

# Uma Arquitetura de Referência para Softwares Assistentes Pessoais Baseada em Agentes e SOA

Saulo Popov Zambiasi<sup>1</sup>, Ricardo J. Rabelo<sup>1</sup>

<sup>1</sup>Centro Tecnológico – Departamento de Automação e Sistemas  
Universidade Federal de Santa Catarina (UFSC) – Florianópolis – SC – Brasil

{popov,rabelo}@das.ufsc.br

**Abstract.** *Much efforts have been made to create computer systems to provide assistance for people in their daily activities, at home and in business. However, they usually follow different approaches. However, they usually follow different approaches that often do not respect the required compatibility and interoperability with each other besides not being connected with legacy systems and new software services. In this way, this paper presents an open reference architecture for personal softwares assistants, allowing the derivation, personalization and deployment of interoperable instances of assistants regarding users' and companies' processes requirements.*

**Resumo.** *Vários esforços vêm sendo feitos para a criação de sistemas computacionais com o objetivo de fornecer assistência às pessoas em suas atividades diárias, em casa e na empresa. Porém, esses geralmente seguem por abordagens distintas, que normalmente não mantêm a compatibilidade e a interoperabilidade uns com os outros, além de não se preocuparem em estarem conectados com os sistemas empresariais legados e serviços novos de software. Neste sentido, este trabalho apresenta uma arquitetura de referência aberta para softwares assistentes pessoais, permitindo que instâncias interoperáveis possam ser derivadas, personalizadas e implantadas consoante as características das pessoas e processos das organizações.*

## 1. Introdução

O conceito de softwares que fornecem assistência às pessoas não é novo e tem influenciado a imaginação de muitos escritores de ficção científica e cientistas da computação [Markoff 2008]. Essa ideia se firma em softwares baseados em conhecimento e que funcionam através da Internet na forma de um tipo de secretário, auxiliando em tarefas como pagamentos de contas, organização de viagens, localização de informações em bibliotecas virtuais na internet, etc. [Michaell 1994].

Durante os últimos anos, diversas pesquisas têm sido efetuadas nesse sentido. Contudo, uma observação feita após uma extensa revisão bibliográfica na área de softwares assistentes é a de que as propostas existentes atacam pontos isolados e que não há nenhuma que seja aberta para permitir a sua integração à ambientes empresariais, i.e. a processos de negócios da empresa [Zambiasi 2010]. Ainda, algumas perspectivas devem ser destacadas. A primeira é a de que assistentes são implementados

como sistemas independentes dos demais da empresa, sem interação ou interoperação alguma, fazendo com que o usuário tenha que navegar por ambientes diferentes, além de que o assistente usualmente atua como um sistema puramente reativo à interação do usuário, e não de forma pró-ativa, executando as tarefas pelo/para ele.

A segunda perspectiva, associada a Tecnologias de Informação e Comunicação, é a interoperação. Os assistentes pessoais costumam ser softwares fechados. Isso porque foram concebidos para realizarem suas atividades sem uma preocupação de projeto de transacionarem com outras aplicações. Essa perspectiva é essencial na ótica de se integrar o assistente ao ambiente geral de execução de processos da empresa, uma vez que normalmente os seus sistemas são distribuídos e heterogêneos.

A terceira perspectiva é quanto a flexibilidade das ações de um assistente. Isso envolve aspectos como adaptabilidade e escalabilidade. No escopo de transações empresariais, cada negócio tem inúmeras particularidades, tanto de contexto como de ambientes computacionais envolvidos. Portanto, há necessidade de maior agilidade geral. Aquelas mesmas transações comerciais não são estáticas. A empresa, seus sistemas e processos de negócios alteram-se ao longo do tempo. Dessa forma, há necessidade de se lidar com a escalabilidade do assistente, que deve estar preparado para flexibilizar suas ações consoante a mudanças e/ou introdução de novos processos de negócios. O aspecto final é que 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 por *software-houses* e por dispositivos computacionais distribuídos. Nesse sentido, o assistente deve poder acessar tais serviços consoante aos 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 para o assistente possa ser montado.

A quarta perspectiva é a de que nenhum dos projetos estudados sugere especificações ou padronizações para interoperabilidade com outras aplicações de forma aberta. E por fim, uma quinta perspectiva, é a de que nenhum dos trabalhos avaliados apresenta um modelo ou arquitetura de referência para o desenvolvimento de softwares assistentes pessoais com padrões suficientes para manter a interoperabilidade.

Dessa forma, este trabalho propõe uma arquitetura aberta de referência para softwares assistentes pessoais, que possa gerar implementações interoperáveis e customizáveis para se adequarem aos processos de negócios da empresa e que possam auxiliar os usuários em suas tarefas diárias.

## **2. Softwares Assistentes Pessoais**

Os conceitos de softwares para assistência pessoal confundem-se em diversas premissas com a tecnologia de agentes de software. Com isso, o presente trabalho se apropria desses conceitos de agentes para tratar dos softwares assistentes pessoais. Um assistente, tal como um agente, pode perceber seu ambiente através de recursos (como a agenda do usuário, preferências, preferências de outros usuários, reputações) e responde com ações de assistência ao usuário. Além disso, um agente de software pode substituir

parcial ou totalmente uma pessoa em determinadas tarefas e em várias situações diferentes, assumindo o papel de um assistente. Estes podem trocar informações automaticamente por meio de negociações eletrônicas [Weiss 1999].

Para Bocionek (1994), os assistentes devem ser autônomos para poder oferecer auxílio no gerenciamento das atividades dos seus usuários, inclusive atividades conflitantes. Devem saber negociar, aprender, ter portabilidade e mobilidade, principalmente no cenário da computação móvel.

Segundo Hunhs (1998), um software assistente pessoal não pode ser visto apenas como um programa de computador personalizado. Isto porque um assistente deve ser baseado em rede, ser interativo, adaptativo, de propósito geral, de execução autônoma e deve poder interagir com outros assistentes. Com a apropriação dessas especificidades, Hunhs (1998) também faz um paralelo dos softwares assistentes pessoais com a tecnologia de agentes, apresentando os assistentes pessoais como softwares que podem representar as pessoas na Internet, auxiliando os usuários nas atividades do dia a dia, especialmente nas que envolvem recuperação de informação, negociação ou coordenação. Ainda, Hunhs (2002) sugere a utilização de serviços web, servindo como uma fonte de investigação para assistentes e agentes. Essa tecnologia, além de possuir características facilmente aplicáveis a agentes, possui flexibilidade e capacidade de trabalhar em conjunto em um ambiente interoperável de empresas.

Na parte de implementação, tanto iniciativas privadas como iniciativas de sistemas *opensources* têm tido influência e destaque no caminho da assistência via software de computador. O Assistente Pessoal da Shelltoys, por exemplo, é um sistema proprietário e fechado para auxílio em tarefas do usuário. Contudo, se limita a apenas gerenciar as tarefas do usuário via uma agenda de compromissos. Este assistente possui uma lista de tarefas e gera lembretes de compromissos quanto a aniversários, atividades, reuniões e outras pequenas funcionalidades [SHELLTOYS 2011].

O projeto Sandy [Mann 2011] foi desenvolvido para auxiliar o usuário a lembrar de compromissos, listas, pendências, contatos, projetos e trabalha com troca de mensagens via SMS, Twitter e até mensagens de e-mail. Para o usuário utilizar o Sandy, basta ele enviar mensagens do tipo: "Lembrar de pegar meu carro em 15 minutos", "O telefone do meu pai é 9999-9999", "lembrar de comprar mantimentos ovos", etc. Contudo, Sandy é um projeto proprietário, fechado e foi descontinuado em 2008.

O projeto Narval (sob licença GNU LGPL) pode ser executado no próprio computador pessoal ou em um servidor remoto, e se comunica por meios normais, como e-mail, web, telnet, celular, GUI específica, etc. O sistema executa sequências de ações descritas pelo usuário, para uma ampla gama de tarefas, como preparar o jornal da manhã, ajudar a navegar na web por meio de filtro de anúncios e lixo eletrônico, cuidar de tarefas repetitivas como responder e-mail, negociar data e hora de reuniões, etc. O software utiliza Inteligência Artificial, Agentes, Redes de Petri, sistemas baseados em regras, programação de contratos, planejamento, aprendizagem automática. Podem ser desenvolvidos *plugins* em Python [Chauvat 2000], [Thenault, 2011]. Contudo, o projeto não tem tido atualizações recentes, indicando uma provável descontinuação.

O Siri (2011) é um projeto bastante recente. Ele se utiliza de informações de preferências pessoais dos indivíduos e de um histórico de interação para ajudá-lo a

resolver tarefas específicas. É baseado na web e links de resultados de buscas, sua interação na forma de conversa com o usuário. O usuário diz para seu assistente o que quer fazer e este faz pesquisa em fontes de informações para poder auxiliar o usuário. O software evolui com a experiência de interação. Ele se utiliza também como base o contexto pessoal (lugar, horário, histórico) do usuário. O Siri foi adquirido pela Apple e é um sistema fechado e proprietário.

Um projeto bastante relevante é o projeto PAL (2011), financiado pela agência americana DARPA. Este projeto tem por objetivo auxiliar os usuários nos trabalhos com sistemas computacionais e automatizar tarefas rotineiras de forma a liberar as pessoas para as tarefas mais importantes [Markoff 2008]. A intenção é criar um sistema cognitivo que pode raciocinar, aprender e lidar com situações imprevistas como forma de fornecer assistência em situações militares. Além disso, o sistema deve poder se adaptar a mudanças em seu ambiente, aos objetivos dos seus usuários e as suas tarefas sem a necessidade de alterar sua programação ou de intervenção técnica. Este projeto está sendo executado por um conjunto de cientistas da computação e pesquisadores em inteligência artificial, percepção, aprendizado de máquina, processamento de linguagem natural, representação de conhecimento, diálogo multimodal, ciberconsciência, interação homem-máquina e planejamento flexível [PAL 2011].

Isso posto, pode-se resumir as características de assistentes pessoais (incluindo a consideração dos requisitos da proposta deste trabalho) em [Bocionek 1994], [Hunhs 1998] e [Zambiasi 2010]:

- Atuar com certa autonomia em suas tarefas de forma que estes possam verificar a situação dos seus usuários e que possam responder apropriadamente a cada situação que aparece;
- Ser flexível para atuar diante de novas situações e cenários de negócios;
- Ser adaptável ao usuário, conforme informações, preferências, necessidades, situação atual, e evolução das informações;
- Interagir com o usuário;
- Ser baseado em rede, para buscar informações na Internet e interagir com outros sistemas;
- Ser de propósito geral, ou seja, não específico à apenas uma atividade ou um grupo de usuários, adaptando-se aos propósitos do seu usuário;
- Se adaptar ao contexto;
- Ser integrado e interoperável aos processos de negócios da empresa.

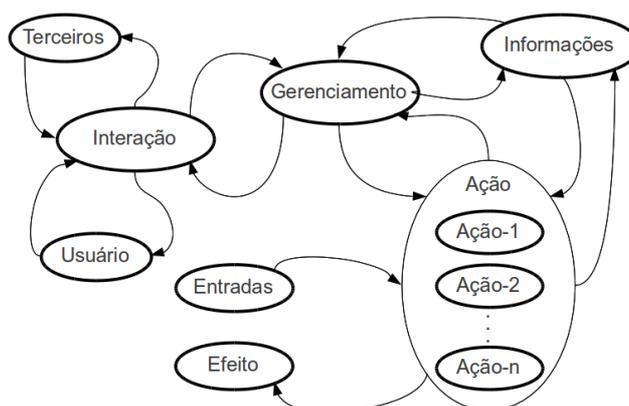
### **3. Proposta**

A concepção de uma arquitetura de referência é vista como uma forma de apresentar um padrão genérico para um projeto e deve abordar os requisitos para o desenvolvimento de soluções, guiado pelo modelo de referência e por um estilo arquitetural, de forma a atender as necessidades do projeto [MacKenzie 2009]. Um modelo de referência é uma divisão de funcionalidades, juntamente com o fluxo de dados entre as partes, e possuem características de domínios maduros, decorrente da experiência sobre este domínio. Ele se caracteriza como um padrão de decomposição do problema. Já o estilo arquitetural descreve os tipos de elementos e suas relações, juntamente com um conjunto de restrições sobre como eles podem ser utilizados, além de padrões de interação entre os

elementos. Tais restrições, sobre a arquitetura e sobre o sistema em si, são vistas na forma de uma imagem da utilização do sistema como um todo [Bass 2003].

### 3.1. Modelo de Referência

Na Figura 1 é apresentado o modelo de referência para assistentes pessoais proposto neste trabalho.



**Figura 1. Modelo de Referência para Softwares Assistentes Pessoais.**

Entre os elementos principais estão a Ação, que se refere aos comportamentos do assistente, a Interação para com o usuário ou outros assistentes ou softwares e o Gerenciamento, para gerenciar a execução do assistente, organizar as informações e o fluxo dessas entre os demais elementos.

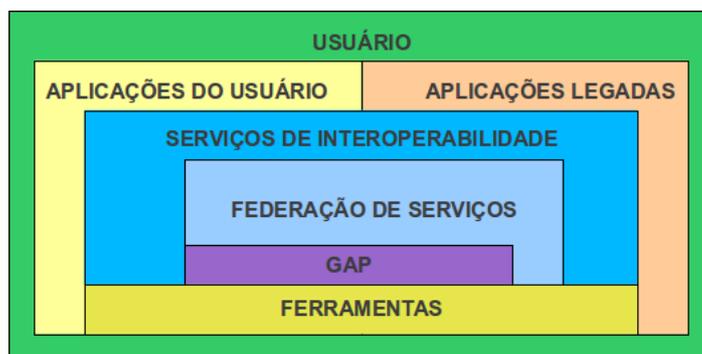
A Ação é a forma como o assistente age, que é a composição do conjunto de  $n$  ações que podem ser utilizadas pelo assistente pessoal. As Informações são os elementos úteis para a execução de uma determinada ação ou de um conjunto de ações. As Entradas, tal qual um agente sente seu ambiente por meio de sensores, refere-se ao conjunto de informações que chegam de elementos externos ao assistente e que servem para iniciar uma determinada ação do assistente. Essa ação reage com um Efeito, que pode ser tanto interno do próprio assistente, como externo, com parceiros, outros assistentes, outros softwares, ou mesmo pela interação com o usuário. A Interação é a forma como se dão as negociações entre o usuário e módulo de Gerenciamento do assistente, ou entre o módulo de Gerenciamento e outros elementos.

O Gerenciamento é responsável pela organização das informações do usuário, pela execução das ações e pelo fluxo dessas informações entre os elementos do modelo. Um conjunto de informações acerca do usuário do assistente e das ações são necessárias para que o assistente gerencie sua execução. Elas devem evoluir com o tempo em concordância com a evolução do usuário, tarefas, preferências, etc. Por meio delas, o Gerenciamento deve decidir quando e como iniciar uma ação, ou seja, tornar uma ação operacional quando certas condições forem satisfeitas.

### 3.2. Arquitetura de Referência

A existência de uma arquitetura de referência para assistentes pessoais é relevante, mas não suficiente para fazer frente àqueles requisitos anteriormente expostos. É importante que o assistente possa ter a característica de comportamentos “plugáveis”, se manter em

um padrão de comunicação e ainda trabalhar com a questão de serviços de software, propiciando assim as bases para a flexibilidade e escalabilidade desejadas. Assim, os comportamentos do assistente pessoal são visto neste trabalho pela invocação de serviços web, distribuídos na Internet, na forma de uma federação de serviços web [Zambiasi 2010]. O usuário pode então escolher os serviços web que melhor se enquadram nas necessidades das tarefas que se quer auxílio. Nesse sentido, os comportamentos do assistente devem poder ser criados ou excluídos de forma flexível. Ainda, isso dá ao usuário do assistente a possibilidade de escolha entre serviços com a mesma funcionalidade, fornecido por diferentes empresas, organizações ou desenvolvedores. Quando um serviço, utilizado por um comportamento, não mais satisfizer as necessidades do usuário, este pode então trocá-lo por outro, que seja mais condizente com sua nova situação/contexto. Ainda, tais serviços devem poder ser utilizados sob variados modelos de negócios, gratuitamente, alugados ou vendidos por terceiros. Com isso o presente trabalho sugere, na Figura 2, uma Arquitetura de Referência para Assistentes Pessoais com estilo arquitetural na Arquitetura Orientada a Serviços (*SOA – Service Oriented Architecture*).



**Figura 2. Arquitetura de Referência para Softwares Assistentes pessoais.**

- **Aplicações Legadas:** Softwares de terceiros, ou desenvolvidos pela empresa, e que o usuário necessita que o assistente gerencie.
- **Aplicações do Usuário:** Aplicações que podem servir para interação com o assistente pessoal. Pode ser, conforme no sistema Sandy (2011), serviços de e-mails, mensagens de Twitter, sistemas de *Instant Messaging*, ou SMS.
- **Ferramentas:** Aplicações de configuração de assistentes pessoais e comportamentos ou ambientes de desenvolvimento.

A Figura 2 mostra uma visão de como o usuário está em contato com o assistente pessoal. O usuário pode estar em vários níveis de conhecimento:

- **Usuário Simples:** cadastra um assistente pessoal, configura comportamentos e as informações dos comportamentos do seu assistente.
- **Usuário Avançado:** pode compor comportamentos mais complexos no assistente, via uma interface de algoritmos ou fluxogramas.
- **Desenvolvedor de serviços web:** Desenvolve serviços web que podem ser utilizados para compor comportamentos.
- **Desenvolvedor de serviços de interoperabilidade:** Desenvolve interfaces de serviços web baseados em SOA da OASIS para sistemas fora desse padrão.
- **Desenvolvedor de Assistentes Pessoais:** Desenvolvedor avançado de

interfaces para configuração de assistentes pessoais, e núcleos de execução.

A **Federação de Serviços** é conjunto de serviços web distribuídos na Internet e que podem ser utilizados nos comportamentos do assistente pessoal isoladamente ou podem servir para compor outros serviços web (orquestração e composição de serviços).

Os **Serviços de Interoperabilidade**, por sua vez, são criados para servirem de interface à outros sistemas que não estão no padrão SOA.

O **Gerenciador de Assistentes Pessoais** é responsável pela coordenação da execução dos comportamentos e suas informações. Os comportamentos podem ter três níveis básicos, apresentados aqui de forma simplificada:

- **Comportamento Simples:** Composto de informações para a chamada do serviço web. Este é executado toda vez que o comportamento é chamado, enviando os parâmetros de entrada e retornando um resultado.
- **Comportamento Condicional:** Composto de informações e especificações condicionais para a execução do comportamento.
- **Comportamento Complexo:** Necessita de um conhecimento mais avançado do usuário. O usuário pode modelar o comportamento na forma de um fluxograma, ou algoritmo, com laços, condicionais e outros elementos.

Os comportamentos são modelados em uma interface de configuração do assistente pessoal, e a forma como isso é feito depende de cada desenvolvedor.

## 4. Verificação

Uma das formas de verificação da Arquitetura de Referência é a criação de uma instância implementada baseada na proposta. Para tal, foi gerado um protótipo completo em cima de um estudo de caso específico. Neste estudo de caso, é considerado o cenário em que um funcionário de uma empresa tem a função de efetuar o gerenciamento do estoque de produtos. Quando o estoque de um produto passa de um limite mínimo, fica a cargo desse funcionário efetuar nova compra. Conforme produtos são retirados do estoque, sua quantidade diminui. No final do dia, o funcionário faz um relatório das atividades realizadas durante todo o dia, desde o início do processo de uma compra, até as modificações em uma ordem de compra e fechamento da ordem.

A possibilidade da utilização de um assistente pessoal é agora inserida no contexto. Neste caso, o assistente deve ser utilizado para auxiliar o funcionário a gerenciar o sistema de controle de estoque e para a criação do relatório das atividades diárias. Ainda, é considerado que o usuário deseja se comunicar com seu assistente pessoal por meio de aplicações que não precisam ser instaladas em seu computador e que podem ser acessadas de qualquer lugar, inclusive do celular. Neste caso, o usuário pretende se comunicar com seu assistente pessoal como se fosse uma outra pessoa por meio de mensagens via Twitter, Gtalk, e-mail, ou mensagem de texto no celular.

### 4.1. Protótipo

Sob o cenário apresentado, e a Arquitetura de Referência proposta, pode-se sugerir a modelagem de alguns elementos de implementação que podem compor o assistente

desse estudo de caso (Figura 3), e que são descritos em sequência. Ou seja, a arquitetura de referência ser instanciada (“populada”) com diferentes serviços (i.e. funcionalidades), modelos e ferramentas de implementação.



Figura 3. Estudo de caso para a implementação baseada na proposta.

As implementações feitas são aqui apresentadas conforme sua classificação na Arquitetura de Referência proposta. Todos os sistemas *daemons* foram desenvolvidos em linguagem de programação Java, os serviços web em linguagem PHP (no estilo SOA da OASIS) e as interfaces em PHP e HTML via web.

- **Aplicações Legadas**
  - **Estoque:** Sistema de controle de estoque. Possui informações dos produtos, fornecedores, quantidades máxima, atual e mínima dos produtos que são utilizadas para iniciar um processo de compra.
  - **Fornecedor:** Controle de produtos no fornecedor. Informa os produtos, quantidades, e preço. Vários fornecedores podem ser cadastrados.
- **Aplicações do Usuário**
  - **Twitter:** Serviço de publicação de mensagens de até 140 caracteres. Acesso via web ou aplicativos especiais para cada sistema operacional.
  - **Celular:** Via Twitter, o usuário pode habilitar receber/enviar mensagens via mensagem de texto no celular.
  - **Cliente IM:** Para troca de mensagens com o assistente pessoal. O usuário vê seu assistente pessoal como uma pessoa de seus contatos conectada ao serviço. É necessário uma conta no Gmail para o assistente pessoal.
  - **Webmail:** Para troca de mensagens de e-mail com o assistente.
  - **Blog:** O assistente envia postagens à um blog para publicar relatórios.
- **Serviços de Interoperabilidade:** Estes são serviços que fazem interface com outros sistemas que não possuem interfaces de serviços web SOA da OASIS.
  - **wsEstoque:** Acessa informações do sistema de controle estoque.
  - **wsOrdem:** Gerenciamento das ordens de compra.
  - **wsVenda:** Acessa o sistema automático de vendas, um *daemon*

executando do lado do sistema de fornecedores.

- **wsTwit:** Acessa uma conta no serviço de microblog Twitter.
- **wsXMPP:** Acessa um *daemon* de conexão ao serviço de IM Gtalk.
- **wsMail:** Acessa os e-mails de um usuário em um servidor.
- **Federação de Serviços**
  - **wsCC:** Simula transações com cartão de crédito.
  - **wsAudit:** Utilizado para armazenar informações de auditoria.
  - **wsReport:** Geração automática de relatórios. Recebe mensagens referente as atividades e quando requisitado retorna o relatório compilado.
  - **wsMBox:** Gerencia mensagens provindas do ISAP.
  - **wsChatter:** Gera respostas para o usuário e quem mais se comunicar com o assistente pessoal tal como um *chatbot*.
  - **wsISAP:** Serviço de interface com o usuário. Se utiliza de serviços como o wsChatter, wsXMPP e wsMail para trocar mensagens com o usuário. Há um sistema *daemon* que faz o gerenciamento das mensagens. Quando um usuário se comunica com a Interface Social para Assistentes Pessoais (ISAP) a mensagem é avaliada e enviada ao GAP. Ela também decide a melhor forma de enviar uma mensagem provinda do GAP ao usuário.
  - **LIBS:** Conjunto de bibliotecas de funções que são utilizadas para criar os comportamentos no GAP (*string, math, datetime, kqml, myMessageTable*).
- **Ferramentas:** Programas de interfaces web para configuração de de alguns sistemas.
  - **iwISAP:** Configuração da ISAP do assistente pessoal.
  - **iwChatbot:** Configuração das conversas do assistente pessoal.
  - **iwRelatorios:** Configuração do gerador automático de relatórios.
  - **iwGAP:** Ambiente de configuração do assistente pessoal e seus comportamentos. Os comportamentos são aqui configurados na forma de um algoritmo. O GAP (Gerenciador de Assistentes Pessoais) é um *daemon* que gerencia a execução autônoma dos assistentes pessoais.

Havendo esse cenário de softwares e serviços web, foi criado um assistente pessoal e alguns comportamentos para efetuar o gerenciamento do sistema de controle de estoque, de responsabilidade do usuário. Em verdade, neste protótipo existem dois caminhos distintos para a criação do comportamento automático de compra:

1. **Usuário avançado:** pode criar comportamentos na forma de algoritmos na interface de configuração do Assistente Pessoal;
2. **Usuário nível básico:** Requisita a um desenvolvedor da empresa um serviço web que efetua o processo automático de compra. O usuário utiliza este serviço web no seu comportamento de compra automática;

Aqui, neste estudo de caso, é considerado que o usuário é do tipo avançado e configurou as informações e algoritmos que vai fazer a compra automática. Neste caso, os seguintes comportamentos foram criados:

- **AtualizaHora:** Invoca operações do serviço web *datetime* para atualizar data e horário para serem utilizadas por outros comportamentos.
- **ISAPalive:** Envia, a cada pouco, uma mensagem ao ISAP, informando que

ele deve manter a interface do assistente funcionando.

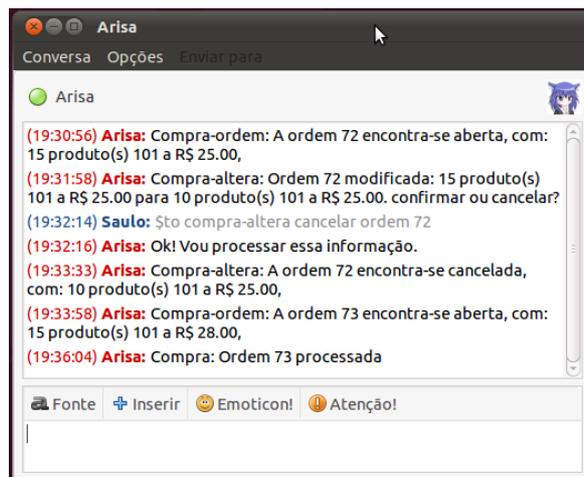
- **mbox:** Invoca operações do wsISAP para verificar e buscar mensagens do usuário para o GAP. As mensagens são identificadas e armazenadas no wsMBox, com informações de qual comportamento essa é destinada.
- **Report:** É verificado se é um horário específico. Caso afirmativo, são requisitados os relatórios das últimas atividades. Relatórios públicos são enviados ao Blog e privados são enviados ao usuário por e-mail.
- **Compra-ordem:** Se um produto está com estoque baixo, gera uma ordem de compra. Essa ordem é enviada ao fornecedor escolhido pelo sistema gerenciador de estoques. Cada nova ordem criada é enviada uma mensagem ao usuário e uma atividade é enviada ao wsRelatorio.
- **Compra:** Se há uma ordem aberta, verifica o estado dela no fornecedor. Se esta foi alterada, altera-a também no wsOrdem e envia uma mensagem ao usuário do assistente pessoal, requisitando confirmação. Se a ordem foi aceita pelo usuário, efetua o pagamento, finalizando a compra. Quando uma ordem é cancelada, o sistema seleciona outro fornecedor para o produto.
- **Compra-altera:** Aguarda a confirmação do usuário de uma ordem alterada. Se o usuário confirma, então a ordem volta a ficar aberta. Se for cancelada, então cancela a ordem no wsOrdem e no wsVenda.

O GAP executa os comportamentos por ciclos. Cada comportamento pode conter diversas atividades e em cada ciclo uma atividade de cada comportamento é executada. Essas atividades podem ser atribuição de valor à uma informação, invocação de uma operação em um serviço web, verificação de um condicional ou de um laço.

#### 4.2. Teste

No sistema de estoque, unidades do produto mouse (código 001), são vendidas, ficando suas informações da seguinte forma: quantidade 5, quantidade mínima 10 e quantidade máxima 20. Ou seja, quando a quantidade deste produto estiver a baixo de 10, são comprados os produtos necessários para chegar à quantidade máxima. Este produto é fornecido pelos fornecedores *John Doe Ltda* e *Sora Konpyuuta* ao valor de 25 reais e 28 reais a unidade, respectivamente. Considere também que o primeiro fornecedor possui apenas 10 produtos em estoque para venda, e que o segundo possui 50. O fornecedor selecionado inicialmente para a compra é o *John Doe Ltda*. As trocas de mensagens entre assistente pessoal e usuário para efetuar o gerenciamento das ordens de compra são vistas na Figura 4.

Quando o assistente pessoal verifica que o estoque está baixo, é iniciado o processo de compra. Como a quantidade do produto 001, do fornecedor *John Doe Ltda* não é o suficiente para o total da ordem de comprar, ou seja, 15 produtos, então a ordem é modificada e o assistente requisita confirmação da alteração. Considere o caso que o usuário necessita comprar todos os produtos para repor o estoque no máximo. Dessa forma, a ordem deve ser cancelada e uma nova criada, a outro fornecedor. Tendo corrido tudo corretamente com a segunda ordem criada, o pagamento é efetuado e a ordem é fechada, atualizando os estoques no fornecedor e no cliente. Por fim, um relatório completo é enviado por e-mail no horário definido pelo usuário.



**Figura 4. Troca de mensagens entre o assistente pessoal e o usuário.**

Quanto a essa execução, é importante se observar alguns pontos. Primeiramente, pode não ser interessante que o usuário fique recebendo informações do assistente pessoal a todo momento, ou mesmo, o usuário pode querer que esse cancelamento de uma ordem de compra, para a geração de uma nova para outro fornecedor, seja automático. Contudo, neste exemplo várias mensagens são trocadas com o usuário para fins de testes e verificação do funcionamento do comportamento. Outro ponto a se observar é que este teste foi realizado com o usuário conectado em seu usuário no Gtalk. Caso o usuário não estivesse conectado, as mensagens seriam trocadas via mensagem de texto do celular.

## **5. Considerações Finais**

Este artigo apresentou uma proposta para uma Arquitetura de Referência para Softwares Assistentes Pessoais. Essa proposta se apropriou da Arquitetura Orientada à Serviços para servir de estilo arquitetural e do conceito de Agentes.

Com base nos resultados de testes sob a implementação final, verificou-se que esta instância se comportou conforme o que foi proposto na arquitetura de referência e executou corretamente as ações associadas aos comportamentos do processo de negócio envolvido no estudo de caso. Todavia, ainda se constata que a área de assistentes pessoais tem ainda enormes desafios à serem enfrentados, incluindo ao que se refere a limitações das TIs atuais para certos propósitos (por exemplo, o problema da interoperabilidade semântica), os impactos organizacionais, a complexidade do problema de composição de comportamentos dentro da abordagem SOA, etc. Dessa forma, este trabalho veio no sentido de oferecer uma contribuição para a área de assistentes pessoais mais flexíveis, mais interoperáveis, e mais integrados ao mundo das empresas/processos de negócios.

## **6. Agradecimentos**

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

## Referências

- Boulic, R. and Renault, O. (1991) “3D Hierarchies for Animation”, In: New Trends in Animation and Visualization, Edited by Nadia Magnenat-Thalmann and Daniel Thalmann, John Wiley & Sons Ltd., England.
- Bass, Len; Clements, Paul and Kazman, Rick. “Software architecture in practice”. 2nd ed. Addison-Wesley, 2003.
- Bocionek, S. “Software secretaries: learning and negotiating personal assistants for the daily office work In Systems, Man, and Cybernetics”, In: Humans, Information and Technology. 1994 IEEE International Conference on, 12 vol.1. 2-5 Oct. 1994.
- Chauvat, N. “Narval, the intelligent personal assistant or how the french Linux Gazette is built”, In: Linux Gazette, issue 59. Nov, 2000. Disponível em <<http://linuxgazette.net/issue59/chauvat.html>>, acessado em 14/02/2011.
- SHELLTOYS. “Personal assistant - day planner and personal information manager”. Disponível em: <[http://www.shelltoys.com/personal\\_assistant/](http://www.shelltoys.com/personal_assistant/)>, acessado em 14/02/2011.
- Huhns, M.N.; Singh, M.P., “Personal assistants”, In: IEEE Internet Computing, Vol. 2, Issue 5, pp. 90-92. Sep./Oct. 1998.
- Huhns, M.N. “Agents as Web services”, In: Internet Computing, IEEE 6, 93-95. 2002.
- MacKenzie, C.; Laskey, K.; McCabe, F. at all. “Reference Model for Service Oriented Architecture 1.0”, In: OASIS Standard, 12 October 2006. <<http://docs.oasis-open.org/soa-rm/v1.0/soa-rm.pdf>> acessado em Fev/2009.
- Mann, H. “Free Personal Assistant - Meet Sandy”. Disponível em <<http://honestholly.com/free-personal-assistant-meet-sandy/>>, acessado 14/02/2011.
- Markoff, J. “A Software Secretary That Takes Charge”, In: New York Times, 13/12/2008. Disponível em: <[http://www.nytimes.com/2008/12/14/business/14stream.html?\\_r=1&scp=7&sq=personal%20assistant&st=cse](http://www.nytimes.com/2008/12/14/business/14stream.html?_r=1&scp=7&sq=personal%20assistant&st=cse)>. Acessado em: 17/03/2009.
- Michael, T.; et all. “Experience With a Learning Personal Assistant”, In: Communications of the ACM, July, 1994.
- PAL, PAL Project, <<http://pal.sri.com/>>, acessado em 14/02/2011.
- SIRI, disponível em <<http://siri.com/>>, acessado em 14/02/2011.
- Thenault, Sylvain. Narval-moved. Logilab Project. Disponível em: <<http://www.logilab.org/908/>>, Acessado em 14/02/2011.
- Weiss, G. “Multiagent systems: A Modern Approach to Distributed Artificial Intelligence”. MIT Press, 1999.
- Zambiasi, S.P.; Rabelo R.J.. “Uma arquitetura de referência para softwares assistentes pessoais baseada na arquitetura orientada à serviços”, In: I2TS'2010: 9th International Information and Telecommunication Technologies Symposium, Rio de Janeiro (RJ), Brasil, 2010.