

### 3 Trabalhos Relacionados

Ao procurar trabalhos relacionados, foi constatado que há pouca literatura confrontando tecnologias de desenvolvimento de software. No contexto de comparação de SMA com OO puro, a existência de trabalhos é ainda mais escassa, limitando-se a trabalhos que fazem comparações puramente qualitativas das características de arquiteturas desenvolvidas com essas tecnologias.

Na literatura existem alguns trabalhos que discutem comparações arquiteturais de outra natureza. Embora essas comparações envolvam tecnologias diferentes de SMA e OO, a metodologia de comparação utilizada nesses trabalhos é relevante para esse trabalho.

#### 3.1. Agentes e objetos

Um dos trabalhos que fazem comparações puramente qualitativas a respeito das características das tecnologias é o livro “*Uma introdução para sistemas multi-agentes*” de WOOLDRIDGE (2002). Esse livro visa introduzir os principais assuntos a respeito de sistemas multi-agentes, de forma a ser acessível para qualquer um que tenha algum conhecimento na área de Ciência da Computação.

No decorrer do livro o autor faz alguns comentários sobre as arquiteturas, mas na seção chamada “Agentes e objetos” são mencionadas as principais diferenças e igualdades das arquiteturas criadas ao usar essas tecnologias.

De acordo com o autor, primeiramente os agentes e os objetos se diferem no grau de autonomia. Os objetos conseguem manter uma autonomia sobre o próprio estado através do encapsulamento. Entretanto, os objetos não apresentam autonomia sobre o próprio comportamento, ou seja, um objeto não pode controlar quando e em que ordem serão utilizados os seus métodos. No caso dos agentes, uma ação pode ser requisitada a um agente, mas o agente requisitado pode decidir agir ou não. Assim, a decisão sobre quem irá ou não executar uma ação no caso da OO é de quem irá invocar o método e no caso da OA é do agente que recebe a requisição.

Segundo WOOLDRIDGE, não há nada que impeça implementar agentes utilizando técnicas orientadas a objetos, um exemplo de aplicação é o framework JADE (2002) utilizado nesta dissertação. Por exemplo, podem ser criados mecanismos de decisão que determinam se irão ou não executar um método do próprio objeto. No entanto, o que o autor quer dizer é que, esse tipo de autonomia não é um componente básico do modelo OO.

Outra distinção importante entre agentes e objetos comentada no livro é a noção de exercer um comportamento autônomo flexível (reativo, pró-ativo, social). O modelo padrão de objetos não diz nada a respeito desses comportamentos. Contudo, como dito anteriormente um sistema OO pode ser desenvolvido para integrar esses tipos de comportamentos, mas o modelo base da OO não lida com esses aspectos naturalmente.

O autor cita uma terceira distinção importante entre o modelo básico de objetos e agentes no livro. Essa diferença é que cada agente tem sua própria *thread* de controle, enquanto no modelo de objetos padrão de maneira geral existe uma única *thread* que controla o sistema.

WOOLDRIDGE resume as diferenças entre as abordagens em:

- Agentes incorporam uma noção de autonomia mais forte do que os objetos, e, em particular, eles decidem por si mesmos se devem ou não realizar uma ação, a pedido de outro agente;
- Agentes são capazes de exercer um comportamento flexível (reativo, pró-ativo, social), e o modelo padrão de objeto não tem nada a dizer sobre esses tipos de comportamento;
- Um sistema multiagentes é inerentemente *multi-threaded*, no qual cada agente deve ter pelo menos uma *thread* de controle.

Uma característica importante que não foi mencionada explicitamente nesse trabalho, mas alguns autores comentam, como pode ser visto no próximo trabalho relacionado, é a possibilidade da OO estar em uma camada inferior à OA. Esse fato é possível de se observar em implementações de sistemas considerando camadas de interpretadores (BERRY, 1978). Uma analogia é a camada procedural, que define o comportamento de um computador, ser inferior à camada OO. Assim, as duas tecnologias não poderiam ser comparadas no nível de código, mas sim no nível de esforço necessário para implementar algo. A OA traz consigo diversas características que na OO necessitariam de um esforço adicional para serem implementados. No entanto, a OA impõe uma série de regras e protocolos de comunicação que não são necessários na OO.

### 3.2. Objetos e agentes comparados

Outro trabalho no contexto de comparação de agentes e orientação a objetos é o de ODELL (2002). Inicialmente ODELL apresenta uma evolução das linguagens de programação e nela a abordagem orientada a agentes é uma evolução da programação orientada a objetos. Segundo o autor, objetos são considerados passivos, pois os métodos são invocados somente quando uma entidade externa envia uma mensagem para o objeto. Já agentes são como “objetos ativos com iniciativa”, pois eles decidem quando e como vão agir.

De acordo com ODELL, um dos conceitos chave dos agentes é a autonomia. No entanto, tal autonomia é melhor caracterizada em graus, ao invés de simplesmente estar presente ou não. Dessa forma, um agente pode ter algum grau de comportamento desde ser simplesmente reativo até ser inteiramente pró-ativo. Assim, agentes podem não só reagir às invocações de método, como aos eventos observáveis gerados pelo meio. Já agentes pró-ativos irão pesquisar no meio por eventos e outras mensagens para determinar que ação o agente deve tomar.

Objetos, por outro lado, são convencionalmente passivos, com seus métodos sendo chamados pela *thread* de controle. Entretanto, UML e Java introduziram frameworks para observar eventos e outros mecanismos permitindo objetos a serem pró-ativos. Dessa forma, os objetos têm atualmente algumas habilidades similares às dos agentes.

Segundo ODELL, a interação envolve a habilidade de se comunicar com o meio e com outras entidades e tal interação também pode ser expressa em graus. Desse modo, a interação pode variar de uma invocação de método, que é a forma mais básica de interação, até a um sistema multi-agentes com a capacidade de se comunicar e observar eventos gerados pelo ambiente. Assim, o autor associa os objetos à forma mais simples de interação que é a invocação de um método e os agentes à forma mais complexa de interação que é a interação em sociedade.

Posteriormente, ODELL apresenta outras formas nas quais agentes podem empregar e desse modo se diferenciar de objetos. Entretanto, nenhuma das formas é universalmente usada por agentes e, além disso, sistemas com objetos ativos podem implementar esses meios também. A seguir são apresentados alguns conceitos apresentados pelo autor:

- Descentralização – Objetos podem ser tanto pensados quanto como organizados de forma central, pois os métodos dos objetos

são invocados por outros componentes no mesmo sistema. Entretanto, ambientes baseados em agentes podem empregar tanto processamento centralizado como descentralizado, provendo um resultado em computação distribuída.

- Pequeno no impacto – Se um objeto é perdido em um sistema, uma exceção é lançada. Em um SMA, devido à facilidade de replicação de agentes, a perda de um agente pode causar um impacto menor no sistema, similar à perda de uma formiga para uma colônia.
- Emergência – Objetos não interagem entre si sem um alto nível de controle, dessa forma emergência não ocorre normalmente. Já a interação entre muitos agentes pode revelar efeitos secundários que podem ser desejados para o sistema.

Segundo o autor, agentes podem ser vistos como objetos com alguns comportamentos adicionados, como por exemplo, mobilidade, inferência, etc. ODELL complementa essa visão supondo que um agente potencialmente perfeito poderia ter todos os comportamentos na lista de atributos do agente. E agentes degradados seriam aqueles que tivessem menos do que todos os atributos. Dessa forma, um objeto seria um agente degradado fazendo tudo o que lhe é pedido. Assim, o autor argumenta que agentes podem ser objetos e que agentes podem ser diferentes de objetos, simultaneamente. Além disso, essas habilidades extras dos agentes podem ser adicionadas nos objetos, no entanto esses objetos com as habilidades se tornariam agentes. Dessa forma, objetos podem encapsular e simular as habilidades nativas dos agentes se tornando *agent-based*.

O autor conclui que objetos e agentes são suficientemente distintos para serem tratados diferentemente e que sistemas podem ser projetados com uma mistura de ambas as abordagens. Além disso, o autor defende que seja usada uma camada adicional baseada em objetos para sistemas multi-agentes. Essa camada serviria de infra-estrutura e iria fortalecer o suporte a agentes nesse tipo de sistema.

Como comentado anteriormente o framework JADE (2002) implementa agentes através da OO. No entanto, o desenvolvimento de sistemas utilizando essa abordagem também pode ser medida em graus. Ou seja, poderíamos utilizar JADE para construir um sistema meramente OO ou extremamente OA. Esse fato é definido pela arquitetura projetada para o sistema e como ela é desenvolvida utilizando o framework, no caso deste trabalho JADE. Nesta

dissertação, foi utilizado o guia de implementação recomendado pelo framework de modo que o sistema tivesse um grau elevado de OA.

Nesse trabalho, algumas diferenças apresentadas pelo autor podem ser questionadas. Um exemplo é o impacto no funcionamento do sistema. Em um sistema OA esse impacto não necessariamente é menor do que em um sistema OO. Existem diversas implementações de agentes que executam papéis de controle, de *logging*, de acesso a dados, entre outros. Nesses casos os sistemas poderiam perder o controle, não fazerem mais log da aplicação e perderem o acesso a dados.

Os dois trabalhos apresentados, ODELL (2002) e WOOLDRIDGE (2002), fazem comparações puramente qualitativas a respeito das arquiteturas resultantes das duas tecnologias. Dessa forma, a escolha do uso de uma ou outra ou das duas arquiteturas se torna subjetivo. Não são apresentados os benefícios reais das arquiteturas ou o porquê uma ou outra ou as duas deveriam ser utilizadas e em quais domínios seriam melhores aproveitadas.

### **3.3. Engenharia de SMA com Aspectos e Padrões**

Os objetivos do trabalho de GARCIA (2002) foram apresentar um método de desenvolvimento de SMA, que explore os benefícios de usar o desenvolvimento orientado a aspectos (DOA); propor um método baseado em padrões para o desenvolvimento de SMA; e a comparação desses métodos avaliando as vantagens e desvantagens relativas em relação à facilidade de compreensão, à facilidade de reutilização, e à manutenibilidade de SMA.

Esse trabalho não faz comparações entre SMA e OO puro, mas envolve as duas arquiteturas e é relevante no âmbito da comparação em si. Esta dissertação se resume a apresentar a comparação presente no trabalho de GARCIA. Para mais detalhes sobre o trabalho vide GARCIA (2002).

Para GARCIA sistemas multi-agentes envolvem múltiplos agentes e objetos, os quais são abstrações distintas usadas para modelar diferentes entidades do domínio do problema. E segundo os autores, agentes são entidades ativas que manipulam objetos e esses objetos são geralmente aplicados para representar entidades passivas no sistema.

No trabalho, os autores utilizaram uma linguagem de programação OO para implementar os agentes. Dessa forma, houve a necessidade de mapear os projetos orientados a agentes em projetos orientados a objetos. No entanto, em nível arquitetural a orientação a agentes foi mantida.

Os autores dividiram o estudo de caso em três fases, realizadas por dois times em paralelo, no qual cada um foi responsável pelo desenvolvimento utilizando uma das técnicas. A primeira fase consistia na construção inicial do sistema. A segunda fase foi composta da modificação subsequente a partir de novos requisitos. Na terceira fase funcionalidades existentes deveriam ser reutilizadas em novos contextos.

Foi criado um processo de medição que considerava critérios qualitativos e quantitativos. As principais propriedades observadas no trabalho foram a facilidade de escrita, a facilidade de entendimento, a facilidade de reutilização e a manutenibilidade.

Os resultados encontrados foram que em todos os critérios analisados a abordagem baseada em aspectos teve desempenho melhor do que a abordagem baseada em padrões. A seguir são apresentados os critérios analisados:

- O método orientado a aspectos suporta uma melhor facilidade na escrita,
- O método orientado a aspectos suporta uma melhor facilidade na reutilização,
- O método orientado a aspectos suporta uma melhor evolução,
- O método orientado a aspectos suporta uma melhor expressividade,
- O método orientado a aspectos suporta uma melhor flexibilidade para acomodar definições distintas de agência.

Com a construção do estudo de caso e a comparação dos sistemas construídos a partir de métodos diferentes, os autores puderam concluir que os mecanismos de composição do DOA provêm um melhor suporte para o tratamento da complexidade dos interesses de agência, produzindo um SMA mais fácil para entender, manter e reutilizar.

Esse trabalho apresenta um ponto positivo em relação aos apresentados. Nesse trabalho foi feita uma avaliação realística e sistemática, dessa forma foi possível verificar os benefícios e desvantagens das técnicas estudadas. Os outros trabalhos se concentram em apresentar os conceitos chaves de cada abordagem, enquanto o trabalho de GARCIA conclui que uma técnica é melhor que a outra. Entretanto, a avaliação foi realizada de acordo com a perspectiva de desenvolvedores de SMA de forma qualitativa. Essa forma de avaliação é útil, mas pode ser subjetiva.

### 3.4. Agentes e Objetos: Um Estudo Empírico no Design e Implementação de SMA

O trabalho de GARCIA (2003) faz a comparação dos mesmos sistemas de desenvolvimento de (GARCIA, 2002), mas de modo empírico e usando a metodologia GQM (BASILI, 1994) assim como foi feito nesta dissertação. Esse trabalho foi planejado como um experimento mais controlado, os autores conduziram a avaliação através de um modelo de qualidade que se focou em dois critérios, a manutenibilidade e o reuso.

Segundo os autores, poucos estudos empíricos na literatura investigam as interações entre abstrações orientadas a agentes e OO puro e abstrações emergentes. Dessa forma, os autores apresentaram um experimento que faz o uso sistemático de duas técnicas OO diferentes para o desenvolvimento de SMA. Nesse experimento, foram aplicadas um conjunto de métricas de forma a prover um melhor entendimento das interações entre abstrações OO e OA. Assim, foi possível avaliar os pontos fortes e fracos das duas técnicas, suas compatibilidades e divergências.

O objetivo principal do GQM definido pelos autores nesse trabalho foi:

Tabela 1 - Objetivo do GQM em (GARCIA, 2003)

Avaliar a facilidade do reuso e a facilidade de evolução dos sistemas multi-agentes implementados de forma a comparar o desenvolvimento orientado a objetos com o desenvolvimento orientado a aspectos.

As perguntas do GQM desenvolvidas nesse trabalho formaram um conjunto hierárquico, que partia de perguntas mais abstratas e mais ligadas ao objetivo até serem refinadas se tornando perguntas mais concretas e próximas da implementação. A seguir são apresentadas algumas perguntas:

Tabela 2 - Perguntas do GQM em (GARCIA, 2003)

1. O quão fácil é evoluir o sistema?
  - 1.1. O quão fácil é entender o sistema?
    - 1.1.1. Quantos componentes existem?
    - 1.1.2. Quantas linhas de código existem?
    - 1.1.3. Quantos atributos existem?

Com o objetivo e as perguntas definidas os autores associaram as perguntas às métricas. Os autores reutilizaram algumas métricas clássicas e definiram um novo conjunto que de acordo com GARCIA capturaria noções importantes no contexto do estudo. A seguir, são apresentadas as métricas

usadas pelos autores e pelo menos uma pergunta que seria respondida pela métrica.

Tabela 3 - Perguntas e métricas em (GARCIA, 2003)

<b>Métricas</b>	<b>Perguntas</b>
Tamanho do Vocabulário (VS)	Quantos componentes existem?
Linhas de Código (LOC)	Quantas linhas de código existem?
Número de Atributos (NOA)	Quantos atributos existem?
Peso dos Métodos por Classe (WMC)	Quantos métodos e <i>advices</i> existem?
Acoplamento entre componentes (CBC)	O quão alto é o acoplamento entre os componentes?
Falta de Coesão nos Métodos (LCOM)	O quão alto é a coesão entre os componentes dos sistemas?
Profundidade na Árvore de Herança (DIT)	O quão alto é o acoplamento entre os componentes?
Difusão de Interesses sobre Componentes (CDC)	Como estão localizados os interesses de agência?
Difusão de Interesses sobre Operações (CDO)	Como estão localizados os interesses de agência?
Difusão de Interesses sobre LOC (CDLOC)	Como estão localizados os interesses de agência?

Além do método GQM, foi criado um framework de avaliação composto de um modelo de qualidade. De acordo com os autores esse modelo é útil para a seleção das métricas e para a interpretação dos dados coletados. Nesta dissertação, em alguns momentos esse modelo de qualidade foi utilizado para a interpretação dos dados coletados.

GARCIA descreve o modelo de qualidade contendo quatro elementos diferentes: qualidades, fatores, critérios e métricas. As qualidades são os atributos que eles desejam primeiramente observar no sistema. Os fatores são atributos secundários de qualidade que influenciam as qualidades primárias. Os critérios são relacionados a princípios de engenharia de software bem estabelecidos que são necessários para a realização das qualidades e dos respectivos fatores.

No modelo de qualidade desenvolvido, os autores definem duas qualidades principais, a facilidade no reuso e a manutenibilidade. Essas qualidades são influenciadas positiva ou negativamente por quatro fatores, a facilidade na compreensão, a flexibilidade, a facilidade de modificação e a facilidade de extensão. Além disso, os autores relacionam a flexibilidade com



três critérios, a separação de interesses, o acoplamento e a coesão. Já a facilidade na compreensão é influenciada pelos mesmos critérios da flexibilidade mais o critério de tamanho.

Com o modelo de qualidade e o plano GQM definidos, os autores passaram para a fase de coleta e interpretação dos dados. Os dados coletados foram analisados e justificados. Na Tabela 4 são apresentadas as conclusões de cada métrica.

Tabela 4 - Resultados das Métricas

<b>Métricas</b>	<b>Conclusões</b>
VS	O SMA orientado a aspectos é mais simples do que o orientado a objetos.
LOC	A implementação OP necessitou de mais linhas de código do que a OA.
NOA	A solução OP necessitou de mais componentes do que a solução OA.
WMC	Em geral a solução OA apresentou componentes mais complexos do que a solução OP.
CBC	A solução OP apresentou componentes mais acoplados do que a solução OA.
LCOM	A solução OP apresentou componentes mais coesos do que a solução OA.
DIT	A solução OP necessitou de hierarquias mais complexas do que a solução OA.
CDC	A solução OP necessitou de mais componentes para a definição dos interesses do SMA do que a solução OA.
CDO	A solução OP necessitou de mais operações para a definição dos interesses do SMA do que a solução OA.
CDLOC	A solução OP necessitou de mais linhas de código para a definição dos interesses do SMA do que a solução OA.

Dessa forma, de acordo com os resultados, a abordagem orientada a aspectos forneceu um suporte melhor para a separação dos interesses do SMA, além disso, produziu um SMA mais conciso e com um menor acoplamento do que a abordagem baseada em padrões. Entretanto a abordagem de padrões produziu um SMA com maior coesão do que a abordagem baseada em aspectos.

Como os objetivos do trabalho eram avaliar o reuso e a evolução dos sistemas, os autores criaram cenários de reuso e evolução de forma que fossem

medidas a facilidade de modificação e a facilidade de extensão dos sistemas, conforme comentado no modelo de qualidade. Assim, em cada cenário foram medidas as seguintes ações:

- Entidades modificadas,
- Operações modificadas,
- Entidades adicionadas,
- Operações adicionadas,
- Relacionamentos modificados,
- Relacionamentos adicionados,
- Linhas de código adicionadas,
- Linhas de código modificadas,
- Entidades copiadas,
- Linhas de código copiadas.

De acordo com GARCIA, em termos dos cenários de reuso e evolução, a técnica orientada a aspectos apresentou melhores resultados do que a técnica orientada a padrões. Essa conclusão foi suportada pelas métricas de entidades modificadas, relacionamentos modificados, relacionamentos adicionados, linhas de código adicionadas, e linhas de código copiadas.

Dessa forma, os autores concluem que a técnica OA permitiu a construção de um SMA reativo com separação dos interesses melhorada, com baixo acoplamento entre os componentes, mais conciso, mais fácil de ser reusado mais manutenível. Além disso, Gacial et al. argumentam que esse é um primeiro estudo exploratório, para melhor confirmar as descobertas, outros experimentos rigorosos e controlados são necessários.

Esse trabalho relacionado é o que se aproxima mais do que é proposto nesta dissertação. Nesse trabalho é feito um estudo empírico através da metodologia GQM, esse estudo tem a finalidade de comparar duas técnicas a fim de verificar os benefícios e as desvantagens de cada uma. Além disso, esse trabalho apresenta um modelo de qualidade, que, como dito anteriormente, em alguns momentos, esse modelo de qualidade foi utilizado para a interpretação dos dados coletados nesta dissertação.

Para esta dissertação esse trabalho não contribui a respeito das técnicas comparadas, mas como são utilizadas metodologias parecidas, esta dissertação pode ser considerada como uma extensão do trabalho de GARCIA incluindo uma comparação entre outras duas técnicas, um modelo de medição adaptado e uma discussão sobre a utilização da metodologia GQM em um projeto real.

Como comentado pelos autores, outros experimentos são necessários, pois este estudo parte do pressuposto que a facilidade de escrita, a facilidade de compreensão e outras qualidades se restringem ao número de linhas de código, ao vocabulário e outros atributos. Nenhum experimento foi feito para avaliar se de fato um texto orientado a aspectos é mais fácil de compreender e escrever do que o mesmo texto utilizando padrões de projeto, ou se é utilizado mais esforço para usar uma ou outra técnica. Além disso, na manutenção dos sistemas atributos como taxa de inserção de defeitos devem ser considerados, pois pode ser fácil fazer uma alteração usando uma técnica, mas isso não indica o quão confiável foi essa alteração.