

7

Conclusão

Simulações computacionais são ferramentas importantes em diversos setores. Dentre as simulações, as de Ambientes Virtuais (VES) tem aplicabilidade especial para jogos sérios e em entretenimento.

O uso de VES em jogos sérios permite realizar treinamentos, testes de planejamento, prospecção de cenários, entre outros, oferecendo excelente relação de custo-benefício.

O uso de simulações baseadas em sistemas multiagentes é considerado promissor para implementar estas simulações. Contudo, o ferramental atual impõe o avanço do tempo de simulação de forma centralizada. Esta decisão restringe a modelagem dos atores do sistema e dos comportamentos como agentes de simulação autônomos, na medida que o tempo é um fator que também deve ser modelado.

A implementação do controle de avanço do tempo de simulação de forma individual por agentes é afetada diretamente por questões relacionadas a tecnologia Java, a qual não foi desenhada para atender o desenvolvimento de sistemas com características de *real-time* e possui *tardiness* por construção.

Esta tese apresentou uma abordagem para tratar este problema. Uma vez que não se verificou a existência de trabalhos anteriores, foi realizada pesquisa sobre as teorias relacionadas ao tema, desenvolvida uma plataforma de testes e realizado um experimento para comprovar o problema e sua relação com abstrações de sistemas multiagentes.

A abordagem proposta para a solução da questão principal de pesquisa apresenta uma arquitetura conceitual e um algoritmo que executa a tarefa de controle do *tardiness* no sistema. Ambos são resultado de experimentação e desenvolvimento incremental desta pesquisa, não se apoiando em trabalhos anteriores.

Os resultados dos experimentos, realizados com a arquitetura instanciada, indicam o sucesso da abordagem. Contudo, admite-se que o algoritmo de controle apresentado não é ótimo e depende de correta parametrização para ser eficiente. Dentro do cenário em que as ferramentas matemáticas de predição para sistemas estocásticos mostram-se limitadas, os resultados foram considerados satisfatórios, conforme apresentado nos resultados quantitativos do capítulo 6.

Abaixo destacam-se as contribuições do trabalho, na visão do autor.

7.1

Contribuições

7.1.1

Desenvolvimento de plataforma para experimentos

A plataforma MASP, apresentada no capítulo 3, foi desenvolvida para apoiar realização de experimentos com medição de *tardiness* em tempo de execução, preenchendo uma lacuna necessária à continuação da pesquisa.

A plataforma usa um mecanismo de inferência sobre o modelo conceitual representado em ontologia, que permite rápida instanciação do sistema, além de implementar a arquitetura proposta para o controle de tempo de simulação e a representação espacial.

Este trabalho foi divulgado no artigo (Taranti et al., 2010), que apresenta uma versão inicial do MASP. O uso de ontologia em associação a técnicas de injeção de dependência utilizada foi desenvolvida para rápida instanciação de simulações, porém a aplicação não é restita a este campo.

7.1.2

Realização de experimento estatístico para estabelecimento de parâmetros

O experimento apresentado no capítulo 4 estabelece relação entre abstrações de agentes em um SMA e o nível de *tardiness*, especificamente o número de agentes em execução; o período entre execuções e a duração da computação.

Esta informação, que deriva de um experimento realizado com grande massa de dados, pode apoiar o desenvolvimento de SMA e modelagem de testes em tempo de execução para verificação da estabilidade do sistema. Os resultados preliminares foram publicados em (Taranti et al., 2010a) e um resumo do experimento final em (Taranti et al., 2011).

7.1.3

Definição de uma arquitetura conceitual para solução e sua implementação

O capítulo 5 apresentou a solução proposta para a principal questão de pesquisa, instanciada sob a forma de uma arquitetura conceitual. A arquitetura atende as questões secundárias e a principal, permitindo o desenvolvimento de MABS cujo o avanço do tempo de simulação possua controle individualizado por agente. A arquitetura suporta simulações cujo avanço do tempo de simulação seja realizado ao próximo evento ou por passos, suportando agentes que adotem abordagens diferentes para seus comportamentos de forma simultânea. A arquitetura foi evoluída e testada para permitir simulações georeferenciadas.

O algoritmo desenvolvido merece destaque, pois sugere uma atende uma lacuna existente nas ferramentas matemáticas disponíveis para predição de valores em sistemas estocásticos em que o erro não é branco e quando não se pode garantir comportamento linear, ainda que local. A aplicabilidade de soluções para este problema ultrapassa a área de simulação.

Os experimentos realizados para validar a solução foram apresentados no capítulo 6, sendo que estes resultados foram também divulgados nos artigos (Taranti et al., 2011b; Taranti et al., 2011a) e na Monografia (Taranti et al., 2012).

7.2

Trabalhos futuros

O trabalho apresentado nesta tese explora questões não abordadas na literatura. A solução apresentada não pretende ser definitiva, pois admite-se que não é ótima. Ao longo da pesquisa, diversos trabalhos foram interrompidos a fim de manter o foco na questão principal de pesquisa, e assim cumprir o cronograma estabelecido. Abaixo, listam-se tópicos de trabalhos futuros que podem ser desenvolvidos a partir do estágio atual desta tese e dos trabalhos decorrentes da mesma:

Uso de ontologias para injeção de dependência

A técnica desenvolvida para instanciar os modelos de execução apresenta o potencial do uso de ontologias para modelar o comportamento de sistemas. O custo computacional é alto, devido a explosão combinatória decorrente da carga da ontologia inferida em memória. Contudo, este custo ocorre apenas no momento da instanciação do sistema.

A técnica apresenta a possibilidade de estabelecer diversos relacionamentos sobre o grafo de objetos, permitindo estabelecer redes complexas de objetos e suas dependências em uma linguagem computável, ie. OWL-DL.

Estas potencialidades podem ser exploradas no campo da Engenharia de Software sob diversos enfoques, tais como testes, sistemas multiagentes e modelagem de domínios.

Plataforma MASP

A plataforma foi desenvolvida ao limite da necessidade para esta pesquisa. Modificações na mesma e um *refactoring* arquitetural permitiriam que a mesma fosse aplicada a simulações reais.

A plataforma também carece de interface gráfica e um mecanismo automatizado de análise de dados. O suporte a visualização de resultados em tempo de execução também é desejável.

Resultados do experimento qualitativo

Os resultados apresentados no capítulo 4 foram utilizados para a pesquisa apresentada, contudo possuem aplicação mais ampla: os resultados podem apoiar decisões de implementação e manutenção de sistemas em tempo de execução. Existe carência de estudos sobre aplicação de sistemas multiagentes em domínios que possuem natureza de tempo-real.

Arquitetura conceitual para o relógio da simulação

A arquitetura foi implementada sobre o MASP, pois era necessário realizar a tarefa de forma controlada, com suporte à coleta de dados em tempo de execução para controlar os resultados no curso da pesquisa.

É desejável que a arquitetura do relógio de simulação seja utilizada para estender uma plataforma madura, como a Mason ou a Swarm, a fim de validar a abordagem nestas. Este trabalho permitiria refinar a arquitetura proposta.

Algoritmo de controle do avanço do tempo de simulação

O algoritmo apresentou comportamento promissor, contudo não restam dúvidas que o mesmo pode ser refinado, seja para o caso geral, seja para situações específicas. São alguns exemplos de tarefas que se vislumbra neste campo:

- ajuste automático dos parâmetros do algoritmo;
- substituição do algoritmo por outra abordagem;
- ajustes no algoritmo atual; e
- especialização do algoritmo para simulações específicas, tais como simulações de apoio a jogos de guerra, para obtenção de resposta mais eficiente nestes casos.

Outra evolução possível é a avaliação de uso de algoritmo semelhante em sistemas de controle aonde o uso de filtros de Kalman, por exemplo, não se adequam devido as características de não normalidade do erro ou pelo fato do erro não possuir uma distribuição branca. São exemplos de aplicações desta natureza, na área de computação:

- distribuição de carga em SMA através de detecção de falta de recursos e migração de agentes para o consequente balanceamento; ou
- distribuição de carga em centros de dados através de migração automática de serviços ou máquinas virtuais.

Versões amadurecidas desta abordagem de controle podem ser avaliadas para sistemas de controle em engenharia de sistemas.