



**Pier–Giovanni Taranti**

**Uma arquitetura para controle de atrasos de  
tempo em simulações baseadas em sistemas  
multiagentes**

**Tese de Doutorado**

Tese apresentada ao Programa de Pós-graduação em  
Informática do Departamento de Informática da PUC–Rio  
como requisito parcial para obtenção Do título de Doutor em  
Informática

Orientador: Prof. Carlos José Pereira de Lucena

Rio de Janeiro  
Março de 2013

**Pier–Giovanni Taranti**

**Uma arquitetura para controle de atrasos de  
tempo em simulações baseadas em sistemas  
multiagentes**

Tese apresentada ao Programa de Pós-graduação em  
Informática do Departamento de Informática do Centro  
Técnico Científico da PUC–Rio como requisito parcial para  
obtenção Do título de Doutor em Informática. Aprovada pela  
Comissão Examinadora abaixo assinada.

**Prof. Carlos José Pereira de Lucena**

Orientador

Departamento de Informática — PUC–Rio

**Prof. Bruno Feijó**

Departamento de Informática – PUC-Rio

**Prof. Helio Côrtes Vieira Lopes**

Departamento de Informática – PUC-Rio

**Prof. Pierre Bommel**

CIRAD - França

**Prof. Ricardo Choren Noya**

Seção de Engenharia de Computação – IME

**Prof. José Eugenio Leal**

Coordenador Setorial do Centro Técnico Científico —  
PUC–Rio

Rio de Janeiro, 27 de Março de 2013

Todos os direitos reservados. É proibida a reprodução total ou parcial do trabalho sem autorização da universidade, do autor e do orientador.

### **Pier–Giovanni Taranti**

O autor obteve o título de Mestre em Sistemas e Computação no Instituto Militar de Engenharia (IME), em 2007, na cidade do Rio de Janeiro, RJ, e possui o título de Bacharel em Ciências Náuticas, com habilitação em eletrônica, pela Escola Naval, em 1995, na cidade do Rio de Janeiro, RJ. Possui experiência como gerente projetos e arquiteto de sistemas, sendo seus principais interesses de pesquisa: engenharia de software aplicada a sistemas multiagentes, simulações, e confiabilidade de sistemas.

#### Ficha Catalográfica

Taranti, Pier–Giovanni

Uma arquitetura para controle de atrasos de tempo em simulações baseadas em sistemas multiagentes / Pier–Giovanni Taranti; orientador: Carlos José Pereira de Lucena. — Rio de Janeiro : PUC–Rio, Departamento de Informática, 2013.

153 f: il. (color.); 30 cm

1. Tese (doutorado) - Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Informática.

Inclui referências bibliográficas.

1. Informática – Tese. 2. Simulação Baseada em Sistemas Multiagentes;. 3. Tardiness;. 4. Tempo de Simulação.. I. Lucena, Carlos José Pereira de. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. III. Título.

CDD: 004

## Agradecimentos

Ao meu orientador, Professor José Carlos Pereira de Lucena pelo apoio, compreensão e conselhos, sem os quais seria impossível a realização deste trabalho.

À PUC–Rio, pelo auxílio concedido e que viabilizaram minha participação no curso.

A todo o pessoal do departamento de Informática pela ajuda e apoio.

Aos meus colegas da PUC–Rio, que sempre mantiveram um ambiente descontraído e voltado a pesquisa, dispostos a ajudar quando necessário.

Aos meus professores e colaboradores, cujo conhecimento, esforço e dedicação são exemplos para mim. Em especial agradeço ao Professor Ricardo Choren, meu orientador de mestrado e que me auxilia desde o término deste, em 2007.

À aqueles que mais me são queridos: minha esposa Raquel, e meus filhos Júlia e Arthur. Por mais que nos esforcemos, não há como evitar as ausências decorrentes da participação em um curso de doutorado. Conciliar os estudos com a rotina familiar é, por vezes, doloroso. Suas compreensão e amor foram fundamentais neste período.

Finalmente, agradeço a Deus, pelo dom da vida e pelas graças que me foram concedidas neste período conturbado.

## Resumo

Taranti, Pier–Giovanni; Lucena, Carlos José Pereira de. **Uma arquitetura para controle de atrasos de tempo em simulações baseadas em sistemas multiagentes**. Rio de Janeiro, 2013. 153p. Tese de Doutorado – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Simulações de Ambientes Virtuais (VES) são um tipo especial de simulação, normalmente utilizado para implementar jogos ou jogos sérios com representação em espaço virtual e utilizando a técnicas de avanço do tempo de simulação para o próximo evento ou dirigida por tempo. Um exemplo de uso é a aplicação em simulações de apoio a Jogos de Guerra. O uso de Simulações Baseadas em Sistemas Multiagentes (MABS) para implementar VES é adequado devido a possibilidade de modelar e implementar o sistema com ênfase nos atores e seus comportamentos dinâmicos. Contudo, quando os agentes da simulação passam a controlar o avanço do tempo de simulação de forma individualizada, surge uma situação semelhante à simulação paralela, o que implica na necessidade de tratar questões como disputa por recursos computacionais pelos agentes, atrasos em tempo de execução e a perda de consistência da própria simulação. A situação torna-se mais complicada em sistemas desenvolvidos com Java, devido a particularidades desta tecnologia. Este trabalho apresenta uma abordagem para controlar estes atrasos em tempo de execução e assim apoiar o desenvolvimento de VES utilizando MABS com Java.

## Palavras–chave

Simulação Baseada em Sistemas Multiagentes; Tardiness; Tempo de Simulação.

## Abstract

Taranti, Pier–Giovanni; Lucena, Carlos José Pereira de (Advisor). **An architecture to tame time tardiness in multiagent based simulations**. Rio de Janeiro, 2013. 153p. DSc Thesis – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Virtual Environment Simulations (VES) are a special type of simulation, often used to implement games and serious games with virtual space representation and using both the next-event or stepped–time simulation time advance approach. An example of serious games is the simulation used to support War Games. Multiagent Based Simulation (MABS) are suitable to implement these simulations because of their ability to handle with complexity and individual actors modeling. However, when agents are responsible for advance their own simulation time, a situation similar to a parallel simulation arises. This implies in treat issues such as delays in performing scheduled actions (i.e tardiness) and its consequences in the virtual space representations. This situation is worst in Java based MABS, because of Java technology particularities. This work presents an approach to tame this tardiness and help the development of these cited VES using agent oriented paradigm.

## Keywords

Multi Agent Based Simulation; Tardiness; Simulation Time.

# Sumário

1	Introdução	<b>14</b>
1.1	Questão de pesquisa	16
1.2	Estrutura da Tese	19
2	Referencial Teórico	<b>20</b>
2.1	Sistemas Multiagentes	20
2.2	Simulação	23
2.3	Tempo Real	29
2.4	Ontologias	31
3	Plataforma MASP	<b>33</b>
3.1	Abordagem	35
3.2	Representando o Modelo Conceitual em Ontologia	37
3.3	Arquitetura de Implementação	41
3.4	Tecnologias utilizadas	47
3.5	Exemplo de Uso	48
3.6	Trabalhos Relacionados	50
3.7	Conclusão do Capítulo	51
4	Verificação de hipóteses	<b>53</b>
4.1	Sistemas Multiagentes Implementados em Java e Tardiness	54
4.2	O estudo quantitativo	56
4.3	Consequências do tardiness e mitigações	71
4.4	Conclusão do Capítulo	73
5	Solução da Questão Principal de Pesquisa	<b>74</b>
5.1	Introdução ao problema	74
5.2	Abordagem para solução das questões	80
5.3	Algoritmo de controle	89
5.4	Conclusão do capítulo	95
6	Análise dos resultados obtidos	<b>97</b>
6.1	Suporte a VES implemetadas com MABS dirigidas por tempo	97
6.2	Suporte a simulações georeferenciadas	103
6.3	Suporte a simulações dirigidas por eventos	111
6.4	Considerações sobre generalização da solução para outras tecnologias	116
6.5	Conclusão do capítulo	116
7	Conclusão	<b>118</b>
7.1	Contribuições	119
7.2	Trabalhos futuros	120
8	Referências Bibliográficas	<b>122</b>

A	Uso de modelos conceituais em OWL-DL para instanciação rápida de sistemas: ontologias em apoio a injeção de dependências	<b>134</b>
A.1	Introdução	134
A.2	Estado da Arte	135
A.3	Abordagem	136
A.4	Exemplo de aplicação da abordagem	139
A.5	Trabalhos Relacionados	151
A.6	Conclusão	152



## Lista de figuras

2.1	Verificação e validação @ (Drogoul et al. 2003)	24
3.1	Arquitetura em alto nível de abstração	36
3.2	Ontologia do modelo conceitual	39
3.3	Arquitetura do MASP - Diagrama de componentes	42
3.4	Ontologia: modelo Conceitual (parte de) de Guerra Anti-Submarina	49
4.1	Diagrama de caso de uso	58
4.2	Diagrama de classe do experimento	59
4.3	Visão gráfica da distribuição dos fatores por níveis	61
4.4	Histograma do nível de tardiness da amostra	63
4.5	Histograma do modelo transformado	64
4.6	Análise gráfica dos resíduos do modelo	64
4.7	Primeiro fator – nível de tardiness x número de agentes (boxplot)	67
4.8	Segundo fator – nível de tardiness x intervalo de execução (boxplot)	69
4.9	Terceiro fator – nível de tardiness x carga dos comportamentos	70
5.1	Diagrama de componentes - arquitetura conceitual da solução	85
5.2	Diagrama de sequência – obtenção do tempo de simulação	86
5.3	Diagrama de sequência – envio de tardiness	86
5.4	Diagrama de sequência – relógio da simulação	87
5.5	Diagrama atividade – controle de tardiness	93
5.6	Diagrama de classes simplificado do controle de $f$	96
6.1	Tardiness - 10 agentes de simulação executando sem controle de $f$ (10 seg. de passo)	99
6.2	Tardiness - 10 agentes de simulação executando sem controle de $f$ (120 seg. de passo)	100
6.3	Tardiness - 10 agentes de simulação executando com controle de valor de $f$ (10 seg. de passo)	100
6.4	Avanço do Tempo de Simulação - MABS com 10 agentes de simulação e com controle de valor de $f$	101
6.5	Variação do valor de $f$ - MABS com 10 agentes de simulação e com controle de valor de $f$	102
6.6	Tardiness em MABS executando com controle de valor de $f$ (2 min de passo - 10 agentes)	102
6.7	Tardiness em MABS executando com controle de valor de $f$ (2 min de passo - 100 agentes)	103
6.8	Amostra de controle – tempo de simulação x tempo real	105
6.9	Amostra principal – tempo de simulação & tempo real	106
6.10	Amostra principal – valor de $f$ x tempo real	106
6.11	Amostra de controle – histograma de tardiness	107
6.12	Amostra de controle – tardiness & tempo real / Boxplot	108
6.13	Amostra principal – histograma de tardiness	108
6.14	Amostra principal – tardiness & tempo de simulação / Boxplot	109

6.15	Amostra de controle – histograma do erro espacial	110
6.16	Amostra de controle – erro espacial & tempo real / Boxplot	110
6.17	Amostra principal – histograma do erro espacial	111
6.18	Amostra principal – erro espacial & tempo de simulação / Boxplot	112
6.19	Modelo conceitual da simulação apresentado em diagrama de classes e indivíduos de ontologia	113
6.20	Histograma de tardiness - Avanço ao Próximo Evento	115
6.21	Histograma de tardiness - Avanço ao Próximo Evento (sem controle)	115
A.1	Arquitetura em alto nível de abstração	140
A.2	Ontologia do modelo conceitual	142
A.3	Arquitetura do MASP - Diagrama de componentes	145
A.4	Ontologia: modelo Conceitual (parte de) de Guerra Anti-Submarina	150

## Lista de tabelas

4.1	Tabela de Responsabilidades	58
4.2	Desvios residuais do GLM	63
4.3	Summary of the transformed GLM, $G = \ln(Y) + Y Y = f(A, B, C)$	66
4.4	Análise de Variância. Model: quasi, link: identity	67
4.5	Análise de <i>Power</i> para o teste ANOVA	67
6.1	Cenário da Simulação	104
6.2	Tardiness	114

## **Lista de Abreviações e Acrônimos**

**ACL** Agent Communication Language

**AGR** Agent group role

**AI** artificial intelligence

**API** application programming interface

**BDI** Belief-Desire-Intention

**COTS** Commercial Off-The-Shelf

**CPU** central processing unit

**CSV** Comma-separated values

**DAML** DARPA Agent Markup Language

**DEVS** discrete–event simulation

**DIS** Distributed Interactive Simulation

**DL** Lógica Descritiva

**DTSS** discrete time system specifications

**ELMS** Environment Description Language for Multi-Agent Simulation

**EUA** Estados Unidos da America

**FIFO** First In, First Out

**FIPA** Foundation for Intelligent Physical Agents

**GIS** geographic information system

**HLA** High Level Architecture

**HLAe** HLA Evolved

**HMM** hidden Markov model

**ISO** International Organization for Standardization

**JADE** Java Agent DEvelopment Framework

**JDK** Java SE Development Kit

**JRE** Java Runtime Environment

**JRS** Java Specification Requests

**JTS** Java Topology Suite

**MABS** multi-agent based simulation

**NTP** Network Time Protocol

**OIL** Ontology Interchange Language

**OWL** Ontology Web Language

**POA** paradigma de orientação a agentes

**POO** orientação a objetos

**RANS** Requisito de alto nível de sistema

**RDF** Resource Description Framework

**RMI** Remote Method Invocation

**RTI** run-time environment

**RTJ** Real-Time Java

**SDK** Software development kit

**SMA** sistemas multiagentes

**SO** sistema operacional

**UDP** User Datagram Protocol

**UML** Unified Modeling Language

**VES** virtual environment simulation

**W3C** World Wide Web Consortium

**XML** Extensible Markup Language