a disponibilidade de mais pessoas para a crescente quantidade de tarefas, assim como para suprir o aumento da complexidade dessas; ou mesmo de algum tipo de secretário ou ajudante para as pessoas em suas atividades [7]. Porém, quando levada em consideração a realidade financeira da maior parte das organizações (micro, pequenas e médias), o custo final disso torna esse caminho usualmente inviável.

Mesmo que intrinsecamente alguns cargos empresariais necessitem fornecer aos seus respectivos responsáveis um ou mais secretários, ou assistentes, isso é suficiente apenas para essas pessoas e não supre a necessidade da grande maioria dos funcionários. Portanto, parece justificável o uso de uma forma alternativa de assistência para as pessoas. Tal assistência poderia ser fornecida, de maneira alternativa, via software de computador, munindo as pessoas de um conjunto de ferramentas para automatização de suas tarefas e assistência na gestão das atividades, muitas vezes tal qual um secretário humano [7].

A utilização de uma ferramenta computacional para fornecer assistência às pessoas inseridas nas empresas segue em conformidade com a evolução das TICs, em que equipamentos cada vez mais potentes, menores, acessíveis e inteligentes permitem que a interação deixe de ser apenas entre parceiros humanos. Tal interação passa agora a ser tratada em negociações entre humanos e computador ou mesmo entre computadores [8], facilitando a interação entre uma pessoa e um assistente pessoal, por exemplo.

2.1. CONCEITOS

O conceito de um software com capacidade de antecipar as necessidades dos usuários e executar ações sem a necessidade de interação, tal como agendar viagens, restaurantes, gerenciar agenda e e-mails, não é novo e tem influenciado a imaginação de muitos escritores de ficção científica e cientistas da computação [9, 10].

É possível imaginar um futuro com assistentes baseados em conhecimento e que funcionam através da Internet na forma de um tipo de software secretário, provendo serviços tanto para tarefas do trabalho como para casa. Contudo, tal como secretários humanos, seu sucesso deve depender do quanto este tem conhecimento sobre os hábitos dos seus usuários [10].

Do ponto de vista organizacional, o conceito de softwares assistentes pessoais pode ser visto como uma ferramenta que fornece assistência em alguns níveis de

auxílio no cenário organizacional como forma de resolver em parte a questão da crescente quantidade de tarefas das pessoas [13].

Todavia, um software assistente pessoal não pode ser visto apenas como um programa de computador personalizado. Isto porque um assistente pessoal deve ser baseado em rede, ser interativo, adaptativo, de propósito geral, de execução autônoma e que possa interagir com outros assistentes pessoais, especificidades correlatas com a tecnologia de agentes de software [11].

Agentes se caracterizam como algo que, provido de sensores, recebe informações do ambiente ao seu redor e toma ações, interagindo com este ambiente por meio de mecanismos denominados de atuadores. Além disso, os agentes devem poder se comunicar com outros para poderem alcançar seus objetivos [12]. Seguindo este conceito, os assistentes pessoais, tal qual um agente, podem perceber seu ambiente através de recursos (como a agenda do usuário, preferências, preferências de outros usuários, reputações) e respondem com ações e assistência ao usuário. Em tempo, um assistente pessoal precisa trabalhar com certa autonomia para cumprir suas tarefas, tal qual um agente [13].

Dessa forma, um agente de software pode substituir parcial ou totalmente uma pessoa para determinadas tarefas e em várias situações diferentes, assumindo o papel de um assistente, muitas vezes sendo necessário se comunicar com outros sistemas, ou outros agentes, sem a intervenção direta ou indireta de um humano [14]. Além disso, os agentes devem refletir a mesma dinâmica que um usuário apresenta na vida real, ou seja, assim como os interesses do usuário mudam com o tempo, os agentes também devem ajustar seus objetivos para corresponder com seus usuários [15].

Assim, os assistentes pessoais podem ser vistos como softwares na forma de agentes, ou agentes pessoais, que fornecem assistência as pessoas em suas atividades diárias. Em verdade, estes já têm sido, e continuarão sendo, utilizados em vários domínios de aplicação [16].

Assim, seguindo a tendência de alguns autores e correlacionando as características de agentes e de assistentes pessoais, este trabalho usa a denominação de Agente Assistente Pessoal (AAP) para um software que fornece assistência à um usuário, aglutinando as terminologias apresentadas por [11] e [16].

Apesar dos avanços na área de TIC em si, uma perspectiva a se chamar a atenção em função da observação dos autores do presente artigo em relação a trabalhos na área, é a utilização ou desenvolvimento de

softwares assistentes de forma não integrada ao ambiente e aos processos de negócios (*business process*) da empresa. No primeiro aspecto, os assistentes são implementados como sistemas independentes dos demais da empresa, sem interação ou inter-operação alguma, fazendo com que o usuário tenha que, explicitamente, navegar por ambientes diferentes e, pior, tenha que redigitar informações e nutrir o assistente de informações contextualizadas. No segundo aspecto, que está intimamente relacionado ao primeiro, o assistente usualmente atua como um sistema puramente reativo à interação do usuário, e não de forma ativa (ou até pró-ativa) mas consoante a uma lógica de negócios expressa nos processos de negócios da empresa, executando as tarefas de forma integrada com outros sistemas dela.

Apesar de estar associada a TICs, uma segunda perspectiva é a inter-operação. Os assistentes pessoais costumam ser softwares concebidos para realizarem suas atividades mas sem uma preocupação de projeto de transacionarem com outras aplicações. Na ótica de se integrar o assistente ao ambiente geral de execução de processos da empresa, essa perspectiva é essencial uma vez que usualmente os sistemas dela são distribuídos e heterogêneos.

A terceira perspectiva a se chamar a atenção é quanto a flexibilidade das ações de um assistente. Isso envolve outros aspectos, como adaptabilidade e escalabilidade. No escopo de transações empresariais, cada negócio tem inúmeras particularidades, tanto de contexto como de ambientes computacionais envolvidos. Por exemplo, um assistente deve oferecer ajuda de forma diferente a uma mensagem referente a um processo de venda de produto (a clientes) a uma de compra de produtos (de fornecedores) [7].

Portanto, há necessidade de adaptabilidade àquelas mesmas transações comerciais que não são estáticas. Tanto a empresa como seus sistemas alteram-se ao longo do tempo com relação aos seus processos de negócios. Dessa forma, há necessidade de se lidar com a escalabilidade do software assistente, em termos de que deve estar preparado para flexibilizar suas ações consoante a mudanças e/ou introdução de novos processos de negócios. O aspecto final é o da distribuição de aplicações na forma de serviços serviços.

As empresas, embora tradicionalmente tenham seus sistemas implantados todos localmente, começam a conviver com ambientes largamente distribuídos, fazendo uso de sistemas externos, disponibilizados por outras empresas que são, principalmente, provedores externos de serviços. Tais provedores são tipicamente formados de *software-houses* e por dispositivos

computacionais distribuídos. Nesse sentido, o software assistente deve poder acessar tais serviços consoante os requisitos funcionais associados aos processos de negócios ora em execução, o que pode levar ao caso da necessidade de composição de diferentes serviços para que o comportamento desejado possa ser montado.

2.2. REQUISITOS GERAIS

Tomando como base os conceitos apresentados, certos requisitos gerais de base desejáveis para os AAPs podem ser identificados.

O AAP deve atuar com certa **autonomia** em suas tarefas de forma que estes possam verificar a situação dos seus usuários (como por exemplo novos e-mails que chegam em sua caixa postal), e que possam responder apropriadamente a cada situação que aparece.

Ele deve ser **flexível** em termos de poder atuar diante de novas situações e cenários de negócios.

Além disso, é necessário ser **adaptável** ao usuário, em conformidade com as informações sobre ele, suas preferências, necessidades, situação atual, e tomando em conta a evolução dessas informações em relação ao tempo.

O AAP também deve **interagir** com seu usuário de forma a tornar possível a assistência a ele.

É importante que seja **baseado em rede**, ou seja, ter a capacidade de buscar informações via redes de computador, tal como a Internet, e interagir com outros sistemas e assistentes pessoais de outros usuários de forma a alcançar os objetivos do seu usuário.

Ainda, deve ser **inter-operável** com os elementos envolvidos nos processos para fornecer assistência pessoal.

É necessário que seja **de propósito geral**, ou seja, não específico à apenas uma atividade ou um grupo de usuários, adaptando-se aos propósitos do seu usuário;

Por fim, o AAP deve ser **adaptável ao contexto e integrado e inter-operável aos processos de negócios da empresa**.

Do ponto de vista funcional, agora mais concretamente para um assistente no ambiente de empresa, são identificadas também algumas macroações que espera-se de um AAP:

- Substituir o usuário em determinadas tarefas no contexto dos processos de negócio da empresa;
- Gerenciar a agenda e atividades do usuário;

- Procurar por informações em repositórios;
- Auxiliar a identificar quais tarefas precisam ser executadas e como;
- Fornecer recomendações para o usuário;
- Filtrar e gerenciar informações providas de redes ubíquas.

2.3. SUB-PROBLEMAS ENVOLVIDOS

Contudo, para um assistente pessoal poder cumprir com os requisitos acima definidos, alguns sub-problemas são identificados.

Relevância das Informações: Identificação de quais informações sobre os usuários são realmente relevantes e como elas podem ser utilizadas na execução da assistência aos usuários [17, 11, 16].

Conhecimento dinâmico: Um AAP precisa se adaptar as mudanças de interesses, objetivos dos usuários e ao contexto dos ambientes em que o usuário transita [4, 15].

Interação Usuário-AAP: O AAP deve poder identificar quando e como a interação com o usuário deve ocorrer [7, 17, 18].

Portabilidade: A interação do usuário com seu AAP deve ser independente de onde ele esteja localizado e do dispositivo computacional que se está utilizando [8, 18].

Disponibilidade: Nos casos dos comportamentos dos AAP estarem localizados em repositórios remotos, deve haver uma forma de tratar indisponibilidades quando os repositórios não estão acessíveis. Este não é um problema com solução trivial [19, 20, 17].

Flexibilidade: Um AAP deve se adaptar, em termos de funcionalidades, as necessidades do seu usuário de forma dinâmica e flexível. O problema está em como tornar possível à um usuário de um AAP reunir um conjunto de comportamentos para se adaptar as suas necessidades e, em certos casos, organizá-los de forma algorítmica para resolver um problema específico [10].

Segurança: A segurança engloba todo o sistema computacional. Porém, a ênfase está na distribuição dos comportamentos do AAP na Internet, na comunicação dos agentes com os sistemas e na interação com o usuário [19].

Conflitos: Mesmo com os comportamentos distribuídos, podem existir problemas de conflitos na ordem da execução destes no caso um depender de outro(s) para executarem seu processamento [7].

Inter-operabilidade: Dever haver inter-operabilidade entre todos os componentes do modelo, tornando possível a utilização de comportamentos já existentes e largamente utilizados na internet (serviços) como fonte de funcionalidades a serem utilizadas pelos AAPs [21, 22].

Previsão e Monitoramento: Encontrar novas oportunidades de negócios não é tarefa fácil. Para tal, é necessário ter como base as informações e preferências dos usuários, eventos e estados locais e atuais, e um certo nível de inteligência no AAP.

Softwares Proprietários: Muitos dos trabalhos na área são proprietários, fechados e não possuem um padrão, não dando suporte às empresas desenvolverem seus próprios assistentes, inter-operáveis com assistentes de outras empresas. Também não fornecem recursos para a utilização explícita de outras funcionalidades (em repositórios de terceiros) [23, 24, 25].

Softwares Monolíticos: Os softwares estudados se mostraram na forma de um sistema de processamento centralizado em um único sistema computacional.

3. PROPOSTA

Este trabalho objetiva fundamentalmente conceber uma arquitetura de referência para assistentes pessoais baseada em agentes e em serviço de softwares, mais especificamente na Arquitetura Orientada a Serviços (SOA). Esta seção descreve a proposta da arquitetura e cada um dos elementos que a compõe.

Uma arquitetura de referência é caracterizada por um padrão genérico para um projeto, abordando requisitos para o desenvolvimento de soluções e que é guiado pelo modelo de referência, de forma a atender as necessidades do projeto [22]. Para compor uma arquitetura de referência, é necessário apresentar os tipos dos elementos envolvidos, como eles interagem e o mapeamento das funcionalidades para estes elementos [26].

De fato, a definição de uma arquitetura de referência para agentes assistentes pessoais que segue conforme especificidades e requisitos apresentados na seção II. B., já se caracteriza como um fator de grande importância para a inovação pretendida. Contudo, são ainda propostos novos requisitos – que devem guiar a definição da arquitetura – de forma a manter uma maior inter-operabilidade entre os elementos envolvidos, além de uma maior flexibilidade na composição dos comportamentos do AAP. Estes requisitos são:

- A arquitetura deve ser baseada em padrões abertos e atuais, independente de tecnologia e que possa ser facilmente adaptável (flexível).
- O AAP deve poder ser implementado em linguagens e tecnologias tanto comerciais quanto gratuitas ou *opensource*.
- O AAP deve permitir que novos comportamentos, processos para execução das atividades do agente, 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 seja necessário reconfigurar o agente, ou que as alterações sejam mínimas.
- Deve haver a possibilidade dos comportamentos do AAP não estarem necessariamente localizados no mesmo computador do núcleo de execução do agente em si, ou seja, poderem estar distribuídos em repositórios de comportamentos na Internet, podendo ser utilizados gratuitamente, alugados ou vendidos por terceiros.
- Por fim, um comportamento deve poder ser invocado pelo AAP apenas quando este estiver habilitado para tal e todas as suas condições para ativação forem satisfeitas.

É importante salientar que o uso de uma arquitetura de referência para o desenvolvimento de AAPs, que segue os requisitos acima citados pode trazer certas vantagens, tais como a inter-operabilidade, a flexibilidade dos comportamentos do agente, a independência de fornecedores de assistentes pessoais e comportamentos, a possibilidade de qualquer pessoa poder ter seu assistente pessoal, com os mais variados tipos de comportamentos, gostos e custos.

3.1. VISÃO GERAL

O Agente Assistente Pessoal (AAP) é, no contexto deste trabalho, um processo computacional criado para representar um usuário na execução de certas tarefas, automaticamente ou com algum grau de intervenção/supervisão humana. Seu macro-comportamento é composto pelo conjunto de comportamentos distribuídos na Internet e que podem ser executados paralelamente.

A visão do macro-comportamento que o AAP possui é o subconjunto dos possíveis comportamentos que podem ser utilizados. Este subconjunto é escolhido pelo usuário e agregado ao AAP como forma de satisfazer suas necessidades. Tal agregação dos comportamentos se dá pela seleção de comportamentos (serviços Web) de repositórios de comportamentos via rede de computadores/Internet, e a uma posterior configuração deste em um ambiente de configuração do AAP. Algumas informações são o endereço do local onde este comportamento está alocado, os parâmetros que precisam ser passados e as informações que possam ser relevantes para a execução de tal comportamento.

Tendo em vista a composição dos requisitos e especificidades dos assistentes pessoais descritas até este ponto, é possível esboçar uma visão geral do cenário dos AAPs, conforme a proposta do presente trabalho, apresentada na Figura 1.

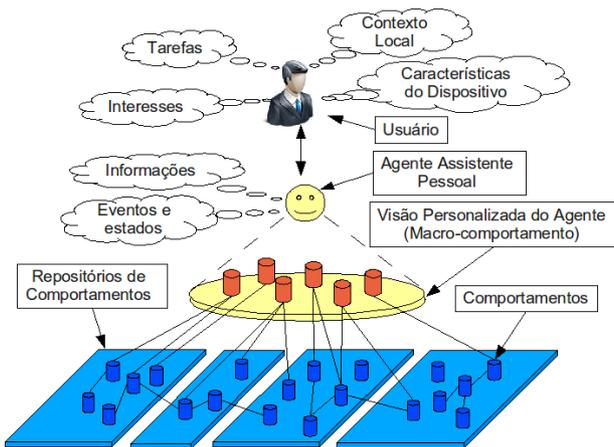


Figura 1: Visão geral do AAP.

Essa visão é composta pelo usuário, inserido em um contexto local de negócios, que possui um conjunto de tarefas que são de sua responsabilidade, interesses pessoais e organizacionais e características específicas do dispositivo computacional que o usuário está utilizando. Por meio desse conjunto de elementos, o usuário interage com seu assistente pessoal. Este, por sua vez, possui uma representação das informações de interesses do usuário, e de um conjunto de informações relativas aos estados e eventos do usuário e dos comportamentos em execução.

Formalizando a visão personalizada do macro-comportamento do AAP da Figura 1, tem-se A um conjunto não vazio, tal que A representa o conjunto de n comportamentos distribuídos em uma rede de computadores $\{a_1, a_2, \dots, a_n\}$. Define-se como B o comportamento resultante (ou macro-comportamento) do AAP, sendo caracterizado pelo conjunto 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 (1), logo, o comportamento de B é caracterizado por

$$B = \sum_{i=1}^n b_i \quad (1)$$

Em suma, o comportamento do Agente Assistente Pessoal, ou macro-comportamento, é o subconjunto de comportamentos selecionados e configurados para este, da totalidade de comportamentos disponíveis e distribuídos em uma rede de computadores, em conformidade com os padrões abertos de conectividade e comunicação aceitos pelo mesmo. Os comportamentos são serviços de software que podem ser formados pela orquestração de outros serviços de software.

3.2. ARQUITETURA

Dando seqüência a proposta deste trabalho, a arquitetura concebida é apresentada na Figura 2. Esta está organizada seguindo um modelo em três camadas. Tal representação possui a vantagem de mostrar separadamente os módulos que devem interagir com o usuário, os módulos de execução dos processos e a representação das informações persistentes. Cada camada contém os elementos que buscam cumprir com os requisitos descritos nas seções 2.2 e 3.

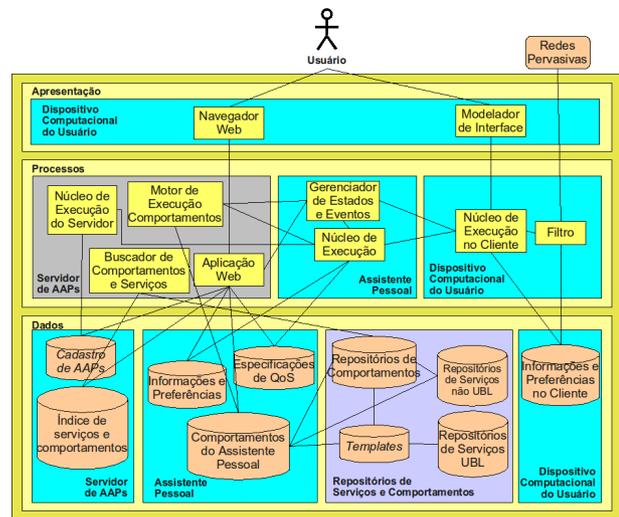


Figura 2: Arquitetura de Referência para AAPs.

Fazendo um paralelo com os requisitos para assistentes pessoais apresentados nas seções 2.2 e 3, esta proposta pretende apresentar soluções para eles da seguinte forma: A autonomia é “resolvida” com um servidor de AAPs, que instancia os AAPs dos usuários, e cada um destes é um agente autônomo. Os requisitos de flexibilidade e adaptabilidade são comportados pela

arquitetura SOA, em que os comportamentos, na forma de serviços Web, podem ser selecionados e orquestrados pelo usuário conforme sua necessidade e suas tarefas. A interação é feita pelo AAP quando necessária e quando um comportamento requerer tal interação. A interoperabilidade e a estrutura baseada em rede fica implícita na utilização de SOA para compor os comportamentos do AAP. A questão da interoperabilidade aos processos da empresa depende destes estarem em conformidade com SOA e com UBL (*Universal Business language*). Por fim, a adaptabilidade de contexto depende de como cada comportamento irá tratar o contexto de informações que são providas de redes ubíquas.

3.3. CAMADA DE APRESENTAÇÃO

A camada de apresentação é caracterizada pela maneira como o usuário interage com o AAP. Esta possui dois elementos básicos, um navegador Web e o Modelador de Interface. Estes são descritos a baixo:

- **Navegador Web:** Localizado no dispositivo computacional do usuário, serve como meio de acesso ao sistema para criação, configuração e manutenção do AAP.
- **Modelador de Interface:** Ajusta o sistema para a interação do usuário com os comportamentos, conforme as particularidades de cada comportamento. Para suportar essa funcionalidade, toma-se como exemplo a proposta de [21], que apresenta uma forma de criação de uma interface gráfica de usuário automática e dinamicamente a partir de uma descrição de um serviço Web e, a partir dessa, uma invocação dinâmica ao serviço Web.

3.4. CAMADA DE PROCESSOS

Camada caracterizada pelos elementos do processamento do AAP.

- **Núcleo de Execução do Servidor:** Programa servidor responsável por instanciar e gerenciar a execução dos AAPs registrados em uso.
- **Núcleo de Execução no Cliente:** Extensão ativa do AAP no cliente para: (i) gerenciar a comunicação entre o usuário e o AAP via modelador de interface, (ii) gerenciar e sincronizar as informações do dispositivo computacional do usuário com o banco de dados no AAP, (iii) interpretar novas informações que

chegam, previamente filtradas pelo “Filtro” e enviar o resultado para o AAP, (iv) informar os estados, eventos, ações e situação do usuário ao “Gerenciador de Estados e Eventos” no AAP.

- **Filtro:** Módulo responsável por filtrar novas informações que chegam, conforme preferências e interesse do usuário ou possível utilização por algum comportamento do AAP. Apenas o que é realmente relevante ao usuário e ao processamento do AAP é processado.
- **Núcleo de Execução:** Núcleo de processamento central do AAP, responsável por: (i) escalonar os comportamentos do AAP na forma de processos no “Motor de Execução de Comportamentos”; (ii) gerenciar os comportamentos que estão em execução; (iii) decidir, por meio da base de informações e do “Gerenciador de Estados e Eventos”, que comportamentos devem entrar em execução e instanciá-los; (iv) gerenciar conflitos de comportamentos; (v) comunicar-se com o “Núcleo de Execução no Cliente” quando necessário interação com o usuário. Devido ao processamento dos comportamentos serem distribuídos, a favor da abordagem SOA, este fica liberado de grande parte do processamento de cada comportamento.
- **Gerenciador de Estados e Eventos:** Módulo funcional para atua tal como um “quadro-negro”, com informações voláteis e atualizadas sempre que possível, de forma a responder ativamente aos demais módulos. Essas informações são provenientes do dispositivo computacional do usuário e da execução ou resultados dos processamento dos comportamentos.
- **Aplicação Web:** Localizado no servidor de AAP e acessado via navegador Web, serve como interface entre a entidade que fornece (aluga, vende, ou distribui gratuitamente) AAPs e os usuários. Por meio desta, o usuário pode conhecer as características da entidade, tipos de serviços que esta fornece e as especificidades dos assistentes pessoais. Além disso, o usuário pode selecionar os comportamentos através de um “Índice de Comportamentos e Serviços

Conhecidos”, com pontuações conforme uma “Especificação de QoS” (*Quality on Service*), e pode configurar cada comportamento do AAP.

- **Motor de Execução de Comportamentos:** O “Núcleo de Execução” verifica as condições para que um comportamento entre em execução e invoca-os via “Motor de Execução de Comportamentos”, da qual mantem o “Gerenciador de Estados e Eventos” informado de qualquer modificação ou situação dos comportamento.
- **Buscador de Comportamentos e Serviços:** Facilita a composição do macro-comportamento do AAP por meio da criação de de um “Índice de Serviços e Comportamentos”, previamente pesquisados na Web. São organizados por funcionalidades e podem ser selecionados pelos usuários no momento de configurar seus AAPs. Isso não impede que o usuário possa agregar ao seu AAP comportamentos que não estejam cadastrados no índice. As funcionalidades extras deste módulo, assim como a inteligência e eficiência utilizada na procura de comportamentos nos repositórios podem ser específicas de cada implementação e de cada entidade fornecedora desta funcionalidade. Este módulo deve fazer uso dos resultados de uma Tese de doutorado [27] em vias de finalização, sendo desenvolvida no Grupo GSIGMA, do DAS.

3.4. CAMADA DE DADOS

- **Cadastro de AAPs:** Controla a execução dos AAPs cadastrados no servidor, conforme particularidades e informações configurações por cada usuário e para cada comportamento que o AAP utiliza.
- **Índice de Serviços e Comportamentos Conhecidos:** Repositório de índice de informações referentes aos comportamentos, facilitando a modelagem do macro-comportamento do AAP.
- **Informações e Preferências:** Base de informações persistente com informações referentes as preferências do usuários de cada comportamento agregados ao AAP. Essas servem para para filtrar novas informações que chegam para selecionar e iniciar a execução dos comportamentos e para conduzir a execução dos comportamentos.
- **Especificações de QoS (*Quality on Service*):** Informações para avaliação dos comportamentos conhecidos no “Índice de Comportamentos” por meio de um sistema de pontuação, auxiliando o usuário na escolha dos comportamentos para compor o macro-comportamento do seu AAP. Estas especificações devem vir essencialmente dos resultados de uma Tese de Doutorado [27] em vias de finalização, sendo desenvolvida no grupo GSIGMA, do DAS.
- **Comportamentos do AAP:** Documentos textos com uma sintaxe estruturada na forma algorítmica para organizar, gerenciar variáveis, orquestrar e instanciar os serviços que serão a composição de cada comportamento do AAP.
- **Repositórios de Comportamentos:** Provedores de serviços que seguem as especificações de SOA. Alguns destes podem passar a produzir serviços específicos para AAPs. Podem ser serviços individuais ou comportamentos compostos pela orquestração de serviços individuais.
- **Repositórios de Serviços UBL:** Fornece padronização para comportamentos de negócios, mantendo uma maior compatibilidade e interoperabilidade com outros sistemas, organizações, empresas ou assistentes pessoais de terceiros. Sugere-se que tais comportamentos sejam baseados em UBL (*Universal Business Language*). Neste momento há uma Dissertação de Mestrado sendo desenvolvida no grupo GSGIMA, do DAS, de implementação de um catálogo de serviços Web com base na especificação UBL.
- **Templates:** Para cada comportamento do AAP que segue a padronização UBL deve existir um ou mais *templates* que visam auxiliar a configuração de tal comportamento.
- **Repositórios de Serviços Não-UBL:** Demais comportamentos que não seguem o padrão UBL. Encontrados em repositórios na internet e

desenvolvidos por terceiros, não tentam manter um padrão de negociação específica.

- **Informações e Preferências no Cliente:** São todas as informações que possuem relevância na execução dos comportamentos do AAP e do filtro. Devem estar sincronizadas com a base de dados localizada no AAP. Referem-se às preferências do usuário e às informações necessárias para a execução de cada comportamento, passadas pelo usuário no momento da configuração deste.

Baseado em uma versão preliminar da arquitetura apresentada em [6], foi desenvolvido um protótipo simplificado com alguns dos componentes citados. Contudo, nem todos os módulos foram implementados e mesmo os que foram desenvolvidos foram feitos de forma bastante simples. Tal protótipo serviu como base para resolver questionamentos que estavam em aberto.

3.5. PONTOS DE ANÁLISE

Até o momento atual deste trabalho, algumas questões foram levantadas e tratadas, mas há aspectos em aberto que devem ser vistos na etapa seguinte, para a finalização da proposta como Tese de Doutorado.

A utilização de serviços Web ainda não se encontra com sua utilização abrangente o suficiente para que seja possível o fornecimento de todas as funcionalidades desejáveis para a composição um conjunto de comportamentos para um bom macro-comportamento do AAP, sendo necessário uma análise prévia de tipos de comportamentos mais gerais e alguns mais específicos que podem ser utilizados para a execução do protótipo pelas pessoas envolvidas no processo de avaliação do sistema.

O usuário deve poder acrescentar um comportamento ao macro-comportamento do AAP de forma dinâmica. Porém, ainda não foi definida a forma como o usuário irá interagir com o sistema caso ele queira criar um novo comportamento, a partir da orquestração de outros comportamentos/serviços. Tendo isso em vista, existem três alternativas a serem avaliadas:

- A utilização de um editor BPM para modelar o comportamento e carregar o documento BPEL (*Business Process Execution Language*) gerado no AAP. Contudo, este caminho limitaria a configuração dos comportamentos dos AAPs à usuários com conhecimentos de BPM/BPEL ou

obrigaria os usuários a terem que conhecer tal tecnologia.

- A criação de uma interface mais simples no próprio sistema para configuração do AAP, que apresente ao usuário uma forma gráfica simplificada de fluxogramas. Porém, isso levaria a um outro ponto que é a necessidade de alocação de tempo para a criação de tal interface ou mesmo para a pesquisa de interfaces que poderiam ser adaptadas para a situação.
- A utilização de uma estrutura algorítmica simples a ser traduzida para BPEL. Neste caso, deve-se criar uma pseudo-linguagem ou utilizar-se de uma biblioteca que permita a utilização de *scripts*. Ainda, esta opção possui o mesmo problema da alocação de tempo que anterior.

O filtro de informações ainda precisa ser definido, modelando este módulo do sistema internamente na forma de trabalhar com as informações de preferências e objetivos do usuário e relacioná-las às informações que são recebidas pelo AAP.

4. ASPECTOS DA IMPLEMENTAÇÃO

Para criar uma primeira instância da Arquitetura, e como forma de fornecer um embasamento prático para a continuidade do trabalho, foi desenvolvido um primeiro protótipo parcial, composto de um comportamento simples para o gerenciamento de uma conta de e-mails de um usuário do AAP [6]. Contudo, como seu comportamento foi implementado em um programa Java, foi necessário o aperfeiçoamento deste para um segundo protótipo, utilizando agora a orquestração dos comportamentos via BPEL, ajustando o protótipo aos novos requisitos da arquitetura agora atualizada.

Os comportamentos do AAP se caracterizam como algoritmos customizados, pré-prontos ou criados pelo usuário do AAP. No protótipo esses algoritmos são desenvolvidos em BPEL devido a sua forma natural de invocação de serviços Web, vistos pelo AAP como comportamentos provindos de repositórios para serem orquestrados em um comportamento do AAP. Ainda, mesmo que um comportamento seja apenas a utilização de um único serviço Web, um documento BPEL deve ser criado para representar tal comportamento no AAP.

Cada um dos comportamentos é visto na forma de serviços disponíveis em Repositórios de Serviços Web e a composição de serviços para a criação de um comportamento para o AAP é também visto como um

serviço (Figura 3).

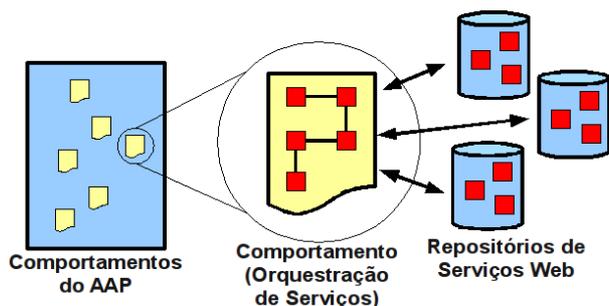


Figura 3: Comportamentos do Agente Assistente Pessoal.

Nesta segunda etapa do protótipo, os elementos implementados foram: o “Núcleo de Execução” no AAP (Java); a “Aplicação Web” no servidor de AAPs para criar e configurar o AAP (PHP e HTML); o comportamento do AAP para o “Gerenciamento de E-mails” (BPEL); um comportamento no repositório de comportamentos que acessa uma conta de um servidor de e-mails (serviço Web em Java); um comportamento no repositório de comportamentos que monitora uma conta de um serviço de *Instant Messages* (IM) (serviço Web).

Estes serviços são selecionados e organizados na forma de um algoritmo (em um documento BPEL). Esta etapa é classificada como a orquestração dos serviços para a criação de um comportamento específico. Este comportamento pode ser fornecido pela própria empresa, por terceiros, ou desenvolvido pelo próprio usuário do AAP e é visto pelo AAP como um serviço que irá compor a gama de comportamentos, ou macro-comportamento, do AAP.

Os elementos utilizados para a execução do protótipo, que não houve a necessidade de reimplementação, foram: um navegador Web (Firefox) para acesso ao Webmail do usuário e à “Aplicação Web”; um servidor de e-mails (Dovecot) com uma aplicação para acesso via Webmail (uebimiau); um servidor Web para as aplicações Web e Webmail (Apache); um cliente de IM (Pidgin); um servidor de IM (Gtalk). Como motor de execução dos comportamentos em BPEL foi utilizado o servidor Apache Tomcat com ODE (*Orchestration Director Engine*). Em tempo, para a implementação, execução e testes do protótipo foram utilizados computadores com sistema operacional GNU/Linux Ubuntu na versão 8.04.

O comportamento do gerenciamento de e-mails funciona em um laço infinito até que o Núcleo de Execução envie um comando para o comportamento

parar (Figura 4).

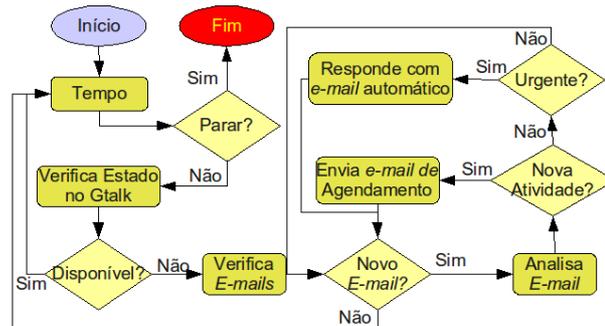


Figura 4: Fluxograma do comportamento de gerenciamento de e-mails.

O algoritmo inicia e se mantém um tempo em espera, de forma a não sobrecarregar o processamento do comportamento. Após isso, é verificado se o comando “parar” foi enviado e, caso tenha recebido este comando, então o algoritmo termina, caso contrário é verificado o estado do usuário no Gtalk. Se o usuário estiver com seu estado disponível, então o comportamento volta ao começo, caso contrário é invocado o processo para verificar e-mails. Caso não existam novos e-mails o algoritmo volta ao início, senão este executa uma análise do novo e-mail. Se este novo e-mail for uma nova atividade enviada para o usuário, então essa atividade é estruturada e enviada para o usuário na forma de um e-mail especial. Esta funcionalidade deve ser mais tarde substituída por um comando para adicionar nova atividade na agenda do usuário.

Ainda, no algoritmo, caso não seja uma nova atividade, então é verificado se o e-mail é urgente, caso seja urgente, é enviado um e-mail resposta para o remetente informando do motivo de não poder responder no momento e informando que logo que possível a mensagem será lida e respondida. Esta funcionalidade também deve ser substituída mais tarde por outra que envia mensagens via SMS (Short Message Service) para o celular do usuário com o assunto do e-mail.

No momento atual, o trabalho encontra-se na etapa de desenvolvimento de um novo protótipo do sistema, agora com todos os módulos da arquitetura implementados e da qual deverão ser aplicados testes para a validação da proposta.

Para tal, deverão ser preparados questionamentos para serem aplicados a usuários selecionados para utilizar AAPs, fazendo uma análise comparativa de como a utilização de um AAP afetou suas atividades rotineiras. Também deverá ser utilizada uma ferramenta/metodologia (ainda a ser definida) para a análise da utilização do tempo dos indivíduos participantes do processo de avaliação.

Assim, indicadores de desempenhos qualitativos e quantitativos serão empregados na avaliação do trabalho. Como qualitativos, por exemplo, deverão ser utilizados indicadores que, de alguma forma, meçam o quanto o trabalho do usuário foi mais facilitado e como quantitativo, que meçam, por exemplo, o quanto do trabalho foi diminuído ou tornado mais rápido.

5. CONSIDERAÇÕES FINAIS

Neste artigo foi apresentada uma proposta de uma Arquitetura de Referência para Assistentes Pessoais baseada em Agentes e Serviços de Software. Estes assistentes pessoais visam fornecer um suporte às pessoas 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 proposta busca fornecer um caminho de inter-operabilidade entre assistentes pessoais via software de computador, mantendo a compatibilidade entre as instâncias que seguem esta arquitetura e tendo os assistentes conectados aos sistemas empresariais.

Com base na arquitetura apresentada em [6], houve refinamentos efetuados na mesma tais como: (i) o acréscimo de um módulo servidor para o gerenciamento de um conjunto de agentes, cada qual com suas próprias informações, usuários e comportamentos; (ii) uma seção para gerenciamento de especificações de QoS em relação aos serviços utilizados pelos AAPs; (iii) um indexador de serviços Web para facilitar a procura de comportamentos para serem acrescentados aos AAPs pelos usuários; e (iv) a utilização de *templates* baseados em UBL para a padronização dos comportamentos relacionados aos processos de negócios da empresa.

Em tempo, tendo a nova arquitetura formulada, o protótipo foi aperfeiçoado, gerando uma segunda versão. Além disso, foram executadas pesquisas relacionadas aos aspectos para a implementação de uma próxima versão do protótipo para a validação da proposta, assim como a forma que esta será validada, com análise da execução do protótipo em um ambiente controlado e em um ambiente real, por meio de *logs* dos usuários, análise de tempo para execução de tarefas (quantitativamente) e questionários aos usuários com opiniões sobre as mudanças ocorridas nos processos de trabalho após a utilização de um AAP (qualitativamente).

Os próximos passos do trabalho incluem ainda um novo refinamento, a apresentação detalhada de cada elemento que compõe a arquitetura e a implementação de uma instância de assistentes pessoais baseado na

arquitetura apresentada. Ainda, deverão ser feitos testes quantitativos e qualitativos em ambiente controlado e ambiente real para a validação do modelo e do protótipo a ser implementado.

AGRADECIMENTOS

Este trabalho é parcialmente financiado pela CAPES – Coordenação de Pessoal de Nível Superior (<http://www.capes.gov.br>).

REFERÊNCIAS

- [1] Loss, L. *Aprendizado de redes colaborativas de organizações: um arcabouço para dar suporte à criação e uso de conhecimento*. 2007. Tese (Doutorado) - PPGEEL-UFSC, Florianópolis, 2007.
- [2] Stojanovic, Z.; Dahanayake, A. *Service-oriented software system engineering: challenges and practices*. Idea Group Inc. Hershey, PA. 2005.
- [3] Lackenby, C.; Seddighi, H. *A dynamic model of virtual organisations: formation and development*. In: Camarinha-Matos, pg. 37–44. 2002.
- [4] Maes, P. *Agents that reduce work and information overload*. Communications of the ACM, nº.7, vol37, NY, USA, 1994.
- [5] Hoyle, M.A.; Lueg, C. *Open Sesame!: A Look at Personal Assistants*. Proceedings of the International Conference on the Practical Application of Intelligent Agents (PAAM97), London, 1997, 51 60, 1997.
- [6] Zambiasi, S.P.; Rabelo, R.J. *Uma Proposta de Arcabouço para Agentes Assistentes Pessoais*. 7th I2TS. Foz do Iguaçu, Brasil, 2008.
- [7] Bocionek, S. *Software secretaries: learning and negotiating personal assistants for the daily office work* In *Systems, Man, and Cybernetics*. 'Humans, Information and Technology'. IEEE. 12 vol.1. 2-5 Oct. 1994.
- [8] Vieira, W. *Agentes Móveis Adaptáveis para Operação Remota*. PhD Thesis, Universidade Nova de Lisboa, Lisboa, 2000.
- [9] Markoff, J. *A Software Secretary That Takes Charge*. New York Times, Dec 13, 2008 <http://www.nytimes.com/2008/12/14/business/14stream.html?_r=1&scp=7&sq=personal

- %20assistant&st=cse> access in Mar 17, 2009.
- [10] Michael, T., Caruana, R., Freitag, D., McDermott, J., Zabowski, D. *Experience With a Learning Personal Assistant*. Communications of the ACM, July, 1994.
- [11] Huhns, M.N.; Singh, M.P. *Personal assistants*. IEEE Internet Computing, Vol. 2, Issue 5, pp. 90-92. Sep./Oct. 1998.
- [12] Russel, S., Norvig, P., *Inteligência Artificial*, 2ªEd, Trad. 2ª ed. RJ: Elsevier, 2004.
- [13] Wong, J.S.K., Mikler, A.R. *Intelligent mobile agents in large distributed autonomous cooperative systems*. The Journal of Systems and Software 47, pg.75-87, 1999.
- [14] Weiss, G. *Multiagent systems: a modern approach to distributed artificial intelligence*. MIT Press, 1999.
- [15] Sensoy, M., Yolum, P. *Evolving service semantics cooperatively: a consumer-driven approach*. Springer Science+Business Media, LLC, Nov 9, 2008.
- [16] Schiaffino, S., Amandi A.; *Polite Personal Agents*. IEEE Intelligent Systems, p12-18. Jan-Fev 2006.
- [17] Bush, J., Irvine, J. and Dunlop, J. *Personal Assistant agent and content manager for ubiquitous services. wireless communication systems*, 2006. ISWCS'06. 3rd International Symposium on , 169-173. 2006.
- [18] Franco, R.D., Neyem, A., Ochoa, S. et al. *Supporting mobile virtual team's coordination with soa-based active entities*. Establishing The Foundation of Collaborative Networks, IFIP TC 5 Working Group 5.5 Eighth IFIP Working Conference on Virtual Enterprises, September 10-12, 2007, Guimarães, Portugal. IFIP 243 Springer 2007.
- [19] Coulouris, G.F.; Dollimore, J.; Kindberg, T. *Distributed systems: concepts and design*. Addison Wesley Longman, 2005.
- [20] Kshemkalyani A.D.; Singhal, M. *Distributed computing principles, algorithms, and systems*. Cambridge University Press. 2008.
- [21] Gesser, Carlos E. *Uma abordagem para a integração dinâmica de serviços web em portais*. Eng. Elétrica - UFSC. Dissertação. 2006.
- [22] MacKenzie, C.; Laskey, K.; McCabe, F. at all. *Reference Model for Service Oriented Architecture 1.0*. OASIS Standard, 12 October 2006. <<http://docs.oasis-open.org/soa-rm/v1.0/soa-rm.pdf>> acessado em Fev/2009.
- [23] Markoff, John. *A software secretary that takes charge*. NY Times, Dec 13, 2008 <http://www.nytimes.com/2008/12/14/business/14stream.html?_r=1&scp=7&sq=personal%20assistant&st=cse> access in Mar17, 2009.
- [24] Vance, A. *Microsoft Mapping course to a jetsons-style future*, NY Times, Mar 1, 2009.<<http://www.nytimes.com/2009/03/02/technology/business-computing/02compute.html?scp=1&sq=jetsons&st=cse>>. Access Mar 8, 2009.
- [25] Mundie, C. *Microsoft to make computers "humanistic" ...meet Laura...*, Interview on Beet.TV, Oct, 2008 <http://www.youtube.com/watch?v=6GOrNU6e_og> Access Mar 17, 2009.
- [26] Hofmeister, C., Nord, R., Soni, D. *Applied software architecture*. Addison Wesley, 2000.
- [27] Tramontin, R.J.; Rabelo, R.J. *A knowledge search framework for collaborative networks*. International Federation for Information Processing-Publications-IFIP, vol.243. 2007.