

# Classificação das Abordagens de Integração de Agentes com Aplicações Heterogêneas

Otávio A. Matoso<sup>1</sup>, Jomi F. Hübner<sup>1</sup>, Maicon R. Zatelli<sup>1</sup>

<sup>1</sup>Universidade Federal de Santa Catarina (UFSC)  
Florianópolis – SC – Brazil

otaviomatoso@yahoo.com.br, {jomi.hubner,maicon.zatelli}@ufsc.br

**Resumo.** *A complexidade dos sistemas atuais demanda soluções computacionais distribuídas e heterogêneas. Em alguns casos, a inclusão de Sistemas Multiagentes (SMA) na solução pode ser promissora. No entanto, a falta de abordagens para integrar agentes com uma grande variedade de aplicações ainda representa um obstáculo à exploração do potencial de SMA em um ambiente distribuído e heterogêneo. Este trabalho investiga o estado da arte relacionado à integração de agentes com aplicações heterogêneas e tem como principal objetivo elaborar uma classificação das abordagens estudadas.*

## 1. Introdução

Um sistema de informação é normalmente composto por inúmeras aplicações, que podem ser oriundas de desenvolvimento interno da própria empresa, adquiridas de terceiros, parte de sistemas legados ou uma combinação dessas opções [Hohpe and Woolf 2004]. Esse sistema deve ter suas aplicações trabalhando em conjunto, na direção de um mesmo objetivo, levando a um fluxo de informações mais dinâmico e fluído. Para que essas aplicações possam interagir entre si, trocando informações de forma eficiente e coordenada, é preciso que elas estejam bem integradas. Porém, um cenário comum é que essas aplicações sejam heterogêneas, ou seja, atuem em diferentes camadas de diferentes sistemas operacionais e possuam suas próprias estruturas de dados e protocolos de comunicação. Como consequência dessa diversidade, a interoperabilidade se mostra bastante prejudicada.

SMA é uma sub-área da inteligência artificial que facilita a resolução de problemas complexos por meio de uma estratégia onde o problema é decomposto em problemas menores e mais simples, que são tratados por entidades autônomas, os agentes. Um agente pode ser denotado como um sistema computacional baseado em software que possui como principais características autonomia, habilidade social, reatividade e proatividade [Wooldridge and Jennings 1995]. Devido às suas particulares características, SMA vem se consolidando como uma importante abordagem para o desenvolvimento de aplicações, se mostrando adequada para ambientes dinâmicos e abertos.

No entanto, a falta de abordagens para integrar agentes com uma grande variedade de aplicações ainda representa um obstáculo à exploração do potencial de SMA em um ambiente distribuído e heterogêneo. O presente trabalho, que é parte de uma pesquisa em andamento, investiga o estado da arte no que se refere à integração de SMA com entidades heterogêneas e apresenta uma classificação, na forma de tabela, das abordagens estudadas, exibindo as principais características encontradas em cada trabalho.

## 2. Trabalhos relacionados

A revisão bibliográfica apresentada nesta seção foi realizada com o intuito de investigar abordagens que geram algum nível de interoperabilidade entre agentes e entidades heterogêneas.

Alguns autores abordaram a integração entre SMA e *Serviços Web* (SW). Contreras e Sheremetov [Contreras and Sheremetov 2008] apresentaram um conjunto de extensões que devem ser feitas em uma plataforma de agentes compatível com as especificações FIPA para gerar interoperabilidade entre agentes e SW, permitindo a implementação de aplicações orientadas a serviços usando agentes e SW já existentes. Seguindo a mesma linha, Coria et al. [Coria et al. 2014] propõem um modelo para a composição dinâmica de SW em um ambiente de computação em nuvem. Neste modelo, um SMA analisa a semântica de um conjunto de especificações dadas pelo usuário, realiza descoberta de SW na nuvem a partir da semântica obtida e compõe o sistema usando os SW disponíveis. Em [Tapia et al. 2009] é descrita uma arquitetura multiagentes onde serviços são gerenciados e controlados por agentes deliberativos. Nessa arquitetura, as funcionalidades dos sistemas são modeladas como serviços distribuídos que podem ser invocados pelos agentes.

A integração de SMA com *sistemas legados* também foi investigada. Estratégias para encapsular tais sistemas de modo que eles possam ser tratados como agentes em um SMA são apresentadas em [Li 2010, Zhao et al. 2008]. Li propõe uma estratégia de integração de dados baseada em SOAP/XML para que sistemas legados sejam encapsulados como agentes e possam cooperar entre si. Já Zhao et al. apresentam um mecanismo de encapsulamento baseado no sistema operacional Windows para gerar interface entre sistemas legados e SMA.

Outros trabalhos focaram na integração de agentes com padrões que geram certo nível de interoperabilidade em domínios específicos. Em [Saleem et al. 2010], as funções de entidades compatíveis com o padrão *IEC 61850* de automação de subestação são implementadas diretamente nos agentes, permitindo que um SMA seja modelado para controle e proteção de sistemas de energia elétrica. Já Miranda et al. [Miranda et al. 2012] utilizam SMA para facilitar a troca de informações entre sistemas e equipamentos de saúde. Na arquitetura proposta, entidades que seguem as normas *HL7*<sup>1</sup> podem ser implementadas sob o paradigma de agentes, facilitando a troca de mensagens entre as entidades.

A integração de agentes com uma variedade de recursos e serviços externos foi investigada em outras pesquisas. Em [Vrba et al. 2014], os autores apresentaram um framework para encapsular um SMA como um serviço a ser usado em um *Enterprise Service Bus* (ESB). Um *gateway* é criado para traduzir mensagens FIPA-ACL em mensagens ESB e vice-versa, permitindo comunicação entre agentes e demais entidades presentes no ESB. Cranefield e Ranathunga [Cranefield and Ranathunga 2013] criaram uma interface entre agentes e o Apache Camel, um framework de integração baseado nos *Padrões de Integração Corporativa* (EIPs) apresentados por Hohpe e Woolf [Hohpe and Woolf 2004]. Com isso, tais agentes podem trocar mensagens com outras entidades que possuem interface com esse framework.

---

<sup>1</sup>Padrão utilizado no setor da saúde para a troca de informações entre aplicações e equipamentos médicos.

### 3. Classificação dos trabalhos

Considerando a visão geral de cada trabalho, foi possível identificar algumas propriedades em comum no que se refere à integração, tornando possível uma certa aproximação entre as propostas. Foram consideradas as seguintes propriedades:

- *Entidades integradas*: o tipo da entidade que foi integrada com os agentes, sendo SW e sistema legado exemplos;
- *Arquitetura dos agentes*: a arquitetura dos agentes utilizada no respectivo trabalho, por exemplo arquitetura BDI;
- *Padrão de comunicação*: o protocolo adotado para comunicação entre agentes e demais entidades;
- *Descrição das entidades*: forma de acesso dos agentes às funcionalidades das entidades que estão sendo integradas;
- *Registro das entidades*: o local onde são registradas as entidades e/ou suas funcionalidades integradas;
- *Encapsulamento das entidades pelo SMA*: indica se a entidade ou funcionalidade foi encapsulada ou não pelo SMA.

A Figura 1 compara as propostas analisadas levando em consideração as propriedades citadas anteriormente. Em alguns trabalhos, não foi possível identificar determinadas propriedades. Em tal situação, utilizou-se o caractere (-) para indicar tal indefinição.

Trabalhos	Entidades integradas	Arquitetura dos agentes	Padrão de comunicação	Descrição das entidades	Registro das entidades	Encapsulamento das entidades pelo SMA
Contreras e Sheremetov 2008	Serviço Web	<i>Behavioral</i>	SOAP	WSDL	<i>Directory Facilitator</i>	<input checked="" type="checkbox"/>
Coria et al. 2014	Serviço Web	-	SOAP	WSDL	Nuvem	<input type="checkbox"/>
Tapia et al. 2009	Serviços local, Web ou <i>Stand Alone</i>	BDI	SOAP	-	Diretório de Serviços	<input checked="" type="checkbox"/>
Vrba et al. 2014	ESB JBoss	<i>Behavioral</i>	FIPA-ACL	WSDL	Diretório de Serviços	<input type="checkbox"/>
Cranefield e Ranathunga 2013	Componentes Camel	BDI	EIPs	-	URI	<input type="checkbox"/>
Xiangyu Li 2010	Sistema legado	-	SOAP	XML	agente	<input checked="" type="checkbox"/>
Zhao et al. 2008	Sistema legado	BDIAIP (BDI Agent Initiative Perception)	COM	DLL/EXE - Componentes COM	<i>Capability Database</i>	<input checked="" type="checkbox"/>
Saleem et al. 2010	Sistemas de automação de subestação	BDI	IEC 61850	Plano de Controle	Papeis	<input checked="" type="checkbox"/>
Miranda et al. 2012	Sistemas de informação em Saúde	<i>Behavioral</i>	HL-7	Ontologia	<i>Directory Facilitator</i>	<input checked="" type="checkbox"/>

**Figura 1. Comparação entre trabalhos de integração**

Com base na revisão do estado da arte e na classificação apresentada, nota-se que grande parte dos trabalhos apresentam soluções mais específicas, integrando agentes com um conjunto de entidades específicas. Contreras e Sheremetov, Coria et al. e Tapia et al. investigaram a interoperabilidade entre agentes e serviços, majoritariamente SW. Miranda

et al. integraram agentes com equipamentos e sistemas de informação que seguem o HL7, um conjunto de normas internacionais na área da saúde. Seguindo a mesma ideia, Saleem et al. discutiram um mapeamento entre uma arquitetura baseada em agentes e o padrão IEC 61850, adotado em projetos de automação de subestação. Já Zhao et al. e Li utilizaram uma estratégia de encapsulamento para que conjuntos específicos de sistemas legados sejam tratados como agentes e compartilhem informações entre si.

Em caso de um cenário onde um SMA necessite trocar informações com uma variedade de entidades heterogêneas, nenhuma das soluções citadas no parágrafo anterior seriam completamente adequadas. Tendo isso em vista, as pesquisas de Cranefield e Ranathunga e Vrba et al. merecem destaque, uma vez que a estratégia adotada nesses trabalhos permite que os agentes troquem mensagens com uma diversidade de aplicações heterogêneas de forma simultânea. Cranefield e Ranathunga desenvolveram uma interface entre uma plataforma de agentes e o framework de integração Apache Camel, de modo a gerar interoperabilidade entre os agentes e qualquer entidade que também tenha interface com o Camel. Enquanto que Vrba et al. encapsularam um SMA em um ESB, permitindo que os agentes troquem mensagens com qualquer outra entidade presente no ESB.

#### **4. Considerações finais**

Este trabalho apresentou uma revisão do estado da arte no que se refere à integração de agentes com uma variedade de entidades heterogêneas, com algumas propriedades sendo analisadas. Verificou-se que grande parte das soluções são direcionadas à integração de agentes com conjuntos específicos de entidades. Nessa ótica, duas pesquisas se destacaram por abrir a possibilidade de integração de agentes com uma gama de tecnologias, em especial a pesquisa de Cranefield e Ranathunga. A escolha desses autores por desenvolver uma interface entre agentes e o Apache Camel se mostrou interessante, pois o Camel é um projeto de código-fonte aberto bastante maduro e fornece conectividade com uma grande variedade de transportes e APIs. Por exemplo, ao integrar agentes com o Camel é possível que esses agentes recebam mensagens de uma fila JMS, realizem requisições HTTP ou consultas SQL.

Uma primeira proposta de trabalho futuro é considerar a dimensão do ambiente como uma abstração de primeira classe em um SMA, como proposto em [Weyns et al. 2007], e investigar seu papel na integração de SMA com entidades heterogêneas, uma vez que este trabalho considerou apenas a dimensão dos agentes. Em [Omicini et al. 2008], ao considerar a integração de SMA com aplicações externas, algumas dessas aplicações podem ser consideradas autônomas, sendo modeladas como agentes, enquanto que outras podem ser não-autônomas e são melhor modeladas como artefatos do ambiente. Seguindo essa ideia e tendo o trabalho de Cranefield e Ranathunga como inspiração, pretende-se propor uma interface entre o Apache Camel e a dimensão do ambiente, acrescentando aos agentes a capacidade de interagir com entidades externas representadas na abstração de artefato do ambiente.

## Referências

- Contreras, M. and Sheremetov, L. (2008). Industrial application integration using the unification approach to agent-enabled semantic soa. *Robotics and Computer-Integrated Manufacturing*, 24(5):680–695.
- Coria, J. A. G., Castellanos-Garzón, J. A., and Corchado, J. M. (2014). Intelligent business processes composition based on multi-agent systems. *Expert Systems with Applications*, 41(4):1189–1205.
- Cranefield, S. and Ranathunga, S. (2013). Embedding agents in business processes using enterprise integration patterns. In *International Workshop on Engineering Multi-Agent Systems*, pages 97–116. Springer.
- Hohpe, G. and Woolf, B. (2004). *Enterprise integration patterns: Designing, building, and deploying messaging solutions*. Addison-Wesley Professional.
- Li, X. (2010). A multi-agent based legacy information system integration strategy. In *2010 International Conference on Networking and Digital Society*, volume 2, pages 72–75. IEEE.
- Miranda, M., Salazar, M., Portela, F., Santos, M., Abelha, A., Neves, J., and Machado, J. (2012). Multi-agent systems for hl7 interoperability services. *Procedia Technology*, 5:725–733.
- Omicini, A., Ricci, A., and Viroli, M. (2008). Artifacts in the a&a meta-model for multi-agent systems. *Autonomous agents and multi-agent systems*, 17(3):432–456.
- Saleem, A., Honeth, N., and Nordström, L. (2010). A case study of multi-agent interoperability in iec 61850 environments. In *2010 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT Europe)*, pages 1–8. IEEE.
- Tapia, D. I., Rodríguez, S., Bajo, J., and Corchado, J. M. (2009). Fusion@, a soa-based multi-agent architecture. In *International Symposium on Distributed Computing and Artificial Intelligence 2008 (DCAI 2008)*, pages 99–107. Springer.
- Vrba, P., Fuksa, M., and Klíma, M. (2014). Jade-jbossesb gateway: Integration of multi-agent system with enterprise service bus. In *2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 3663–3668. IEEE.
- Weyns, D., Omicini, A., and Odell, J. (2007). Environment as a first class abstraction in multiagent systems. *Autonomous agents and multi-agent systems*, 14(1):5–30.
- Wooldridge, M. and Jennings, N. R. (1995). Intelligent agents: Theory and practice. *The knowledge engineering review*, 10:115–152.
- Zhao, C., Li, Q., Wang, M., Wang, Y., and Li, Y. (2008). An agent based wrapper mechanism used in system integration. In *2008 IEEE International Conference on e-Business Engineering*, pages 637–640. IEEE.