

## 6 Conclusões e Trabalhos Futuros

Apesar da engenharia de software ter evoluído no sentido de mostrar que a modelagem é ponto chave para o sucesso de um projeto de sistemas, ainda há quem diga que esta é a principal causa de atrasos em projetos.

Com o aumento da pesquisa em engenharia de software orientada a agente aumentaram também as necessidades deste grupo e isto obriga que se tenham novos métodos e adaptações no que diz respeito à modelagem e a implementação.

Esta dissertação apresenta o método GP-MVAP que possibilita modelar SMA de maneira mais flexível de forma que fica claramente visível à linha de produto de tal sistema, o que possibilita o desenvolvimento e a modelagem de *frameworks*.

Um outro ponto importante é o fato do método GP-MVAP não ter nenhuma dependência com meta-modelos ou algo feito anteriormente e com isto a solução proposta pode ser facilmente acoplada a linguagens de modelagem de agentes, sendo utilizada como uma extensão, com o propósito de tornar possível a representação de pontos que antes não eram expostos por tais linguagens. Isto ocorre porque as linguagens de modelagem de agentes existentes não prevêm uma estrutura de flexibilização situada nos planos e ações dos agentes; outra questão é que as linguagens baseadas em OO não são adequadas para representação de agentes acarretando problemas semânticos.

A solução descrita aqui na dissertação visa solucionar o problema de representação, através de diagramas e técnicas de documentação e por meio destes será possível representar varias instâncias para um sistema multi-agente.

A notação proposta traz uma forma simples de representação, através da qual é possível mostra que um plano pode herdar características de um outro e através desta idéia será possível representar as possíveis instâncias para um dado escopo. Com as ações ocorre algo semelhante já que estas também poderão herdar características de uma dada ação, porém suas instâncias terão maneiras distintas

de serem executadas, em função das distintas formas que uma aplicação tem de ser executada.

Para tanto o método GP-MVAP oferece diagramas e *tags* para elucidar todas as características do sistema.

Uma outra questão abordada é que o método GP-MVAP também oferece ao responsável pela modelagem uma técnica de documentação a qual serve como guia para as futuras instâncias da aplicação.

## **6.1. Contribuições**

A dissertação apresenta uma abordagem para a modelagem de agentes de software, podendo ser também utilizada juntamente com as linguagens de modelagem.

Como contribuição acredita-se que a dissertação venha trazer uma forma de nova de modelar agente de software que possibilitará a representação de pontos de flexibilização de um agente num dado sistema que conseqüentemente serão pontos de flexibilização do próprio sistema. Os pontos de flexibilização de um agente de software poderão ser seus planos e as ações, os quais dependerão de características do sistema para serem utilizados como tal. Assim os SMA que podem ser utilizados em vários escopos distintos, ou seja, podem ter instancias distintas, terão suas representações através de seus pontos de flexibilização, ou seja, através de seus planos e ações. Com o método GP-MVAP será possível representar todos os possíveis escopos para uma aplicação baseada em SMA.

A modelagem de tais pontos traz a possibilidade de ter um agente com um *framework* onde os planos e as ações são utilizados como *hotspots* o que resultaria na instância de uma aplicação. Desta forma, cada aplicação teria a sua implementação de planos e ações.

A dissertação contribui com os seguintes pontos:

- O método GP-MVAP possibilita que o sistema multi-agente tenha uma flexibilidade documentada o que facilitará as futuras modificações;

- Com o método GP-MVAP o sistema terá uma melhor adaptação ao processo incremental;
- A proposta dá ao sistema uma possibilidade de reutilização do design (facilidade de manutenção);
- A proposta permite que seja dada uma maior atenção na área específica;
- O método GP-MVAP pode ser facilmente utilizado de modo concomitante com outras linguagens de modelagem de agentes;
- A proposta traz uma técnica de documentação que permitirá que o desenvolvedor tenha um guia para novas instâncias;
- O uso da idéia de *framework* para agentes de software, trazendo características de re-uso.

## 6.2. Trabalhos Futuros

Alguns pontos não foram abordados nesta dissertação e ficarão para pesquisas futuras, a seguir serão apresentados tais pontos.

Uma questão que é de grande valia é a modelagem completa do sistema *LearnAgents* com o método GP-MVAP, mostrando que dessa forma todo o sistema (e não só um agente) pode ser beneficiado, já que existem vários escopos a serem tratados os quais podem uma linha de produto com varias aplicações.

Um segundo passo para pesquisas futuras será o desenvolvimento de um script para facilitar a geração das instâncias. Desta forma, será possível automatizar e otimizar um processo que muitas das vezes gera problemas já que o responsável pela modelagem geralmente não é o responsável pelo desenvolvimento.

Após resolver todas estas questões o terceiro passo será dar continuidade as pesquisa de agentes aplicados à área médica e certamente utilizando todas as pesquisas feitas anteriormente.

Uma questão para o futuro é que com a utilização do método GP-MVAP podem surgir problemas relacionados à integração de *framework*, assim como inversão de controle, integração com sistema legado e ferramentas existentes, integração de funcionalidades entre diferentes *frameworks*, etc. como acontece em

OO, uma abordagem destes problemas em OO pode ser vistos em (Mattsson et al., 1999) e será pensada para sistemas baseados em agentes de software.