

# VIRTUALIZAÇÃO DE COLABORADORES NA MANUFATURA: UM MODELO BASEADO EM *AGENT BOTS*

SAULO P. ZAMBIASI, RICARDO J. RABELO

*GSIGMA*, Departamento de Automação e Sistemas, Universidade Federal de Santa Catarina  
Caixa Postal 476, 88040-900 Florianópolis, SC, Brasil  
E-mails: popov@gsigma.ufsc.br, rabelo@das.ufsc.br

**Abstract**— Due to the market competitiveness, enterprises need continuously to adapt to new technologies. Their employees also need resources to fulfill their tasks and processes in time, as well as to be agile and efficient. This paper presents a model in order to partially replace employees during the execution of some process and tasks. It is done when the employees can not be available. A scenario comprising enterprises integrated by IT and ubiquitous computing as well.

**Keywords**— Agent Bot, Ubiquitous Computing, context-awareness.

**Resumo**— As empresas necessitam continuamente de novas tecnologias para poderem se adaptar a um mercado cada vez mais competitivo, assim como os colaboradores destas empresas precisam de recursos para poderem cumprir suas tarefas, metas e executar seus processos em tempo hábil, de forma ágil e eficaz. Este artigo apresenta um modelo para representar colaboradores durante a execução de determinados processos e tarefas. Isto deve ocorrer quando um ou mais colaboradores não se encontrarem disponíveis. Este artigo também apresenta um cenário de empresas que são providas de tecnologias de informação integradas e computação Ubíquia.

**Palavras-chave** – *Agent Bot*, Computação Ubíquia, Consciência de Contexto.

## 1 Introdução

Para sobreviverem em um mercado cada vez mais competitivo, as empresas têm tido que aperfeiçoar seus processos em vários níveis, implantando novos métodos de trabalho e uma série de tecnologias de informação e comunicação (TIC). Tais medidas, em suma, visam torná-las mais ágeis em várias frentes. Cobrem as atividades normais de operação da manufatura e seus procedimentos de supervisão, passando por dar o devido suporte a introduções de inovações, e vão até ao encontro de respostas mais rápidas e confiáveis a clientes e fornecedores.

A prática tem mostrado que essas mudanças, se por um lado têm facilitado o trabalho das pessoas, por outro as têm feito terem que ser mais produtivas, acarretando um aumento ainda maior de tarefas e responsabilidades. Elas têm estado cada vez mais ocupadas e mergulhadas em diversos problemas, muitas vezes não conseguindo cumprir as tarefas em tempo hábil e/ou com a qualidade esperada.

As pessoas são o mais nobre recurso para as empresas. Portanto, a sua participação nas tarefas deveria ser direcionada para a realização de atividades tão essenciais hoje em dia quanto a operação dos processos em si, ou seja, para atividades que agregam maior valor: a realização de negócios, o exercício da criatividade para inovação, e a aprendizagem/aperfeiçoamento. Porém, a rotina diária das empresas obriga-os a tarefas repetitivas, enfadonhas e muitas delas sem grande valor. Mesmo nas tarefas de grande valor, a pessoa está ocupada com outras coisas e, assim, não consegue executá-las a tempo e com qualidade. Por conseguinte, isso afeta diretamente a agilidade e a eficiência da empresa como um todo, principalmente se se considerar que uma empresa funciona cada vez mais de forma integrada, com as suas várias atividades cada vez mais dependentes umas das outras.

Isto se torna ainda mais problemático à medida que se observa que as pessoas estão envolvidas em diversas tarefas simultaneamente, por vezes de diferentes setores de atuação, com terminologias próprias e com vários contextos de execução. Portanto, as pessoas têm que lidar cada vez com mais tarefas – e em geral simultaneamente – nas suas rotinas diárias, têm que executar as tarefas com crescente eficácia e entender e saber lidar com as particularidades de cada tarefa.

Do ponto de vista computacional, várias abordagens têm sido propostas para lidar com essa complexidade. No entanto, todas elas atacam apenas a questão de potencializar a execução de um processo de forma mais correta, mas sempre requerendo a presença do usuário.

Neste sentido, este presente trabalho propõe uma nova abordagem para o problema, que é a utilização de *Agent Bots*. Neste trabalho, um *Agent Bot* é tido como um processo computacional virtual criado para representar um ou vários usuários na execução de certas tarefas, automaticamente ou com algum grau de intervenção/supervisão humana. Auxiliados por *Agent Bots*, colaboradores podem se concentrar nas tarefas essenciais e mais nobres, que realmente requeiram sua participação direta.

Este artigo está organizado da seguinte maneira: A seção 2 resume o nível atual atingido pelas principais abordagens existentes na direção do problema em questão. A seção 3 apresenta uma visão geral do cenário dentro do qual a abordagem é proposta, bem como faz um resumo de relevantes tecnologias de informação associadas a esta visão. A seção 4 descreve o modelo conceitual proposto, apresentando os componentes relevantes da arquitetura do *Agent Bot*, complementando com uma preliminar análise das tecnologias potenciais para a implementação do *Agent Bot*. A seção 5 apresenta a primeira versão de um protótipo baseado no modelo

conceitual mostrado na seção 4. Na seção 6 são apresentadas considerações finais e próximos passos.

## 2. Breve revisão de abordagens

Há bastante tempo têm-se utilizado *Sistemas de Suporte à Decisão* e *Sistemas Especialistas* para auxiliar na execução de tarefas. Porém, são capazes de dar algum apoio apenas na solução de problemas bem específicos e pré-determinados, e requerem sempre a presença de um humano.

Mais recentemente, tem surgido a figura dos *Assistentes*. Porém, estes também dão uma ajuda limitada (em relação a problemática em questão) uma vez que apenas prestam algum auxílio para se tirar dúvidas sobre como uma dada ação deve ser feita. Já outros tipos de assistentes ficam programados para buscar certas informações em certos *sites* da Internet ou bancos de dados. Também não resolvem o problema, senão apenas prestam alguma ajuda para se ter certas informações disponíveis e se executar a tarefa com mais eficácia. Porém, é a pessoa quem tem que fazer a tarefa. Já as ferramentas de “Trabalho em Grupo” (*Groupware*) oferecem apenas um ambiente mais adequado para que as pessoas colaborem entre si.

Numa direção mais avançada e relacionada com a problemática de base tratada neste artigo, surge a abordagem dos sistemas multiagentes (Weiss, 1999). Nesta, por exemplo, agentes de negociação eletrônica realizam parte das trocas de informação com outros agentes automaticamente, sem intervenção humana. Porém, atacam apenas o processo “negociação” e para cenários de negócios rigorosamente definidos.

A praticamente inexistência de soluções para a problemática abordada não é por acaso. A complexidade para se representar um ser humano em variadas situações é enorme. Neste sentido, o que se deseja é ir além das soluções ora propostas, não só no sentido de se inserir situações empresariais concretas e práticas, como também de se criar um arcabouço aberto e genérico onde novas situações possam ser adicionadas ao longo do tempo. É importante ressaltar novamente que não se deseja substituir o ser humano na execução dos processos. Deseja-se essencialmente diminuir o seu excesso de trabalho, potencializando que se dedique majoritariamente a tarefas mais nobres caso o usuário assim o deseje.

## 3. Agent Bots

Antes de explicar o significado de *Agent Bot* focado neste trabalho, é importante apresentar o significado de cada um dos seus termos.

Um agente é, segundo Russell e Norvig, tudo aquilo que pode, através de sensores, perceber o ambiente em que se encontra e agir autonomamente sobre este por meio de atuadores (Russell, 1995). Já a palavra *Bot* é a abreviação da palavra *Robot*, que quer dizer *trabalho* em tcheco (Primo, 2002).

No cenário de manufatura estudado neste trabalho, pode-se definir um *Agent Bot* como um processo virtual (módulo de *software*) que representa

um colaborador em algum processo, garantindo a execução de suas atividades necessárias via um conjunto de serviços, com a devida segurança, restrições de ações e adaptação às condições vigentes do ambiente computacional. Mais especificamente, deseja-se contemplar ter-se um *Agent Bot* atuando em sete macro tipos de ações:

- 1- substituir um usuário na execução de algumas tarefas;
- 2- auxiliar um usuário na execução de algumas tarefas;
- 3- decifrar para um usuário na forma de uma seqüência de ações uma tarefa abstrata recebida;
- 4- ajudar um usuário no planejamento das suas atividades;
- 5- gerenciar a agenda do usuário;
- 6- processar as informações provenientes de redes ubíquas e contextualizá-las, filtrá-las e apresentá-las ao usuário;
- 7- buscar informações e documentos em certos repositórios.

Essas ações podem ter três níveis de autonomia: *total* (decide e age sozinho), *parcial* (decide e age sozinho, mas com possibilidade de intervenção humana e/ou de aprovação de algum outro sistema) ou *sem autonomia* (decide e age de acordo com configurações previamente determinadas).

O *Agent Bot* deve ter a capacidade de se comunicar com os demais membros da empresa através de uma plataforma computacional interoperável, fornecendo e requisitando informações, processos, atividades e serviços. Dessa forma, os colaboradores, ou seus *Agent Bots*, podem colaborar entre si para a execução de determinados processos ou tarefas. Isso implica na figura de um *plano*, onde o *Agent Bot* deve *raciocinar* sobre o *que* fazer, *como* fazer, *quando* fazer, *com quem* se comunicar, *como* interagir, *o que* adaptar, *quais* recursos usar, *quanto* usar e *de quem* usar.

A comunicação entre os *Agent Bots* e os sistemas da empresa pode se dar através de serviços, em que um cliente faz uma requisição e o serviço envia uma resposta. Esses serviços precisam manter um protocolo padrão de comunicação, onde o *Agent Bot* possa trocar informações, processando e respondendo a requisições de forma padronizada.

### 3.1. Computação em Grade

Além de poder ter autonomia quanto a como fazer uma tarefa, um *Agent Bot* pode ter também autonomia para decidir onde a tarefa será computacionalmente executada. Isso visa maximizar a possibilidade de se cumprir a data de entrega da tarefa e, ao mesmo tempo, racionalizar a utilização de recursos computacionais, que usualmente estão em um bom número subutilizados em termos de uso de CPU e capacidade de armazenamento.

“Grades Computacionais” (*Grid Computing*) (Roure, 2003) tem emergido como uma tecnologia de crescente relevância e que agora começa a ganhar força também nas empresas. Essencialmente, consiste num conjunto virtual lógico e distribuído de computadores que, juntos, formam temporariamente

um processador com grande poder de desempenho. Graças a plataformas de *Grid* hoje já disponíveis (Globus (Globus, 2007), por exemplo), consegue-se configurar um *grid* de computadores e se ter a flexibilidade de se decidir onde um determinado programa será executado, independentemente de onde ele se encontra fisicamente, além de fazer com que, transparente aos usuários, suas máquinas (CPU e/ou disco) possam ser temporariamente compartilhadas com outros sistemas/usuários (Pinheiro, 2005).

Do ponto de vista funcional, plataformas *Grid* disponibilizam suas funcionalidades na forma de “serviços” de software, que podem ser invocados quando se desejar ou de acordo com uma dada lógica do processo. Para o *Agent Bot*, este serviço é visto apenas como mais um dos serviços disponíveis para que ele possa alcançar seus objetivos (Figura 1, Provedores de Serviços).

### 3.2. Computação Ubíqua

Para que um colaborador possa estar informado de todos os processos/tarefas em andamento, dos resultados ora já atingidos e demais informações, é necessário que essas informações possam ser acessadas a qualquer tempo e em qualquer lugar.

Dispositivos computacionais distribuídos por toda a empresa podem trocar informações entre si, assim como os sistemas integrados podem fornecer informações contextuais de diversas fontes e podem possuir diversos tipos de estruturas (Abowd, 1997).

Esse conjunto de dispositivos cria um ambiente de diversas fontes de informação, sensores, sistemas embarcados, altamente distribuídos. A tecnologia de “Computação Ubíqua” (*Ubiquitous Computing*) (Henricksen, 2006) consiste em ambientes/redes onde os dispositivos são integrados e, assim, se possa ter acesso às informações por eles capturadas.

Aplicando este conceito neste trabalho, um *Agent Bot* pode se comunicar com essas redes ubíquas a fim de complementar suas informações e, assim, tomar uma melhor decisão. Ainda, se implantado em dispositivos móveis (PDA, celular, etc.), um *Agent Bot* pode buscar e/ou filtrar informações captadas 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, auxiliados por seus *Agent Bots*, sem que tenham que estar presentes na empresa.

### 3.3. Context Awareness

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. (Henricksen, 2006). Portanto, se devidamente configurados, os dispositivos móveis podem reagir de acordo com essas especificações (Abowd, 1997).

## 4 Proposta de Modelo de *Agent Bot*

O *Agent Bot* deve possuir características que lhe permita executar processos com diversos níveis de autonomia e independência. Essas características compõem o modelo do *Agent Bot* e seus sistemas auxiliares necessários para seu ciclo de vida, como pode ser observado na Figura 1.

O ambiente em que o *software* do *Agent Bot* deve executar precisa estar disponível para que ele permaneça em operação, mesmo quando o usuário estiver desconectado ou não estiver disponível. O *Agent Bot* deve enviar informações ou requisições ao usuário quando solicitado, ou quando necessário para que um processo seja executado.

### 4.1. Base de Dados de Informações e Necessidades

Cada colaborador, possuidor de um *Agent Bot*, deve configurar as informações que são relevantes em conformidade com suas próprias necessidades e requisitos dos processos em que o agente possa se envolver. Estas informações devem ser armazenadas em um repositório de dados, que deve permanecer conectado à rede sem interrupções e que possa ter acesso à Internet para fornecer aos *Agent Bots* as informações necessárias quando necessário.

Um computador servidor, conectado à rede da empresa e com um banco de dados, pode fornecer essa base para as informações do *Agent Bot*. Este banco de dados pode ou não estar no mesmo computador que provê os serviços de *Agent Bots*.

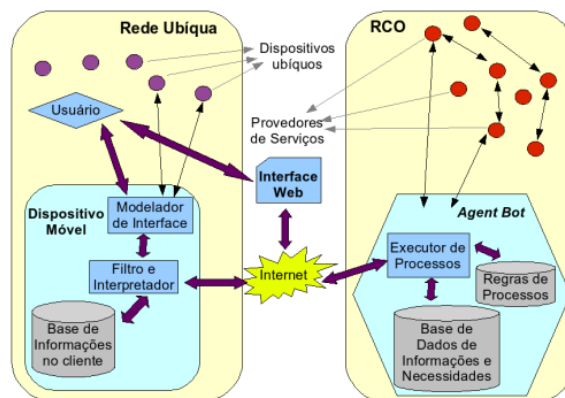


Figura 1. Arquitetura Genérica do Modelo de Representação.

#### 4.2. Regras de Processos

Faz-se necessário que as regras dos processos de negócio (i.e. o que fazer em cada situação esperada) sejam implementadas conforme as atribuições e necessidades dos usuários dos *Agent Bots*. Estas regras podem ser modulares e configuradas nos próprios *Agent Bots* na forma de *plug-ins*, ou mesmo podem ser criadas novas regras para suprir necessidades não atendidas por softwares adquiridos de terceiros ou padronizados.

#### 4.3. Executor de Processos

Para a execução dos processos/tarefas do *Agent Bot* é necessária a utilização de uma linguagem e ontologia para descrever como os vários tipos de processos serão entendidos e posteriormente executados. As informações armazenadas no *Agent Bot*, juntamente com as Regras de Processos, devem ser utilizadas na execução dos processos de negócios.

#### 4.4. Base de Informações no Cliente

O computador do usuário ou dispositivo móvel pode receber uma grande quantidade de informações que provêm de dispositivos das redes ubíquas da empresa. Essas informações necessitam ser filtradas com base em um conjunto de informações do usuário, conforme seus interesses, necessidades e objetivos. Essa base de informações fornece a um determinado filtro para a escolha de quais informações serão realmente relevantes à ele na execução de uma tarefa.

A base de informações em questão deve estar localizada no próprio local ou dispositivo móvel do colaborador. As informações relevantes que devem ser utilizadas para a filtragem podem ser buscadas no banco de dados, sincronizado cada vez que essas informações são alteradas e o dispositivo móvel estiver disponível para as trocas das informações.

#### 4.5. Filtro e Interpretador

Ainda do lado do colaborador, deve existir um interpretador que receba as informações providas do *Agent Bot*, dos sensores de redes ubíquas da empresa e de entradas do usuário. Este interpretador compila as informações de forma que ele possa modelar os processos para o colaborador.

#### 4.6. Modelador de Interface

Para cada tipo de processo, uma interface diferente pode ser necessária. Dessa forma, é definido um módulo que deve ajustar a interface ao colaborador (usuário do dispositivo móvel) baseado no conjunto de informações enviadas pelo *Agent Bot*.

#### 4.7. Interface Web

Para que o colaborador possa acessar e configurar suas informações e regras de processos no *Agent Bot* de forma segura e ágil, deve existir uma interface que possa ser acessada de qualquer lugar, por meio de uma interface Web. Esta fornece ao colaborador as opções e operações que nem sempre é possível no

caso de dispositivos móveis, devido a limitações visuais dos mesmos.

#### 4.8. Tecnologias de Informação para Implementação

Desejando-se criar um *Agent Bot* que possa ser também acessado e operado pela Web, várias tecnologias de informação podem ser utilizadas. No entanto, com base em um extenso estudo, algumas tecnologias têm sido vistas como mais apropriadas.

A tecnologia de “serviços web” (*web services*), amplamente usada atualmente e quase tida como padrão de fato para soluções distribuídas de fraco acoplamento, pode fornecer o padrão necessário para manter a compatibilidade entre estes serviços. Serviços web são vistos como pequenos módulos de software, acessíveis por outras aplicações por meio de um conjunto de protocolos Web, através de uma URL e baseados em um outro conjunto amplo de padrões já consolidados, principalmente a linguagem XML (*Extended Markup Language*), a forma de se descrever a interface de um serviço - WSDL (*Web Services Description Language*), o protocolo SOAP (*Simple Object Access Protocol*) e o UDDI (*Universal Description, Discovery and Integration*). Através do uso desses padrões, constroem-se aplicações modulares, abertas, orientadas à Internet e com interfaces baseadas em padrões (Alonso, 2004).

A execução dos processos/tarefas realizada pelo *Agent Bot* também precisa ser padronizada, de forma a se adaptar às diversas tarefas que possam ser requeridas. A “Linguagem de Execução de Processos de Negócios para Serviços Web” (*Business Process Execution Language for Web Services – BPEL4WS*) provê uma especificação formal para processos e protocolos de interação (IBM, 2003). O BPEL4WS é organizado de forma estruturada em documentos XML, representando sua especificação e facilitando a integração dos processos, tanto dentro da própria organização, como inclusive no espaço de *business-to-business*, B2B (Akehurst, 2004).

A tecnologia de agentes surge como a mais apropriada tecnologia para suportar este “objeto inteligente”, que seria o *Agent Bot*. Inicialmente desenvolvidos em ambientes próprios e com linguagens de comunicação proprietárias, atualmente já começam a fazer uso de tecnologias mais recentes e padrões, como XML e serviços web. Sendo assim, agora podem se comunicar usando SOAP e expressar suas performativas em XML. Dessa forma, os comportamentos dos agentes podem ser modelados como serviços, podendo interoperar com serviços web, serviços de *grid* e redes *peer-to-peer*. Um outro ponto a favor é que já existem diversas plataformas de sistemas multiagentes, sendo que muitas delas já possuem um bom nível de maturidade e robustez.

## 5 Protótipo

Para uma primeira versão de protótipo, foi definido um comportamento que faz a gerência de uma conta de e-mail do usuário pelo *Agent Bot*. Ou seja, foi implementada uma instância do macro tipo de ação I

mencionado na seção 3, derivada da arquitetura genérica mostrada na figura 1. Essa versão é composta por: (i) uma interface web desenvolvida em linguagem PHP para a criação e configuração do agente, (ii) uma base de dados em MySQL para o armazenamento das informações deste, e (iii) a implementação, propriamente dita, do agente em linguagem de programação Java.

Por via de regra, foi definido que o protótipo deveria ser o menos intrusivo possível, ou seja, nenhum *software* extra deveria ser instalado no *desktop* do usuário. Sendo assim, o agente é executado em um servidor situado remotamente e este controla as informações do usuário.

No protótipo, o agente permanece em execução, verificando informações como o *status* do usuário no Gtalk<sup>1</sup> (*conectado, desconectado, ocupado*), a chegada de novas mensagens em uma conta de e-mail do usuário, e tratando essas por meio de um conjunto de regras definidas pelo usuário.

A Figura 2 apresenta, de um modo geral, o fluxo de execução dos passos apresentados abaixo, envolvidos no estudo de caso:

- 1- O usuário cadastra suas preferências no *AgentBot*;
- 2- Chega uma tarefa ou mensagem;
- 3- Detectada a tarefa ou mensagem, o *AgentBot* verifica no banco de dados os critérios de ativação e sua autonomia em relação a esta;
- 4- É montado o plano de execução pelo *AgentBot* conforme suas regras de negócios;
- 5- Por fim, o *AgentBot* executa o plano;

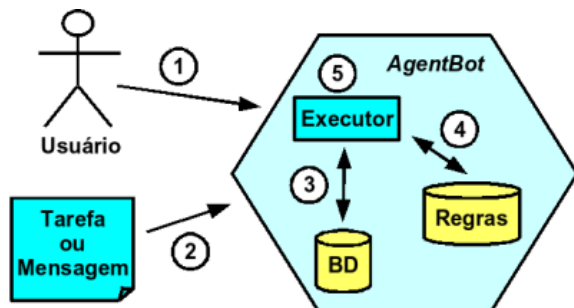


Figura 2. Fluxo de execução do Protótipo.

Uma sintaxe de regras para a execução do comportamento do *Agent Bot* foi definida preliminarmente para este protótipo, sendo que na próxima versão deve ser substituída por documentos BPEL. As figuras 3 e 4 apresentam parte de um código XML exemplo que definem regras de como o agente deve se comportar quando uma nova mensagem chegar na caixa de entrada do usuário.

Na Figura 3, o usuário define uma regra para verificar a procedência da mensagem, o *status* do usuário no Gtalk e a existência de uma determinada palavra chave no *subject* da mensagem recebida. Caso a expressão seja satisfeita, um *reply* deve ser enviado ao remetente da mensagem. Neste caso, se o *status* do Gtalk for *unavailable*, se o remetente da mensagem

for *popov@das.ufsc.br* e se no *subject* existir a palavra chave URGENTE, o *Agent Bot* deve responder ao remetente com a mensagem “*Desculpe, no momento não posso responder!*”, juntamente com o corpo original da mensagem.

```
<rule>
  <expression>
    <gtalkStatus>unavailable</gtalkStatus>
    <emailfrom>popov@das.ufsc.br</emailfrom>
    <emailSubject>URGENTE</emailSubject>
  </expression>
  <then>
    <reply>
      Desculpe, no momento não
      posso responder!
    </reply>
  </then>
</rule>
```

Figura 3. Primeira regra exemplo do protótipo, em XML.

Já na regra apresentada na Figura 4, é verificado se a mensagem refere-se a uma nova atividade a ser agendada, por meio de uma sintaxe pré-definida no *subject* das mensagens. Neste protótipo, apenas uma mensagem é enviada ao usuário do *Agent Bot* para alertá-lo sobre o compromisso, sem ainda fazer uso de uma agenda de compromissos onde essas atividades estejam armazenadas.

É importante frisar que o arcabouço é aberto no sentido de que facilmente outras regras podem ser acrescentadas conforme as necessidades do usuário.

```
<rule>
  <expression>
    <gtalkStatus>unavailable</gtalkStatus>
    <emailfrom>rabelo@das.ufsc.br</emailfrom>
    <emailSubject>FAZER</emailSubject>
  </expression>
  <then>
    <sendEmail>
      <to>popov@das.ufsc.br</to>
      <subject>NOVA TAREFA</subject>
      <content>emailSubject</content>
      <content>emailContent</content>
    </sendEmail>
  </then>
</rule>
```

Figura 4. Segunda regra exemplo do protótipo, em XML.

## 6 Considerações e Perspectivas Futuras

O presente trabalho apresentou uma proposta de um sistema tipo *Agent Bot*, que visa minimizar o tempo gasto dos usuários na execução de algumas das suas tarefas, especialmente as repetitivas, que não agregam valor, ou mesmo quando o colaborador não está disponível e/ou muito ocupado. Desta forma, podem dedicar mais tempo a atividades mais importantes, como novos negócios, melhorias de qualidade, inovações e aprendizagem e, paralelamente, têm maiores chances de garantir o cumprimento das tarefas no prazo estipulado e até com maior potencial qualitativo.

É apresentado um modelo para representação de um colaborador na execução de determinados

<sup>1</sup>Gtalk é um *software* de comunicação entre usuários conectados à internet. Link: <http://www.google.com/talk/intl/pt-BR/>

processos de negócios. Sua arquitetura é demonstrada em um nível conceitual, sendo indicados os módulos relevantes para a execução e ciclo de vida dos processos do *Agent Bot*. Uma análise preliminar de tecnologias potenciais para implementação e um protótipo inicial foram igualmente mostrados. Este protótipo consiste na implementação parcial de um caso de uso de um processo de negócios; no caso, de recebimento de uma mensagem eletrônica (e-mail).

Os próximos passos do trabalho irão na direção de refinamentos no modelo conceitual e na arquitetura genérica, de detalhamento dos casos de uso dos sete tipos de processos de negócio que deverão ser tratados, e análise das tecnologias e padrões para implementação.

### **Agradecimentos**

Este trabalho foi parcialmente suportado 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>).

### **Referências Bibliográficas**

- Abowd, G. D. et al. Cyberguide: a mobile context-aware tour guide. *Wireless Network*, n. 3, p.421-433, 1997.
- Akehurst, D. H.; *Validating BPEL Specifications using OCL*, Published by the Computing Laboratory, University of Kent, Canterbury, 2004.
- Alonso, G. *At all; Web Services: concepts, architecture and applications*; Springer Verlag NY, 1ª ed. 2004.
- Globus, The Globus Alliance, <http://www.globus.org> (acessado em 03/07).
- Henricksen, K., Indulska, J., *Developing context-aware pervasive computing applications: model and approach*. *Pervasive and Mobile Computing*; pp. 37-64. <http://www.sciencedirect.com>. (acesso em 12/06).
- IBM, *Business Process Execution Language for Web Services*; <http://www-128.ibm.com/developerworks/library/specification/ws-bpel/> (acessado em 03/07).
- Pinheiro, F. J. R., Rabelo, R. J. *Towards Flexible Infrastructures for Supply Chain Management: a Grid Computing based approach*. In: 38th CIRP - International Seminar on Manufacturing Systems, Florianópolis-SC, Brazil, May 2005.
- Roure, D, at all; *The Evolution of the Grid*. In: BERMAN, Fran, at all; *Grid Computing: Making The Global Infrastructure a Reality*; Publisher: John Wiley & Sons; (April 8, 2003), pp 65-100.
- Russel, S., Norvig, Peter, *Artificial intelligence: a modern approach*; Prentice-Hall, pp 773-814, EUA 1995.
- Weiss, G., *Multiagent systems: a modern approach to Distributed AI*. Weiss, Gerhard, MIT, 1999.
- Primo, A., Coelho, L. R. *Comunicação e inteligência artificial: interagindo com a robô de conversação Cybelle*. In: MOTTA, L. G. M. et al. (Eds.). *Estratégias e culturas da comunicação* ed.Brasília. Brasília: Editora UnB, 2002. pp. 83-106.