

Massyve-Kit: Um Ambiente de Suporte ao Desenvolvimento de Aplicações Multiagente

André Jacomino
jacomino@gsigma-grucon.ufsc.br

Michel B. Geszychter
geszychter@gsigma-grucon.ufsc.br

Universidade Federal de Santa Catarina – UFSC
Grupo de Sistemas Inteligentes de Manufatura - GSIGMA
Departamentos de Automação e Sistemas e de Engenharia Mecânica
Caixa Postal 476 Cep: 88040-900 Florianópolis (SC) - Brasil
Telefone: (+ 55 48) 331 9387 Telefax: (+ 55 48) 234 1519

Resumo

O desenvolvimento de soluções de *software* abertas e ao mesmo tempo feitas sob medida para os cada vez mais complexos sistemas tem surgido como uma forte tendência. A criação de ambientes computacionais de suporte constitui uma importante contribuição para, entre outras, a diminuição do tempo de desenvolvimento, documentação e manutenção do sistema e, por conseguinte, dos seus custos gerais de desenvolvimento. Dentro deste contexto, o presente artigo apresenta um ambiente computacional (um “kit”) de suporte ao desenvolvimento de aplicações distribuídas e multiagente, denominado Massyve-Kit.

Palavras-chave: Sistemas Multiagente, Agentes, Sistemas Distribuídos.

Abstract

The development of open software solutions and at the same time custom-tailored for the even more complex systems have emerging as a strong tendency. The creation of supporting and user-friendly environments constitutes an important contribution, among other, to decrease the time spent with development, documentation and system maintenance and, consequently, to save overall costs. Within this context, this paper presents a system (a “kit”) that supports the “derivation” of distributed and multiagent applications, that is called Massyve-Kit.

Keywords: Multiagent Systems, Agents, Distributed Systems.

1. INTRODUÇÃO

Analisando a evolução das várias metodologias e abordagens de desenvolvimento de sistemas, verifica-se que, assim como ocorre em outras áreas, elas têm sofrido os efeitos da “Lei do Pêndulo”. Na situação-extremo inicial, os sistemas eram desenvolvidos totalmente “à mão” e à medida para cada cliente, abordagem que veio a ser posteriormente abandonada devido aos altos custos de desenvolvimento que isto representava. Desta situação passou-se para o outro ponto extremo. Neste, os sistemas passaram então a ser desenvolvidos como soluções gerais, comercializados como se servissem para todos os casos, com custos bem

menos elevados (tanto de desenvolvimento como de venda), porém como verdadeiras "caixas pretas" [Rabelo, 97].

Nesta direção, e tendo em consideração os hoje "obrigatórios" requisitos de soluções abertas, integradas, modulares e reutilizáveis, a abordagem de "derivação de sistemas" vem ganhando muita força e tem sido vista como uma abordagem emergente em termos de desenvolvimento de sistemas complexos [Dietrich, 94][Szelke, 94]. Esta abordagem consiste em, basicamente, derivar sistemas particulares ("instâncias" de sistemas) a partir de um sistema genérico ("modelo de referência"), criando uma espécie de imagem coerente, configurada e adaptada deste para uma empresa / sistema em específico, na direção daquela solução ideal. Ao longo do processo de derivação (ou antes dele ser iniciado) são feitas parametrizações e / ou especificações que servem para guiar o processo até que, no final, um sistema particular seja derivado.

Dentro da filosofia de "derivação de sistemas" e da crescente necessidade por ambientes computacionais que ofereçam grande conforto ao usuário ao longo do projeto, desenvolvimento e operação dos sistemas, surge a noção do "kit". O Massyve-Kit é um kit gráfico-interativo leve, pequeno, facilmente instalável e configurável, que permite uma rápida derivação de infra-estruturas para o desenvolvimento de sistemas multiagente particulares não complexos. Tem como principais objetivos servir como ferramenta de base para o ensino e pesquisa, no âmbito de disciplinas de graduação e pós-graduação afetadas pelo emergente tema, e também de base para o treinamento de pessoas no referido tema, dificultado pela inexistência de ferramentas com todas aquelas características no mercado e *freeware*¹.

No caso específico de aplicações apoiadas na tecnologia de sistemas multiagente, o quesito "conforto" têm sido pouco priorizado nos vários kits hoje disponíveis na Internet, principalmente porque têm sido idealizados para outros fins que não aqueles desejados no Massyve-Kit. Paralelamente a isto, muitos desses kits estão na fase de protótipo e, por também tentarem ser ferramentas de propósito geral e extremamente robustas em termos de mecanismos de tolerância à falhas, tornam-se ambientes muito "pesados" e difíceis de instanciar / configurar. Outros tantos não são ferramentas *freeware* ou *shareware*, que possam ser facilmente adquiridas por instituições de ensino e pesquisa.

Também é importante ressaltar que o Massyve-Kit está incluído no âmbito de um projeto de pesquisa aplicada de cooperação internacional, denominado Massyve (*Multiagent Agile Manufacturing Scheduling Systems for Virtual Enterprise*) [Massyve, 98], financiando pela União Européia e pelo CNPq/Brasil (no âmbito de outro projeto).

2. PRINCIPAIS CARACTERÍSTICAS DO MASSYVE-KIT

A construção de Sistemas Multiagente exige a consideração de uma variedade de aspectos, incluindo a definição do domínio de conhecimento, a concepção e implementação das suas funcionalidades associadas (comportamentos), o desenvolvimento da arquitetura dos agentes (modelos mental, de ação e de cooperação) e a definição dos protocolos e infra-estruturas de comunicação que os agentes precisarão para se comunicar com outros agentes (ou sistemas).

¹ Em: <http://www.gsigma-grucon.ufsc.br/massyve/links.htm#kits> é apresentada uma extensiva relação de kits voltados para Sistemas Multiagente.

Numa análise geral, pode-se dizer que um sistema multiagente “ideal” provê mecanismos complexos e robustos de suporte à tolerância à falhas, configuração gráfica e interativa dos protocolos de alto nível de interação multiagente assim como bibliotecas de “comportamentos” codificados, agentes para plataformas heterogêneas (tais como agentes Java), entre outras facilidades. Todavia, algumas dessas características vão de certa forma contra aos objetivos fundamentais do Massyve-Kit, que é o de se ter uma ferramenta leve, rápida, fácil e pequena, essencialmente direcionada a ensino e treinamento. Outras dessas características ainda podem ser consideradas como “estado-da-arte”, requerendo um maior amadurecimento de algumas tecnologias de suporte. Nesse sentido, Massyve-Kit não possui nenhum mecanismo sofisticado de recuperação e tolerância à falhas, usualmente muito “pesados” (o que implica em maior dimensão dos agentes e lentidão do sistema) e de difícil configuração e gerenciamento por um usuário não especialista em sistemas distribuídos e redes. Todavia, oferece-se um mecanismo simples de salvamento dos agentes (e assim dos seus estados vigentes) - em alguns casos com o seu automático relançamento e anúncio - e algumas funcionalidades de apoio à "depuração", especialmente no tocante à troca de mensagens entre os agentes e ao estado geral de consistência do sistema multiagente.

O Massyve-Kit possui as seguintes principais funcionalidades:

- ◆ Criação de modelos / arquiteturas de referência;
- ◆ Criação e configuração de arquiteturas de agentes;
- ◆ Salvamento de arquiteturas de agentes;
- ◆ Carregamento de arquiteturas de agentes;
- ◆ Criação e configuração de agentes;
- ◆ Salvamento de agentes;
- ◆ Edição dos atributos dos agentes;
- ◆ Edição / configuração das conexões dos agentes;
- ◆ Exclusão de agentes;
- ◆ Lançamento de um sistema particular (todos os agentes criados) e / ou de um agente individualmente em máquinas distintas de uma rede local;
- ◆ Troca (envio e recebimento) de mensagens entre os agentes lançados;
- ◆ Estabelecimento, modificação e exclusão de conexões com agentes já lançados;
- ◆ Encerramento da execução de um sistema.

Os agentes gerados pelo Massyve-Kit são agentes de *software* [Bradshaw, 97], não móveis, distribuídos em um ambiente homogêneo (PCs / Windows[®] NT), e que apresentam as três propriedades primordiais de um agente (autonomia, sociabilidade e independência). São modelados como objetos, inspirado no paradigma de programação orientada por agentes (AOP) [Shoham, 93], programados em C++. Além disso, dado que o Massyve-Kit visa suportar a geração de sistemas (distribuídos ou) multiagente particulares, compete ao usuário especificar e programar tanto o protocolo de cooperação que deva ser adotado na solução particular assim como os vários “comportamentos” que os agentes deverão exibir. No caso do protocolo, foi desenvolvido um protocolo simples, livre de contexto, sobre o qual uma semântica / ontologia particular pode ser introduzida (considerando que o Massyve-Kit é uma ferramenta para ensino e treinamento) ou mesmo um protocolo do tipo o “padrão” KQML [Finin, 94] possa ser implantado.

Portanto, o Massyve-Kit oferece um ambiente de suporte à geração da infraestrutura bem como um conjunto de funcionalidades de base para a comunicação em um sistema multiagente/distribuídos particulares. Por outras palavras, pode-se dizer que o Massyve-Kit provê, em linhas gerais, o “alicerce” do sistema e os serviços da camada de cooperação dos agentes, competindo ao usuário o desenvolvimento do *kernel* propriamente dito (consoante a aplicação desejada) dos agentes, onde os vários comportamentos associados à sociabilidade, autonomia e independência (além de outros) deverão ser programados nas classes da arquitetura de referência a ser criada e baseada na qual um sistema particular será derivado.

3. IMPLEMENTAÇÃO

O Massyve-Kit foi definido considerando-se essencialmente a necessidade de derivar sistemas multiagente particulares, utilizando-se aplicações com interfaces gráficas, que possam construir e gerenciar todo o ciclo de vida de um sistema multiagente. Para isso, foram definidas três entidades (aplicações) fundamentais que compõe o Massyve-Kit:

- ◆ **Kit Manager:** é a entidade responsável por, gráfica e interativamente, permitir ao usuário derivar e controlar o sistema multiagente particular durante todo o seu ciclo de vida, envolvendo assim toda a comunidade de agentes e suas relações;
- ◆ **Agent:** é a entidade fundamental de um sistema multiagente. Provê todos os métodos para a cooperação com outros membros da comunidade; contem os métodos específicos para a realização de tarefas de um sistema multiagente particular; permite que o usuário gerencie e controle suas interações através uma interface gráfica;
- ◆ **Address Server:** esta entidade tem como tarefa fundamental o controle de todo o sistema multiagente particular durante sua execução, garantindo a manutenção da consistência global do sistema ora em execução. Para tal "conhece" todos os agentes da comunidade, bem como fornece todas as informações requisitadas pelos mesmos.

Além das três entidades fundamentais do Massyve-Kit acima descritas, foram definidas outras quatro entidades com funções muito específicas e também de relevante importância durante o ciclo de vida de um sistema multiagente particular. Estas entidades foram definidas como:

- ◆ **Reference Architecture Configuration Module:** entidade dedicada a configuração de arquiteturas de referência para a posterior geração de sistemas particulares no **Kit Manager**;
- ◆ **Distribution Manager Process:** entidade dedicada a auxiliar o **Kit Manager** no gerenciamento dos agentes que são lançados em máquinas distintas de uma determinada rede local;
- ◆ **Receiver Process:** entidade que suporta a toda a comunicação dos agentes e dos **Distribution Manager Process**'s, tendo como propósito principal permitir que estas entidades recebam mensagens;
- ◆ **Agent Killer:** entidade dedicada à fase de encerramento de um sistema multiagente particular gerado pelo Massyve-Kit, tendo como propósito principal permitir que uma entidade *Agent* seja terminada.

A figura 1 ilustra o inter-relacionamento entre as várias entidades do sistema. As três entidades fundamentais interagem entre si, indicando o funcionamento cooperativo das entidades *A (Agent)* que são geradas pelo *KM (Kit Manager)* e que são controladas pelo *AS (Address Server)*, mostrando também as interações realizadas entre as demais entidades que compõe o *Massyve-Kit*.

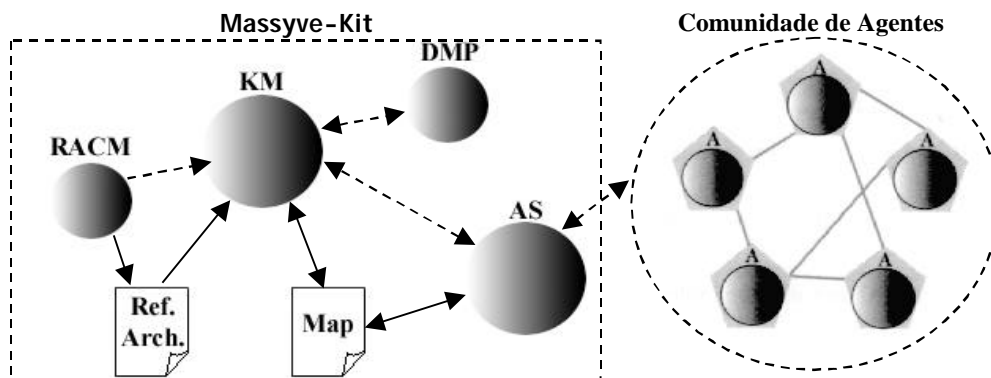


Figura 1 - Modelo Funcional do Massyve-Kit

No *KM* é derivado o sistema multiagente particular onde são representadas as entidades *Agent* e todas as suas relações. Esses dados são salvos em um arquivo tipo texto, que será toda a fonte de informação para o controle a ser realizado pelo *AS* durante a execução deste sistema particular. Este arquivo texto, que será definido como *Map*, conterà sempre o *status* atual do sistema, quer este estando em fase de derivação, quer de execução.

No *RACM (Reference Architecture Configuration Module)* é configurada uma nova arquitetura de referência, e os dados são armazenados em um arquivo tipo texto (*RefArch*). Este arquivo é a fonte de informações utilizada pelo *KM* durante a derivação do sistema multiagente particular. A interação $KM \leftrightarrow RACM$ é feita de forma indireta através do arquivo *RefArch*, de forma similar à interação $KM \leftrightarrow AS$, feita via arquivo *Map*. Por outro lado, o *KM* também interage com o *DMP (Distribution Manager Process)* de forma direta, via troca de mensagens, permitindo que um sistema particular seja lançado de forma distribuída em outros computadores de uma rede local, como ilustra a figura 1.

A figura 2 ilustra como é feita a comunicação entre as entidades *Agent*, bem como suas interações com a entidade *RP (Receiver Process)*. Note-se que a elipse tracejada delimita a entidade lógica do agente, que é uma combinação dessas duas entidades e de um arquivo.

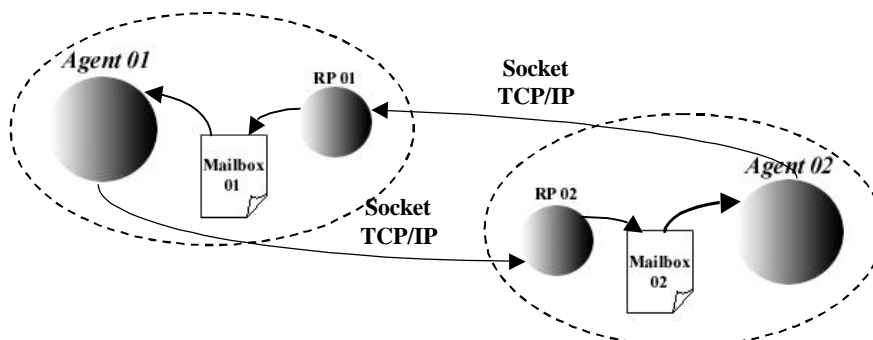


Figura 2 - Modelo Funcional das Entidades *Agent* e *Receiver Process* do Massyve-Kit

Cada entidade *Agent* possui obrigatoriamente uma entidade “escrava” RP, que recebe as mensagens endereçadas àquele agente e as armazena no arquivo de mensagens chamado *Mailbox*. A utilização desta estratégia visa evitar que um agente possa “travar” quando do recebimento de uma mensagem e, assim, que todo o sistema fique “pendurado”.

4. CRIAÇÃO DE SISTEMAS MULTIAGENTE PARTICULARES

O ciclo de vida de um sistema multiagente particular gerado a partir do Massyve-Kit é compreendido por três grandes fases: derivação, lançamento (execução) e encerramento do sistema.

4.1. Derivação do Sistema

Esta é a primeira fase do ciclo de vida. É nela que serão configuradas as arquiteturas de referência para a geração de instâncias de sistemas particulares. Uma vez configurada esta arquitetura, será gerada também a instância propriamente dita, ou seja, uma arquitetura (lógica) de agentes particular. Isso permite uma flexibilidade na definição dos sistemas particulares, sendo que qualquer tipo de arquitetura de referência pode ser facilmente criada. Tipicamente a criação de novas classes de agentes (para novos sistemas) é feita "reutilizando" a classe "*Agent*" (ver capítulo 3) e assim introduzindo-se os métodos particulares desejados.

Uma vez configurada uma arquitetura de referência, todas as demais fases do ciclo de vida deverão ser orientadas pelas regras definidas. Uma arquitetura de referência no Massyve-Kit é composta por *classes de agentes*, *regras de conectividade* e *regras de cardinalidade local e global*. A entidade (aplicação) responsável por permitir a definição de uma arquitetura de referência é a RACM.

Na continuidade da derivação de um sistema multiagente particular, é criada uma instância de sistema baseada em uma dada arquitetura de referência previamente configurada. Para se criar uma instância de um sistema multiagente particular no Massyve-Kit é necessário representar logicamente os agentes e suas conexões, ou seja, toda a arquitetura de agentes. No Massyve-Kit, esta tarefa será realizada através da entidade (aplicação) *Kit Manager*.

4.2. Lançamento do Sistema

Após a fase de derivação do sistema particular ter sido concluída com sucesso, torna-se possível lançar este sistema como um todo, iniciando a execução das entidades fundamentais *Address Server* e *Agents* (além da entidade DMP) que, juntamente com o *Kit Manager*, irão constituir o sistema em execução, e darão suporte à gestão do sistema multiagente lançado pelo Massyve-Kit.

Uma vez o sistema multiagente particular instanciado e apto a ser lançado, uma série de sub-etapas devem ser inicialmente executadas pelo *Kit Manager* para o lançamento deste sistema ser completado com sucesso. São elas: o salvamento do arquivo *Map*, inicialização do *Address Server*, salvamento / lançamento dos agentes e troca inicial de mensagens.

Após a conclusão das quatro sub-etapas iniciais, o Massyve-Kit está apto a suportar todas as interações entre os agentes da comunidade do sistema atual, tais como: criação, mudança e

exclusão de conexões entre agentes, entrada e saída de agentes do sistema e troca de mensagens em geral entre as entidades do sistema lançado.

4.3. Encerramento do Sistema

Esta é a terceira e mais simples fase do ciclo de vida de um sistema multiagente particular gerado pelo Massyve-Kit. Considerando-se que o sistema foi derivado e lançado corretamente, esta etapa será realizada com o objetivo de finalizar todos os agentes (utilizando-se a entidade *Agent Killer*) bem como o *Address Server* e os DMPs.

5. PRÓXIMOS DESENVOLVIMENTOS DO MASSYVE-KIT

Uma série de complementos ao sistema ora implementado foi identificada ao longo do seu desenvolvimento. Neste sentido, como sugestão para futuras implementações, pode-se citar alguns que são considerados como os mais interessantes do ponto de vista conceitual.

Pode-se citar o desenvolvimento de um ambiente interativo, nos moldes do Massyve-Kit, para a criação de agentes, configurando-se graficamente suas propriedades (de autonomia e independência), sem a necessidade de atividades de programação a partir de um código genérico já existente. Além disso, permitir aos agentes criados uma interoperabilidade com relação as mais diversas plataformas computacionais existentes atualmente.

Também surge como proposta para trabalhos futuros a definição de formas de comunicação entre os agentes através de múltiplos canais (portas), possibilitando a implantação da tecnologia de Sistemas Multiagente em aplicações que necessitem de grande capacidade na troca de informações, tais como aplicações multimídia. Uma API para uma utilização opcional do protocolo "*standard*" KQML [Finin, 94] é igualmente prevista.

Ainda, o desenvolvimento de um módulo opcional de suporte à recuperação e tolerância a falhas baseado em CORBA [CORBA] capaz de garantir a integridade de sistemas mais complexos de forma robusta e, ao mesmo tempo, facilmente configurável e gerenciável para usuários não especialistas em sistemas distribuídos e redes.

O Massyve-Kit encontra-se disponível na página oficial do projeto Massyve (<http://centaurus.dee.fct.unl.pt/~massyve>) para *download*.

6. CONCLUSÕES

Baseado em todo o estudo que foi realizado para o desenvolvimento do Massyve-Kit, a utilização do conceito de sistemas multiagente evidenciou a aplicabilidade desta tecnologia emergente em problemas do mundo real. O desenvolvimento de soluções de *software* abertas e ao mesmo tempo feitas sob medida têm surgido como uma das tendências mais importantes para a criação dos mais diversos ambientes computacionais.

Deve-se destacar também que os kits desenvolvidos até o momento não são ferramentas de fácil utilização e tentam ser ferramentas de propósito geral, tornando-se ambientes "pesados" demais, além de serem difíceis de instanciar e customizar. Muitos destes produtos estão na

fase de protótipo, devido ao fato da tecnologia MAS estar ainda em fase de amadurecimento, embora seja uma das áreas mais promissoras do momento.

De acordo com as características e tendências acima apresentadas, fica evidente o valor acrescentado com o desenvolvimento do Massyve-Kit, uma ferramenta de fácil utilização, leve, oferecendo não apenas um alto nível de conforto ao usuário, mas que permite uma rápida confecção de sistemas particulares, além de estar voltada primordialmente ao ensino e treinamento. Outro aspecto importante é que esta ferramenta é baseada na filosofia de derivação de sistemas, voltada para aplicações multiagente independentes de domínio. Apesar dos aspectos que essa ferramenta pode ser mais refinada e estendida, crê-se fortemente que ela oferece uma importante contribuição na área e favorece uma maior disseminação da tecnologia de sistemas multiagente.

AGRADECIMENTOS

O desenvolvimento deste projeto foi orientado pelo Prof. Ricardo Rabelo, do Departamento de Automação e Sistemas. Os autores agradecem ao CNPq (projeto Prodnet-II ProTeM-CC 680120/96-3) e à Comunidade Européia (projeto INCO-DC MASSYVE 962219) pelo suporte financeiro.

REFERÊNCIAS BIBLIOGRÁFICAS

- [Bradshaw, 97] BRADSHAW, Jeffrey M., *Software Agents*, MIT Press, pp.9-18, 1997.
- [Dietrich, 94] Dietrich, B. - *Automation in Manufacturing, Control versus Chaos*, in *Advances in Agile Manufacturing* (ed. P.T. Kidd and W. Karwowski), IOS Press, 1994.
- [Finin, 94] Finin, T.; Frizson, R. - *KQML - A Language and Protocol for Knowledge and Information Exchange*, Technical Report CS-94-02, Computer Science Department, University of Maryland.
- [Massyve, 98] <http://centaurus.dee.fct.unl.pt/~massyve>.
- [Rabelo, 97] Rabelo, R.J. - *A Framework for the Development of Manufacturing Agile Scheduling Systems – A Multi-agent Approach*, Ph.D. Thesis, New University of Lisbon, Portugal.
- [Shoham, 93] Shoham, Y. - *Agent-Oriented Programming*, Artificial Intelligence, N 60, pp-51-92, Elsevier, 1993.
- [Szelke, 94] Szelke, E.; Kerr, R. - *Knowledge-Based Reactive Scheduling*, Int. Journal of Production Planning & Control, Vol 5 N 2, Taylor & Francis, pp. 124-145, 1994.
- [CORBA] <http://www.omg.org>