



Davy de Medeiros Baía

**Modelagem de Contextos Dinâmicos em
Simulação de Gestão de Projetos de Software
Baseada em 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 2016



Davy de Medeiros Baía

**Modelagem de Contextos Dinâmicos em
Simulação de Gestão de Projetos de Software
Baseada em 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. Alessandro Garcia

Departamento de Informática — PUC-Rio

Prof. Julio Cesar Sampaio do Prado Leite

Departamento de Informática — PUC-Rio

Prof. Jean-Pierre Briot

CNRS Brasil

Prof. Paulo Alencar

School of Computer Science — UW

Prof. Márcio da Silveira Carvalho

Coordenador Setorial do Centro

Técnico Científico — PUC-Rio

Rio de Janeiro, 08 de Março de 2016

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

Davy de Medeiros Baía

Davy de Medeiros Baía graduou-se em Bacharelado em Sistema de Informação na Faculdade de Alagoas em 2006. Especializou-se em Engenharia de Software para Web na Faculdade de Alagoas em 2007. Também concluiu MBA em Gerenciamento de Projetos na Fundação Getúlio Vargas em 2010. Recebeu o título de Mestre em Modelagem Computacional de Conhecimento pela Universidade Federal de Alagoas em 2013, com um semestre cursado na PUC-Rio.

Ficha Catalográfica

Baía, Davy de Medeiros

Modelagem de Contextos Dinâmicos em Simulação de Gestão de Projetos de Software Baseada em MultiAgentes / Davy de Medeiros Baía; orientador: Carlos José Pereira de Lucena . — 2016.

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

1. Tese (doutorado) - Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Informática, 2016.

Inclui bibliografia.

1. Informática – Teses. 2. Gestão de Projetos de Software. 3. Modelagem de Contexto de Projeto de Software. 4. Sistemas Multiagentes. 5. Simulação Baseada em Multiagentes. 6. Tomada de Decisão para Gestão de Projetos de Software. I. Lucena, Carlos José Pereira de. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. III. Título.

Agradecimentos

Ao professor Carlos José Pereira de Lucena, meu orientador e referência, com quem eu tive a oportunidade de aprender e interagir, tornando o trabalho mais valioso e gratificante. Obrigado pela oportunidade de aprendizado e interação durante estes quatro anos intensos.

A todos os membros da banca por aceitarem o convite para participar da banca e por ajudaram a enriquecer o trabalho com suas críticas.

À minha família e principalmente aos meus pais, Manoel e Luciana, e as minhas irmãs Bruna e Kyriane por estarem sempre ao meu lado e incentivando.

À minha namorada Lorena, por todo amor, carinho, paciência e presença durante esses anos.

Aos meus amigos da PUC pelas palavras de incentivo e pela grande ajuda concedida durante todo o doutorado. Em especial, ao Rafael Rocha que por repetidas vezes leu os capítulos desta tese.

Aos meus amigos externos a PUC por toda a distração que me propiciaram esquecer os momentos mais difíceis.

À amiga Vera Menezes por toda a ajuda e carinho.

À Fundação Padre Leonel Franca, à PUC-Rio, à Capes e ao CNPq pelos auxílios concedidos, sem os quais este trabalho não poderia ter sido realizado.

A Deus, por simplesmente tudo.

Resumo

Baía, Davy de Medeiros ; Lucena, Carlos José Pereira de. **Modelagem de Contextos Dinâmicos em Simulação de Gestão de Projetos de Software Baseada em MultiAgentes**. Rio de Janeiro, 2016. 201p. Tese de Doutorado — Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Gestão de Projetos de software não é uma tarefa trivial, especialmente com as alterações que ocorrem ao longo de sua execução. Geralmente, um projeto de software possui elementos como tarefas e recursos humanos. Cada um destes elementos possui características próprias e estão relacionados uns com os outros. Estes elementos, suas características e seus relacionamentos são definidos como contexto. Definimos como contexto dinâmicos as alterações que ocorrem no contexto ao longo da execução do projeto. Os envolvidos na tomada de decisão de projeto precisam lidar com este contexto dinâmico, desta forma aumentando a complexidade da gestão do projeto. Simulações geralmente são utilizadas para auxiliar a responder questões ou fenômenos específicos sobre um domínio, por meio de análise de resultados de experimentos ou execuções. Para isto, simulações usam modelos que capturam detalhes de um domínio específico. Modelagem em sistemas multiagentes fornecem modelos robustos para representar ambientes do mundo real que são complexos e dinâmicos. Uma das vantagens ao utilizar simulação baseada em multiagentes é a sua capacidade de apoiar aspectos realistas da gestão de projetos, incorporando seus elementos por meio de agentes. Porém, existe a necessidade de uma abordagem para simular a gestão de projetos de software, para representar o seu contexto, executar cenários que auxiliem no processo de tomada de decisão e para apoiar os contextos dinâmicos que ocorrem ao longo da execução do projeto. Neste contexto, propomos um modelo conceitual baseado em sistemas multiagentes e simulação de gestão de projetos de software, o ProMabs. Este modelo conceitual contém cinco componentes para modelar o contexto e sua dinâmica, com isto, executar simulações de cenários que auxiliam no processo de tomada de decisão. Como contribuições, esta tese apresenta o ProMabs com três instanciações baseadas em tecnologias diferentes: uma plataforma de programação multiagentes, um ambiente de simulação, e uma ferramenta de simulação. Pode-se com isto, avaliar o ProMabs sob três perspectivas diferentes. A utilização do ProMabs com estas tecnologias, permite representar elementos e seus relacionamentos, criar cenários para tomada de decisão e apoiar projetos adaptativos de software, ou seja, com contexto dinâmico. Por fim, apresentamos um experimento com análise qualitativa e quantitativa,

com a aplicação do ProMabs para representar o contexto, apoiar a sua dinâmica e auxiliar a tomada de decisão por meio de cenários. Os resultados do experimento apresentam indícios positivos de que a instanciação do ProMabs por meio de uma ferramenta de simulação, auxilia os participantes na tomada de decisão.

Palavras-chave

Gestão de Projetos de Software ; Modelagem de Contexto de Projeto de Software ; Sistemas Multiagentes ; Simulação Baseada em Multiagentes ; Tomada de Decisão para Gestão de Projetos de Software .

Abstract

Baía, Davy de Medeiros ; Lucena, Carlos José Pereira de (Advisor). **Dynamic Context Modelling in Software Project Management Simulation Based on Multi-Agent**. Rio de Janeiro, 2016. 201p. PhD. Thesis — Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Software Project Management is not a trivial task, especially with the changes that occur during the course of its execution. Generally, software project has elements such as tasks and human resources. Each of these elements has its features and their relationships with each other. These elements, their features and their relationships defines a context. We define as dynamic context changes that occur in the context during execution of the project. Persons involved in project decision-making need to deal with this dynamic context, thus increasing the complexity of the project management. Simulations often apply to support answer specific questions or phenomena on a domain, through analysis of experiments or executions results. For this, simulations use models that capture details of specifics domains. Multi-agent systems modeling provides robust models to represent real-world environments that are complex and dynamic. One of the advantages of using multi-agent-based simulation is its ability to support realistic aspects of project management, incorporating its elements through agents. However, there is a lack of an approach to simulate the software projects management, to represent the context, execute scenarios to assist in the decision-making process and to support the dynamic context that occur throughout the project execution. In this context, we propose a conceptual model based on multi-agent system and software project management simulation, the ProMabs. This conceptual model contains five components to model the context and its dynamics, thus, execute simulations of scenarios that assist in the decision-making process. As contributions, this thesis presents ProMabs with three instances based on different technologies: a multi-agent programming platform, a simulation environment, and a simulation tool. Therewith, evaluate the ProMabs from three different perspectives. The application of ProMabs with these technologies, allows to represent elements and their relationships, create scenarios to assist decision-making and support adaptive software project, i.e., with dynamic context. Finally, we present an experiment with qualitative and quantitative analysis, with application of the ProMabs to represent the context, support its dynamic and assist decision-making by means of scenarios. The experimental results are positive indications that the instantiation of the

ProMabs as supported by a simulation tool assists the participants in decision making.

Keywords

Software Project Management ; Software Project Context Modelling ; Multi-Agent System ; Multi-Agent-based Simulation ; Decision-Making in Software Project Management .

Sumário

1	Introdução	16
1.1	Definição do Problema e Limitações Atuais da Área	17
1.2	Questões de Pesquisa	23
1.3	As Principais Contribuições	23
1.4	Organização da Tese	24
2	Fundamentação Teórica e Trabalhos Relacionados	26
2.1	Modelando Simulação de Gestão de Projeto de Software	26
2.2	Simulação Baseada em Multiagentes para Processo de Desenvolvimento de Software	27
2.3	Simulação Baseada em Multiagentes para Gestão de Projetos de Software	29
2.4	Abordagens que não utilizam Simulação baseada em MultiAgentes para gestão de Projetos de Software	31
2.5	Ambiente de Simulação MultiAgentes, Plataforma de Programação MultiAgentes e Ferramenta de Simulação	31
2.6	Considerações Finais	33
3	ProMabs: Simulação de Gestão de Projeto Baseada em Multiagentes	34
3.1	Interação com Ambiente Externo	34
3.2	ProMabs	35
3.2.1	Filtro de Informação	36
3.2.2	Modelo de Simulação para o Planejamento e Execução	37
3.2.3	Cenários de Execução	38
3.2.4	Analisador	39
3.3	Simulação baseada em Multiagentes para Gestão de Projetos de Software	40
3.3.1	Domínio de Projeto de Software com EAP	41
3.3.2	Diagramas Estruturais do ProMabs para Projetos de Software	42
3.3.3	Diagramas Dinâmicos do ProMabs para Projetos de Software	51
3.3.4	Dados de Instanciação e Situacional	54
3.3.5	Gerador de Cenários de EAP	55
3.4	Considerações Finais	62
4	Representação dos Elementos de Projeto de Software	63
4.1	Gestão de Projetos de Software e seus Elementos	63
4.2	Trabalhos Relacionados	65
4.3	ProMabs para um Plataforma de Programação MultiAgentes	67
4.3.1	Dimensão Agente e ProMabs	69
4.3.2	Dimensão Ambiente e ProMabs	70
4.3.3	Dimensão Organizacional e ProMabs	71
4.4	Exemplo Ilustrativo	72
4.4.1	Planejar o Gerenciamento do Escopo	73
4.4.2	Coletar os Requisitos	74
4.4.3	Definir o Escopo	76
4.4.4	Criar a Estrutura Analítica do Projeto (EAP)	76

4.4.5	Validar Escopo	77
4.4.6	Controlar Escopo	79
4.4.7	Relação entre PMBOK, JaCaMo e JaCaMoPM	80
4.4.8	Estudo Exploratório	81
4.5	Considerações Finais	84
5	Execução de Cenários para Gestão de Projetos	85
5.1	Cenários para Tomada de Decisão	85
5.2	Trabalhos Relacionados	87
5.3	ProMabs para um Ambiente de Simulação	89
5.3.1	Descrição do ProMabs no CORMAS	91
5.3.2	Modelo estrutural da Simulação	93
5.3.3	Gerenciamento de Escopo, Tempo, custo e Recurso Humanos	96
5.4	Estudo de Caso	99
5.4.1	Problema escolher uma Equipe	100
5.4.2	Orientações gerais	101
5.4.3	Os dados iniciais para executar a simulação	101
5.4.4	Cenários	102
5.4.5	Resultado da Simulação e Discussão	104
5.5	Considerações Finais	106
6	Suporte ao Gerenciamento de Projetos Adaptativo de Software	108
6.1	Projetos Adaptativos de Software	109
6.2	Trabalhos Relacionados	110
6.3	ProMabs para Ferramenta de Simulação	110
6.3.1	Ambiente Externo e o Filtro de Informação	112
6.3.2	Modelo de Simulação de Planejamento e Execução	115
6.3.3	Cenários de Execução	118
6.3.4	Analisador	124
6.4	Estudo de Caso	129
6.4.1	Visão Geral do Estudo de Caso	129
6.4.2	Exemplos Ilustrativos	129
6.4.3	Simulações e Resultados	130
6.5	Considerações Finais	135
7	Experimento - Tomada de Decisão	136
7.1	Planejamento e Execução do Estudo	136
7.1.1	Dados de Instanciação	137
7.1.2	Objetivos Específicos	137
7.1.3	Questionário e Aplicações do Questionário	138
7.1.4	Métricas	139
7.2	Análise de Dados e Resultados	139
7.2.1	Perfil dos Participantes	140
7.2.2	Analisando os Dados e Resultados das Atividades	142
7.2.3	Analisando os Dados e Resultados por Etapas e Grupos	154
7.3	Possíveis Ameaças	165
7.4	Considerações Finais	166
8	Conclusão e Trabalhos Futuros	167

8.1	Contribuições do trabalho	168
8.2	Limitações da Tese	170
8.3	Trabalho Futuros	171
9	Referências Bibliográficas	173
A	Questionários Utilizados no Estudo	181
A.1	Questionário para Grupo A	181
A.1.1	Formação Acadêmica, Conhecimentos Específicos e Participação/Experiência	181
A.1.2	Primeira Etapa	185
A.1.3	Segunda Etapa	186
A.2	Questionário para Grupo B	190
A.2.1	Formação Acadêmica, Conhecimentos Específicos e Participação/Experiência	190
A.2.2	Primeira Etapa	193
A.2.3	Segunda Etapa	194
B	Cenários em Simulação e Simulados	198

Lista de figuras

1.1	Cenário típico de gestão de projetos de software	18
1.2	Exemplo do Contexto baseado na Figura 1.1.	19
3.1	Interação com Ambiente Externo.	35
3.2	Os componentes do Modelo Conceitual ProMabs.	36
3.3	Os relacionamentos e as entidades do TAO (Silva, 2004).	41
3.4	Exemplo de uma EAP.	42
3.5	Diagrama de Organização.	43
3.6	Projeto de Software - Ambiente.	43
3.7	EAP - Organização Principal.	44
3.8	Classe do Papel de Agente - Projeto.	45
3.9	Classe do Papel de Agente - Pacote de Trabalho.	45
3.10	Classe do Papel de Agente - Tarefa.	46
3.11	Classe do Papel de Agente - Recurso Humano.	46
3.12	Classe do Papel de Agente - Gerente de Projeto.	46
3.13	Classe do Papel de Objeto - Horário de Trabalho.	47
3.14	Diagrama de Papel.	47
3.15	Classe Agente Projeto.	48
3.16	Classe Agente Pacote de Trabalho.	48
3.17	Classe Agente Tarefa.	49
3.18	Classe Agente Recurso Humano.	50
3.19	Classe Agente Gerente de Projeto.	50
3.20	Classe Calendario.	51
3.21	Diagrama de Classes.	51
3.22	Diagrama de Sequencia - Parte 1.	52
3.23	Diagrama de Sequencia - Parte 2.	53
3.24	Diagrama Entidade-Relacionamento.	54
3.25	Exemplo de arquivo de configuração para instanciar o Gerador.	57
3.26	Pseudocódigo do Gerador de EAP - Parte 2.	59
3.27	Pseudocódigo do Gerador de EAP - Parte 2.	59
3.28	Pseudocódigo do Gerador de EAP - Parte 3.	60
3.29	Pseudocódigo do Gerador de EAP - Parte 4.	61
4.1	Trabalhos existente e evolução para nossa abordagem.	67
4.2	Componentes Instanciados do ProMabs.	68
4.3	Meta-Modelo do JaCaMo (Boissier et al., 2013).	69
4.4	Grupos e Processos relacionados do PMBOK.	72
4.5	EAP criada pela simulação.	80
5.1	Componentes Instanciado do ProMabs.	90
5.2	MABS em Projetos de Software.	91
5.3	Diagrama de Classe para o modelo de Simulação de Gestão de Software.	94
5.4	Diagrama de Atividade do agente gerente de projeto.	95

5.5	Grupos e Processos relacionados a tempo, custo, recurso humano e escopo.	97
5.6	Dados da instanciação do Exemplo.	102
5.7	Avaliação do Conhecimento e Desempenho do Recurso Humano.	102
5.8	Report with the summary of the simulation.	105
5.9	Equipes Projeto x Cenários.	105
6.1	Componentes Instanciado do ProMabs no Anylogic.	111
6.2	Pseudocódigo para calcular distância entre nós.	115
6.3	Pseudocódigo para calcular a posição dos nós.	116
6.4	Exemplo de EAP para Cenário de Execução e Analisador.	117
6.5	Estados do Agente Projeto.	119
6.6	Estados do Agente Pacote de Trabalho.	120
6.7	Estados do Agente Tarefa.	121
6.8	Estados do Agente Recurso Humano.	122
6.9	Estados do Agente Gerente de Projeto.	124
6.10	Representação Visual do Agente Projeto.	125
6.11	Representação Visual do Agente Pacote de Trabalho.	126
6.12	Representação Visual do Agente Tarefa.	126
6.13	Gráfico - Estado da Tarefa x Dias.	127
6.14	Gráfico - Custo da Tarefas x Dias.	128
6.15	Gráfico - Recurso Humano por Tarefas x Dias.	128
7.1	<i>Latin Square</i> projetado para o estudo.	139
7.2	Anos de Experiência dos participantes em Projetos de Software.	140
7.3	Perfil dos participantes do Estudo.	141
7.4	Recursos Humanos, Habilidade (Skill) e Nível de Conhecimento (Level).	142
7.5	EAP para o desenvolvimento do Software.	143
7.6	Taxa de acerto por tempo da Atividade 1.	145
7.7	Taxa de acerto por tempo da Atividade 2.	147
7.8	Taxa de acerto por tempo da Atividade 3.	149
7.9	EAP para a Atividade 4.	151
7.10	Taxa de acerto por tempo da Atividade 4.	153
A.1	Recursos Humanos, Habilidade (Skill) e Nível de Conhecimento (Level).	183
A.2	EAP para o desenvolvimento do Sistema	184
A.3	EAP para redução tempo	188
A.4	Recursos Humanos, Habilidade (Skill) e Nível de Conhecimento (Level).	191
A.5	EAP para o desenvolvimento do Sistema	192
A.6	EAP para redução tempo	196
B.1	Gráfico - Estado da Tarefa x Dias. (Alta Resolução)	199
B.2	Gráfico - Custo da Tarefas x Dias. (Alta Resolução)	200
B.3	Gráfico - Recurso Humano por Tarefas x Dias. (Alta Resolução)	201

Lista de tabelas

3.1	Nome dos Campos do Arquivo de Configuração e seus Parâmetros Associados.	58
3.2	Exemplos de Arquivos Gerados.	61
3.3	Arquivo Projeto	61
3.4	Arquivo Gerente de Projeto	61
3.5	Arquivo Pacote de Trabalho	61
3.6	Arquivo Tarefa	61
3.7	Arquivo Recurso Humano	61
4.1	Relação entre PMBOK, JaCaMo, and JaCaMoPM	81
6.1	Exemplos dos Dados do Projeto.	112
6.2	Exemplos dos Dados do Gerente de Projeto.	112
6.3	Exemplos dos Dados do Pacote de Trabalho.	113
6.4	Exemplos dos Dados do Recurso.	113
6.5	Exemplos dos Dados da Tarefa.	114
6.6	Resultado da Simulação com a Estratégia Custo.	131
6.7	Resultado da Simulação com a Estratégia Custo e Tempo.	132
6.8	Resultado da Simulação com a Estratégia Nível de Conhecimento.	133
6.9	Resultado da Simulação com a Estratégia Nível de Conhecimento associada à taxa de performance.	134
7.1	Médias das Métricas.	155
7.2	Suposições de Taxa de Acerto - Horizontal.	159
7.3	Suposições de Tempo - Horizontal.	160
7.4	Suposições de Nível de Dificuldade - Horizontal.	161
7.5	Suposições de Taxa de Acerto - Vertical.	162
7.6	Suposições de Tempo - Vertical.	163
7.7	Suposições de nível de Dificuldade - Vertical.	164
A.1	Formação Acadêmica	181
A.2	Conhecimentos Específicos	182
A.3	Participação em Projetos de Software	182
A.4	Formação Acadêmica	190
A.5	Conhecimentos Específicos	190
A.6	Participação em Projetos de Software	191

“Omnia possum in eo qui me confortat.”

Philippians 4:13

1

Introdução

Gerenciar um projeto de software não é uma tarefa trivial, especialmente com as alterações que ocorrem durante a sua execução. Geralmente, um projeto de software possui um escopo que contém as tarefas e informações necessárias para realização do projeto. Porém, ao longo de um projeto existem diversas alterações no escopo que resultam no aumento da complexidade do projeto. Métodos e ferramentas de simulação baseado em Sistemas Multiagentes estão sendo aplicados para simular o comportamento de projetos de software ou para antecipar acontecimentos (Agarwal e Umphress, 2010, Cherif e Davidsson, 2010, Joslin e Poole, 2005, Nienaber e Barnard, 2007, Araújo et al., 2010, Wu et al., 2009, Athavale e Balaraman, 2013). Outras abordagem que não aplicam sistemas multiagentes também estão sendo utilizadas para auxiliar na gestão de projetos de software (Alba e Chicano, 2007, Yannibelli e Amandi, 2011, Hanne e Nickel, 2005, Xiao et al., 2013, Pfahl et al., 2001, Rodríguez et al., 2004, Lakey, 2003, Haapio e Menzies, 2009, Balsera et al., 2012) . Em geral, os objetivos dos trabalhos aqui discutidos são fornecer, de forma simples, uma solução para testar diferentes configurações de escopo de projetos e entender seus efeitos.

De forma mais geral, simulações usam modelos que capturam aspectos de um domínio específico. Estas simulações geralmente são utilizadas para tentar responder questões ou fenômenos específicos sobre o domínio, através da análise de resultados por meio de experimentos ou execuções. No domínio de gestão de projetos a simulação precisa envolver inúmeros e variados elementos, como por exemplo, recursos, tarefas, gerentes, as partes interessadas, patrocinadores e o ambiente do projeto. Cada um destes elementos tem suas propriedades e se relacionam entre si de muitas formas, e estes relacionamentos capturam as interações necessárias para o cumprimento das tarefas, resultando na finalização do projeto. No planejamento de projeto, estes elementos, suas propriedades e seus relacionamento são estáticos, pois o projeto não começou e portanto não sofrem alterações, chamamos a isto de contexto estático do projeto. Por outro lado, durante o tempo de execução

do projeto, eventualidades acontecem e estes elementos, suas propriedades e suas interações sofrem alterações, chamamos a isto de contexto dinâmico do projeto. Desta forma, baseando-se nas definições de contextos estático e dinâmico propostas por O'Donnell (O'Donnell, 1999). Tais contextos precisam ser observados para serem modelados e representados em uma simulação.

Segundo Luck et al. (Luck e McBurney, 2008), Simulações Baseadas em Multiagentes (SBMA) oferecem modelos para representar ambientes do mundo real com um grau adequado de complexidade. Assim, para acompanhar os acontecimentos de projetos do mundo real, torna-se necessário modelar os elementos, suas propriedades, suas interações e suas alterações de forma individual, apoiados por uma SBMA. Com esta abordagem de modelo é possível executar cenários por meio de simulações de acordo com contextos específicos. A execução destes cenários e seus resultados contém informações que auxiliam os envolvidos na gestão de projeto ao longo das etapas que envolvem alguma tomada de decisão. Desta forma, no domínio de gerenciamento de projetos de software, SBMA fornece soluções para que os gestores de projeto possam experimentar diferentes configurações de contextos que envolvem o projeto e, com os resultados, compreender seus efeitos.

1.1

Definição do Problema e Limitações Atuais da Área

Existe uma ampla aceitação de SBMA para simular gestão de forma mais geral (Adamatti et al., 2005, Briot et al., 2007, Page et al., 2012), bem como de forma mais específica em projetos de software (Agarwal e Umphress, 2010, Cherif e Davidsson, 2010, Joslin e Poole, 2005, Nienaber e Barnard, 2007, Szymczak et al., 2007). Apesar disto, um modelo conceitual para simular gestão de projetos de software com contexto dinâmico ainda não foi tratado. Existe a necessidade de uma abordagem para criar simulações que se configuram com estas alterações. Desta forma, apoiar a criação de simulações de gestão de projetos de software e com isto, auxiliar na execução de cenários a partir das alterações relevantes de contexto externo a simulação, com dados de projetos do mundo real.

Existe uma lacuna na modelagem de simulação de gestão de projetos adaptativos de software em sincronização com dados externos baseada em multiagentes. Definimos como projetos adaptativos os que possuem contexto dinâmico, ou seja, os que sofrem alterações ao longo de sua execução, como por exemplo, uma nova tarefa no escopo do projeto

após ser inicializado. Existe a necessidade de modelar e representar estes acontecimentos ou alterações, relacionadas com o que ocorre em projetos do mundo real, que são os dados externos da simulação, para apoiar na execução de novos cenários utilizando o mesmo modelo de simulação. Isto permite fornecer um suporte para simulação de contexto estático e dinâmico, sendo (auto)adaptativo em relação ao contexto dinâmico. Ao perceber estas alterações o simulador interpreta e simula de acordo com os novos contextos, envolvendo dados situacionais de momento e alterações do projeto do mundo real. Desta forma, esta tese de doutorado se concentra na **criação de um modelo conceitual baseado em sistemas multiagentes que oferece suporte a contextos dinâmicos para apoiar na concepção de simulações de gestão de projetos de software que precisam estar de acordo com o dinamismo do ambiente externo.**

A seguir, apresentamos um exemplo típico envolvendo gerenciamento de projetos de software. Este exemplo é descrito para ilustrar a necessidade de um modelo conceitual com contexto dinâmico para simulações de gestão de projetos de software.

PUC-Rio - Certificação Digital Nº 1212395/CA

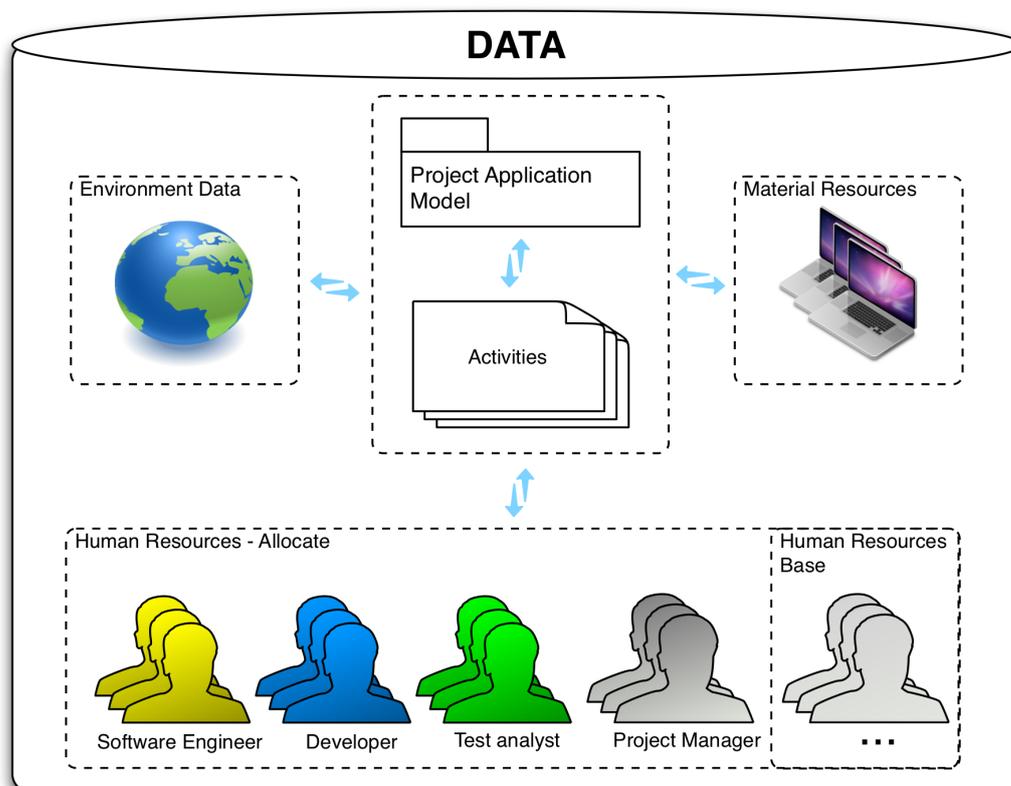


Figura 1.1: Cenário típico de gestão de projetos de software

A Figura 1.1 ilustra um exemplo em que uma SBMA de projeto de

software com modelagem de contexto dinâmico pode ser aplicada. Esta figura contém exemplos de elementos necessários para uma abordagem que modele e represente gestão de projetos de software e seus contextos dinâmicos.

Em um projeto de software temos os elementos, como ilustrado na Figura 1.1, que são os recursos humanos e materiais, as tarefas, o modelo de aplicação de projeto, e o ambiente, e para cada um destes elementos existe uma coleção de propriedades relacionadas. Por exemplo, a tarefa pode ter como propriedades (i) os recursos materiais necessários para sua execução, (ii) a área de conhecimento necessária que um recurso humano precisa possuir para realizá-la, assim como (iii) o nível de conhecimento desejado do recurso humano. Por sua vez, o recurso humano pode ter como propriedades a disponibilidade, o custo, a área de conhecimento e também seu nível de conhecimento. Já o modelo de projeto de aplicação pode ter como propriedade um conjunto de tarefas ou processos a serem seguidos dado um tipo específico de projeto. Finalmente, o ambiente pode ter como propriedades os recursos disponíveis e a configuração do espaço físico. Estas são algumas das propriedades e relacionamentos dos elementos do projeto do mundo real que chamamos de contexto e fazem parte da instanciação da nossa abordagem, os dados do projetos que estão no ambiente externo.

PUC-Rio - Certificação Digital Nº 1212395/CA

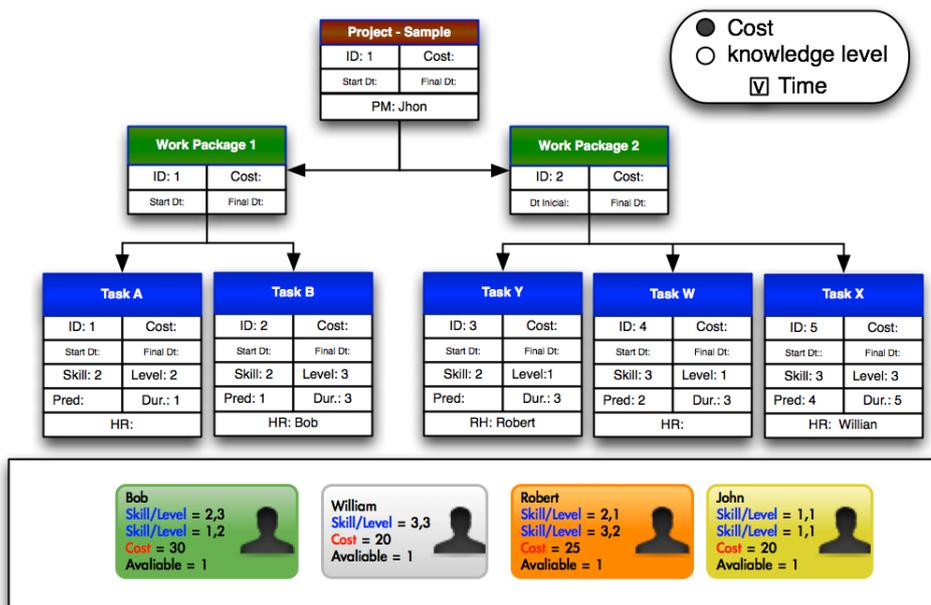


Figura 1.2: Exemplo do Contexto baseado na Figura 1.1.

Geralmente este contexto é descrito em um escopo de projeto que pode incluir uma Estrutura Analítica do Projeto (EAP). A figura 1.2 ilustra este contexto que são os elementos e suas propriedades em um modelo de aplicação

que neste exemplo é uma EAP. A EAP geralmente é organizada por níveis, no primeiro nível está o projeto como um todo e em seus níveis subseqüente estão os pacotes de trabalhos que possuem tarefas necessarias a serem realizadas para conclusão do pacote de trabalho.

Ao longo da execução do projeto este contexto ilustrado na Figura 1.2 geralmente sofre alterações por conta de situações que podem ocorrer pelos mais diversos motivos, geralmente incontroláveis, ou até mesmo por conta da intervenção direta do gestor do projeto. Por exemplo, o recurso humano Robert por algum motivo fortuito é desvinculado do projeto. Neste caso, as tarefas atribuídas a este recurso humano precisam ser realocadas a outros recursos humanos para que a execução do projeto continue, porém respeitando as características exigidas da tarefa. Um outro exemplo, seria a mudança da ordem das tarefas que pode influenciar em alguns fatores do projeto como tempo e custo. Além disso, esta mudança poderá impactar nos relacionamentos desta tarefa, como por exemplo, disponibilidade de recursos, pois as atividades possuem um relacionamento com os recursos e com a mudança os recursos podem estar ocupados em outras tarefas. Outro exemplo, seria a propagação da mudança de um elemento no projeto como um todo, uma tarefa atrasa, então as datas de inicio das tarefas sucessoras irão ser postergadas, ou seja, o projeto poderá atrasar. Assim, todo o escopo tem que ser revisto. Estes exemplos supracitados ao longo da execução do projeto, chamamos de contexto dinâmico, pois sofreram alterações após iniciar o projeto.

Projetos de software, geralmente enfrentam, como foi descrito, alterações que podem levar a modificações do plano do projeto. No entanto, as abordagens existentes de SBMA de projetos listadas na nossa investigação não apoiam métodos de adaptação para lidar com essas alterações. Com isso, existe a necessidade de um modelo conceitual para simulação de gestão de projetos que possua componentes necessários que garantam a adaptação, apesar das alterações. Com isto, executar cenários com informações relevantes para tomada de decisão, para que as metas do projeto sejam alcançadas pelos gestores de projeto. Nesta conjuntura, a capacidade de adaptar um plano do projeto às alterações de contexto que acontecem durante a execução do projeto, muitas vezes distinguem os projetos entregues conforme as suas metas daqueles que não são entregues.

Falta apoio para tomada de decisão em simulações com contextos estáticos e dinâmicos. Usar uma SBMA em projetos de software requer a observação das alterações geradas por acontecimentos na execução do projeto ou realizadas pelo gestor no projeto. Com isso, existe a necessidade de

um modelo conceitual que contemple estas alterações de contexto de projeto de software. Além disto, aplicar um modelo conceitual em simulações de gestão de projetos de software para auxiliar no processo de tomada de decisão. Para isto, uma EAP pode ser aplicada para visualizar o fluxo de uma tarefa para outra à medida que cada tarefa é executada pelo recurso associado, como também para verificar os impactos das alterações, com informações relevantes para atividades que envolvem alguma tomada de decisão. A simulação precisa executar cenários para ilustrar os impactos causados pelas alterações de contexto, por exemplo, se aumenta o custo, tempo ou altera a equipe de desenvolvimento. Desta forma, permite aos envolvidos no processo de tomada de decisão do projeto acompanhar ou verificar os impactos destas alterações no escopo do projeto, auxiliando na validação, monitoramento e no seu controle.

Geralmente atividades que envolvem alguma decisão possuem enormes quantidades de dados que geram informação. Em tais atividades, o registro e análise manual de dados não são viáveis, devido ao risco de erro humano e limitações de custos e tempo globais. Pesquisas recentes indicam que projetos de Tecnologia da Informação (TI) continuam a ter uma elevada taxa de insucesso (GROUP, 2013),(Rubinstein, 2007); E também há evidências crescentes de que as atividades e habilidades do gerente de projeto podem ser cruciais para o desempenho eficiente e eficaz da equipe do projeto em melhorar os seus resultados (Flyvbjerg e Budzier, 2011), (Taylor e Woelfer, 2011). Assim, o gerente de projeto precisa de um conjunto de tecnologias apropriadas para apoiar a sua gestão.

Os gerentes de projetos precisam de informações para apoiar as fases de planejamento e execução, e estas possuem informações estáticas e dinâmicas, respectivamente. Os gerentes de projeto podem executar simulações relacionadas com vários planos. Desta forma, os gerentes de projeto executam cenários com diferentes contextos para auxiliar na escolha de um plano. Previsão e gestão de projetos podem prover resposta às alterações e incertezas que acontecem durante a fase de execução do projeto, como também auxiliar na fase de planejamento. Com isso, existe a necessidade de suporte para geração desses planos nestas fases, mas os trabalhos pesquisados ainda não implementam este suporte.

Para lidar com os aspectos dinâmicos de projeto, os dados situacionais precisam ser incluídos como parte da SBMA, de modo que as anomalias de execução de projetos, como por exemplo, perda de recurso, comum em projetos realistas, possam ser percebidas e simuladas. Assim sendo, os dados de execução de projeto do ambiente externo, precisam ser incorporados a

SBMA. Estes dados podem ser usados para criar estratégias de execução de projetos que podem ser percebidos após o modelo de simulação receber estes dados situacionais. Deste modo, uma abordagem para modelagem de contexto dinâmico em SBMA pode ser aplicada para simular aspectos complexos e dinâmicos do mundo real. Com isso, fornecer cenários para os problemas que acontecem ao decorrer do projeto. Porém, para isso existe a necessidade de representar e simular as alterações, que não seriam de outro modo resolvíveis utilizando apenas os dados estáticos isoladamente.

Em geral, a modelagem de contexto dinâmico em SBMA deve apoiar aspectos realistas do domínio do problema, que neste caso é a gestão de projeto do software. Em primeiro lugar, SBMA fornece recursos de modelagem na qual os agentes podem ter seus tipos individuais. Por exemplo, um agente pode representar um recurso humano do projeto, assumindo um papel de desenvolvedor que precisa executar determinadas tarefas e que possui suas características individuais. Em segundo lugar, os agentes são autônomos e podem (auto)adaptar-se em função de fatores, por exemplo, a disponibilidade de recursos, que neste caso pode se relacionar ao fato do agente desenvolvedor estar indisponível. Esta forma de adaptação pode acontecer quando um recurso que foi originalmente disponível para uma tarefa do projeto se torna indisponível. Assim, o agente gerente de projeto atribui outro agente desenvolvedor disponível, sem interação humana. Com isso, os gerentes de projeto podem usar a abordagem de simulação. Esta abordagem permite verificar o impacto em um projeto quando uma equipe de desenvolvimento específica for alterada ou selecionada, sob certos fatores como prazos, custos e requisitos de qualidade, que dependem dos recursos humanos escolhidos. Em terceiro lugar, a execução de projeto geralmente é um processo estocástico, uma das características para utilização de sistemas multiagentes. Por exemplo, a perda de um recurso humano é algo extremamente incerto, a simulação percebe este evento e atribui outro recurso, que no caso da simulação é um agente.

Finalmente, um modelo conceitual e SBMA para gestão de projetos de software com contexto dinâmico, pode ser utilizada por gerentes de projeto para visualizar execuções de cenários específicos, por meio das representações dos elementos da gestão de projeto, a fim de tomar decisões relacionadas ao planejamento e execução do projeto.

1.2

Questões de Pesquisa

Baseado na definição dos problemas e limitações descritos na seção 1.1 foram definidas as questões de pesquisas apresentadas a seguir.

Questão de Pesquisa (QP1) Como projetar e implementar uma abordagem SBMA para simular gestão de projetos de software?

- (i) Como **representar** os elementos necessários para simular a gestão de projetos de software?
- (ii) Como **executar cenários** para auxiliar o gerente de projetos com a tomada de decisão?
- (iii) Como **apoiar** o gerenciamento de projetos adaptativos de software com o uso de SBMA?

1.3

As Principais Contribuições

- **Um modelo conceitual baseado em sistemas multiagentes e simulação de gestão de projetos de software.** Uma abordagem para simular gestão de projetos de software, o ProMabs. Esta abordagem, permite por meio de seus componentes instanciados, representar elementos e seus relacionamentos, executar cenários para auxiliar a tomada de decisão, e apoiar à simulação de gestão de projetos adaptativos de software. Para isto, o ProMabs possui 5 componentes, cada um com um objetivo e se baseia em simulação multiagentes.
- **Representar Elementos e seus Relacionamentos de Gestão de Projetos de Software.** Por meio da instanciação do modelo conceitual ProMabs em uma plataforma de programação multiagentes, representamos elementos do projetos baseados nas noções de crenças, metas e planos como atitudes mentais. Com isto, orientar a realização dos comportamentos dos agentes, que são representações dos elementos.
- **Cenários para Auxiliar a Tomada de Decisão.** Por meio da instanciação do modelo conceitual ProMabs em um ambiente de simulação, executamos cenários baseados em estratégias. Desta forma, o mecanismo social incorporado na simulação multiagentes executa alguns cenários simultaneamente, por meio de um modelo único. Estes cenários

são baseados em estratégias que auxiliam no processo de tomada de decisão.

- **Apoiar a Simulação de Gestão de Projetos Adaptativo de Software.** A fim de lidar com alterações que acontecem ao longo do projeto de software, instanciamos o modelo conceitual ProMabs em uma ferramenta de simulação. Com isto, obtivemos um modelo para simular a gestão projetos de software que permite contextos estáticos e dinâmicos. Esta instanciação é baseada nos componentes do ProMabs e cada um deles tem uma função para o modelo. Possibilitando a simulação adaptar-se, de acordo com o ambiente externo.
- **Experimento com o Modelo Conceitual Instanciado para Tomada de Decisão.** A fim de analisar uma instanciação do modelo conceitual para a tomada de decisão, realizamos um experimento com análise qualitativa e quantitativa da instanciação do ProMabs em uma ferramenta de simulação para auxiliar em atividades que necessitam de tomada de decisão.

1.4

Organização da Tese

Esta tese é estruturada como segue:

- No Capítulo 2 são apresentados conceitos e alguns trabalhos relacionados.
- O Capítulo 3 apresenta o ProMabs e seus componentes e SBMA para gestão de projetos de software.
- No Capítulo 4 apresenta-se uma instanciação do ProMabs em uma plataforma de programação multiagentes, o JaCaMoPM. Com este, aplicamos um estudo exploratório para descrever a representação dos elementos e relacionamentos da gestão de projetos de software, no âmbito de escopo.
- No Capítulo 5 apresenta-se uma instanciação do ProMabs em um ambiente de simulação. Com esta, aplicamos um estudo de caso para descrever como execução de cenários auxilia no processo de tomada de decisão.
- No Capítulo 6 apresenta-se uma instanciação do ProMabs em uma ferramenta de simulação. Com esta, exemplificamos como apoiar à gestão de projetos adaptativos de software.

- No Capítulo 7 apresenta-se um estudo de caso com análise qualitativa e quantitativa, por meio de um *Latin Square*.
- O Capítulo 8 apresenta as conclusões finais e os trabalhos futuros.

2

Fundamentação Teórica e Trabalhos Relacionados

A análise dos trabalhos relacionados a esta tese cobrem as propostas que definem um modelo para simulação de gestão de projetos de software. Algumas abordagens diferentes de simulação baseada em multiagentes também foram analisadas. Neste contexto, ressaltamos que não temos a intenção de avaliar diferentes abordagens, apenas identificar as vantagens de usar nossa simulação baseada em multiagentes. A fim de identificar as vantagens da simulação baseada em multiagentes discutimos por que simular e quais os benefícios da simulação baseada em multiagentes. Como fundamentação teórica, apresentamos as definições para gestão de projeto de software, modelo e simulação e seus significados quando usadas em conjunto.

2.1

Modelando Simulação de Gestão de Projeto de Software

Nesta seção iremos discutir as palavras chaves da nossa abordagem, gestão de projeto de software, modelo e simulação e seus significados quando usadas em conjunto. Uma *gestão de projeto de software* envolve a gestão de todos os aspectos e questões que estão envolvidas no desenvolvimento de um projeto de software. Isto inclui, a identificação de escopo e objetivo, planejamento, avaliação, abordagens de desenvolvimento de projetos, esforço de desenvolvimento e estimativa de custos. Além disto, existem o planejamento, monitoramento e controle de atividades, gestão de riscos, alocação e controle de recursos, bem como gestão de contratos, equipes e qualidade (Nienaber e Cloete, 2003). Concluímos que a gestão de projeto de software é a gestão de tarefas, recursos e pessoas com objetivo comum de desenvolver um software de forma estruturada. Para isto, segue um planejamento, execução, monitoramento e controle. Então, esta tese foca em gestão de projetos de software, porém existem trabalhos relacionados que focam em áreas transversais da gestão de projetos como processo de desenvolvimento de software. Estes trabalhos serão discutidos posteriormente na seção 2.2.

Um *modelo* é uma representação simplificada de um sistema complexo real ou conceitual. Um modelo é criado para apresentar características significativas e propriedades do sistema que se pretende estudar, prever, modificar ou controlar (Kellner et al., 1999). Porém, um modelo geralmente não inclui todos os aspectos do sistema que está sendo modelado, apenas os mais importantes para o estudo. Desta forma, o modelo se torna valioso quando fornece informações úteis, previsões e respostas para perguntas para o qual foi modelado. Assim, existe a necessidade de se representar os elementos básicos da gestão de projetos de software para criação do modelo. Com isto, fornecendo cenários com informações úteis, previsões e respostas para as perguntas as quais a gestão de projeto de software necessita para a tomada de decisão.

Segundo Kerllner et al. (Kellner et al., 1999), um *modelo de simulação* é um modelo computacional que possui as características acima descritas e que representa algum sistema dinâmico, fenômeno ou comportamento dinâmico de sistemas específicos. Uma das principais motivações para o desenvolvimento de um modelo de simulação é fornecer uma alternativa viável para obter *insight* importantes quando os custos, riscos ou logística de manipular o sistema real são proibitivos. As simulações são geralmente utilizadas quando a complexidade do sistema que está sendo modelado está além da representação útil dos modelos estáticos ou outras técnicas. Assim, em gestão de projetos de software existe a necessidade de modelos de simulação que sincronizam com o ambiente externo. Além disto, modelos que possuam componentes nos quais os gestores de projetos possam simular fenômenos ou comportamentos dinâmicos por cada um dos elementos do projeto. Esta é uma das vantagens da nossa abordagem.

Finalmente, o Modelo de Simulação de Gestão de Projetos de Software é uma representação computacional de um sistema complexo real, que envolve os aspectos e questões que estão envolvidas na gestão de um projeto de software. Com isto, oferecer um suporte a gestão de projetos de software com *insights* que auxiliem nas etapas que envolvem alguma tomada de decisão.

2.2

Simulação Baseada em Multiagentes para Processo de Desenvolvimento de Software

Processo de desenvolvimento de software é uma área transversal da Gestão de Projetos de Software, que pode determinar quais são as regras e atividades necessárias e suas saídas para o desenvolvimento de software.

Segundo nossa investigação na literatura, o primeiro trabalho aplicando simulação baseada em sistema multiagentes foi em 1990 por Peiwei Mi e Walt Scacchi (Mi e Scacchi, 1990). Os autores descrevem um processo de desenvolvimento de software baseado em conhecimento e regras para modelagem e simulação com multiagentes, evoluindo em (Mi e Scacchi, 1993). Os autores provêm uma visão de arquitetura geral que identifica alguns componentes principais, detalhando o meta-modelo de conhecimento, o simulador e a base de conhecimento. Porém só alguns dos elementos são agentes, os recursos humanos, com isto existe a necessidade de criar níveis de abstração para o meta-modelo, o que dificulta a compreensão do processo de simulação. Torna-se necessário especificar as entradas e saídas de cada atividade para montar o processo de desenvolvimento de software exigindo, assim, um alto-grau de nível de especificação do processo a ser simulado. Por outro lado, nossa abordagem necessita de uma conexão com o ambiente externo que possua o nível de abstração necessário para representar a execução do projeto simulado, ou seja, estrutura analítica de projeto que contém algum processo de desenvolvimento de software. Desta forma, nossa abordagem foca no nível de gestão de projetos de software e não no nível de processo de desenvolvimento de software.

Um outro trabalho (Wickenberg e Davidsson, 2002), descreve os motivos para a utilização de sistemas multiagentes para simulação de processo de software. Os autores afirmam que apesar do processo de desenvolvimento de software ser realizado por um conjunto de indivíduos que colaboram entre si, a maioria das abordagens para simular este processo usam uma visão baseada em atividades centralizadas ao invés de uma visão baseada em distribuição. Este é um dos nossos motivos para aplicar multiagentes em Gestão de Projetos de Software. Trata de uma sociedade que colabora para desenvolver um software. Bojan and Onggo (Spasic e Onggo, 2012) propuseram uma função de esforço prático para estimar o comportamento dos desenvolvedores. Este artigo foca na fase de construção de um *Rational Unified Process* adaptado.

Agarwal and Umphress (Agarwal e Umphress, 2010), em um trabalho inicial, aplicam agentes com o papel de desenvolvedores em simulação de processo de desenvolvimento de software, com o objetivo de verificar o impacto da mudança de quantidades de agentes desenvolvedores durante a simulação. Em qualquer tempo de simulação, o usuário pode alterar o número de desenvolvedores ou atribuir desenvolvedores a diferentes fases do desenvolvimento de software, dependendo de suas características. Já Cherif and Davidsson (Cherif e Davidsson, 2010) desenvolvem um modelo também

para uma simulação de processo de desenvolvimento de software baseado em multiagentes. Além disso, fazem comparações entre o modelo com abordagem de Dinâmica de Sistemas (DS) e MultiAgentes. Com isto, descrevendo como SBMA refletem o domínio do problema de forma mais realista do que DS.

Apesar da modelagem do processo de desenvolvimento de software concentrar um grande número de trabalhos (Zhang et al., 2008, Zhang et al., 2008, Zhang et al., 2010, Ali e Petersen, 2012) poucos trabalhos aplicam a simulação baseada em agente. Boa parte dos trabalhos tem como aplicação eventos discretos e dinâmica de sistemas. Os trabalhos listados nesta seção focam apenas na concepção de plano e definição de processo de desenvolvimento de software (SDP). Porém o processo de desenvolvimento de software adotado é um dos elementos para definir um escopo de projeto de software e geralmente este SDP sofrem alterações. Nossa abordagem está no nível de gestão de projeto de software e parte do pressuposto que existe uma Estrutura Analítica de Projeto. Desta forma, a EAP aplica de alguma forma um processo de desenvolvimento de software, o que então nos faz acreditar que estes trabalhos podem ser uma entrada para a nossa abordagem. Por esta razão, abstraímos o processo de desenvolvimento de software e focamos na gestão de projeto de software e nos impactos de suas alterações.

2.3

Simulação Baseada em Multiagentes para Gestão de Projetos de Software

Segundo a nossa pesquisa, o primeiro trabalho que introduziu multiagentes para gestão de projeto teve início com Brazier e Treur (Brazier et al., 1996). Os autores descrevem um modelo de cooperação de agente, especificando, especializando e instanciando. Porém, diferente do nosso trabalho a interatividade com o gestor de projeto não fica clara, assim como, as alterações na estratégia podem impactar a simulação do projeto.

Joslin e Poole (Joslin e Poole, 2005) descrevem uma tentativa preliminar de adaptar um algoritmo de planejamento baseado em simulação, que foi desenvolvido para o planejamento de atividades experimentais de robôs em Marte para o problema de planejamento de gerenciamento de projetos de software.

Nienaber and Barnard (Nienaber e Barnard, 2007) utilizam agentes para ajudar com a melhoria dos processos de gerenciamento de projetos de software embora ainda em fase inicial. Um trabalho na linha de gestão de portfólio está em (Araújo et al., 2010), ou seja, na gestão de vários projetos.

Os autores, propõem um sistema multiagentes, onde os projetos negociam a aquisição de recursos através de um mecanismo de leilões ao longo da execução do portfólio. Ambos, projetos e recursos são modelados como agentes. Este trabalho permite o gerenciamento de portfólio de projetos e a avaliação na decisão de aceitação ou rejeição de novos projetos. Nossa abordagem também aceita mais de um projeto compartilhando recursos, porém, além de focar no planejamento focamos também na execução, o que diferencia o nosso trabalho. Além disto, modelamos nossas tarefas e pacotes de trabalhos também como agentes, assim carregando características próprias, podendo, por exemplo, se auto-verificar, como iremos demonstrar nos próximos capítulos.

Wu et al. (Wu et al., 2009) apresentam um sistema multiagentes baseado em métrica, com o objetivo de ajudar os gerentes na compreensão e visualização de processo de gestão de projetos de software definido em um plano de projeto de software. Diferente do nosso trabalho os autores aplicam os agentes para coletar dados, calcular atributos e monitorar o projeto. Nossa abordagem simula a gestão do projeto de software quando executado por meio de agentes que representam os elementos do projeto.

Baseado na modelagem e simulação do comportamento humano, Athavale e Balaraman (Athavale e Balaraman, 2013) desenvolveram um modelo computacional do comportamento humano no contexto de um projeto de desenvolvimento de software aplicando MultiAgentes. Os autores modelam aspectos humanos que afetam o desempenho da tarefa em um projeto de software. Demonstram o potencial impacto de aspectos humanos nos resultados do projeto de software, que pode levar a *insights* de gestão. Diferente da nossa abordagem os autores focam apenas na modelagem do comportamento humano, por meio da simulação de eventos que influenciam na atividade dos agentes humanos. Como não queremos verificar esse nível, esta modelagem não foi amplamente abordada no nosso trabalho. Porém, caso exista a necessidade, é algo que pode ser aplicado em nossa abordagem.

Finalmente, nossa abordagem, se diferencia das demais supracitadas por focar na simulação de um planejamento e na execução do projeto de software, modelando os principais elementos tais como, pacote de trabalho, tarefa, gerente de projeto, e recursos como agentes. Para isto, os agentes seguem uma arquitetura BDI (do inglês Belief-Desire-Intention) (Rao e Georgeff, 1991), que se baseia nas noções de crenças, desejos e intenções como atitudes mentais que orientam a realização dos comportamentos dos agentes. Com isso, pode-se verificar os comportamentos de cada elemento e os impactos das alterações destes elementos.

2.4

Abordagens que não utilizam Simulação baseada em MultiAgentes para gestão de Projetos de Software

Alguns trabalhos tentam resolver o problema de cronograma em Projetos de software, parte da gestão de projeto, utilizando algoritmos genéticos (Alba e Chicano, 2007, Yannibelli e Amandi, 2011), evolutivos multiobjectivo (Hanne e Nickel, 2005), otimização da colônia de formigas (Xiao et al., 2013), de acordo com o número de recursos e tarefas. Porém a nossa abordagem não é procurar a melhor configuração para um cronograma e sim verificar fenômenos que podem acontecer na gestão de projetos que com outra técnica fica difícil, pois modelamos o comportamento individual dos elementos envolvidos. Além disto, estes trabalhos tendem a piorar a performance quando aumentamos o número de recursos e tarefas, pois geram grande número de opções para serem verificadas. Como na simulação multiagentes trabalhamos com *steps* este problema é mitigado. Porém estendemos o gerador de tarefas do artigo (Alba e Chicano, 2007) para gerar EAPs que são aplicadas nas simulações desta tese, que será apresentada no capítulo 3.

Outras abordagens utilizam a modelagem em dinâmicas de sistema (Pfahl et al., 2001, Rodríguez et al., 2004), diferente de MultiAgentes que é essencialmente descentralizada, pois foca no individualismo (agente). Dinâmicas de sistema e eventos discretos também estão sendo utilizado de forma híbrida (Lakey, 2003), para auxiliar nas estimativas.

Mineração de dados é também utilizada na gestão de projetos (Haapio e Menzies, 2009, Balsera et al., 2012), porém só é eficiente quando possui se dados históricos suficientes. Já em simulação baseada em multiagentes está técnica poderia ser aplicada para melhorar as informações para o modelo, entretanto não é um pré-requisito.

2.5

Ambiente de Simulação MultiAgentes, Plataforma de Programação MultiAgentes e Ferramenta de Simulação

Podemos encontrar alguns ambientes e ferramentas de simulação baseada em multiagentes (Kravari e Bassiliades, 2015). Estudamos alguns, porém iremos descrever o Cormas e o Anylogic, pois aplicamos nossos estudos nestas tecnologias. Além disto, aplicamos também no JaCaMo, uma plataforma de programação multiagentes. Desta forma, utilizamos três diferentes tecnologias em nossos estudos. Discutiremos em capítulos futuros os motivos destas

escolhas.

Cormas (Bousquet et al., 1998) é uma plataforma de simulação que usa o ambiente de programação VisualWorks e que permite o desenvolvimento de aplicações na linguagem orientada a objeto Smalltalk. Cormas tem entidades pré-definidas que são classes genéricas de Smalltalk podendo ser, especializadas e aperfeiçoadas. Assim, os usuários podem criar entidades específicas para o seu próprio modelo. Este ambiente de simulação, facilita a construção de modelos baseados em agentes, com o projeto, monitoramento e análise de cenários de simulação baseados em agentes. Cormas foi orientada principalmente para a representação das interações entre as partes interessadas na utilização dos recursos naturais renováveis. Nós utilizamos o Cormas em um momento em que existia a necessidade de executar cenários baseados em estratégias, que utilizavam recursos renováveis, ou seja, agentes representando recursos humanos. Como resultado, os cenários representava um conjunto de recurso humano para executar determinada tarefas.

Anylogic (Borshchev, 2013) é uma plataforma de simulação multi-método, onde um desses métodos é o multiagentes. A plataforma foi desenvolvida em Java e seu ambiente de desenvolvimento é um plugin para o eclipse. Além disso, o modelo de paradigma de design orientado a objeto apoiado por Anylogic prevê a construção modular, hierárquica e incremental de grandes modelos. A interface gráfica, ferramentas e objetos de biblioteca de Anylogic permitem a utilização por usuários de forma rápida, quando existe um modelo. No nosso caso os usuários são as pessoas envolvidas no processo de tomada de decisão.

Para o nível de abstração utilizamos a plataforma JaCaMo (Boissier et al., 2013), que reúne três paradigmas de programação multiagentes, programação orientada a agente, programação orientada a organização e programação orientada a ambiente. O JaCaMo é baseado em três plataformas existentes: Jason (Bordini et al., 2007) para a programação de agentes autônomos, Moise (Hubner et al., 2007) para programação de organizações de agentes, e Cartago (Ricci et al., 2009) para programação de ambientes compartilhados. Utilizamos o JaCaMo para a formalizar os conceitos de autonomia, organização e ambiente compartilhado dos agentes. Com isto, representamos os elementos e seus relacionamentos da gestão de projeto de software.

2.6

Considerações Finais

Neste capítulo, introduzimos a fundamentação teórica e apresentamos os trabalhos relacionados com esta tese de forma mais ampla. Para a fundamentação teórica, descrevemos a modelagem de simulação de gestão de projeto de software. Para os trabalhos relacionados, listamos alguns trabalhos encontrado na literatura que utilizam sistemas multiagentes e outros que não utilizam. Além disto, descrevemos tecnologias para utilizar multiagentes.

3

ProMabs: Simulação de Gestão de Projeto Baseada em Multiagentes

Este capítulo define um modelo conceitual para simulação de gestão de projeto de software baseado em multiagentes, chamado ProMabs. O modelo conceitual é composto por componentes que desempenham papéis específicos na simulação. Estes componentes são detalhados em termos de suas funções para a simulação. A partir destes componentes, descreve-se a aplicação do ProMabs utilizando Estrutura Analítica de Projeto modelado na linguagem MAS-ML, para instanciar os agentes representando elementos da EAP. Finalmente, um Gerador de Cenários de EAP é apresentado como forma de obter dados para nosso estudo. Porém, neste capítulo o ProMabs é descrito apenas em termos de especificação no nível conceitual como base para os estudos, que serão apresentados nos próximos capítulos. Neste contexto, este capítulo se concentra na primeira questão de pesquisa desta tese:

(QP1). Como projetar e implementar uma abordagem SBMA para simular gestão de projetos de software?

Para isto, na seção 3.1 descrevemos a interação do ProMabs com ambiente externo. Na seção 3.2 descrevemos o ProMabs e seus componentes. Na seção 3.3 descrevemos o ProMabs aplicando uma EAP modelado em MAS-ML. E finalmente na seção 3.4, apresentamos a conclusão e algumas discussões.

3.1

Interação com Ambiente Externo

Para desenvolver uma simulação de gestão de projetos de software em cooperação com o ambiente externo, ou seja, que suporte a dinâmica de um projeto nas fases de planejamento e execução, um modelo conceitual precisa possuir os componentes necessários para auxiliar estas duas fases. Com este objetivo, foi desenvolvido o modelo conceitual chamado ProMabs.

ProMabs é capaz de apoiar dados externos à simulação, esses dados são de instanciação do modelo e dados situacionais, com isso, obtém-se a robustez e a flexibilidade suficientes para lidar com eventos inesperados. Para isso, parte-se do pressuposto de que existem informações do projeto que estão em uma base de dados continuamente alimentada por uma ferramenta de gestão de projeto. Desta forma, o ProMabs utiliza-se destes dados que parametrizam a simulação. Com isto, a simulação pode ser executada e os envolvidos na tomada de decisão podem observar o comportamento da execução simulada da gestão projeto de software. Caso exista a necessidade, por motivos de alteração no projeto do ambiente externo, o ProMabs e seus componentes permitem executar uma nova simulação com novos dados que são alimentados continuamente por meio de uma ferramenta de gestão adotada. Estes dados chamamos de contexto e suas alterações chamamos de contexto dinâmico. A figura 3.1 ilustra esta interação entre o ambiente externo e o modelo ProMabs.

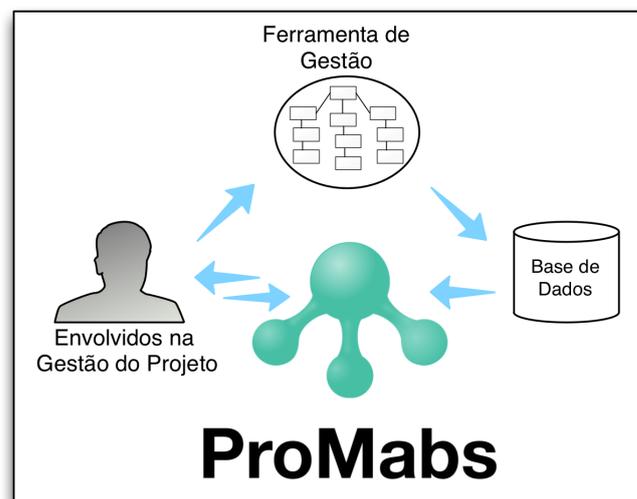


Figura 3.1: Interação com Ambiente Externo.

3.2

ProMabs

ProMabs é um modelo conceitual para simulação de gestão de projeto de software baseada em multiagentes, que é definida e projetada a partir da solução do problema identificado na seção 1.1 do capítulo 1.

A figura 3.2 ilustra o ProMabs que contém cinco componentes principais: o filtro de informação (*Information Filter*), o modelo de simulação para o planejamento (*Planning Simulation Model*), o modelo de simulação para a

execução (*Execution Simulation Model*), os cenários de execução (*Execution Scenarios*) e o analisador (*Analyze*). Detalhes deste modelo conceitual e seus componentes são discutidos separadamente para uma implementação em tecnologias específicas, como por exemplo, plataforma de programação multiagentes, ambiente de simulação e ferramenta de simulação.

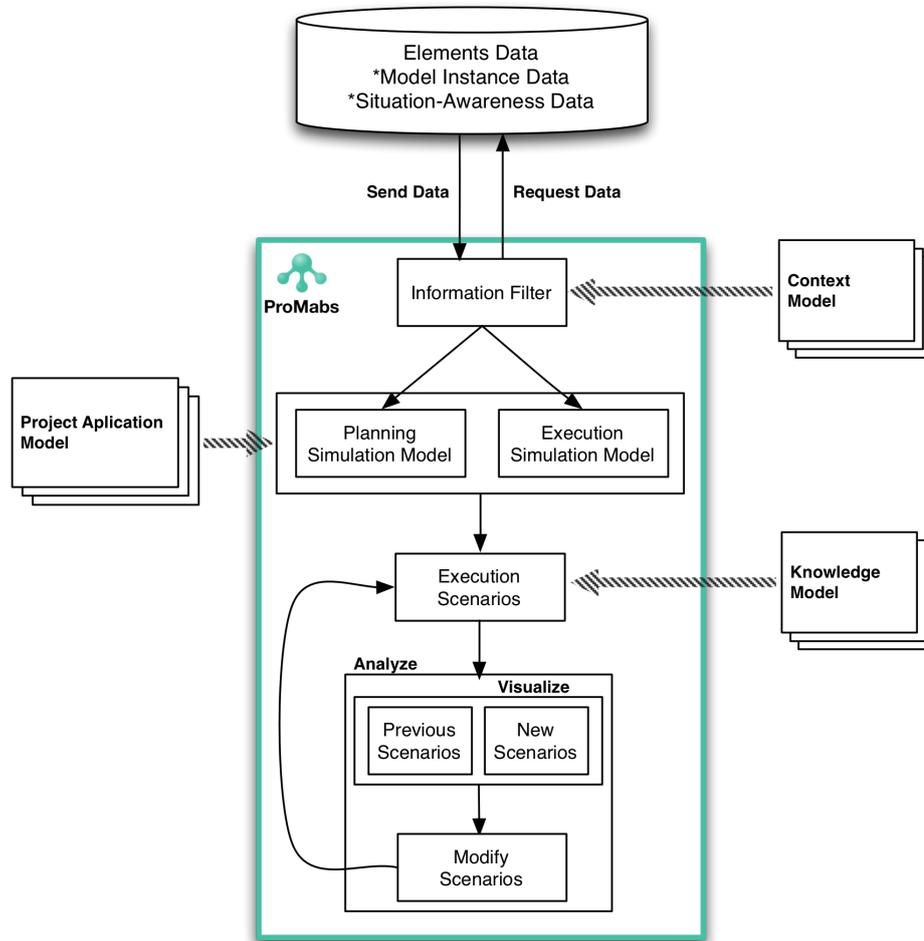


Figura 3.2: Os componentes do Modelo Conceitual ProMabs.

3.2.1

Filtro de Informação

O Filtro de Informação (*Information Filter*) como ilustrado na figura 3.2 é o componente responsável por solicitar e receber os dados do ambiente externo que são importantes para a simulação. Um elemento fundamental para Filtro de Informação é o Modelo de Contexto adotado para verificar as alterações que podem impactar o projeto. Por exemplo, quando temos na fase de planejamento uma atribuição de um recurso humano para uma tarefa e na fase de execução do projeto este recurso humano está inativo, precisamos

informar ao modelo de contexto (*Context Model*) que esta é uma mudança importante e na simulação terá um impacto no projeto. Este tipo de mudança no ambiente externo, a qual chamamos de contexto dinâmico, é comum em projetos e geralmente causa um impacto significativo em termos de custo ou tempo. Impacta o custo, pois o novo recurso humano pode ter um custo maior ou menor, ou impactando no tempo, pois o novo recurso humano poderá estar ocupado em outra atividade.

O modelo de contexto precisa ser definido para solicitar e receber as informações necessárias da base de dados e com isto instanciar ou parametrizar a simulação. Por exemplo, precisa-se definir qual é o contexto do projeto, no caso de projeto de software que aplica-se uma Estrutura Analítica de Projeto (EAP), dividem-se as tarefas em pacotes de trabalhos. Cada tarefa contém informações importantes para sua execução, como sua predecessora, duração, habilidade exigida e nível de conhecimento recomendado para sua execução. Sendo estes, alguns dos exemplos de informação relevantes sobre o qual o ProMabs e seus componentes são estruturados para serem robusto o suficiente para inserir estas informações, e caso necessário, adaptá-las. Para isso, o modelo de contexto usa um conjunto de pares com campo e valor, como estrutura para os dados da EAP. Desta forma, quando existe uma alteração de contexto (contexto dinâmico), o componente percebe e solicita tais alterações, e assim, modifica o contexto na simulação.

3.2.2

Modelo de Simulação para o Planejamento e Execução

Os modelos de simulação para o Planejamento e para Execução que são o *Planning Simulation Model* e o *Execution Simulation Model*, respectivamente ilustrado na figura 3.2, são responsáveis por executar o modelo apropriado de acordo com o estado dos elementos informado pelo Filtro de Informação. Por exemplo, na seção 3.2.1 descrevemos que um projeto contém um conjunto de tarefas divididas em pacotes de trabalho e que para cada tarefa existe um conjunto de informações. Uma destas informações é o seu estado no ambiente externo, ou seja, se já foi executado ou não. Caso alguma das tarefas possua o estado de execução indicando positivo o modelo utilizado para a simulação é o *Execution Simulation*, pois esta tarefa terá que se manter na simulação, porém como item informativo que não poderá sofrer nenhuma simulação/alteração devido esta já ter sido executada no ambiente externo não permitindo mudanças. Isto acontece pelo fato que ela já foi concluída e não pode alterar suas atribuições. Por outro lado, todas as tarefas que estão

em estado de não execução no ambiente externo pode sofrer alteração, assim a simulação possui autonomia para alterá-la caso necessite. Para isto, utiliza-se o modelo para a planeamento (*Planning Simulation*).

Um elemento fundamental para os modelos de simulação de planeamento e execução é o modelo de aplicação de projeto (*Project Application Model*) ilustrado na figura 3.2. O elemento aplicação de projeto contém informações ou regras de como as tarefas serão executadas no simulador. Por exemplo, se aplicamos uma EAP que possui pacotes de trabalho com um conjunto de tarefas, e estas tarefas possuem ligações umas com as outras, como, os pacotes de trabalhos. Então, temos que a **tarefa A** é predecessora da **tarefa B** de forma sequencial, assim sendo, a **tarefa B** só pode começar quando a **tarefa A** termina, sendo este um exemplo de regras utilizada com as informações das tarefas. De forma semelhante, os pacotes de trabalho também podem ter ligações. Este é um tipo de aplicação que faz parte do modelo de aplicação de projeto.

3.2.3

Cenários de Execução

O Cenários de Execução (*Execution Scenario*) ilustrado na figura 3.2 é responsável pela criação dos cenários de execução da simulação de acordo com o modelo de simulação descrito anteriormente na subseção 3.2.2. Para isto, adota-se estratégias pertencentes ao seu modelo de conhecimento (*Knowledge Model*). Por exemplo, ao atribuir um recurso humano para uma tarefa em simulação de planeamento com o menor custo. Adota-se a estratégia de encontrar os recursos humanos com habilidade para realizar a tarefa e verifica-se qual destes recursos humanos tem o menor custo.

O modelo de conhecimento (*Knowledge Model*) contém as estratégias para execução da simulação. Um gerente de projeto ou um especialista cria as estratégias de acordo com seu conhecimento para determinadas situações, para posteriormente serem aplicadas durante a simulação. Cada estratégia tem um objetivo próprio e responde as parametrizações ou acontecimentos provenientes do ambiente externo.

O fundamento principal para criar os cenários de execução é que qualquer elemento do projeto que está no ambiente externo é um agente de software na simulação. Aproveitando-se assim as vantagens de simulação baseadas em multiagentes. Por exemplo, em uma EAP cada um de seus elementos é um agente. Então, temos os *agentes* do tipo projeto, tarefa, pacote de trabalho,

recursos humano e gerente de projeto. Com isto, cada tipo de *agente* adota uma arquitetura BDI (*Belief-Desire-Intention*) (Rao e Georgeff, 1991), que se baseia nas noções de crenças, desejos e intenções como atitudes mentais que orientam a realização de comportamentos dos agentes baseados nos elementos do mundo real. Assim, modelando a complexidade de projetos do mundo real de forma a individualizar as características de cada elemento do ambiente externo em agentes, com o objetivo de autonomia dos agentes na simulação. A seção 3.3 descreve mais detalhadamente as vantagens da utilização de multiagentes e a arquitetura BDI para cada tipo de agente.

3.2.4

Analizador

O Analizador (*Analyze*) é o componente responsável por apresentar recursos visuais da execução da simulação para os envolvidos nos projetos. Assim, auxiliando os responsáveis pela tomada de decisão a analisar os cenários resultantes da simulação. Por exemplo, se aplicarmos novamente uma EAP que já possui cenários de execução como o descrito na subseção 3.2.3, então cada elemento do projeto é um agente autônomo que assume um papel definido na EAP. Com isso, o componente oferece um suporte para visualizar a interação dos agentes e seus comportamentos. Assim, uma tarefa pode ser visualizada em termos de estado, visualizando qual o Agente Recurso Humano que está atribuído e executando uma tarefa, se um pacote de trabalho foi finalizado e qual o valor de uma tarefa em um determinado dia, por exemplo.

O analisador precisa possuir dois tipos de cenários, o já simulado (*Previous Scenarios*) e o que está sendo simulado (*New Scenarios*). Então, quando um cenário está simulado possui o estado final de cada agente e suas informações em relação aos dados fornecidos pela componente de cenários de execução. Por outro lado, o cenário que está sendo simulado apresenta as interações dos agentes ao longo do tempo. Por exemplo, após a definição do cenário de execução entrar para uma simulação de cenários. O componente analisador irá demonstrar essa execução para auxiliar na tomada de decisão em termos de tempo. Após a finalização da simulação, o cenário simulado auxilia também a tomada de decisão pois demonstra todas as informações ao finalizar o cenário em termos de custo final ou total de tempo, por exemplo.

Um elemento importante para o componente analisador é a modificação de cenário (*Modify Scenarios*). A possibilidade de modificar o cenário facilita a gera novas opções de cenários para execução. Por exemplo, na primeira

criação de cenário de execução optou-se por estratégia de menor custo e agora irá modificar para a estratégia de melhor recurso humano para execução desconsiderando o custo. Com isto, simulando-se uma nova opção para execução do projeto, gera-se mais informações para a tomada de decisão.

3.3

Simulação baseada em Multiagentes para Gestão de Projetos de Software

O ProMabs é baseado em agentes atuando como elementos dos Projetos, como mencionado anteriormente. Os agentes começam sua atuação no componente de Execução de Cenário descrito na subseção 3.2.3 e atua até a finalização da simulação do projeto, ou seja, quando todas as tarefas e os pacotes de trabalho do projeto estão finalizadas.

A fim de modelar a atuação dos agentes em ProMabs, uma linguagem de modelagem chamada MAS-ML (Silva et al., 2008) foi utilizada para abstrair os aspectos dinâmicos e estruturais dos Agentes em ProMabs. O MAS-ML estende UML com base nas propriedades do metamodelo de TAO (Silva e Lucena, 2004). Os diagramas estruturais que compõem a MAS-ML são o diagrama de classes estendido e dois novos diagramas chamados diagrama de organização e diagrama de papel. Esses três diagramas, diagrama de classes, organização e papel, mostram todas as entidades e relacionamentos definidos em TAO.

A principal aplicabilidade do metamodelo TAO é oferecer um *framework* conceitual para entender as abstrações e seus relacionamentos a fim de oferecer suporte ao desenvolvimento de Sistemas MultiAgentes (SMAs). O TAO conecta ontologia de abstrações consolidadas, como objetos e classes, a outras abstrações como agentes, papéis e organizações, que são as bases da engenharia de software orientada a objetos e agentes (Silva e Lucena, 2004). A figura 3.3 ilustra os relacionamentos e as entidades do TAO, sendo estes a base para o ProMabs. Detalhes das entidades do ProMabs são discutidos separadamente para uma implementação em um ambiente específico de projetos de softwares.

O objetivo para a utilização da modelagem em MAS-ML é auxiliar um projetista a instanciar o ProMabs descrevendo assim o que é necessário ser especificado para sua instanciação. Para isto iremos utilizar o domínio de projeto de software que utiliza um Estrutura Analítica de Projeto (EAP) como ferramenta para gestão de projeto de software.

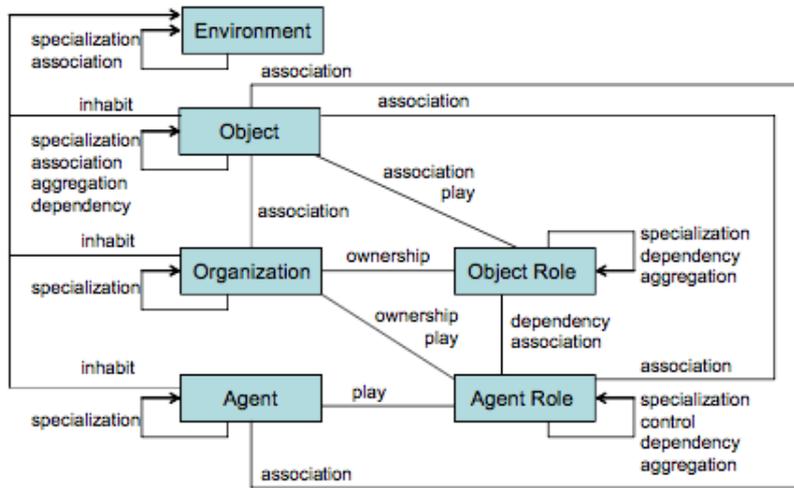


Figura 3.3: Os relacionamentos e as entidades do TAO (Silva, 2004).

3.3.1

Domínio de Projeto de Software com EAP

Os projetos de software possuem geralmente o escopo que contém os pacotes de trabalhos e as tarefas necessárias para o desenvolvimento do Software. Desta forma, o conjunto de pacotes de trabalhos, de tarefas e as propriedades de ambas formam uma EAP (Tausworthe, 1980).

Uma EAP geralmente é organizada por níveis, no primeiro nível está o projeto como um todo e em seus níveis subseqüente estão os pacotes de trabalhos que possuem tarefas a serem realizadas para conclusão do pacote de trabalho. Conseqüentemente com a conclusão dos pacotes de trabalho, o projeto é finalizado. A figura 3.4 ilustra esta estrutura. Para cada elemento da EAP existe um conjunto de propriedades com informações para realização do mesmo. Por exemplo, um pacote de trabalho tem um conjunto de tarefas, cada tarefa é realizada por um recurso humano, que possui as características necessárias para realizar essa tarefa. O gerente de projeto tem como uma de suas atribuições, alocar recursos humanos para cada tarefa. Porém, na fase de execução do projeto a EAP geralmente muda. Esta dinâmica não é trivial de ser modelada. Existe a necessidade de identificar e modelar estes elementos e sua dinâmica para um suporte à simulação com elementos que estão em constante mudança, como por exemplo em gestão de projetos de software. O foco do ProMabs é oferecer um suporte para esta identificação e modelagem.

O ProMabs cria agentes do tipo projeto, gerente de projeto, pacote de trabalho, tarefa e recursos humano. Além disto pode criar objetos necessários para a simulação do projeto, como por exemplo, calendário do projeto ou de

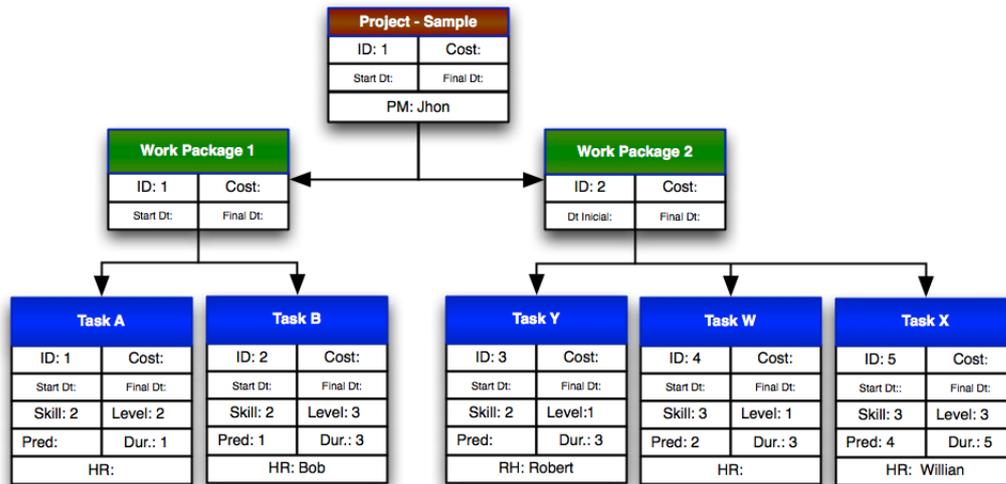


Figura 3.4: Exemplo de uma EAP.

um agente específico no qual o agente irá atuar quando estiver disponível de acordo com o seu calendário.

O domínio de Projeto de Software aplicando EAP é expresso seguindo a abordagem de modelagem definida por MAL-ML (Silva e Lucena, 2004), com o objetivo de criar diagramas estruturais e dinâmicos para este domínio no ProMabs e seus elementos.

3.3.2

Diagramas Estruturais do ProMabs para Projetos de Software

Segundo Silva (Silva e Lucena, 2004), podemos usar o diagrama de organização, papel e classes para modelar as principais partes estruturais de sistemas multiagentes. O diagrama de organização precisa definir as classes de organização e ambiente, de papel de objeto e agente, de agente e classes que estão exercendo as classes de papel. A figura 3.5 ilustra a classe da organização principal e do ambiente, junto com as classes de papel, de agente e classes que pertencem ao domínios de projeto de software.

Como ilustrado na figura 3.5, temos a EAP (WBS) como Organização principal que está no ambiente de Projeto de Software (*Software Project*). Além disto, foram modelados cinco tipos de agentes: Agente Projeto (*Project Agent*), Agente Pacote de Trabalho (*WorkPackage Agent*), Agente Tarefa (*Task Agent*), Agente Gerente de Projeto (*Project Manager Agent*) e Agente Recurso Humano (*Resource Human Agent*). O diagrama também ilustra as classes de papéis exercidas por agentes supracitados e também o objeto cronograma (*Schedule*) e papel para esse objeto, o horário de trabalho (*Worker*

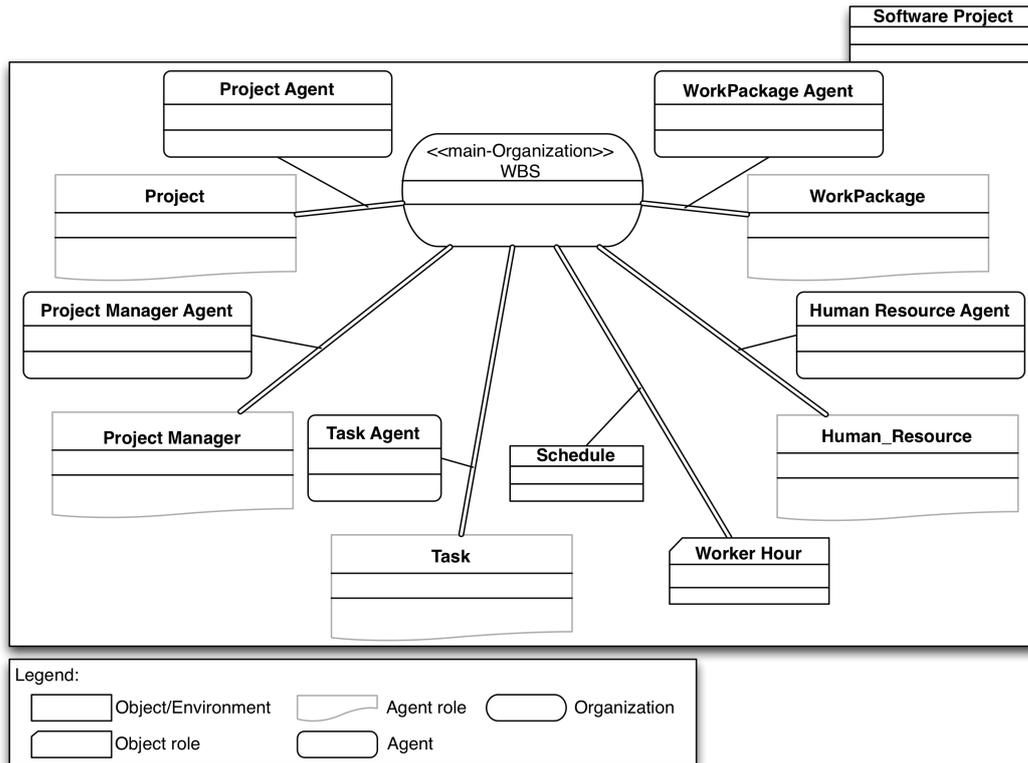


Figura 3.5: Diagrama de Organização.

Hour).

Projeto de Software - Ambiente

O Ambiente Projeto de Software armazena os elementos da EAP, como todos os projetos, pacotes de trabalho, tarefas, recursos humanos e gerentes de projetos. Possuindo assim todas as informações para instanciar os elementos como agentes. Estas informações são carregadas através do ambiente externo, como discutido na seção 3.1. Como ilustrado na figura 3.6 o principal atributo do ambiente é a árvore do projeto que é a estrutura de uma EAP.



Figura 3.6: Projeto de Software - Ambiente.

EAP - Organização Principal

A Organização Principal EAP (*WBS*) representa uma EAP de um Projeto de Software. Porém, podemos também ter uma EAP de um projeto principal com várias sub-organizações de EAPs de sub-projetos, formando um portfólio ou programa de projetos. Entretanto, para simplificar o nosso exemplo focamos apenas em um Projeto que possui apenas uma EAP. O principal *objetivo* da organização EAP é sua execução de forma a respeitar as propriedades de estrutura pertencentes entre os elementos Projetos, Pacotes de Trabalhos e Tarefas. Para isto, a organização EAP define *planos* para (i), criar os elementos, (ii) conectar os elementos e (iii) atualizar informações dos elementos. Para garantir que a organização EAP receba as informações relativas aos elementos é definido o *axioma* Árvore do Projeto. As *crenças* da organização EAP estão relacionadas às informações relativas aos projetos, pacotes de trabalho e tarefas. A figura 3.7 ilustra a Organização Principal EAP.

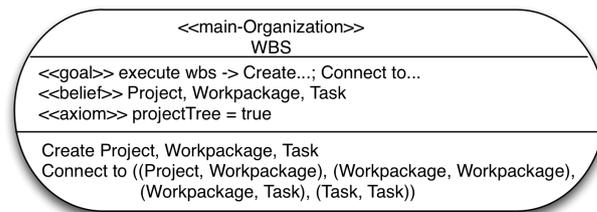


Figura 3.7: EAP - Organização Principal.

Papéis de Agentes e Objetos

Conforme já ilustrado na figura 3.7, a Organização Principal EAP define os papéis de agentes Projeto (*Project*), Pacote de Trabalho (*WorkPackage*), Tarefa (*Task*), Recurso Humano (*Human Recource*), Gerente de Projeto (*Project Manager*). Além disto, o papel de objeto horário de trabalho (*Worker Hour*).

A figura 3.8 ilustra a classe de papel **Projeto**, cujo *objetivo* é finalizar os pacotes de trabalho. Para alcançar o seu objetivo, o papel autoriza os pacotes de trabalho a iniciar sua execução, porém apenas os pacotes de trabalho de primeiro nível, ou seja, os pacotes de trabalho sem pacote de trabalho predecessor. O seu *dever* é verificar o estado dos pacotes de trabalho. O papel define um *protocolo* que verifica o estado de seus pacotes de trabalhos, ou seja, se eles foram finalizados. Quando um pacote de trabalho finaliza ele informa

ao projeto que possui o *direito* de ser informado sobre a finalização. Assim, quando todos os pacotes de trabalhos forem finalizados o projeto é finalizado.

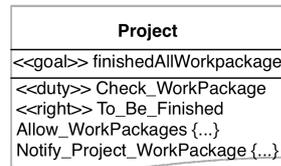


Figura 3.8: Classe do Papel de Agente - Projeto.

A figura 3.9 ilustra a classe de papel **Pacote de Trabalho** cujo *objetivo* é finalizar todas as suas tarefas. Para alcançar o seu objetivo, o papel autoriza as tarefas a iniciar sua execução, porém apenas as tarefas de primeiro nível, ou seja, as tarefas sem tarefas predecessoras. O seu *dever* é verificar o estado das suas tarefas. O papel define um *protocolo* que verifica o estado de suas tarefas, ou seja, se elas foram finalizadas. Quando uma tarefa finaliza, ela informa ao pacote de trabalho que possui o *direito* de ser informado sobre a finalização. Assim, quando todos as tarefas forem finalizadas o pacote de trabalho é finalizado.

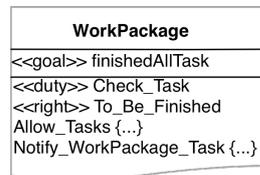


Figura 3.9: Classe do Papel de Agente - Pacote de Trabalho.

A figura 3.10 ilustra a classe de papel **Tarefa** cujo *objetivo* é ser executada. Para alcançar o seu objetivo, o papel informa o recurso humano que pode executar a tarefa, porém, se não possuir um recurso humano, o papel solicita ao gerente de projeto que atribua um recurso humano para execução. O seu *dever* é informar ao pacote de trabalho o seu estado. O papel define um *protocolo* que verifica o seu estado e informa o pacote de trabalho quando finalizado. O papel possui o *direito* de ter um recurso humano para sua execução.

A figura 3.11 ilustra a classe de papel **Recurso Humano** cujo *objetivo* é executar uma tarefa. Para alcançar o seu objetivo, o papel precisa estar disponível para execução. Porém se estiver indisponível, a tarefa fica aguardando até que o seu estado fique disponível, podendo então executar a tarefa na qual foi solicitado. O seu *dever* é informar a tarefa o seu estado disponível ou indisponível. O papel define um *protocolo* que notifica a tarefa

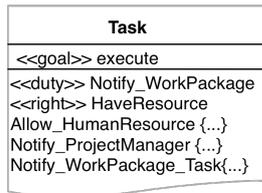


Figura 3.10: Classe do Papel de Agente - Tarefa.

quando é finalizada. O papel possui o *direito* de executar tarefa apenas em seu horário de trabalho.

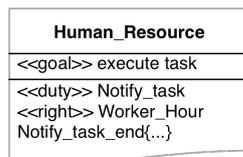


Figura 3.11: Classe do Papel de Agente - Recurso Humano.

A figura 3.12 ilustra a classe de papel **Gerente de Projeto** cujo *objetivo* neste exemplo é apenas definir recurso humano para tarefas que solicitam o recurso. Para alcançar o seu objetivo, o papel atribui um recurso humano como resposta para solicitação. O seu *dever* é escolher um recurso humano de acordo com uma estratégia definida. O papel define um *protocolo* que notifica um recurso humano sobre execução de uma tarefa. O papel possui o *direito* de conhecer todos os recursos humanos.

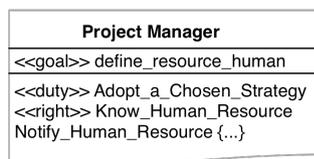


Figura 3.12: Classe do Papel de Agente - Gerente de Projeto.

A figura 3.13 ilustra a classe de papel de objeto **Horario de Trabalho** que permite que o recurso humano defina seu horário de trabalho, em dias da semana e horários a longo do dia, assim como também intervalos durante o dia de trabalho, porém retorna apenas se em um determinado tempo o recurso está disponível ou não.

Diagrama de Papel

Segundo Silva (Silva, 2004), um diagrama de papel possui os relacionamentos entre as classes de papel e classes. A figura 3.14, ilustra

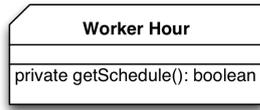


Figura 3.13: Classe do Papel de Objeto - Horário de Trabalho.

o diagrama de papel do domínio exemplificado de Projeto de Software que utiliza EAP. Assim, temos relacionamentos associativos (association) entre os papéis que trocam mensagens, e também temos relacionamento de controle entre um papel que controla um outro papel. Então, Projeto está associado com Pacote de trabalho e com Gerente de projeto. Pacote de Trabalho está associado com Tarefas. Já a Tarefa está associada com Gerente de Projeto e controla Recurso Humano. Já Recurso Humano está associado com Horário de Trabalho e pode ser controlado também pelo Gerente de Projeto.

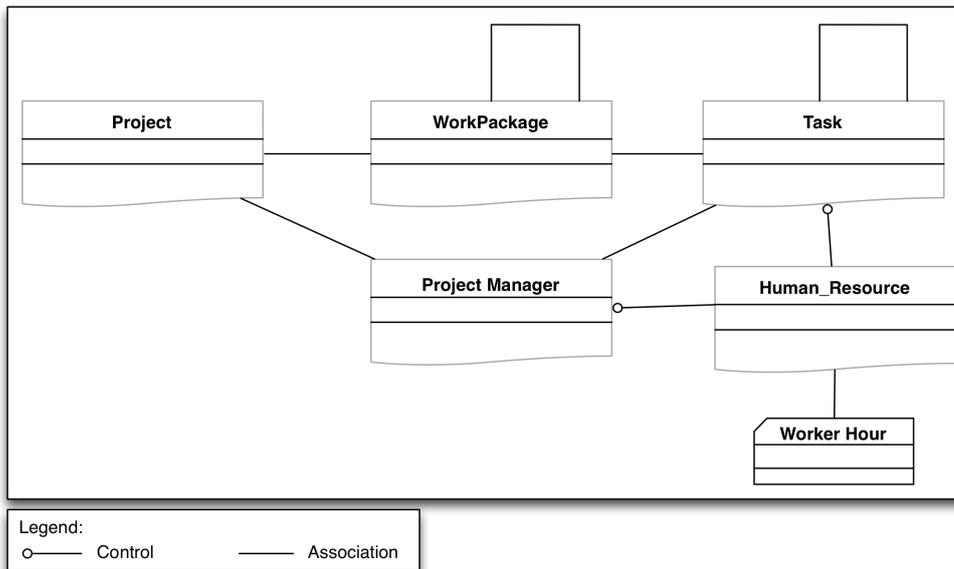


Figura 3.14: Diagrama de Papel.

Classes e Agentes

Conforme descrito anteriormente há cinco elementos principais na EAP. Estes elementos já foram discutidos em termos de papéis, porém é necessário também descrever os agentes que exercem estes papéis. A classe Agente Projeto (*Project Agent*) ilustrada na figura 3.15 representa um projeto que tem como *objetivo* a finalização de todos os seus pacotes de trabalho. Para isto, possui um *plano* que verificar quais são os seus pacotes de trabalho e envia um mensagem autorizando sua inicialização. Além disto, a classe define como

crenças os seus pacotes de trabalho. As crenças são usadas para armazenar quais são os pacotes de trabalho e suas propriedades pertencentes ao projeto, pois pode existir vários projetos em simulação simultânea. Para alcançar seus objetivos, o Agente Projeto precisa exercer os papéis de Projeto, já discutidos anteriormente.

Para exemplificar, temos uma lista de Pacotes de Trabalho e suas propriedades que estão em suas crenças, o Agente Projeto dispara um plano para verificar quais são os seus pacotes de trabalho. Em seguida, envia uma mensagem para os agentes pacotes de trabalho iniciarem. Ao finalizar, o Agente Projeto recebe uma notificação de finalização do pacote de trabalho.

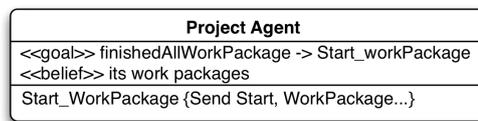


Figura 3.15: Classe Agente Projeto.

A classe Agente Pacote de Trabalho (*WorkPackage Agent*) ilustrada na figura 3.16 representa um pacote de trabalho que tem como *objetivo* a finalização de todas as suas tarefas. Para isto, possui o *plano* que verifica quais são as suas tarefas e envia uma mensagem autorizando sua inicialização. Além disto, a classe define como *crenças* as suas tarefas. As crenças são usadas para armazenar as suas tarefas e suas propriedades pertencentes ao pacote de trabalho, pois existem várias tarefas na simulação e o pacote de trabalho só pode autorizar as suas tarefas. Para alcançar seus objetivos, o Agente Pacote de Trabalho precisa exercer os papéis de Pacote de Trabalho, já discutido anteriormente.

Para exemplificar, temos uma lista de tarefas e suas propriedades que estão em suas crenças, o Agente Pacote de Trabalho dispara um plano para verificar quais são as suas tarefas. Após isto, envia uma mensagem para os Agentes Tarefas iniciarem, ao finalizarem o Agente Pacote de Trabalho recebe uma notificação de finalização das tarefas.

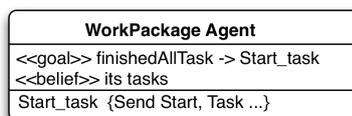


Figura 3.16: Classe Agente Pacote de Trabalho.

A classe Agente Tarefa (*Task Agent*) ilustrada na figura 3.17 representa uma tarefa que tem como *objetivo* ter um recurso. Para isto, possui o *plano* que procura um recurso humano. Além disto, a classe define como *crenças* as suas propriedades, como por exemplo, tarefa predecessora, habilidade exigida e duração, porém não se limitando a apenas estas. Para alcançar seus objetivos, o Agente Tarefa precisa exercer os papéis de tarefa, já discutido anteriormente.

Para exemplificar, temos uma tarefa e suas propriedades que são suas crenças, o Agente Tarefa dispara um plano para verificar se possui um recurso humano. Caso possua, envia uma mensagem autorizando o Agente Recurso Humano a executar a tarefa. Caso não possua, envia uma mensagem para o Agente Gerente de Projeto solicitando um recurso humano apropriado para execução da tarefa. Após isto, o Agente Gerente de Projeto envia uma mensagem para o Agente Recurso Humano para executar a tarefa. Quando o Agente Recurso Humano finaliza, o Agente Tarefa recebe uma notificação de finalização da Tarefa.

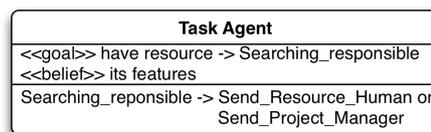


Figura 3.17: Classe Agente Tarefa.

A classe Agente Recurso Humano (*Human Resource Agent*) ilustrada na figura 3.18 representa um recurso humano que tem como *objetivo* executar as tarefas para qual foi solicitado. Para isto, possui o *plano* que verifica se tem tarefa e autorização para execução. Além disto, a classe define como *crenças* as propriedades do Recurso Humano, como, habilidade e custo, porém não se limitando apenas a estas. Também possui como *crença* um horário de trabalho. Para alcançar seus objetivos, o Agente Recurso Humano precisa exercer os papéis de Recurso Humano, já discutido anteriormente.

Para exemplificar, temos um Recurso Humano e suas propriedades que são suas crenças, o Agente Recurso Humano é solicitado para executar uma tarefa e dispara um plano para carregar as propriedades das tarefas na qual foi solicitado executar. Após isto, executa a tarefa respeitando as propriedades das tarefas e seu horário de trabalho. Ao finalizar envia uma mensagem para o Agente Tarefa informando sobre a finalização.

A classe Agente Gerente de Projeto (*Project Manager Agent*) ilustrada na figura 3.19 representa um gerente de projeto que tem como *objetivo* alocar recurso humano, quando necessário. Para isto, possui o *plano* que verifica quais

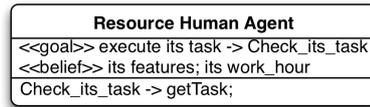


Figura 3.18: Classe Agente Recurso Humano.

são os recursos humanos disponíveis e que possua as características necessárias para executar a tarefa que solicitou um Recurso Humano. Além disto, a classe Agente Gerente de Projeto define como *crenças* a tarefa que solicitou um recurso humano, a estratégia que foi adotada, e a lista de recursos humanos. Estas crenças são usadas para armazenar qual é a tarefa e suas propriedades, a estratégia escolhida e a lista de recursos humanos e suas propriedades. Para alcançar seus objetivos, o Agente Gerente de Projeto precisa exercer os papéis de Gerente de Projeto, já discutido anteriormente.

Para exemplificar, temos uma tarefa e suas propriedades, uma estratégia escolhida e uma lista de recursos humanos disponíveis que estão em sua crença, o Agente Gerente de Projeto dispara um plano para encontrar um recurso humano apropriado para executar a tarefa, para isto, ele escolhe, seguindo uma estratégia que pode ser, por exemplo, por menor custo ou por um nível de conhecimento mais próximo ao desejado pela tarefa, sendo este nível de conhecimento uma propriedade da tarefa. Após isto, envia uma mensagem para o Agente Recurso Humano para executar a tarefa.

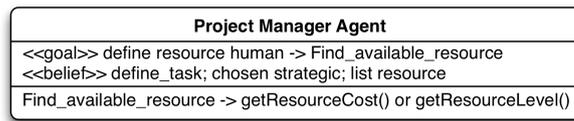


Figura 3.19: Classe Agente Gerente de Projeto.

A classe Calendário (*Schedule*) ilustrada na figura 3.20 representa um Calendário que tem como atributos um Agente Recurso Humano e uma lista de dias disponíveis para executar tarefas, seu calendário. Tem como método a configuração do seu calendário. A classe Calendário precisa exercer os papéis de horário de Trabalho, já discutido anteriormente. Para exemplificar, temos um recurso humano que possui um Horário de Trabalho, e é a classe Calendário que contém os dias de trabalho configurado para um determinado recurso humano.



Figura 3.20: Classe Calendario.

O diagrama de classes ilustrado na Figura 3.21 completa a modelagem dos aspectos estruturais do sistema descrito anteriormente.

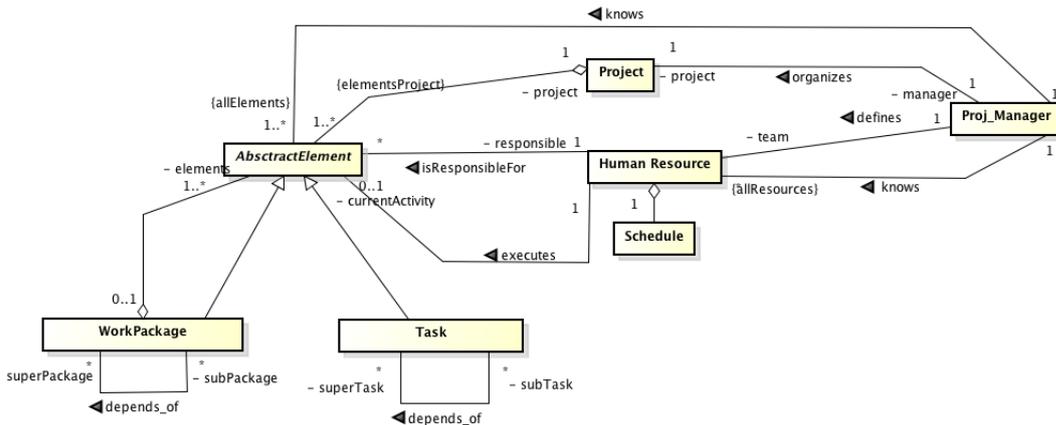


Figura 3.21: Diagrama de Classes.

3.3.3

Diagramas Dinâmicos do ProMabs para Projetos de Software

Segundo Silva (Silva, 2004), os aspectos dinâmicos de SMAs modelam as interações entre agentes, organizações, ambientes e objetos. Com o foco em expressar os aspectos dinâmicos da simulação do domínio de Projeto de Software, foi criado um diagrama de sequência dividido em duas partes. A primeira parte ilustrada na figura 3.22 tem o LES como um ambiente de Projeto de Software que está conectado a um repositório de dados, como visto na dinâmica entre ambiente externo e o ProMabs discutida na seção 3.1. A receber estes dados que estão em constante mudança, o componente Filtro de Informação passa para o componente apropriado que é o modelo de simulação para o Planejamento ou para a Execução, dependendo em qual tipo de modelo será necessário simular a EAP, que foram discutidos na seção 3.2. Após isto, seguimos estes passos ilustrado na figura 3.22:

- i. O Ambiente Projeto de Software (*Software Project*) o qual chamamos de LES recebe os dados e envia para a criação de uma Organização da EAP

(WBS) a qual chamamos de wbsMed;

- ii. A organização wbsMed envia uma mensagem para criar o Agente Projeto que exerce o papel de Projeto o qual chamamos de med, autorizando seu inicio.
- iii. O Agente Projeto Med, envia uma mensagem para criar os Agentes Pacote de Trabalho que exercem o papel de Pacote de Trabalho. A figura ilustra apenas um dos Agentes Pacote de Trabalho o qual chamamos de *SoftwareBuild*, autorizando seu inicio.
- iv. O Agente Pacote de Trabalho *SoftwareBuild*, envia uma mensagem para criar os Agentes Tarefas que exercem o papel de Tarefa. A figura ilustra apenas um dos Agentes Tarefas o qual chamamos de *SoftwarePrototyping*, autorizando seu inicio.
- v. Deste ponto em diante discutiremos a parte dois da figura 3.23. Porém o Agente *SoftwarePrototyping* envia um mensagem de retorno quando finaliza para o Agente *SoftwareBuild*, assim como todos os outros Agentes Tarefas fazem quando são finalizados.
- vi. Quando o Agente *SoftwareBuild* recebe as mensagens de finalização de todos seus Agentes Tarefas, envia uma mensagem de retorno para o Agente Med. Assim, quando todos os Agentes Pacotes de Trabalho finalizarem o Agente Projeto Med finaliza também.

PUC-Rio - Certificação Digital Nº 1212395/CA

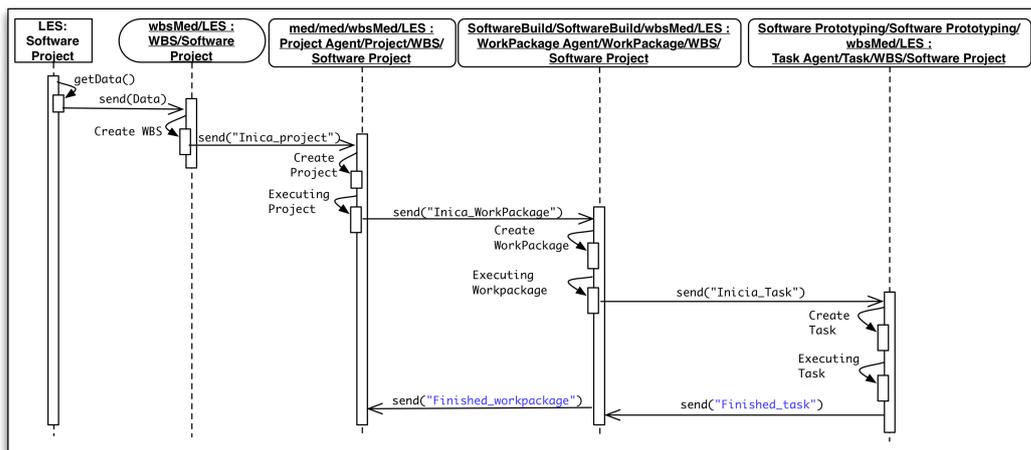


Figura 3.22: Diagrama de Sequencia - Parte 1.

A segunda parte do digrama ilustrada na 3.23 possui a sequencia para atribuição de um recurso humano, caso uma Tarefa não possua um recurso humano alocado para executa-la. Para isto segue:

- i. Após o Agente *SoftwarePrototyping* receber a autorização para iniciar, o Agente verifica se tem recurso humano. Neste exemplo o Agente *SoftwarePrototyping* não possui um recurso humano. Então solicita um recurso humano para o Agente Gerente de Projeto o qual chamamos de Elder.
- ii. Quando o Agente Elder recebe a mensagem, ele define através de uma estratégia adotada qual o recurso humano irá executar a tarefa. Após isto, envia as informações sobre a tarefa para o Agente Recurso Humano definido.
- iii. O Agente Elder, envia a mensagem para o Agente Recurso Humano. Neste exemplo foi atribuído para o João. O agente João executa a Tarefa *SoftwarePrototyping* e ao finalizar envia mensagem com as informações para o Agente Tarefa *SoftwarePrototyping*. A parti deste ponto em diante começam as mensagens de retorno, como mencionado anteriormente na primeira parte do diagrama.

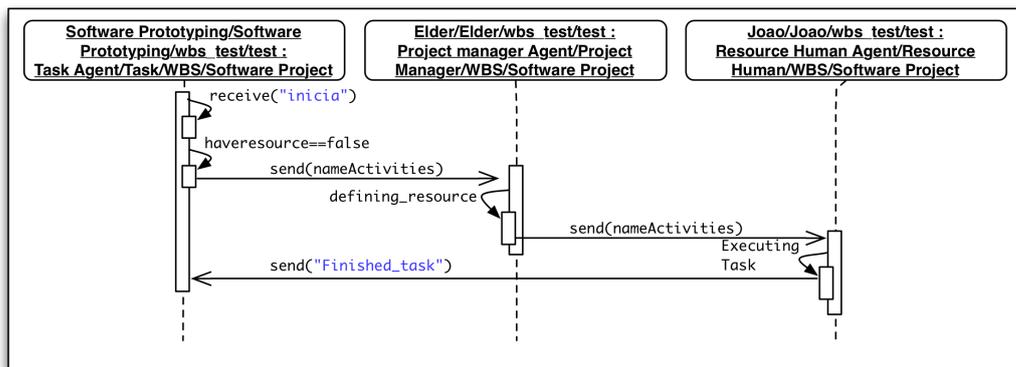


Figura 3.23: Diagrama de Sequencia - Parte 2.

Seguindo esta sequencia o ProMabs simula um EAP sendo executado, porém são necessários os dados de entrada vistos na figura 3.2, que são os dados de instanciação e situacionais da simulação. Estes dados devem seguir uma estrutura que permita ao ProMabs relacionar os dados com os tipos agentes configurados para simulação. A próxima subseção ilustra os dados para o exemplo de projetos de Software que utiliza EAP.

3.3.4

Dados de Instanciação e Situacional

Os Dados de Instanciação e Situacional são definidos de acordo com o que precisa ser estudado através dos fenômenos evidenciados na simulação. Dando continuidade ao exemplo de uma EAP é necessário identificar as propriedades de cada elemento da EAP. A figura 3.24 ilustra um diagrama Entidade-Relacionamento com suas entidades, atributos e relacionamento básico.

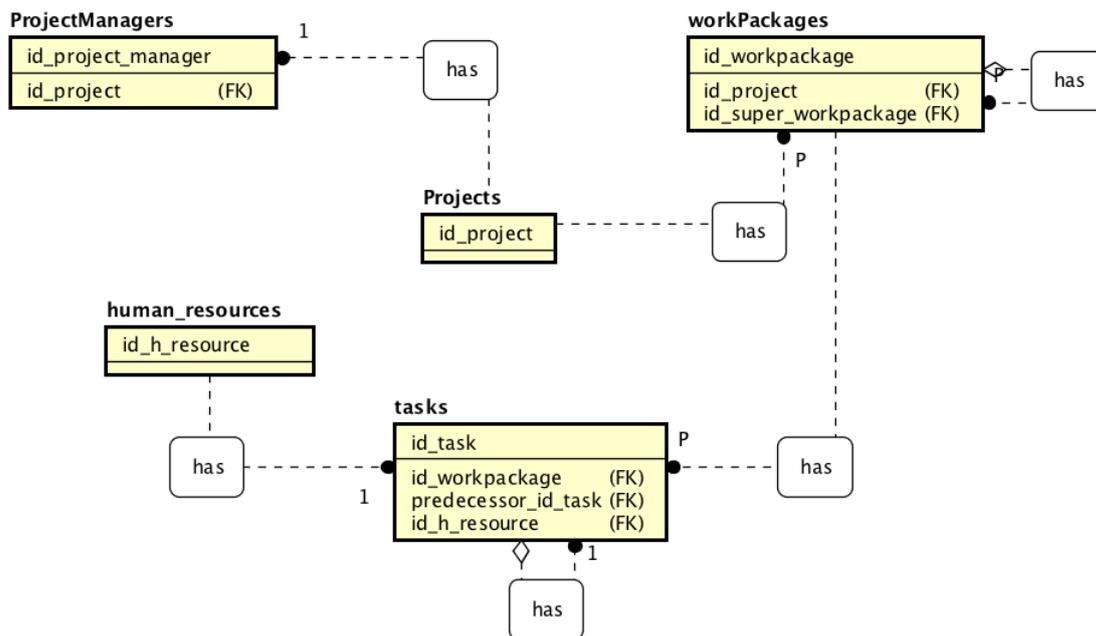


Figura 3.24: Diagrama Entidade-Relacionamento.

Como descrito anteriormente na seção 3.2, o ProMabs possui o componente Filtro de Informação que é responsável em verificar se os dados são para o modelo de simulação de planejamento ou execução. Além disto o Filtro de Informação contém o modelo de contexto que faz parte da identificação das alterações nos elementos da EAP. A estas alterações chamamos de contexto dinâmico, constantes em ambiente externo.

Os dados de Instanciação para o modelo de planejamento são geralmente os dados padrão para iniciar um planejamento de projeto. Como o projeto no ambiente externo não começou, o simulador executa a simulação sem restrições pois os elementos não foram finalizados e com isto pode existir alterações de planejamento. Para executar uma simulação que será utilizada no planejamento é necessário que Filtro de Informação receba: (i) Projetos (*Projects*); (ii) Cada Projeto possua um Gerente de Projeto (*Projectmanager*);

(iii) Cada Projeto tem um ou mais Pacotes de Trabalho (*WorkPackages*); (iii) Um Pacote de Trabalho pode possuir ou não outros Pacotes de Trabalho; (iv) Cada Pacote de Trabalho tem duas ou mais Tarefas (*Tasks*); (v) Tarefas podem ter Tarefas predecessoras ou não; (vi) E finalmente, uma lista de Recursos Humanos (*HumanResources*).

Os dados de Instanciação para o modelo de execução são geralmente os dados do projeto do ambiente externo que já possuem elementos finalizados, então o componente Filtro de Informação precisa identificar estes dados e o simulador precisa seguir de forma restrita o mesmo acontecimento do ambiente externo com os elementos finalizados. Por exemplo, quando os dados são para o planejamento um atributo com data de finalização da tarefa pode ser simulada e ter como resultado valores diferentes dependendo dos fenômenos que aconteceram na simulação. Já os dados de Instanciação para o modelo de execução não podem diferenciar dos acontecimentos do ambiente externo, se uma tarefa já terminou em uma determinada data no ambiente externo, o simulado precisa terminar na mesma data.

Os dados Situacionais são dados que representam alguma alteração no ambiente externo. Então quando temos um dado de instanciação que altera ao longo do projeto no ambiente externo, temos um dado situacional. Por exemplo, em uma determinada tarefa foi atribuída uma estimativa de duração, mas por alguma motivo essa estimativa foi alterada no ambiente externo, assim o simulador percebe esta alteração de contexto e simula com esta nova estimativa. Em consequência disto, pode existir alteração em outros elementos do projeto. Estas alterações já classificamos anteriormente como contexto dinâmico.

Para aplicarmos o ProMabs em uma simulação de gestão de projeto de software que tem como entrada uma EAP, com os dados supracitados, é necessário um Gerador que de suporte a diferentes configurações para representar Projetos do mundo Real. A próxima subseção descreve um gerador de Cenários de EAP que dá suporte a diferentes configurações.

3.3.5

Gerador de Cenários de EAP

O Gerador de Cenários de EAP cria dados necessários para instanciar domínios que aplicam EAP. Com o Gerador de Cenários, podemos instanciar várias EAP fictícias em vez de se concentrar em apenas uma EAP. Para isto, implementamos um gerador de EAP baseado no Gerador de Instancias de Alba

e Chicano (Alba e Chicano, 2007). Porém, diferenciando do trabalho deles, que se concentraram apenas em instanciar tarefas em um grafo, o Gerador de Cenários de EAP além de criar as tarefas e suas ligações também organiza por pacote de trabalho e projeto, sendo esta a principal diferença. Então, temos um grafo dentro de uma EAP.

Para criar as EAPs o usuário precisa definir um conjunto de parâmetros, tais como o número de projetos, número de pacote de trabalho, número de tarefas, o número de recursos humanos e etc. Assim, o Gerador de EAP é uma ferramenta de simples parametrização que são configuradas de acordo com a necessidade. O principal objetivo da utilização do gerador é mitigar o problema de possibilidades limitadas quando geramos as EAPs fictícias de forma manual. Com um gerador podemos criar várias EAPs fictícias de configurações diferentes, mitigando o problema de coletar várias EAPs de configuração diferentes do mundo real.

Os elementos da instância do gerador são: projetos, pacotes de trabalho, tarefas, recursos humanos e gerente de projeto. Estes elementos possuem um conjunto de parâmetros que determina como será configurada cada EAP. Existem três tipos de valores para os parâmetros que são data, número inteiro e número fracionado que representa uma porcentagem.

```
Arquivo de Configuração para Gerador de Cenário
#Projeto
n_projects = 1
date.start = 10/10/2015
date.end = 10/01/2016

#Pacote de Trabalho
n_workpackages = 4
workpackage_pred = 0.8

#Habilidade e Nível
n_skill_types = 10
n_skill_levels = 3

#Tarefa
n_tasks=10
task_min_skill = 1
task_max_skill = 1
task_min_duration = 1
task_max_duration = 7
task_hresource_prob = 1
task_pred = 1
task_max_pred = 1

#Recurso Humano
n_resources = 5
hresource_cost_min = 10
hresource_cost_max = 50
hresource_min_skill = 2
```

Figura 3.25: Exemplo de arquivo de configuração para instanciar o Gerador.

A figura 3.25 ilustra um exemplo de arquivo de configuração para instanciar o gerador de EAP. Neste arquivo contém todos os campos necessários e suas atribuições para gerar os cenários de EAP. Por exemplo, temos o campo `n.task` que é o número de tarefas que irá ser criado para serem distribuídas em pacotes de trabalho. Já o `task_pred` é a probabilidade de uma tarefa possuir uma tarefa predecessora. A tabela 3.1 ilustra todos os campos e seus parâmetros.

Tabela 3.1: Nome dos Campos do Arquivo de Configuração e seus Parâmetros Associados.

Nome do Campo	Parâmetro
n_projects	Quantidade de projetos
date_start	Data de início do projeto
date_end	Data de término do projeto
n_workpackages	Quantidade de pacotes de trabalho
workpackage_pred	Probabilidade de ter pacote de trabalho predecessor
n_skill_types	Número de tipos de habilidade
n_skill_levels=3	Número de níveis de habilidade
n_tasks=10	Quantidade de tarefa
task_min_skill	Número mínimo de habilidade por tarefa
task_max_skill	Número de habilidade por tarefa
task_min_duration	Quantidade mínima de duração
task_max_duration	Quantidade máxima de duração
task_hresource_prob	Probabilidade de tarefa possuir recurso humano
task_pred	Probabilidade de ter uma tarefa predecessora
task_max_pred	Número máxima de predecessora
n_hresource	Quantidade de recursos humanos
hresource_cost_min	Custo mínimo do recurso humano
hresource_cost_max	Custo máximo do recurso humano
hresource_min_skill	Número mínimo de habilidade do recurso humano

O gerador interpreta o arquivo de configuração e em seguida gera as instâncias ou cenários de EAPs. O pseudocódigo do gerador de EAP é ilustrado em partes para demonstrar como é criada a EAP de um Projeto. A figura 3.26 descreve a criação de Projetos e Gerente de Projeto, para cada projeto cria um gerente de projeto e os conecta. Após isto, são criados os pacotes de trabalho e em paralelo são conectados a um projeto. Em seguida conecta pacotes de trabalho a pacote de trabalho predecessor dependendo da parametrização.

A figura 3.27 descreve a criação das tarefas já atribuindo habilidades e níveis de conhecimento necessários para a atividade. Finalmente, algumas tarefas são conectadas a tarefas predecessoras dependendo da parametrização.

```

▷ Criando Projetos, Gerente de Projetos e Conectando
for  $i = 1$  to (n_project) do
   $project[i].manager = i$ 

▷ Criando Pacotes de trabalho para diferentes projetos e Conectando
for  $i = 1$  to (n_workpackages) do
   $p = random(n\_projects)$ 
   $wp = wp \cup workpackage(i, p)$ 

▷ Conectando Pacotes de Trabalho
for  $i = 1$  to (n_workpackages) do
  if  $uniform(0,1) < (workpackage\_pred)$  then
     $candidates = \emptyset$ 
    for  $j = 1$  to (n_workpackages) do
      if  $wp_i.project == wp_j.project$  and  $i \neq j$  then
         $candidates = candidates \cup wp_j$ 
       $wp_i.parent = choose\_random(candidates)$ 
      if  $loop(wp_i)$  then
         $wp_i.removeLoop()$ 

```

Figura 3.26: Pseudocódigo do Gerador de EAP - Parte 2.

```

▷ Criando tarefas
for  $i = 1$  to (n_tasks) do
   $task_i.workpackage = workpackages[i \bmod (n\_workpackages)]$ 
   $task_i.duration = random((task\_min\_dur),(task\_max\_dur))$ 
   $S = random((task\_min\_skill),(task\_max\_skill))$ 
  for  $j = 1$  to  $S$  do
     $type = random(1, (n\_skill\_types))$ 
     $level = random(1, (n\_skill\_levels))$ 
     $task_i.skill = task_i.skill \cup (type, level)$ 

▷ Conectando tarefas
for  $i = 1$  to (n_tasks) do
  if  $uniform(0,1) < (task\_pred)$  then
     $pred = random(1,(task\_max\_pred))$ 
    for  $p = 1$  to  $pred$  do
       $parent = random(1,(n\_tasks))$ 
      if  $sameProject(task_i, task_{parent})$  then
         $task_i.predecessors = task_i.predecessors \cup task_{parent}$ 
      if  $loop(task_i)$  then
         $task_i.removeLoop()$ 

```

Figura 3.27: Pseudocódigo do Gerador de EAP - Parte 2.

Na figura 3.28, temos a computação de habilidades e níveis necessários para executar o projeto, que será utilizado na atribuição para os recursos humanos. Após isto, os recursos humanos são criados e atribuídos custos para os mesmos. Em seguida são atribuídas as habilidades e níveis para os recursos humanos, caso necessário podendo existir um complemento de cota.

```

▷ Computando habilidades e níveis necessários para executar o projeto
reqLevel = [0]*(n_skill_types)
reqSkills = ∅
for task in tasks do
    for skill in task.skill do
        reqSkills = reqSkills ∪ task.skill
        if (skill[1] > reqLevel[skill[0]]) then
            reqLevel[skill[0]] = skill[1]

▷ Criando recursos humanos e atribuindo custos
for i = 1 to (n_hresources) do
    hresourcei.cost = random((hresource_cost_min),(hresource_cost_max))

▷ Atribuindo habilidades e níveis necessários para recursos humanos
for skill in reqSkills do
    reqLevel = reqLevel[skill]
    hr = hresource[skill mod (n_hresources)]
    hr.skills = hr.skills ∪ (skill, reqLevel[skill])

▷ Completando cota de habilidades de recursos humanos
for hr in hresources do
    while hr.skills < (hresource_min_skill) do
        candidates = skills - hr.skills
        skill = choose_random(candidates)
        level = random(1,(n_skill_levels))
        hr.skills = hr.skills ∪ (skill, level)

```

Figura 3.28: Pseudocódigo do Gerador de EAP - Parte 3.

Finalmente na figura 3.29, os recursos humanos são atribuídos a tarefa. Porém para isto fazendo uma comparação da tupla (habilidade, nível) da tarefa com recurso humano. Entretanto, podendo em alguns casos não

encontrar recursos humanos para todas as tarefas dependendo do arquivo de configuração.

```

▷ Atribuindo recursos humanos à tarefas
for task in tasks do
  for skill in task.skills do
    if uniform(0,1) > (task.hresource_prob) then
      continue
    for hr in hresources do
      if skill in hr.skills then
        task.hresource = task.hresource ∪ hr
      break
    
```

Figura 3.29: Pseudocódigo do Gerador de EAP - Parte 4.

Após a leitura do arquivo de configuração e percorrer o pseudocódigo ilustrado anteriormente, o gerador cria cinco arquivos, *projects*, *project_managers*, *workpackages*, *tasks* e *human_resources*, nesta ordem. Cada um deste arquivo representa um tipo de elemento da EAP. A estrutura de cada arquivo com um exemplo é ilustrado na tabela 3.2.

Tabela 3.2: Exemplos de Arquivos Gerados.

Tabela 3.3: Arquivo Projeto

id_project	name_project	dateStart	dateEnd
1	project1	10/10/15	10/01/16

Tabela 3.4: Arquivo Gerente de Projeto

id_project_manager	name_project_manager	id_project
1	manager1	1

Tabela 3.5: Arquivo Pacote de Trabalho

id_workpackage	name_workpackage	id_project	id_super_workpackage
1	workpackage1	1	3

Tabela 3.6: Arquivo Tarefa

id_task	name_task	id_workpackage	skills_type	skill_levels	duration
1	task1	1	1	1	5
	predecessor_id_task	id_h_resource	state		
	2	1	0		

Tabela 3.7: Arquivo Recurso Humano

id_h_resource	name_h_resource	skills_type	skill_levels	cost
1	humanResource1	1	1	10

O arquivo de configuração e o gerador estão disponível em nosso

repositório¹. Com isto, pode-se utilizar em outros estudos e também ser modificado para adicionar novos atributos necessários em estudo específicos. Os atributos ilustrados na tabela 3.2 foram selecionados a partir dos trabalhos relacionados listado no capítulo 2 e serve como bases para estudos de casos dos próximos capítulos.

3.4

Considerações Finais

Este capítulo apresentou o modelo conceitual ProMabs para projetar simulação de gestão de projetos de software baseada em agentes capaz de lidar com a dinâmica do ambiente externo. O ProMabs foi apresentado em fase inicial no simpósio de doutorado (Baia, 2015) e estendido neste capítulo. Foram descritos os componentes principais do ProMabs e como estes são instanciados em um domínio de Projetos de Software que aplica EAP para auxiliar na gestão do projeto. Neste exemplo da EAP, cada um de seus elementos é um tipo de agente que possui uma arquitetura BDI. Estes tipos de agentes são modelados em MAS-ML em termos estrutural e dinâmico. Porém, estes agentes precisam de dados para serem parametrizados e instanciados. Para isto, precisam seguir uma estrutura de dados descrita em um diagrama de Entidade-Relacionamento, para que o ProMabs possa identificar e representar na simulação. Finalmente, para mitigar o problema de dados para simulação foi descrito um gerador de cenários de EAP que é parametrizado de acordo com a configuração de EAP necessária, assim podendo gerar diversos e diferentes EAPs.

Como contribuição deste capítulo temos: (i) o ProMabs com seus componentes e a modelagem em MAS-ML para o domínio de Gestão de Projeto de Software que aplica EAP; (ii) E um gerador de Cenário de EAP. Com estas contribuições os próximos capítulos focam em implementar o ProMabs em uma plataforma de programação orientada a multiagentes chamada JaCaMo, em um ambiente de simulação chamado Cormas e finalmente em uma ferramenta de simulação chamada Anylogic.

¹<http://www.inf.puc-rio.br/~dbaia/generator>

4

Representação dos Elementos de Projeto de Software

Este capítulo apresenta uma aplicação do modelo conceitual ProMabs em uma plataforma de programação multiagentes, a qual chamamos de JaCaMoPM, com o objetivo de identificar e compreender os elementos incorporados na Gestão de Projeto de Software. A partir destes elementos, representamos suas características e seus relacionamentos, por meio do JaCaMoPM. Desta forma, compreendemos os elementos e suas alterações que geralmente ocorrem em projetos de software, para auxiliar na tomada de decisão. Para isto, utilizamos uma plataforma de programação multiagentes, para compreender como agentes executam tarefas, de acordo com uma organização social e de ambiente, para simular uma gestão de projeto de software, aplicando os processos de gestão de escopo. Neste contexto, este capítulo se concentra na primeira questão de pesquisa desta tese:

*QP1(i). Como **representar** os elementos necessários para simular gestão de projeto de software ?*

Para isto, na seção 4.1 introduzimos os elementos para gestão de projetos. Na seção 4.2 descrevemos os trabalhos relacionados que utilizam simulação para auxiliar projetos. Na seção 4.3 descrevemos como uma plataforma de programação multiagentes instancia o ProMabs. Na seção 4.4 apresentamos um exemplo ilustrativo através da instanciação de uma simulação, com foco na gestão de escopo. E finalmente na seção 4.5, apresentamos a conclusão e algumas discussões.

4.1

Gestão de Projetos de Software e seus Elementos

Gerenciamento de projetos de software tornou-se uma tarefa essencial em muitas organizações, especialmente com o aumento do tamanho e da complexidade de sistemas de software atuais. A gestão de projetos de software

envolve diversos elementos que incluem os recursos, os gerentes de projetos, as tarefas, as partes interessadas, os patrocinadores e o ambiente do projeto, e esses elementos muitas vezes mudam durante o processo de desenvolvimento. Estes elementos são relacionados uns aos outros de várias formas. Para uma simulação apoiar a gestão de projeto de software existe a necessidade de representar estes elementos e seus relacionamento. Porém, estes elementos possuem características próprias e seus relacionamentos dependem de alguns fatores, como por exemplo, um modelo de gestão que configura estes elementos.

Modelagem baseada em Multiagentes, provê a capacidade de capturar os atributos relevantes no processo de desenvolvimento de software, incluindo processos, produtos e pessoas, e como estes se relacionam com o planejamento, monitoramento e controle (Agarwal e Umphress, 2010). Em relação a gerenciamento de escopo, as simulações devem basear-se na representação das relações entre tarefas e suas dependências, e sequências de fluxo de tarefas, assim como, os recursos necessários para executar as tarefas. Os modelos de simulação também podem ser aplicados em simulações envolvendo recursos finitos, os atrasos e os custos estimados. Estas simulações são úteis para executar cenários que apoiem as tomadas de decisões, tais como as relacionadas com o custo e cronograma estimado. Os gerentes de projeto podem utilizar essas simulações para planejar e coordenar os seus esforços atentamente, para que seus projetos finalizem de acordo com o esperado. (Agarwal e Umphress, 2010). No entanto, há uma necessidade de representar os elementos e suas relações do projeto, com isto, fornecer suporte por meio de visualização dos cenários de simulação. Além disto, novos métodos que dão suporte a informação estática e dinâmica dos elementos, precisam ser introduzidos em Simulações Baseadas em MultiAgentes (SBMA) para apoiar a gestão do planejamento e a execução de projetos. Como mencionado anteriormente, informações estáticas, são os dados para o planejamento, e informações dinâmica, são informações que sofreram alterações devido as alterações que acontecem quando o projeto está sendo executado. Especificamente nos concentramos nas informações que abrangem as questões de escopo, englobando tempo, custo e recursos humanos.

Gestão de escopo de projetos de software precisam representar um grande número de elementos tais como, gerente de projeto, recursos, tarefas, e suas inter-relações complexas. Porém, a configuração destes elementos depende de modelos de processos de desenvolvimento de software específico, assim como, de um modelo de gestão, que são transversais. Ao longo do projeto, tanto estes elementos como suas relações sofrem alteração. Com isto, tornando

o gerenciamento de projeto complexo e dinâmico. As simulações precisam identificar e se adaptar a essas mudanças em tempo real e para isto os dados situacionais são necessários para capturar as alterações relevantes. Neste capítulo os dados situacionais são as modificações das representações dos elementos. Com o objetivo de representar os elementos por meio de uma programação multiagentes e verificar como esta representação simula uma gestão de projeto.

Estas representações são baseada no ProMabs, descrito no capítulo 3. Para isto, aplicamos o ProMabs em uma plataforma de programação orientada a multiagentes, o JaCaMo (Boissier et al., 2013), que utiliza uma combinação de plataformas para programar (i) os agentes e as suas interações, (ii) as organizações, e (iii) os ambientes. Como resultado desta instanciação, os envolvidos na tomada de decisão do projeto podem simular representações e seus resultados auxiliam em suas atividades de gestão.

4.2

Trabalhos Relacionados

O principal objetivo da representação, por meio de uma programação de sistema multiagentes, é auxiliar os envolvidos na tomada de decisão com um modelo de simulação que dê suporte, as atividades de gestão de escopo. Neste contexto, são três as principais linhas de investigação que estão relacionadas com este capítulo: (i) agentes no processo de gestão empresarial; (ii) agentes em gerenciamento de projetos; e (iii) ferramentas e aplicativos baseados em agentes. Em primeiro lugar, diferentes sistemas de *workflow* foram propostas para automatizar os processos de uma empresa (Muller et al., 2004), (Jennings e Wooldridge, 1998).

Em segundo lugar, os agentes de software estão sendo aplicados de forma diferentes para apoiar as atividades de gerenciamento de projetos. Existem trabalhos focados em auxiliar o processo de gestão (Nienaber e Barnard, 2007) e no agendamento de tarefas dentro da cadeia de fornecimento no ambiente de gerenciamento de projetos (Sauer e Applerath, 2003). Da mesma forma, os agentes foram usados industrialmente na gestão da cadeia de abastecimento (Wu e Simmons, 2000, Parunak et al., 1997). Além disso, métodos e ferramentas foram desenvolvidas para apoiar o planejamento e gestão de fluxo de trabalho, especialmente nos domínios de concepção (*design*) (Maurer, 1996). Alguns pesquisadores também se concentraram na gestão de recursos em organizações virtuais (Szymczak et al., 2007) e em um plano de

acompanhamento (Wu e Simmons, 2000). Planejamento para gerenciamento de projetos de software tem aplicado algoritmos adaptados de planejamento baseados em simulação (Joslin e Poole, 2005). Com relação ao controle e monitoramento, os agentes de software têm sido aplicados em vários locais para auxiliar o processos de engenharia de software distribuídos (Gaeta e Ritrovato, 2002). Em relação à garantia de qualidade, foi fornecido um framework com ambiente multiagentes para apoiar o processo e garantia de qualidade, em que um agente faz o papel de um gerente de projeto e ajuda atribuir e atividades de apoio, tais como testes e avaliações (Leung e Poon, 1999). Além disso, existem outras abordagens que dependem de agentes de software para apoiar a gestão de riscos em sistemas de robôs (Korb et al., 2003) e para prevenir os riscos técnicos em pequenas e médias empresas (Rigaud e Guarnieri, 2002).

Finalmente, várias ferramentas e aplicativos baseados em agentes são fornecidos, incluindo o trabalho em espaços *co-working* inteligentes (Kameas, 2010), infraestrutura de gerenciamento de máquina-a-máquina baseada em agente (Persson et al., 2011), a gestão do conhecimento de base ontológica para *workflows* de conhecimento intensivo em processos de negócios (Toledo et al., 2011), bem como a plataforma de programação baseada em multiagentes JaCaMo (Boissier et al., 2013) que estamos usando entre outras em nosso trabalho.

A Figura 4.1 mostra a relação entre os métodos existentes (a) e a nossa abordagem (b e c). Em (a) as informações sobre elementos no processo de gerenciamento de projeto são fornecidas, em seguida a simulação é executada e os resultados são verificados. Em nossa abordagem representado por (b) e (c) na Figura 4.1 apoiamos não só a criação das representações do processo, mas também as mudança para estas representações que são o resultado de interações humanas ou através da entrada de dados de tempo real e de dados situacionais. Pode existir a necessidade de modificar as representações e verificar o impacto dessa mudança. Com isto, as simulações dão suporte a auto-adaptação que envolve agentes buscando cumprir metas específicas, tais como tempo e custo, como descrito em (Baia et al., 2014). Finalmente, a nossa abordagem apoia a visualização da dinâmica do projeto. Neste capítulo, vamos mostrar resultados relacionados a uma abordagem que apoia (b). Nos próximos capítulos, nos concentramos em (c), incorporando simulações sincronizadas com o ambiente externo, com isto, apoiando dados situacionais e (auto)-adaptação. Para isto, é necessário entender os agentes, seu mecanismo social e como estes se organizam com o ambiente para simular a gestão de projetos de software.

Em geral, em comparação com outros trabalhos relacionados, a nossa

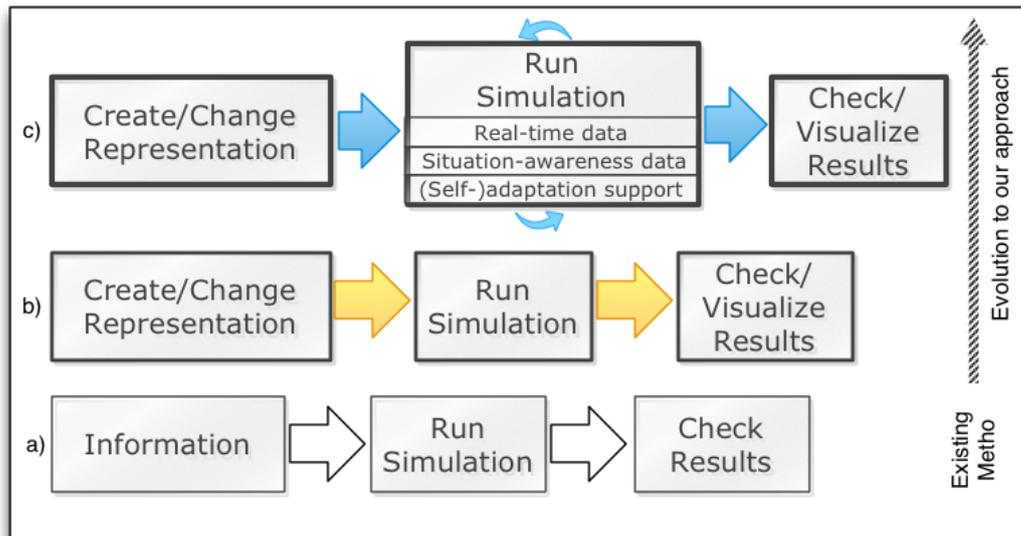


Figura 4.1: Trabalhos existente e evolução para nossa abordagem.

abordagem utiliza o ProMabs, aplicando os diferentes recursos previstos em JaCaMo, que utiliza dimensões para representar diferentes aspectos dos agentes e suas interações, organização e o ambiente. Ao final, estas representações usufruem dos recursos das dimensões do JaCaMo, para auxiliar no processo de desenvolvimento que é complexo e dinâmico, pois envolve relações e interações que devem ser explicitamente representadas. Desta forma, melhora-se o entendimento sobre inter-relacionamentos, que envolvem elementos como recursos, gerentes de projetos, tarefas e ambiente, com sua representações e visualizações. Assim, pode-se auxiliar os processos de gerenciamento de projetos de software que são complexos e dinâmicos.

4.3

ProMabs para um Plataforma de Programação MultiAgentes

O objetivo desta seção é aplicar o modelo conceitual ProMabs a uma plataforma de programação multiagentes. Com isto, pode-se compreender e representar os elementos necessários para gestão de projeto de software, no âmbito de gestão de escopo. O ProMabs descrito no capítulo 3 possui cinco componentes principais e é baseada em multiagentes. A figura 4.2 ilustra estes componentes. Instanciamos os componentes, modelo de simulação de planejamento, cenários de execução e analisador, em vermelho. Com o objetivo de auxiliar na criação de um modelo que incorpora os elementos de gestão projeto.

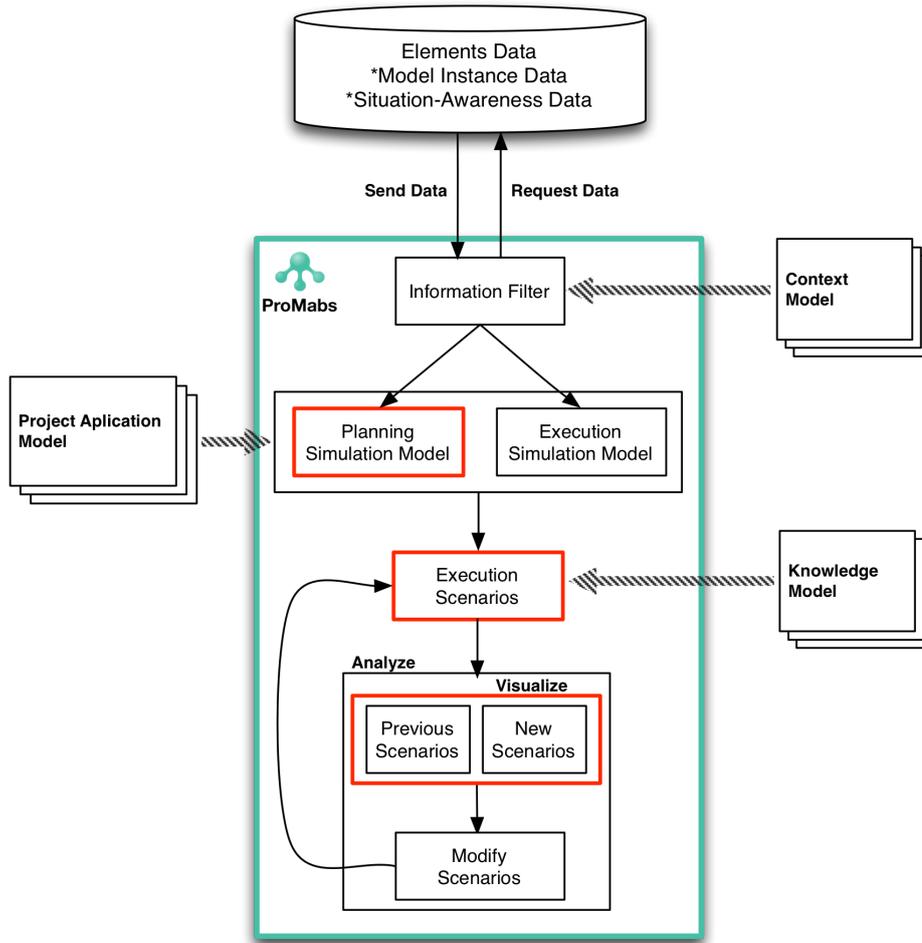


Figura 4.2: Componentes Instanciados do ProMabs.

Para auxiliar na compreensão e representação destes elementos, instanciamos os componentes supracitados do ProMabs em uma plataforma de programação multiagente, o JaCaMo. Porém, existiu a necessidade de criar uma extensão do JaCaMo para dar suporte ao componente analisador da simulação de gestão de projeto proposto pelo ProMabs. Este componente foi implementado para visualizar uma Estrutura Analítica de Projeto (EAP). A esta extensão chamamos de JaCaMoPM. Além disto, descrevemos como JaCaMoPM pode ser mapeado para os processos de gerenciamento de escopo baseado no Guia PMBOK (Project Management Body of Knowledge) (PMI, 2013), com os elementos identificados e representados, para auxiliar a tomada de decisão.

JaCaMo é uma plataforma para apoiar à programação orientada a multiagentes. Esta plataforma foi construída sobre três outras plataformas: Jason (Bordini et al., 2007) para a programação de agentes autônomos, Moise (Hubner et al., 2007) para programação de organizações de agentes, e

Cartago(Ricci et al., 2009) para programação de ambientes compartilhados. Como resultado, JaCaMo unifica programação de agentes, organizações e ambientes. Com isto, tem-se uma plataforma com três dimensões para instanciar os componentes do ProMabs. A figura 4.3 ilustra o meta-modelo das três dimensões, agentes, ambiente e organizacional.

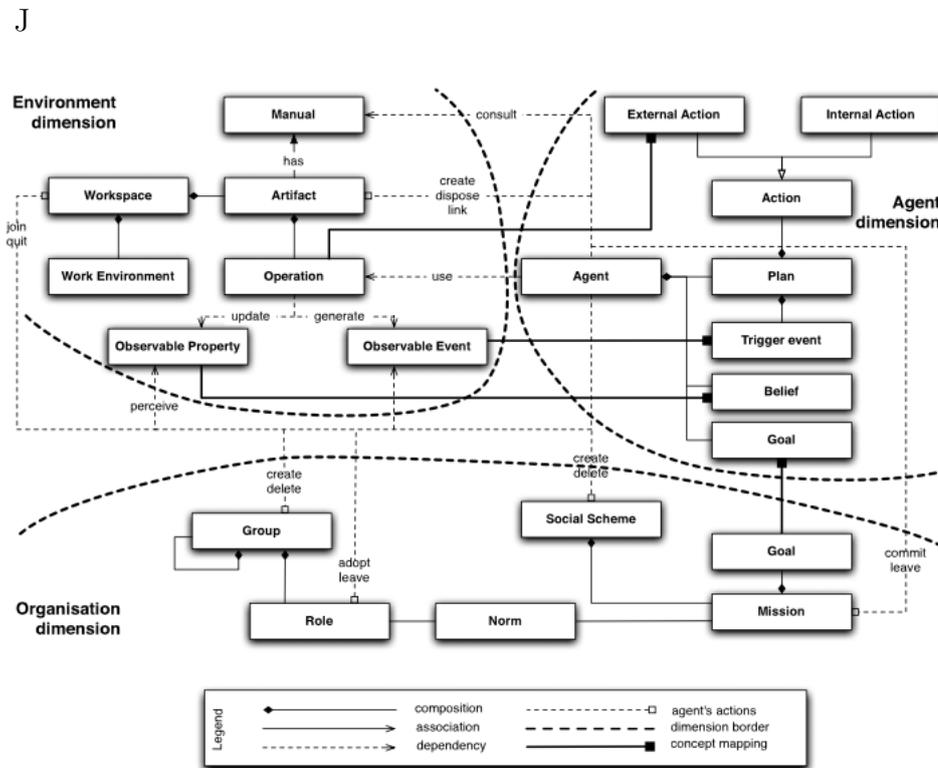


Figura 4.3: Meta-Modelo do JaCaMo (Boissier et al., 2013).

4.3.1

Dimensão Agente e ProMabs

Como mencionado no capítulo 3, o componente Modelo de Planejamento é responsável pela configuração do componente Cenários de Execução. As simulações são executadas de acordo com estas configurações que possuem informações sobre os elementos do projeto. Após isto, o componente analisador, que visualiza os cenários, auxilia os envolvidos na tomada de decisão na verificação do comportamento dos agentes simulados com estas configurações do projeto. Para o ProMabs, os elementos, recursos humanos e gerentes de projeto, são agentes que possuem como informação suas crenças, metas e planos. De acordo com Boissier et al (Boissier et al., 2013), um agente em Jacamo é uma entidade composta por (i) um conjunto de crenças representando o estado atual do agente e conhecimento sobre o ambiente do agente, (ii) um

conjunto de metas que correspondem às atividades que o agente tem de realizar, e (iii) um conjunto de planos que são cursos de ação, internos ou externos. Estes planos podem ser compostos dinamicamente pelos agentes, instanciados e executados para alcançar seus objetivos, podendo ser desencadeados por eventos. Os eventos podem estar relacionados às mudanças na base de crenças do agente ou nos seus objetivos. A figura 4.3 ilustra o meta-modelo do JaCaMo, com recursos da dimensão agente no lado direito superior. Por exemplo, um agente recurso humano possui crenças específicas para o estado disponível. Neste estado, o agente precisa executar uma das tarefas para qual foi atribuído. Com isto, alcança seu objetivo que é realizar a tarefa com base em uma estratégia ou plano. Geralmente o agente está indisponível, quando está realizando uma tarefa. Quando suas crenças se modificam para estado indisponível, o agente não pode executar outra tarefa.

4.3.2

Dimensão Ambiente e ProMabs

Em Jacamo (Boissier et al., 2013), a dimensão do ambiente é composta por um ou mais espaços de trabalho que são utilizados para definir a topologia do ambiente. Cada área de trabalho é um lugar lógico que contém um conjunto dinâmico de artefatos que definem a estrutura e comportamento do ambiente, representando os recursos e ferramentas que os agentes podem criar, descobrir e usar em tempo de execução. Cada artefato fornece um conjunto de operações e propriedades observáveis que definem a interface de uso de um artefato. Essa interface é usada por agentes para observar e operar os artefatos. A execução das operações pode gerar atualizações nas propriedades observáveis e eventos observáveis específicos. A figura 4.3 ilustra o meta-modelo do JaCaMo, com o recurso da dimensão ambiente no lado direito superior. Por exemplo, em um projeto de desenvolvimento de software, os artefatos são tarefas que possuem como operação e propriedades. A execução da tarefa é a operação que o agente recurso humano pode realizar. As propriedades são as características observáveis da tarefa que o agente gerente de projeto observa para verificar e alocar um recurso humano apropriado para sua execução. Cada conjunto de tarefas estrutura uma área de trabalho, que em nosso exemplo é uma EAP. Esta EAP é o ambiente no qual os agentes interagem para execução da mesma. No ProMabs este ambiente faz parte do componente modelo de planejamento que será utilizado no componente cenário de Execução.

4.3.3

Dimensão Organizacional e ProMabs

Na dimensão organizacional do JaCaMo(Boissier et al., 2013), uma organização é descrita a partir do ponto de vista estrutural, em termos de grupo e papel de entidades, especificando o quanto e quais recursos são necessários para um projeto. Missão e meta das entidades são introduzidos a partir de um ponto de vista funcional em termos de um esquema social. A principal meta do projeto pode ser dividida em várias submetas. No entanto, existem missões que podem ser relacionadas às tarefas, e um conjunto de missões pode ser uma sub-meta. De um ponto de vista normativo onde os papéis são ligados a missões, é possível limitar o comportamento de um agente no que diz respeito a conjuntos de metas que devem ser alcançadas quando um agente for escolhido para entrar em um grupo e desempenhar um certo papel no grupo. No exemplo ilustrativo iremos descrever estes conceitos instanciados. A figura 4.3 ilustra o meta-modelo do JaCaMo, com recurso da dimensão organizacional na parte inferior. Por exemplo, um agente gerente de projeto possui a norma do tipo obrigação que está relacionada à missão, que deve ser gerenciada no desenvolvimento do projeto. Esta dimensão organizacional auxilia na configuração do componente modelo de planejamento proposto pelo ProMabs. Com isto, por exemplo, determinando configurações e aplicando missões para tipos de agentes. Temos que o agente gerente de projeto tem a missão de gerenciar o projeto. O agente recurso humano com o papel de testador tem a missão de conduzir testes no software.

Por fim, estas três dimensões incorporam recursos importantes para auxiliar a simulação de gestão de projetos de software, baseada em ProMabs, e seus relacionamentos complexos. Visto que JaCaMo depende dessas dimensões, usamos esta plataforma para apoiar e representar o relacionamento de elementos da gestão de projetos a partir de uma perspectiva orientada a agentes. Porém, JaCaMo não dá suporte ao componente analisador do ProMabs. Com isto, existe a necessidade de estender-lo para dar suporte à visualização de uma EAP, a qual chamamos de JaCaMoPM. Para verificar como o JaCaMoPM auxilia na representação da gestão de projeto de software, realizamos um estudo exploratório com um exemplo ilustrativo. Com isto, aplicamos uma abordagem baseada em técnicas e ferramentas de apoio para auxiliar os gerentes de projeto nos processos de escopo.

4.4

Exemplo Ilustrativo

De acordo com o PMBOK (PMI, 2013) "Gerenciamento do Escopo do Projeto inclui os processos necessários para assegurar que o projeto inclua todo o trabalho necessário, e somente o trabalho necessário, para completar o projeto com sucesso." Gerenciamento do escopo do projeto para o PMBOK, envolve seis processos, que são o Plano de Gerenciamento do Escopo, a Coleta de Requisitos, a Definição de Escopo, a Criação da Estrutura Analítica do Projeto (EAP), a Validação de Escopo, e o Controle do Escopo. Cada um destes processos tem as suas informações de entradas relacionadas com elementos do projeto. Com isto, ferramentas e técnicas são aplicadas as estas entradas para gerar resultado que auxiliam na gestão do escopo. A figura 4.4 ilustra estas entradas. Utilizando o JaCamoPM para formular as entradas de cada um destes processos do PMBOK. Assim, criando uma nova ferramenta de apoio ao gerente de projeto na tomada de decisões, com seus resultados da simulação. Além disto, auxiliando a compreensão sobre a utilização de agentes para simular gestão de projetos de software.

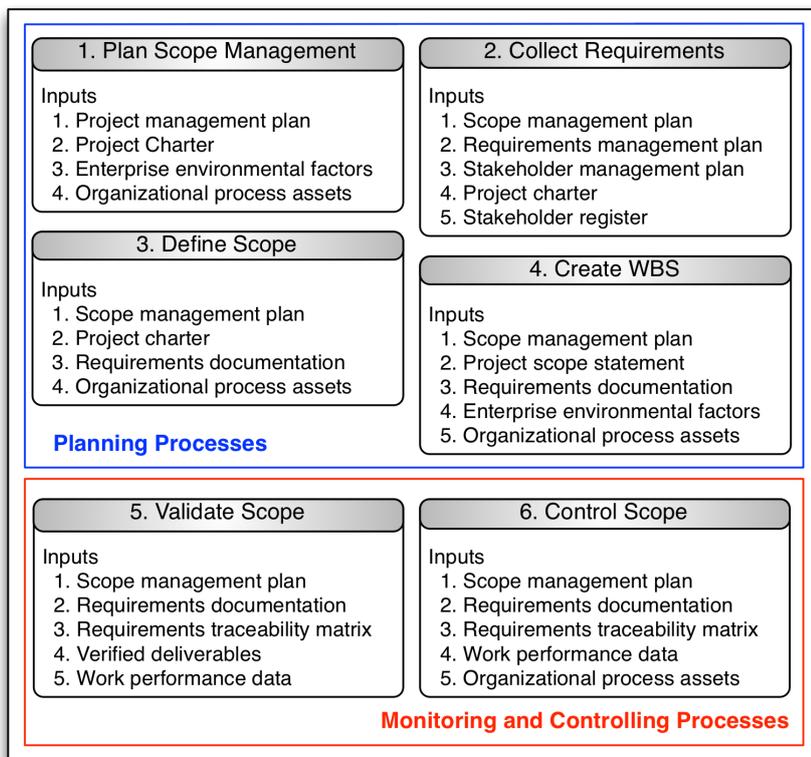


Figura 4.4: Grupos e Processos relacionados do PMBOK.

4.4.1

Planejar o Gerenciamento do Escopo

Planejar o Gerenciamento do Escopo é um processo da fase de planejamento que documenta como o escopo do projeto será definido, validado e controlado. De acordo com as práticas descritas no PMBOK (PMI, 2013), o principal benefício desse processo é fornecer orientação sobre como o escopo será gerenciado ao longo do projeto. A Figura 4.4 ilustra as entradas associadas a este processo.

Podemos utilizar os recursos das três dimensões mencionadas anteriormente para formalizar estas entradas para a simulação e, com isto, auxiliar na instanciação do componente modelo de planejamento da simulação. Por exemplo, podemos utilizar a dimensão organizacional para formalizar os papéis dos participantes do projeto. Esta formalização tem recursos relacionados com os fatores da empresa e ambiente. Por exemplo, para uma certa empresa, um projeto específico precisa possuir um grupo com no mínimo um gerente de projeto, um analista de requisito e um programador. A Listagem 4.1 ilustra esta formalização.

Listagem 4.1: Definição dos Papéis do Projeto.

```

1 <role-definitions>
2 <role id="project_owner" />
3 <role id="sponsor" />
4 <role id="building_resource" />
5 <role id="software_requirement_analyst" >
6   <extends role="building_resource" />
7 </role>
8 <role id="programmer" >
9   <extends role="building_resource" />
10 </role>
11 </role-definitions>
12
13 <group-specification id="project_group">
14 <roles>
15 <role id="project_owner" min="1" max="1" />
16 <role id="software_requirement_analyst" min="1" max="2" />
17 <role id="programmer" min="1" max="4" />
18 <role id="sponsor" min="1" max="4" />
19 </roles>
20 </group-specification>

```

A tag *role-definitions* define os papéis. Além disto, esta tag pode conter papéis com extensões. Por exemplo, o papel *software_requirement_analyst* estende *building_resource*. Desta forma, o *software_requirement_analyst* tem todas as características relacionadas com o *building_resource*, além de suas próprias características. Precisamos também definir grupos, com número de participantes por papéis para cada grupo. A tag *group-specification* atribui papéis a grupos e especifica o número de participantes.

O gerente de projeto pode usar estas representações e relacionamentos na simulação e com isto, apoiar a sua gestão de escopo, com uma configuração de papéis e suas quantidades para um projeto. Assim, identificando-se os papéis e suas quantidades de participantes é satisfatório para a meta do projeto, por meio do componente execuções de cenários.

4.4.2

Coletar os Requisitos

A coleta de requisitos é um processo da fase de planejamento que define, documenta e gerencia as necessidades e exigências dos *stakeholders* no cumprimento dos objetivos do projeto. De acordo com as boas práticas no PMBOK (PMI, 2013), o principal benefício desse processo é fornecer a base

para definir e gerenciar o escopo do projeto e do produto. A Figura 4.4 mostra as entradas associadas a este processo.

Os *stakeholders* têm um papel importante na gestão de projetos. A sua participação ativa pode contribuir para o sucesso do projeto, pois o projeto é criado para eles e com eles. Do ponto de vista dos *stakeholders*, precisamos descobrir e decompor as suas necessidades em requisitos específicos. No entanto, precisamos também formalizar estes requisitos, criando uma estrutura dos objetivos a serem alcançados, a fim de cumprir os requisitos. Para este efeito, podemos usar a dimensão organizacional e seus recursos, com isto, formalizando as necessidades do projeto, como entrada. Com isto, auxiliar na instanciação do componentes modelo de planejamento e cenários de execução, pois existe um *scheme* a ser seguido. Por exemplo, vamos usar as informações encontradas no termo de abertura do projeto (*Project Charter*), para formalizar as metas, que são os resultados esperados, decompondo em tarefas. Assim, podemos criar tarefas a serem executadas para atingir essas metas. A Listagem 4.2 ilustra esta formalização.

Listagem 4.2: Definição de *scheme* do Projeto.

```

1 <scheme id=" build_project_sch">
2   <goal id=" project_built">
3     <plan operator=" sequence">
4       <goal id=" sw_Requirement_Specification" ttf="3 days" />
5       <goal id=" software_Prototyping_done" ttf="3 days" />
6     </plan>
7   </goal>
8 </scheme>

```

O *build_project_sch* é definido por um *scheme*. Este incorpora a meta do *Project_built* que possui um plano que está em conformidade com as necessidades do projeto. Estes planos podem possuir metas sequenciais ou paralelas. Elas também, definem um tempo para cada meta. Além disto, precisa-se também definir as tarefas relacionadas a cada uma destas metas. Para este fim, usamos a dimensão ambiente. Na Subseção 4.4.3, vamos mostrar essa definição.

Os *stakeholders* em conjunto com o gerente de projetos, podem realizar rodadas de simulações que utilizam esta formalização. Com isto, incorporá-las em ferramentas ou técnicas existentes, tais como técnicas de tomada de decisão em grupo ou criação de protótipos. Desta forma, apoiando os gerentes de projeto e os *stakeholders* na criação de documentação de requisitos e uma matriz de rastreabilidade de requisitos por meio de metas alcançadas.

4.4.3

Definir o Escopo

Depois de completar o processo de coleta de requisitos, é necessário definir o escopo. A definição do escopo descreve detalhadamente o projeto e ou o produto. De acordo com o PMBOK (PMI, 2013), o principal benefício desse processo é que este descreve os limites do produto, serviço ou resultado, definindo qual dos requisitos coletados serão incluídos ou excluídos do escopo do projeto. A Figura 4.4 mostra as entradas associadas a este processo.

Definir o escopo é um processo da fase de planejamento. Este é um importante processo para preparação detalhada da declaração do escopo do projeto, sendo fundamental para o sucesso do projeto. Ele deve conter as principais entregas, premissas, e restrições que estão documentadas durante a iniciação do projeto. Para este objetivo, podemos usar a dimensão do ambiente e seus recursos para formalizar a definição do escopo, com isto, auxiliar na configuração do modelo de planejamento do ProMabs. Por exemplo, podemos usar a documentação de requisitos, que é uma das entradas, para apoiar a criação das tarefas. Assim, podemos simular as tarefas necessárias para os requisitos e verificar se as metas são cumpridas com estas tarefas. A Listagem 4.3 mostra esta formalização.

Listagem 4.3: Tarefas por Objetivo.

```

1 +!sw_Requirement_Specification <- specify_software_requirement .
2 +!software_Prototyping_done <- softwarePrototype .

```

Como já mencionado anteriormente, as metas do projeto estão relacionadas a tarefas específicas e para cada meta existe uma lista de tarefas. Por exemplo, na Listagem 4.3, para cumprir com o objetivo `sw_Requirement_Specification` a tarefa `specify_software_requirement` precisa ser executada. Assim, podemos usar essas especificações para apoiar a criação da declaração do escopo do projeto e para atualizar a documentação dos produtos, como o registro do *stakeholder* e documentação de requisitos, assim auxiliando a definição do escopo.

4.4.4

Criar a Estrutura Analítica do Projeto (EAP)

Criação de EAP é o último processo da fase de planejamento. Este processo subdivide as entregas do projeto em tarefas, com isto, resultando em componentes menores e melhor gerenciáveis. De acordo com PMBOK (PMI,

2013), o principal benefício deste processo é fornecer uma visão estruturada do que deve ser realizado. A EAP é um escopo hierarquicamente organizado do trabalho a ser realizado pela equipe do projeto para cumprir os objetivos do projeto e criar as entregas necessárias. A Figura 4.4 mostra as entradas associadas a este processo. A EAP utiliza a formalização de todos os outros processos mencionados anteriormente, como dados de entrada.

Listagem 4.4: Papel apropriado para tarefa.

```

1 task_roles ("Software_Requirement_Specification",
2   [software_requirement_analyst]).
3 task_roles ("Software_Prototyping", [programmer]).

```

A Listagem 4.4 ilustra as informações adicionais necessárias para a criação da WBS. Cada tarefa está associada a um papel, sendo uma regra específica que precisa ser seguida. Usamos a dimensão organização e seus recursos para formular estas regras. Deste modo, os relacionamentos de projeto, entre tarefas e recursos humanos, podem ser definidos como regras e o sistema tem que segui-las. Por exemplo, existem uma regra que define que apenas um *software_requirement_analyst* pode realizar a tarefa *Software_Requirement_Specification*, então esta regra deverá ser seguida. Desta forma, só os agentes recursos humanos que assumem este papel podem executar a tarefa.

O processo de criar EAP usa todas as dimensões para fornecer recursos suficientes para incorporar as relações necessárias entre os elementos do projeto de software. Desta forma, temos uma formalização que pode ser usada no componente modelo de planejamento, conseqüentemente, no componente cenário de execução do ProMabs. Apoiado a criação da linha de base do escopo e as atualizações de documentos do projeto.

4.4.5

Validar Escopo

Validar escopo é um processo da fase de monitoramento e controle de projeto. Este processo formaliza a aceitação das entregas finalizadas do projeto. De acordo com o PMBOK (PMI, 2013), o principal benefício desse processo é ser objetivo em relação ao processo de aceitação. A Figura 4.4 mostra as entradas associadas a este processo.

Clientes ou patrocinadores precisam rever e verificar os produtos do projeto, a fim de garantir que estes sejam concluídos de forma satisfatória.

Este processo utiliza saídas dos outros processos mencionados anteriormente, para realizar a validação e aceitar o projeto ou produto final. Por exemplo, podemos formular as entradas deste processo e os entregáveis verificados com os dados de desempenho de trabalho, utilizando o ambiente de organização. Além disso, podemos formular as expectativas do cliente ou patrocinador com relação a cada entregável usando missões. Também, normas podem ser usadas para especificar limitações que envolvem papéis e a missão do projeto. A Listagem 4.5 mostra esta formalização.

Listagem 4.5: MissMission for role .

```

1 <mission id="management_of_project_building" min="1" max="1">
2   <goal id="project_built" />
3 </mission>
4 <mission id="specify_SoftwareRequirement" min="1" max="1">
5   <goal id="software_Requirement_Specification_done" />
6 </mission>
7 <mission id="software_Prototype" min="1" max="1">
8   <goal id="software_Prototyping_done" />
9 </mission>

```

A tag *management_of_project_building* define a missão do projeto. Esta incorpora a meta *project_built* que tem de ser cumprida por um papel. Este papel pode ser definido usando normas. Além disso, podemos definir os papéis que podem executar a missão. Precisamos também definir as normas relativas a essa missão. Para definir estas normas usamos a dimensão ambiente. Podemos observar esta definição por meio da listagem 4.6.

Listagem 4.6: Normas por papel.

```

1 <normative-specification>
2 <norm id="n1" type="obligation"
3   role="project_owner"
4   mission="management_of_project_building" />
5 <norm id="n2" type="obligation"
6   role="software_requirement_analyst"
7   mission="specify_SoftwareRequirement" />
8 <norm id="n3" type="obligation"
9   role="programmer" mission="software_Prototype" />
10 <norm id="n4" type="obligation"
11   role="programmer2" mission="software_Unit_Code" />
12 <norm id="n5" type="obligation"
13   role="programmer3" mission="software_Unit_Debug" />
14 <norm id="n6" type="obligation"
15   role="tester" mission="unit_Test_Conduct" />
16 </normative-specification>

```

Usando estas características, o processo de validar o escopo possui a formalização necessária que pode ser usada no componente cenários de execução do ProMabs, conseqüentemente, no componente analisador, onde visualiza-se os cenários. Desta forma, para cada tarefa na EAP sendo simulada, podemos usar uma representação diferenciada para as as tarefas que cumpriu suas missões. Por exemplo, para as tarefas que não cumpri a sua missão, ela recebe uma cor Vermelha, por outro lado, se a tarefa cumpriu a sua missão, recebe a cor verde. Essas representações podem apoiar os gerentes de projeto no trato com informações sobre o desempenho do trabalho, solicitações de mudança e aceitação dos entregáveis.

4.4.6

Controlar Escopo

Controlar Escopo é um processo da fase de monitoramento e controle de projeto. Este processo monitora o estado do escopo do projeto e do produto, e suas mudanças de acordo com uma linha base. De acordo com o PMBOK (PMI, 2013), o principal benefício deste processo é permitir que a linha base do escopo possa ser acompanhada e se está sendo realizada ao longo do projeto. A Figura 4.4 ilustra as entradas associadas a este processo.

O gerente de projeto utiliza o processo controle de escopo para avaliar a evolução do projeto e gerenciar as alterações. Para apoiar este processo, é necessária a formalização de todos os outros processos mencionados anteriormente para que atuem como entradas para o processo controle de escopo. Após esta etapa, o simulador é capaz de instanciar as formalizações previamente definidas. Assim, simulações envolvendo controle do escopo podem ser executadas. Para isto, é necessário utilizar a abordagem proposta pelo componente analisador do ProMabs, para visualizar os cenário.

A Figura 4.5 ilustra a simulação sendo executada, de acordo com o componente analisador proposto pelo ProMabs. O simulador cria uma EAP com todas as tarefas necessárias a serem realizadas. Assim, a simulação pode fluir de uma tarefa para outra quando cada tarefa é executada. Nesta figura, cada caixa azul representa uma tarefa agendada, que possui um nome e que por enquanto não tem recursos humanos associados. Depois de uma tarefa ser executada, a sua caixa associada muda de cor (de azul para verde) e também recebe o nome do recurso humano que realizou a tarefa. Deste modo, JaCaMoPM dá suporte a representação das características necessárias requeridas no processo de gestão de escopo, mas também pode

proporcionar uma visualização de EAP adequada. Uma característica especial da visualização da EAP é que ela permite que os gerentes de projeto, por meio de alterações na cor, possam acompanhar a dinâmica das sequências de fluxo de atividades, o que auxilia na validação e controle do escopo do projeto.

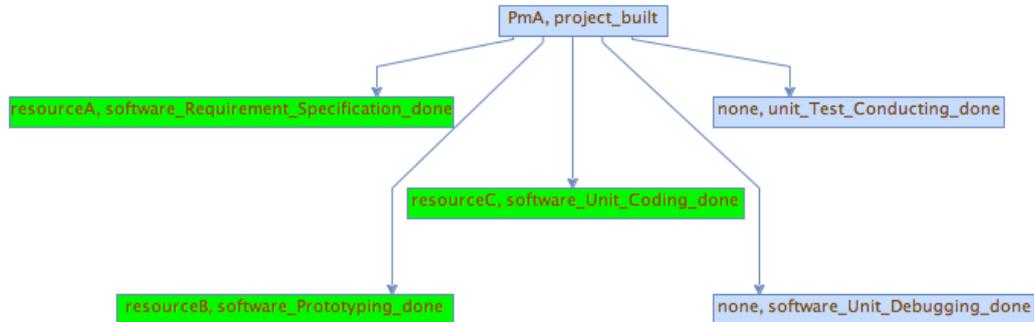


Figura 4.5: EAP criada pela simulação.

Análise de Variância é uma técnica utilizada neste processo. O objetivo da análise de variância é determinar a causa e o grau de diferença entre a linha base e o desempenho real (PMI, 2013). As linhas de base podem ser geradas pelo simulador por meio das especificações de EAPs. Desta forma, o gerente de projeto tem o apoio necessário para realizar comparações relevantes, e a simulação e seus recursos podem apoiar a execução das saídas deste processo, tais como solicitações de mudança e atualizações do plano.

4.4.7

Relação entre PMBOK, JaCaMo e JaCaMoPM

JaCaMoPM é a instanciação do ProMabs, para representar elementos da gestão de escopo que estamos descrevendo nesta seção. Como mencionado anteriormente o JaCaMoPM é uma extensão do JaCaMo, uma plataforma baseada em agentes que possui três dimensões necessárias para representar inicialmente os processos de gerenciamento do escopo. Porém, em termos do componente analisador, para a visualização da EAP, foi necessário a extensão. A Tabela 4.1 mostra a relação entre PMBOK, JaCaMo, e JaCaMoPM.

Assumindo que o foco está no de gerenciamento de escopo, esta tabela lista todos os seus processos (P1-P6) definidos no PMBOK. A tabela 4.1 apresenta também, o mapeamento de todos os processos PMBOK (P1-P6) mapeados para recursos de representação JaCaMo. Em JaCaMoPM os resultados e a visualização podem ser mapeados, ou seja, existe uma visualização de EAP e uma apresentação de estado da EAP.

Tabela 4.1: Relação entre PMBOK, JaCaMo, and JaCaMoPM

PMBOK	JaCaMo
P.1 Planejar o Gerenciamento do Escopo	Role-definitions, Group specification (P.1)
P.2 Coletar os Requisitos	Scheme, Goal, Plan (P.1, P.2)
P.3 Definir o Escopo	Tasks for goal (P.3, P.4)
P.4 Criar a EAP	Role appropriate to task (P.3, P.4)
P.5 Validar o escopo	Mission rules (P.5,P.6)
P.6 Controlar o Escopo	
	JaCaMoPM
	Visualizar EAP(P.4, P.5, P.6)
	Visualizar os Estados da EAP (P.6)

4.4.8

Estudo Exploratório

Com esta representação de processos da gestão de escopo em JaCaMoPM, realizamos um estudo exploratório por meio de exemplo de problema real. Para isto, seguimos uma abordagem qualitativa para descrição da utilização do JaCaMoPM, com as representações supracitadas. Nosso estudo descreve como estas representações propostas, auxiliam os processos de gestão de escopo, no âmbito de tomada de decisão. A estrutura usada no estudo exploratório envolve um ambiente em que são descritas as características do estudo, um problema a ser resolvido, uma fase de simulação que é realizada para executar os cenários que serão utilizados na tomada de decisão e, finalmente, a avaliação dos resultados.

O estudo foi conduzido no Laboratório de Engenharia de Software (LES) da Pontifícia Universidade Católica do Rio de Janeiro. O LES se concentra no desenvolvimento de sistemas de software para a web e desktop em diferentes setores como petróleo e e-commerce, assim como, para trabalhos científicos. Usamos parte de um projeto real do LES em nosso estudo. A equipe de LES que participou deste projeto envolve três desenvolvedores e um gerente de projeto que também é desenvolvedor.

Um dos objetivos do estudo é representar e visualizar os elementos que envolvem uma gestão de projeto de software, tais como recursos, tarefas e seus relacionamentos. Na perspectiva do líder de projeto, um dos objetivos é visualizar as tarefas necessárias para execução do projeto e a sequência de fluxo de tarefas e, desta forma, entender os fenômenos que acontecem durante a simulação do projeto. Com isto, auxiliando o líder do projeto no processo de tomada de decisões.

Criação de EAP e Alocação de Recurso Humano para Tarefas

LES possui muitos projetos, e cada um envolve no mínimo um gerente de projeto com sua equipe de desenvolvimento. Uma das atribuições do líder de projeto é criar uma EAP que satisfaça as exigências do projeto. Além disto, alocar recursos humanos para cada tarefa da EAP. Quando este líder não possui uma certa experiência, estas atribuições se tornam difíceis. Pois uma criação de uma EAP que não atende os requisitos do projeto, podem levar ao fracasso do projeto. Atribuir uma tarefa a um recurso que não se encaixa nas características exigidas, pode causar problemas graves, como atrasos nas entregas, custos mais elevados e produtos de má qualidade. A questão principal é como apoiar o líder de projeto na criação EAP e atribuição de tarefas para recursos com o auxílio de cenários de simulação.

Diretrizes Gerais

Primeiro, nós projetamos e apresentamos o JaCaMoPM para ProMabs, além disto, utilizando as diretrizes do PMBOK, conforme descrito anteriormente nesta seção. Então, utilizamos o JaCaMoPM para implementar e executar a simulação de uma EAP. Para avaliar a eficácia, as simulações foram instanciadas utilizando como exemplo uma parte de um projeto de desenvolvimento do LES, usando elementos recursos humanos e tarefas. Com isto, verificamos se a EAP simulada estava de acordo com os requisitos do projeto. E se os recursos atribuídos para tarefas, atendiam suas exigências.

Dados iniciais, Criação da EAP e Alocação de Recursos

O ProMabs instanciado precisa de informações de entrada sobre o projeto. Estas informações são fornecidas pelo gerente do projeto ou líder de projeto. Para apoiar esta fase, utilizamos documentos disponíveis do Projeto. Aplicando as representações de elementos já descritos com as informações padrão necessárias. Informação padrão, são dados para inicializar a simulação, porém, com a execução da simulação, podem existir alterações. Por exemplo, cada tarefa precisa de um nome, um nível de conhecimento ou papel requerido e uma duração padrão. A tarefa precisa também saber qual é a sua tarefa predecessora, caso possua, e o plano ou pacote de trabalho a que pertence. Para recursos humanos, é necessário saber o nome do recurso, o nível de conhecimento e o custo. Depois de ter inserido os dados necessários para a simulação, executamos e verificamos os seus resultados, conforme ilustrado

na EAP da figura 4.5. A execução destes cenários (EAPs), permitem ao líder de projeto analisar e acompanhar a dinâmica do projeto com diferentes configurações de EAP. Assim, o JaCaMoPM auxilia na criação de uma EAP, de forma a permitir ao líder de projeto testar EAPs diferentes e verificar em conjunto com sua equipe e ou *stakeholders* se estão de acordo, por meio de protótipos de EAPs. Desta forma, permitindo alterações no plano e auxiliando na visualização das consequências destas alterações.

Depois de aplicar uma sequência de ações para criar a Estrutura Analítica de Projeto (EAP), o simulador encontra e atribui agentes recursos humanos para as tarefas, seguindo o fluxo das tarefas. A escolha dos recursos depende da estratégia escolhida pelo líder de projeto. Para este projeto, utilizamos apenas a menor custo, assim, o simulador checa quais os recursos são capazes de executar uma determinada tarefa e escolhe o recurso com o menor custo, seguindo o fluxo mencionado anteriormente.

Com esta simulação, o líder de projeto é auxiliado no processo de tomada de decisão durante a fase de planejamento do projeto. O líder do projeto pode usar a simulação para criar uma EAP baseado em uma estratégia. Por exemplo, se a prioridade é o tempo, o gerente de projeto pode alterar o plano para executar algumas tarefas em paralelo. Assim, o gerente de projeto pode obter sugestões para uma EAP e simular resultados que mostram a complexidade e dinâmica de todo o projeto desejado. Desta forma, a simulação mostra o fluxo da EAP e o impacto da mudança.

Discussões do Estudo

Anteriormente, o líder do projeto no LES, não dispunha de uma abordagem para auxiliar a criar EAP e nem para atribuir recurso humano de acordo com o escopo e com isto auxiliar o processo de gestão de escopo. Porém, para utilizar o JaCamo é necessário um conhecimento significativo sobre a plataforma, assim, dificultando o seu uso. Entretanto, o JaCaMoPm, auxiliou a avançar a nossa pesquisa em termos de entender como os agentes podem representar e simular os elementos do projeto. Além, disto, identificamos a importância de as tarefas serem representadas por agentes. Pois, pode existir um comportamento dependendo da tarefa. Como por exemplo, em projetos com novas tecnologias incorporadas, uma tarefa que programa uma funcionalidade em uma linguagem recente para uma equipe, pode precisar de uma margem de segurança para determinar o prazo para realizar este tipo de tarefa. Podemos simular um comportamento para esta tarefa por meio de

atrasos aleatórios com base em um histórico de atrasos da equipe em novas tecnologias. Com isto, verificar uma possível margem de segurança.

4.5

Considerações Finais

Neste capítulo, apresentamos e detalhamos como o ProMabs, instanciado no JaCaMoPM, representa os elementos do projeto e seus relacionamentos. Para isto, aplicamos e utilizamos como exemplo ilustrativo os processos de gestão de escopo. Ao final, utilizamos um exemplo ilustrativo real para verificar como esta representação pode auxiliar na criação de uma EAP, alocação de recursos humanos, assim como, o entendimento dos seus elementos, suas representações e seus relacionamentos. O JaCaMoPM com este exemplo está disponível em nosso repositório ¹.

Este capítulo se baseia no relatório técnico (Baia et al., 2014), que auxiliou na compreensão de como os agentes podem simular gestão de projetos, por meio de suas crenças, metas e planos, avançando assim a nossa pesquisa. Além disto, a instanciação do ProMabs em uma plataforma de programação multiagentes, reforça nossa contribuição sobre a aplicação do ProMabs para simular gestão de projetos. Nos próximos capítulos 5 e 6, iremos aplicar o conhecimento adquirido, neste capítulo, para primeiro, criar cenários que auxiliam a tomada de decisão e segundo apoiar projetos de software adaptativos, simulando gestão de projetos. Para isto, será necessário instanciar os componentes do ProMabs, que não foram utilizados neste capítulo. Para criar uma conexão com o ambiente externo, para receber e solicitar dados. Além, disto utilizar o ProMabs para criar uma nova técnica ou ferramenta que auxilie o gerente de projeto, sem a necessidade de um conhecimento significativo em uma plataforma de programação.

¹<http://www.inf.puc-rio.br/~dbaia/jacamopm/>

5

Execução de Cenários para Gestão de Projetos

Este capítulo apresenta uma aplicação do modelo conceitual ProMabs em um ambiente de simulação multiagentes. Com o objetivo de executar cenários por meio de simulação de gestão de projetos de software, para auxiliar no processo de tomada de decisão. Desta forma, por meio do ambiente de simulação multiagentes, compreender como um único modelo simula diferentes cenários. Neste contexto, este capítulo se concentra na segunda questão de pesquisa desta tese:

QP1(ii). Como executar cenários para auxiliar o gerente de projetos com a tomada de decisão?

Para isto, na seção 5.1 discutimos como cenários auxiliam a tomada de decisão. Na seção 5.2 descrevemos os trabalhos relacionados que utilizam cenários para auxiliar em projetos. Na seção 5.3 descrevemos como um ambiente de simulação instancia o ProMabs. Na seção 5.4 descrevemos um estudo de caso por meio da instanciação de uma simulação. E finalmente na seção 5.5, apresentamos a conclusão e algumas discussões.

5.1

Cenários para Tomada de Decisão

Desenvolvimento de software em organizações de grande porte, envolvem equipes que trabalham em colaboração para resolver problemas que muitas vezes são bastante complexos. Os membros destas equipes normalmente seguem processos de desenvolvimento definidos pela organização em que trabalham. Neste contexto, o gerente de projeto desempenha um papel fundamental em todo o processo, sendo responsável por atividades como a alocação de recursos, a definição de prioridades, o acompanhamento e a avaliação dos progressos realizados, e também mantém a equipe focada nos objetivos do projeto. Para realizar estas atividades de forma eficaz, o gerente

de projeto precisa conhecer o andamento do projeto e recursos relacionados. Uma atividade chave do gerente de projeto é atribuir tarefas a recursos humanos, formando assim uma equipe para o projeto. Porém, uma equipe precisa atender os objetivos do projeto, como tempo e custo. Para garantir uma formação satisfatória, o gestor necessita conhecer previamente os recursos humanos disponíveis e as suas capacidades. Por exemplo, em um projeto de desenvolvimento de software o recurso desenvolvedor tem características específicas, tais como conhecimento, custo e níveis de desempenho.

Atribuir uma tarefa a um recurso que não se encaixa nas características exigidas pela tarefa pode causar problemas graves, como atrasos nas entregas e produtos de má qualidade. Além disso, as atividades do gerente de projeto geralmente envolvem enormes quantidades de informação, em especial no caso de projetos de software grandes e complexos. Em tais casos, o registro e análise manual de dados não são viáveis, devido ao risco de erro humano e limitações de custos e tempo globais. Pesquisas recentes indicam que projetos de Tecnologia da Informação (TI) continuam a ter uma elevada taxa de insucesso (GROUP, 2013),(Rubinstein, 2007); E também há evidências crescentes de que as habilidades do gerente de projeto podem ser cruciais para o desempenho eficiente e eficaz da equipe do projeto em melhorar os seus resultados (Flyvbjerg e Budzier, 2011), (Taylor e Woelfer, 2011). Assim, o gerente de projeto precisa de um conjunto de ferramentas apropriadas para apoiar a sua gestão.

Os projetos de desenvolvimento de software muitas vezes não atingem as metas previstas, como custo e tempo em sua conclusão. Isto geralmente é resultado de práticas de gestão inadequadas associadas a uma situação altamente volátil. As abordagens existentes muitas vezes apoiam soluções mais estáticas, mas não incorporam os aspectos dinâmicos de gerenciamento de projetos para os cenários de simulação. Neste contexto, a motivação para usar Simulação Baseada em Multiagentes (SBMA) decorre do fato de que modelagem em Sistemas MultiAgentes fornece modelos robustos para representar ambientes do mundo real que são complexos e dinâmicos (Luck e McBurney, 2008) . As vantagens da utilização SBMA são a sua capacidade para apoiar aspectos realistas do domínio do problema incorporando características baseadas em agentes. Por exemplo, um agente toma uma decisão baseado em regras, delegando tarefas para outros agentes, isto é um recurso de autonomia de agente. Já para um exemplo de auto-organização do ambiente, temos agentes executando tarefas de outros agentes, com isto temos adaptação dinâmica entre agentes para executar tarefas.

Neste contexto, acreditamos que uma simulação baseada no ProMabs, descrito no capítulo 3, reúne componentes necessários para executar cenários, para auxiliar os envolvidos no processo de tomada de decisão. O ProMabs incorpora elementos básicos da gestão de projeto de software, tais como, tarefas, recursos, gerente de projeto. Assim como, oferecer suporte as alterações destes elementos, que acontecem ao longo do projeto. Estas alterações geralmente causam impactos no resultado do projeto, podendo refletir no custo, tempo e qualidade. Desta forma, existe a necessidade de simular os elementos e suas alterações, executando cenários de acordo com estratégias para atingir objetivos específicos. Para isto, o ProMabs instanciado por um ambiente de simulação, executa cenários com base em estratégias específicas criando uma equipe, por meio de alocação de tarefas com recursos humanos. Estas estratégias são baseadas nas áreas do PMBOK como custo, tempo, escopo e recursos humanos. A instanciação do ProMabs possui agentes representando as características de um ser humano, no papel de um dos recursos humanos. Por exemplo, um agente como um desenvolvedor ou como um gerente de projeto, e também, com a capacidade de tomar decisões de forma autônoma.

5.2

Trabalhos Relacionados

A literatura sobre o uso de simulação para auxiliar projetos de desenvolvimento de software apareceu pela primeira vez em 1999 (Kellner et al., 1999), (Rus et al., 1999). Kellner et al. descreve a modelagem de simulação de processo de desenvolvimento de software em termos de por quê, o quê e como. O artigo aponta os benefícios da simulação em apoiar os vários aspectos da gestão de desenvolvimento de software, tais como as melhorias de processos e para o treinamento de gerenciamento de projetos de software. Rus et al. descreve o uso de um simulador de processos para apoiar o planejamento e gestão de projetos de software. O modelo foca na confiabilidade do software, mas é limitado aos fatores de custo e cronograma. Os trabalhos focam no processo de desenvolvimento de software, diferenciando de nosso trabalho que é simulação da gestão de projetos de software.

SBMA também são aplicados em outras áreas. Os SBMA têm sido utilizados para melhorar os tipos anteriores de simulações, tais como micro-simulação, orientada a objetos ou para simulação baseada em indivíduo. Segundo Drogoul (Drogoul et al., 2003), uma das características positivas de SBMA é a capacidade de modelar indivíduos. Assim, esta abordagem é capaz de incorporar aspectos do mundo real, tais como a adaptabilidade dos sistemas

sociais. Esta flexibilidade contribui para a escolha destas características de simulação em sistemas complexos.

Recentemente, artigos de Wickenburg e Davidsson (Wickenberg e Davidsson, 2003), Agarwal e Umphress (Agarwal e Umphress, 2010), Cherif e Davidsson (Cherif e Davidsson, 2010) propõem abordagens para apoiar o processo de desenvolvimento de software e gerenciamento de projetos de software usando SBMA. Wickenburg e Davidsson (Wickenberg e Davidsson, 2003) fornecem um conjunto de diretrizes gerais sobre quando usar SBMA e também ilustram três exemplos concretos, onde a aplicação SBMA parece particularmente promissora.

Agarwal e Umphress (Agarwal e Umphress, 2010), descrevem formas nas quais SBMA podem ser aplicadas ao processo de desenvolvimento de software. Os autores expõem que é possível simular o desempenho de equipes de desenvolvimento de software. Para isto, usando os dados de cada membro da equipe de desempenho em projetos de software de pequena escala. Esta abordagem permite que os gerentes configurem o modelo dinamicamente sem ter de alterar o código de simulação. Por exemplo, ao permitir a adição ou remoção de desenvolvedores da equipe de desenvolvimento, tornando possível entender o impacto de tais mudanças no processo, executando cenários.

Cherif e Davidson (Cherif e Davidsson, 2010), apresentam uma aplicação de SBMA a modelagem de simulação de processo de software. Os autores usam um modelo básico para executar em simuladores SBMA e Dinâmica de Sistemas (DS). A abordagem permite comparações relevantes entre SBMA e DS.

Outra abordagem relacionada a este capítulo envolve o aprendizado de gerenciamento de projetos de software por meio de simulações. Artigo Mittermeier et al. (Mittermeier et al., 2003) descreve o framework AMEISE que é um sistema cliente/servidor que utiliza um mecanismo de simulação chamado SESAM, (Engenharia de Software simuladas por modelos animados), que foi desenvolvido na Universidade de Stuttgart (Drappa e Ludewig, 2000). A equipe AMEISE adotou o modelo educacional proposto por SESAM e usa o sistema como um protótipo inicial, evoluindo em uma série de iterações para produzir a versão do sistema AMEISE atualmente disponíveis. A abordagem AMEISE é diferente da nossa abordagem na medida em que exige regras para descrever eventos possíveis e depende de interações passo-a-passo com os atores que desencadeiam as regras.

Em contraste com trabalhos relacionados, neste capítulo nos

concentramos em um modelo de simulação baseada em multiagentes que abordam aspectos dinâmicos de gerenciamento de projetos incorporados dentro dos cenários de simulação. O modelo contém agentes de software, que capturam comportamentos para representar um ser humano. Este agente assume um papel de recurso humano. Por exemplo, como um desenvolvedor ou como um gerente de projeto, e também, com a capacidade de tomar decisões de forma autônoma.

5.3

ProMabs para um Ambiente de Simulação

O objetivo desta seção é aplicar o ProMabs a um ambiente de simulação. Com isto, executar cenários baseados em estratégias, para auxiliar a gestão de projetos no processo de tomada de decisão. Os cenários são executados baseado em quatro áreas de concentração do PMBOK (PMI, 2013). O ProMabs descrito no capítulo 3 possui cinco componentes principais e é baseada em multiagentes. A figura 5.1 ilustra estes componentes. Instanciamos estes componentes usando um ambiente de simulação chamado CORMAS (Bousquet et al., 1998) que é um software livre e de código aberto dedicado à gestão dos recursos. CORMAS é construído sobre VisualWorks, uma aplicação multi-plataforma da linguagem Smalltalk. O objetivo de instanciar o ProMabs no ambiente de simulação CORMAS, é executar cenários diferentes a partir de um modelo único. Além disto, verificar como a execução destes cenários podem auxiliar em estratégias como tempo, custo e qualidade.

Tal ambiente de simulação provou ser útil na compreensão das interações complexas entre o ambiente e a dinâmica social. Optamos em utilizar o ambiente de simulação CORMAS porque o ProMabs propõe um conjunto de agentes que interagem e modificam seu ambiente, no caso, as tarefas. Além disto, como os agentes representam recursos humanos, estes agentes são limitados. Por exemplo, um agente recurso humano só pode realizar uma tarefa por vez. E também só pode realizar tarefas na qual possui conhecimento. Para isto, utilizamos e representamos um conjunto de elementos de gestão de projeto de software, de acordo como descrito no capítulo 4.

Os princípios em que se baseia o CORMAS são expressos por diferentes escolhas. Ao construir um modelo em CORMAS, deve-se considerar três aspectos do padrão MVC (do inglês Model-View-Controller) (Krasner e Pope, 1988). O primeiro aspecto, modelo, relaciona-se aos agentes e suas interações com o meio ambiente ou através de comunicação direta entre os agentes. Este

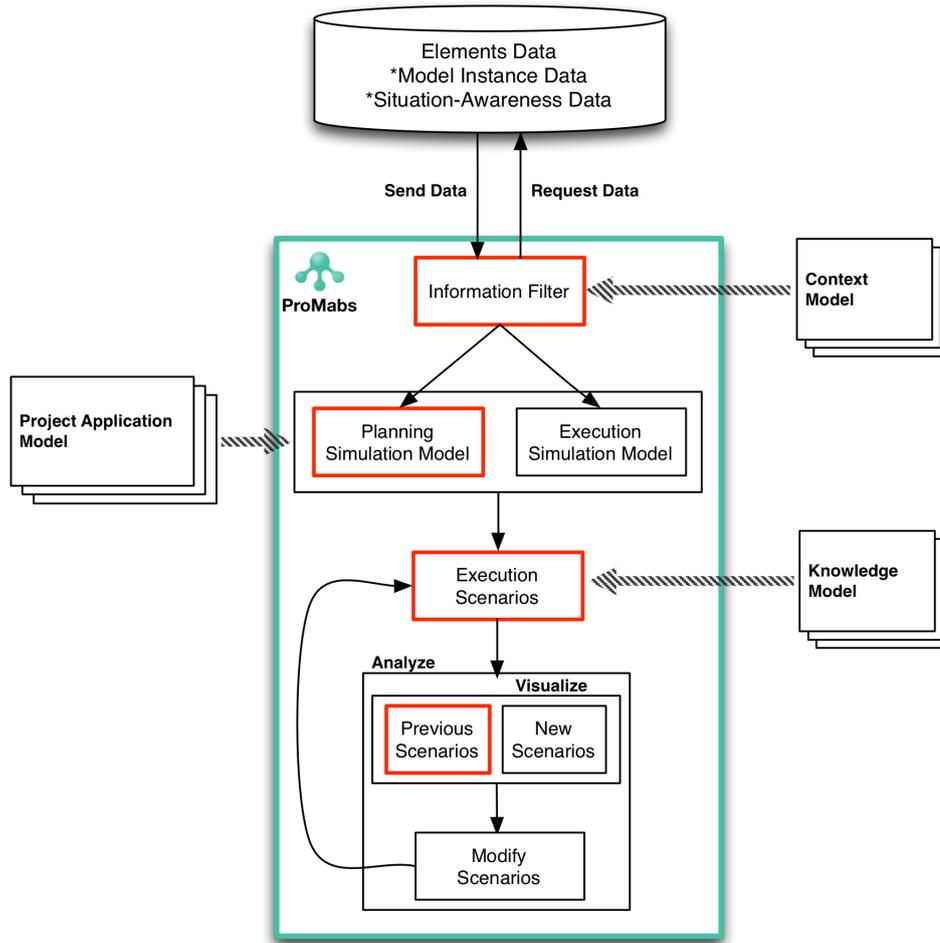


Figura 5.1: Componentes Instanciado do ProMabs.

aspecto auxilia na instanciação do componente Modelo de Planejamento e na conexão com o componente filtro de informação. O segundo aspecto, visão, define uma ou várias observações do sistema por meio do conceito de visão. Este aspecto auxilia na instanciação do componente analisador, para este caso foram criadas tabelas com os resultados da simulação. A seção 5.4.5 ilustra estes resultados. Finalmente, o terceiro aspecto, controlador, refere-se ao controle global da dinâmica da simulação. Este aspecto auxilia na instanciação do componente cenários de execução, o qual possui estratégias associadas para simular os cenários. Desta forma, realizando as mediações do componente filtro de informação, modelo de planejamento e cenários de execução. Assim, possuindo características iniciais suficientes para representar e instanciar os componentes do ProMabs. A descrição da execução do modelo de simulação a seguir auxilia a compreender como estes componentes foram instanciados no CORMAS, descrevendo os agentes e entidades.

5.3.1

Descrição do ProMabs no CORMAS

Como descrito no capítulo 3, o componente filtro de informação é responsável por receber as informações(dados) do ambiente externo. Após isto, o componente modelo simulação de planejamento é instanciado com estes dados fornecidos pelo componente filtro de informação. Como não usamos o modelo de simulação de execução, o papel do filtro de informação é conectar-se ao ambiente externo para receber os dados. Na sequência, os agentes são criados com suas características representando um ser humano. Além disto, as entidades tarefas são criadas de acordo com as características das tarefas do ambiente externo. A figura 5.2 descreve as relações dos agentes e as entidades criadas para instanciar o ProMabs no CORMAS.

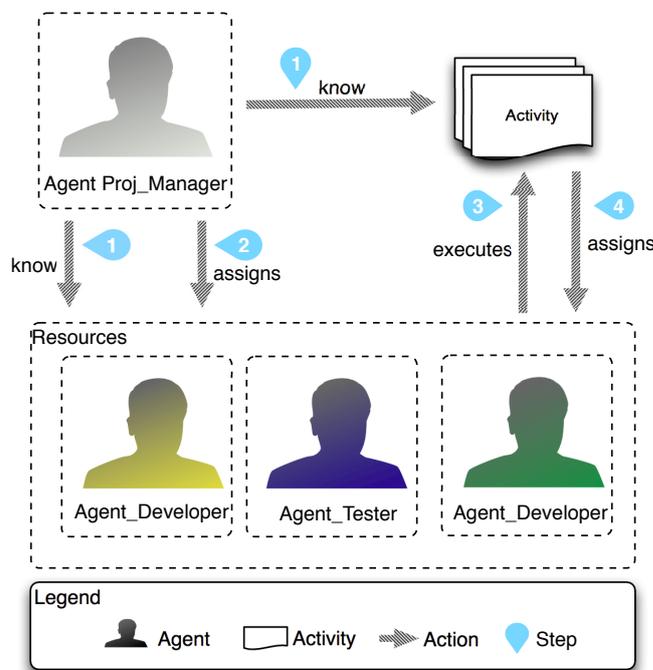


Figura 5.2: MABS em Projetos de Software.

Para cada agente e entidade existe um conjunto de características, que são:

- Recurso Humano (agente): Cada recurso humano é modelado por um agente reativo simples, aplicando regras de ação, tais como o mecanismo do agente para a execução de uma tarefa. Cada agente recurso humano incorpora características sobre sua natureza, tais como custo, conhecimento e performance. O agente gerente de projeto precisa conhecer os recursos disponíveis e as suas características. O gerente de

projeto também precisa definir essas características de recursos humanos através do uso de alguma ferramenta no ambiente externo, como por exemplo, OpenProject¹. Com isto, o componente filtro de informação pode acessá-la.

- Gerente de Projeto (agente): Na fase inicial, o agente gerente de projeto atribui agentes recursos humanos para as tarefas e define datas para as tarefas. Estas datas incluem datas estimadas, ou seja, as datas previstas para a tarefa ser inicializada e finalizada. O agente gerente também é responsável pela alocação das tarefas apropriadas para os recursos humanos que possuem as características desejadas pela tarefa. Se um agente recurso humano está atribuído a uma tarefa e este agente recurso humano não está disponível, o gerente de projeto deve procurar outro recurso disponível, se a estratégia tempo estiver ligada.
- Tarefa (entidade): O modelo precisa de informações sobre cada uma das Tarefas. Isto pode ser coletado através da utilização de uma ferramenta no ambiente externo, tais como OpenProject. Esta ferramenta deve possuir todas as informações necessária sobre a tarefa, para que o componente filtro de informação possa acessá-la. A figura 5.6 lista os dados necessários.

A Figura 5.2 ilustra também a sequência de etapas da simulação, que são:

1. Na primeira etapa, o agente gerente de projeto precisa conhecer tanto as entidades tarefas quanto os agentes recursos humanos. Cada agente recurso humano e cada entidade tarefa tem suas características individuais. A próxima subseção 5.3.2 descreve cada um deles.
2. Na próxima etapa, o agente gerente de projeto atribui as entidades tarefas aos agentes recursos humanos. Para isto, utiliza as características das entidades tarefas para combinar com o perfil dos agentes recursos humanos, delegando um agente recurso humano para executar tarefas específicas. Este passo reduz a possibilidade de atribuir uma tarefa para um recurso que não é apropriado em uma determinada simulação. Esta etapa também define a data prevista para o final e início do projeto, das tarefas e dos pacotes de trabalho. Assim, criando/formalizando um Estrutura Analítica de Projeto (EAP).
3. No terceiro passo, os agentes recursos humanos executam as tarefas.

¹<https://www.openproject.org>

4. A quarta etapa acontece de forma paralela, assim, quando cada entidade tarefa termina, libera o seu agente recurso para outras tarefas. Além disso, a entidade tarefa envia um alerta para todas as tarefas sucessoras, informando-as de que foi concluída. Assim, todas as tarefas sucessoras estão autorizadas a ser executadas. Além disso, se no momento inicial da execução de uma tarefa o seu agente recurso humano atribuído está indisponível, o agente gerente de projeto precisa determinar outro agente recurso humano que está disponível e que possua o melhor perfil para executar a tarefa. Esta escolha é baseada em um conjunto de regras relacionadas com as estratégias adotadas. Por exemplo, se a escolha foi a estratégia de custo então o agente gerente de projeto precisa encontrar um agente recurso humano apropriado que possua o menor custo. Os agentes recursos humanos podem ser dinamicamente alocados para uma tarefa específica, dependendo de fatores como a disponibilidade. O agente gerente de projeto realiza essas alocações. Isto garante a autonomia para o sistema se auto-adaptar. A terceira e quarta etapas são repetidas até que todas as entidades tarefas estejam concluídas.

Estas etapas estão associadas ao componente cenários de execução que possui a configuração fornecida do componente modelo de planejamento de simulação. Na próxima subseção descrevemos o modelo estrutural da simulação que auxilia na compreensão da instanciação dos componentes supracitados.

5.3.2

Modelo estrutural da Simulação

O diagrama de classes da simulação tem quatro classes principais: Projeto (*Project*), Gerente de Projetos (*Proj_Manager*), Recursos Humanos (*Resources*) e Tarefa (*Activity*). A Figura 5.3 ilustra este conjunto de classes básicas para a simulação e seus relacionamentos.

Para inicializar, precisamos definir o conjunto de tarefas, recursos humanos e gerente de projetos e suas respectivas características. Os atributos são:

- Project - name: nome do projeto; startDate: data de inicio do projeto; finalExpectedDate: data esperada para finalizar o projeto;
- AbstractElement – É uma classe abstrata que pode ser usada para criar tarefas filhas (sub-activities) e pacotes de trabalhos filhos (sub-workpackages).

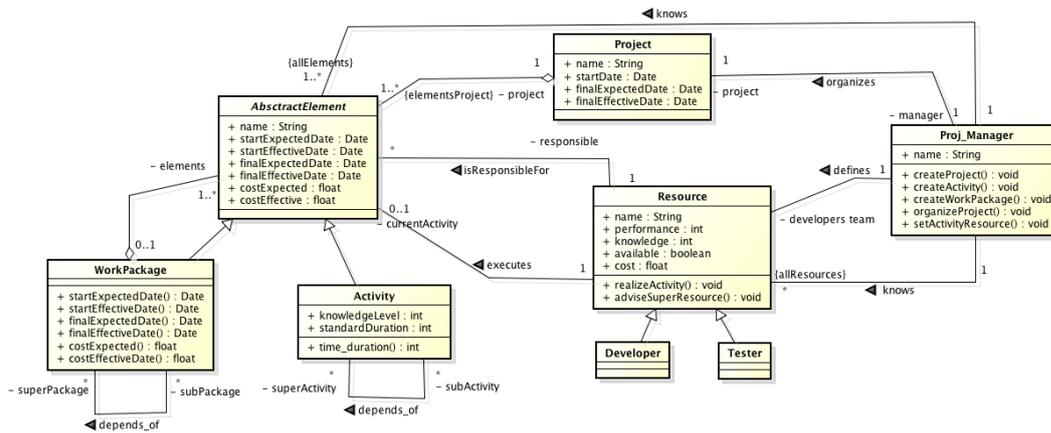


Figura 5.3: Diagrama de Classe para o modelo de Simulação de Gestão de Software.

- Activity - name: nome da tarefa, knowledgeLevel: nível de conhecimento esperado do recurso humano que irá realizar a tarefa, standardDuration: duração estimada, superActivity: a tarefa predecessora de quem esta tarefa depende para ser inicializada;
- Resource - name: nome do recurso humano, performance: desempenho do recurso humano, knowledge: conhecimento do recurso humano, available: disponibilidade do recurso, inicialmente todos estão disponíveis;
- Proj_Manager - name: nome do gerente de projeto.

Após o primeiro passo, mostrado na Figura 5.2, o segundo passo em diante define alguns atributos, tais como:

- Activity - startExpectedDate: data estimada para inicializar, finalExpectedDate: data estimada para finalizar, startEffectiveDate: data de inicio, responsible: recurso humano responsável, costExpected: custo estimado, costEffective: custo real;
- WorkPackage - startExpectedDate: data estimada para iniciar o pacote de trabalho, finalExpectedDate data estimada para finalizar o pacote de trabalho, costExpected: custo estimado do pacote de trabalho, costEffective: custo real do pacote de trabalho.

Durante a simulação, alterações podem ser feitas em alguns destes atributos. Por exemplo, se uma entidade tarefa precisa de um agente recurso humano para iniciar e está sem ou o agente recurso humano responsável está indisponível, então, o agente gerente de projeto irá atribuir um agente recurso humano para a tarefa. Sendo esta uma das vantagens de usar SBMA:

a possibilidade de auto-adaptação do sistema e a autonomia dos seus agentes. Para isto, o agente de gerente de projeto precisa seguir estratégias baseadas em regras. E são estas estratégias que auxiliam a executar vários cenários com o mesmo modelo. A seguir utilizaremos um diagrama de atividade para descrever como o agente gerente de projeto segue as estratégias.

Para ilustrar as segunda, terceira e quarta etapas da Figura 5.2, criamos um diagrama de atividade do agente gerente de projeto ilustrado na figura 5.4. Para cada tarefa na qual o agente gerente de projeto é solicitado auxiliar, um conjunto de ações baseadas nas estratégias são escolhidas. Ou seja, o agente gerente de projeto é capaz de fazer escolhas com base em suas estratégias incorporadas.

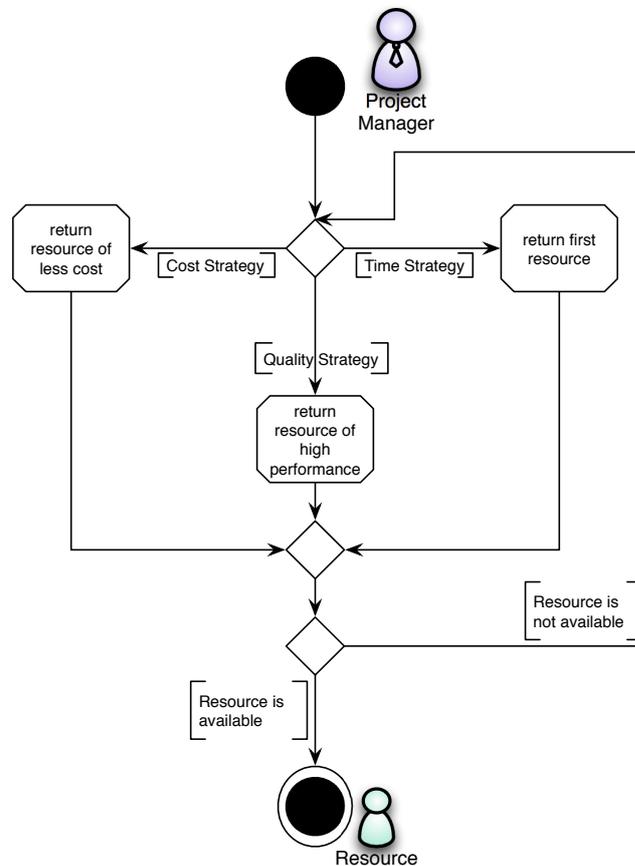


Figura 5.4: Diagrama de Atividade do agente gerente de projeto.

Como podemos observar na figura 5.4, inicialmente, o agente gerente de projeto recebe uma atividade, por exemplo atribuir um recurso humano para uma tarefa, em seguida, escolhe uma estratégia (custo, qualidade ou tempo) para realizar a atribuição. Cada estratégia tem uma regra a ser seguida. Ao escolher o agente recurso humano baseado nas regras das estratégias, a disponibilidade do agente recurso humano associado/escolhido é verificada. Se

está indisponível, o agente gerente de projeto escolhe outro recurso. Por outro lado, se o recurso está disponível, a tarefa será executada por este recurso. Estas estratégias são executadas de forma paralela, o agente pode criar três cenários ao mesmo tempo. Estes cenários foram criado para auxiliar a gestão de projeto de software, por meio de execução de cenários, em quatro áreas de concentração do PMBOK (PMI, 2013), gerenciamento de escopo, tempo, custo e recurso humano.

5.3.3

Gerenciamento de Escopo, Tempo, custo e Recurso Humanos

A importância de uma gestão do projeto de qualidade tem sido reconhecida por várias empresas de todos os setores como um fator essencial para o sucesso de suas iniciativas. Um projeto é um esforço temporário em que pessoas estão envolvidas para criar um produto, serviço ou resultado (PMI, 2013). O resultado do projeto é proposto para resolver um problema concreto, transformando ideias em ações. Alguns projetos são definidos através de suas metas ou objetivos e dentro de suas limitações de recursos, custo e tempo. Os objetivos dos projetos devem ser claros e viáveis. Pois cada projeto tem um início, meio e fim. Alguns projetos têm características específicas, por exemplo, restrições à data de início e / ou a data de entrega, ou custo total do projeto. A data de entrega ou de início específica o prazo de entrega ou início do produto (s) e / ou serviço (s), como previsto por partes interessadas no projeto. Outro fator fundamental é o conjunto de recursos humanos que devem ser perfeitamente inseridos para garantir que o produto ou serviço possua uma qualidade adequada e seja entregue no prazo e dentro do orçamento, conforme especificado pela parte interessada.

A gestão tenta minimizar a possibilidade de falha de um projeto por meio do uso de um plano que abranja todos os entregáveis do projeto. O plano leva em conta monitoramento e controle das tarefas do projeto, com a expectativa de minimizar os erros que podem exceder os custos estimados e datas de entrega. O gerenciamento de projetos é a aplicação de conhecimentos, habilidades, ferramentas e técnicas às atividades do projeto a fim de atender aos requisitos das partes interessadas (PMI, 2013). Para isso, o guia PMBOK (PMI, 2013) propõe dez áreas de conhecimento para gestão de projetos, que são, integração, escopo, tempo, custo, qualidade, recursos humanos, comunicação, riscos, aquisições e partes interessadas. Os processos de gerenciamento de projetos do PMBOK são compostos por cinco grupos de processos: a inicialização, planejamento, execução, monitoramento e controle, e

encerramento. Porém, para este estudo focamos nas áreas de gestão de escopo, tempo, custo e recursos humanos. A Figura 5.5 ilustra os grupos e processos relacionados a estas áreas. Podemos observar que estas áreas concentram seus processos em três grupos, como ilustrado na figura 5.5.

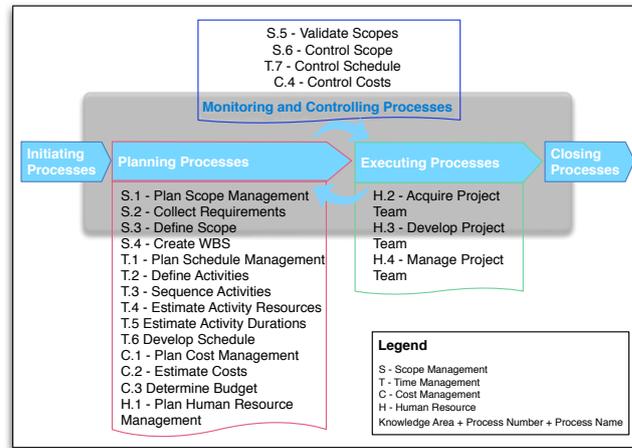


Figura 5.5: Grupos e Processos relacionados a tempo, custo, recurso humano e escopo.

De acordo com o PMBOK (PMI, 2013), "o gerenciamento do escopo de projeto inclui os processos necessários para assegurar que o projeto abranja somente o trabalho necessário para completar o projeto com sucesso. O principal interesse é definir e controlar o que é ou o que não é uma tarefa incluída no projeto". Ao definir o que constitui o conjunto de atividade do projeto, o gerente de projeto pode determinar todas as tarefas necessárias para formular um apropriado plano operacional para o projeto. O modelo proposto neste capítulo dá suporte à alguns dos processos ilustrado na figura 5.5. No grupo de planejamento da área de escopo, o modelo pode apoiar as atividades como gerenciamento do escopo do plano, coleta de requisitos, definição de escopo, e criação da Estrutura Analítica do Projeto (EAP). O simulador instancia uma EAP através da coleta de todas as tarefas e define datas correspondentes para cada tarefa. A EAP é um dos principais documentos ou ferramentas nesta área de gerenciamento de projetos, uma vez que se refere ao trabalho que precisa ser realizado para entregar um produto, serviço ou resultado com as características e funcionalidades especificadas.

Na área de tempo, o objetivo é assegurar que o projeto termina no tempo previsto. O grupo de planejamento para tempo possui processos tais como, plano de gerenciamento do cronograma, definição das tarefas, sequência das tarefas, estimar recursos necessários para cada tarefa, estimar a duração de cada tarefa e desenvolver um cronograma. Gerenciamento de tempo é uma

habilidade importante para qualquer gerente de projeto bem-sucedido. Sendo bastante comum o custo real ultrapassar o custo estimado, previsto no processo de planejamento do projeto. Sendo esta uma das possíveis razões para esta situação, a administração do tempo inadequada. Essa má gestão pode causar prejuízos financeiros e até mesmo inviabilizar o projeto. Para auxiliar nestes pontos, o nosso modelo auxilia alguns processos nesta área. Por exemplo, no processo de atribuição de recursos humano das tarefas, o nosso modelo cria um cenário para dar prioridade a tempo. Assim, verificando a disponibilidade de recursos e escolhendo o melhor recurso para cumprir com a estimativa de tempo.

Na gestão de custos, o objetivo é desenvolver e controlar o orçamento do projeto. Os processos no grupo de planejamento para o custo são a gestão do plano de custo, estimativa de custos e determinação do orçamento. De acordo com o PMBOK (PMI, 2013), o gerenciamento de custos do projeto gerencia principalmente o custo dos recursos necessários para terminar as tarefas programadas. No entanto, também deve-se considerar o efeito das decisões de projeto sobre o custo de operação, manutenção e suporte do produto, serviço ou resultado do projeto. Para isto, o nosso modelo apoia processos, tais como, a estimativa de custos e determinar orçamentos. Por exemplo, o modelo é capaz de simular vários cenários de equipes com custos distintos, assim, trazendo mais um tipo de apoio ao processo de decisão do gerente do projeto. Com isto, o gerente de projeto pode escolher uma opção que atenda o orçamento para o projeto.

Gerenciamento de recursos humanos de projetos envolve os processos que organizam e gerenciam a equipe do projeto (PMI, 2013). A equipe do projeto é composta de pessoas com habilidades e responsabilidades. Os membros da equipe devem estar envolvidos em grande parte do processo de planejamento dos projetos e também na tomada de decisões. Gestão de recursos humanos no grupo de planejamento tem apenas um processo que incide sobre o plano de gestão de recursos humanos. O nosso modelo auxilia neste processo, pois ao longo do projeto os recursos humanos podem sofrer alterações. Por exemplo, o tipo e o número de membros da equipe do projeto muitas vezes muda à medida que o projeto evolui. Assim, existe a necessidade de especificar as características dos recursos. Com isto, podemos simular vários cenários de equipes de acordo com a disponibilidade de recursos com o objetivo de cumprir com o tempo ou o custo, e por fim, para apoiar o gerente de projeto na tomada de decisão sobre qual recurso humano utilizar e em qual momento.

Estas quatro áreas e seus respectivos processos consomem uma grande

quantidade de tempo do gerente de projeto. Além disso, a tomada de decisão incorreta pode gerar resultados inesperados. As simulações da gestão destas áreas podem auxiliar os gerentes de projetos, a verificar possíveis alternativas antes de tomar uma decisão, por meio de cenários. Para isto, aplicamos um estudo de caso que descreve como aplicar a instanciação do ProMabs no CORMAS, para executar cenários para auxiliar a tomada de decisão.

5.4

Estudo de Caso

Segundo Runeson e Host (Runeson e Host, 2008), embora exista uma tendência em engenharia de software para os estudos empíricos, abordagens que aplicam estudos de caso estão cada vez mais sendo adotados. De acordo com Runeson e Host (Runeson e Host, 2008), em engenharia de software o método de estudo de caso pode ser aplicado em algumas situações, como no caso dos objetos sob investigação serem fenômenos contemporâneos ou de difícil estudo em isolamento. Nosso estudo foi realizado em um ambiente real que envolveu pessoas que trabalham em projetos de desenvolvimento de software realista em que a aplicação de um experimento controlado não é trivial. Nosso estudo de caso segue uma abordagem qualitativa que utiliza a descrição de estudo de caso para verificar se o nosso modelo de simulação, auxilia o gerente do projeto na tomada de decisões. Especificamente, Runeson e Host (Runeson e Host, 2008) propõem uma estrutura para relatar um caso de estudo em engenharia de software. Esta estrutura envolve, basicamente, um ambiente que descreve as características do caso, um problema que descreve o que deve ser resolvido, uma fase de coleta de dados que é realizada para gerar os resultados e, finalmente, uma fase de análise de dados, na qual os participantes do estudo de caso avaliam os resultados.

O ProMabs foi implementado em CORMAS e aplicado a um estudo de caso em um projeto realista do Laboratório de Engenharia de Software (LES) da Pontifícia Universidade Católica do Rio de Janeiro. Os projetos do LES focam na coordenação e realização de desenvolvimento de sistemas de software para a web e desktop em diferentes domínios, como no petróleo e e-commerce, e também em trabalhos científicos. Usamos parte de um projeto real do LES em nosso estudo de caso. A equipe do LES que participou deste projeto envolveu desenvolvedores, testadores e um gerente de projeto. Um dos objetivos do gerente de projeto foi definir uma equipe para desenvolver parte do software. No entanto, escolher a equipe não era uma tarefa trivial, devido à grande quantidade de dados que precisava ser examinado, por exemplo, características

das tarefas, características dos recursos humanos e as metas do projeto. Para automatizar esse processo, nós aplicamos nossa instanciação do ProMabs no CORMAS, para verificar como apoiar o gestor do projeto em fazer escolhas de equipe.

Para entender melhor o estudo de caso, adotamos a estrutura proposta por Runesson e Host, para relatar um estudo de caso em engenharia de software. Além disso, dividimos o estudo de caso em seções tais como o problema, as orientações gerais, os dados iniciais para começar a simulação, e o resultado da simulação seguido por uma discussão.

5.4.1

Problema escolher uma Equipe

Muitos projetos são realizados no LES, e cada um envolve um gerente de projeto e uma equipe de desenvolvimento. O gerente do projeto normalmente só conhece sua equipe. Além disso, os gerentes muitas vezes têm dificuldades de alocar um recurso humano que não pertence à sua equipe, uma vez que eles não conhecem quais recursos humanos e suas características estão disponíveis em outras equipes. Como também, os recursos humanos disponíveis para os projetos futuros, que podem ser fornecidos por meio de um registro de projeto e ou registro de recursos humanos. Neste contexto, o total de recursos humanos no LES são aproximadamente mais de cem pessoas. A dinâmica e complexidade associadas com disponibilidade de recursos humanos e suas características em situações específicas, é um dos problemas que os gerentes de projeto enfrentam ao escolher uma equipe. Assim, os gerentes de projetos precisam conhecer, por exemplo as características dos desenvolvedores para alocá-los a tarefas da melhor maneira possível de acordo com uma determinada estratégia. Atribuir uma tarefa a um recurso que não se encaixa as características exigidas, pode causar problemas graves, como atrasos nas entregas, maior custo e produtos de uma qualidade não esperada. A questão principal é como apoiar o gerente de projeto em sua tomada de decisão na atribuição de tarefas aos recursos humanos com base na estratégia, por meio de cenários. O problema central é como incorporar os aspectos dinâmicos da simulação gestão do projeto e dos seus recursos para os cenários de simulação.

5.4.2

Orientações gerais

Primeiro, nós instanciamos o ProMabs no CORMAS, conforme descrito na seção 5.3 e também aplicando as diretrizes do PMBOK, conforme descrito na seção 5.3.3. Em seguida, utilizado o ambiente CORMAS para executar a simulação. Para avaliar a eficiência do nosso SBMA, as simulações foram instanciadas a partir de um projeto de desenvolvimento do LES usando elementos como recursos humanos e tarefas. Com isto, discutimos e alisamos quais seriam os benefícios e desvantagens de usar a abordagem de cenários como suporte para os gerentes de projeto no processo de tomada de decisão.

5.4.3

Os dados iniciais para executar a simulação

Dados são necessários para inicializar a simulação. Em nosso estudo de caso, os dados correspondem a tarefas, pacotes de trabalho, recursos humanos, e gerente de projeto de um projeto real. Para alguns destes elementos, informações padrões são necessárias. Por exemplo, cada tarefa necessita de um nome, um nível de conhecimento e uma duração estimada, assim como a sua tarefa predecessora e o pacote de trabalho a que pertence. Apenas o nome é necessário para pacotes de trabalho e para o gerente de projeto. Para recursos humanos, é necessário conhecer o nome do recurso, performance, conhecimento e custo. Finalmente, o projeto precisa de um nome e uma data de início. Para apoiar esta fase, foi utilizada uma ferramenta para armazenar as tarefas, chamado OpenProject. As tarefas foram criadas baseadas no processo de desenvolvimento de software do LES. Estes dados estão disponíveis na página do CORMAS ². Além disto, a figura 5.6 ilustra estes dados. Depois de ter inserido os dados necessários para a simulação, executamos e verificamos seus resultados. A execução de simulação segue os passos de simulação já mencionado no item 5.3.1. A próxima subseção apresenta e discute os resultados.

O modelo precisa de informações sobre os recursos e as tarefas e estas informações são adicionados pelo gerente de projeto através de uma ferramenta (OpenProject), como mencionado anteriormente. Para esta instanciação deste estudo de caso as principais características são o conhecimento de recursos humanos e seus desempenhos. O gerente de projeto classifica o conhecimento

²http://cormas.cirad.fr/fr/applica/MABS_Project_Management.htm

Activities				
Name_Activities	level_knowledge	standard_duration	superActivity	workpackage
Software Requirement Specification	9	2		Software Design
Software Prototyping	7	4		Software Design
Software Unit Coding	3	4	Software Prototyping	Software Design
Software Unit Debugging	3	3	Software Unit Coding	Software Build
Unit Test Conduct	9	5	Software Unit Debugging	Software Build

Resource			
Name_Resource	performance	knowledge	cost
Davy	1	9	50
Marx	2	4	20
Rafael	2	9	60

Figura 5.6: Dados da instanciação do Exemplo.

de recursos entre 1 e 10. Cada número é uma área de conhecimento em que o recurso humano é capaz de executar tarefas. As tarefas possuem também o atributo de conhecimento. Existem também a classificação para o desempenho de recursos humanos. Os valores situam-se entre 1 e 3, em que 1 significa baixo desempenho e 3 de alto desempenho. Esses recursos são classificados de acordo com a experiência de um gerente de projeto, onde o mesmo atribui para o recurso humano. A Figura 5.7 descreve estes atributos.

Name_Knowledge_Area	Knowledge_level
Test	1
Coding (Java)	3
Software architecture	7
Software requirement. de Software	9

Performance	Performance_level
Low	1
Standard	2
High	3

Figura 5.7: Avaliação do Conhecimento e Desempenho do Recurso Humano.

5.4.4 Cenários

Para este estudo de caso, criamos três estratégias, que permitem o gerente de projeto analisar diferentes opções de cenários. Desta forma, auxiliando o gerente de projeto na tomada de decisões. Os cenários foram criados para executar estratégias para tempo, custo e qualidade.

O algoritmo 1 descreve o pseudocódigo das atribuições realizadas pelo agente gerente de projeto. Este algoritmo avalia todas as estratégias e atribui recursos para tarefas baseadas no custo, ou seja, o recurso de menor custo, no tempo, ou seja, o primeiro recurso disponível de uma lista, e na qualidade, ou seja, o recurso com o maior desempenho para realizar a tarefa. O cenário de custo utiliza uma estratégia focada em um custo mais baixo. Na segunda etapa, ilustrado na Figura 5.2, o agente gerente de projeto verifica quais os

agentes recursos humanos são capazes de executar a tarefa e escolhe o agente recurso humano com o menor custo. Ao proceder desta forma, o agente gerente de projeto cria uma equipe para realizar as tarefas com o menor custo.

Algorithm 1 Atribuir Recurso para Tarefa

```

1: function ASSIGNRESOURCE(Senarios, Activities, Resources)
2:   for all scenario S : (Cost, Time, Quality) do
3:     for all activity A  $A_i : A_n$  do
4:       resourceR.knowledge meet activityA.knowledge;
5:       assign activity A to the resourceActivity(S, A, resource_enable)
6:   function RESOURCEACTIVITY(S, A, resource_enable)
7:     if S=Cost then return MIN resource.cost
8:     if S=Quality then return MAX resource.performance
9:     if S=Time then return FIRST Resource

```

O cenário de tempo usa uma estratégia com foco no tempo. Na segunda etapa ilustrada na Figura 5.2, o gerente de projeto verifica os recursos disponíveis apropriados para executar a tarefa e escolhe o primeiro recurso humano. Algoritmo 1 descreve pseudo-código que especifica essa atribuição. Isso permite o agente gerente de projeto reduzir a duração do projeto.

Finalmente, o cenário de qualidade usa uma estratégia focada no desempenho. Na segunda etapa ilustrada na Figura 5.2, o agente gerente projeto verifica os agentes recursos humanos apropriado para executar a tarefa, escolhendo o agente recurso humano com o desempenho superior. Assim, o agente gerente do projeto cria uma equipe com alto desempenho para realizar as tarefas. Algoritmo 1 contém pseudocódigo que inclui esta atribuição. Porém, estas estratégias podem também verificar a disponibilidade dos recursos, em um determinando momento. Desta forma se o agente recurso humano está executando outra tarefa ou se tornar indisponível, o agente gerente de projeto escolhe outro recurso disponível seguindo uma estratégia. Algoritmo 2 ilustra o pseudocódigo que especifica esta abordagem.

Algorithm 2 Agente Recurso Humano indisponível.

```

1: if Resource is not available then return
2:   function ASSIGNRESOURCE(Senarios, Activities, Resources)

```

Com estes algoritmos executam-se diferentes cenários com base em estratégias específicas e oferecem-se sugestões de equipe. Esta é uma das vantagens desta instanciação. Na subseção seguinte, vamos analisar e discutir os resultados da simulação.

5.4.5

Resultado da Simulação e Discussão

A simulação é executada passo a passo. No primeiro passo, o agente gerente de projeto identifica todas as tarefas e agentes recursos humanos. Depois disso, o agente gerente de projeto cria as EAPs e define as datas iniciais e finais estimadas, ordenadas pela sequência de tarefas. No segundo passo, o simulador encontra e atribui os agentes recursos humanos para tarefas. Esta escolha de recursos humanos depende da estratégia e cenário que o gerente de projeto pretende criar. As configurações para estratégia e cenário são alguns dos recursos de suporte do simulador para o processo de tomada decisão do gerente de projeto, pois uma vez configurado o simulador executa os cenários automaticamente. Mas esta configuração pode não ser a melhor opção, porque o simulador considera alguns detalhes, como por exemplo, agente recurso humano está ocupado. Existe uma solução que descreveremos adiante. Uma das desvantagens para executar essas configurações é a necessidade das características de recursos e tarefas. Esta é uma limitação da simulação.

O ciclo que envolve a terceira e quarta etapas é executado até que todas as tarefas terminem a sua execução. O agente recurso humano recebe uma tarefa e a executa. No cenário de tempo, se um recurso está ocupado, o agente gerente de projeto recebe uma mensagem do agente recurso humano, notificando que não pode realizar a tarefa. Então, o agente gerente de projeto verifica se há outros recursos disponíveis. Se um recurso for encontrado, o agente gerente de projeto verifica se o recurso tem as características apropriadas para executar a tarefa, e delega a execução desta tarefa para este agente recurso humano. Desta forma, está é outra vantagem do modelo para apoiar a tomada de decisão, porque o simulador se auto-organiza, criando cenários alternativos. Ao definir a configuração inicial para a estratégia de custo, o simulador encontra um recurso de custo inferior. Para a estratégia de qualidade o simulador encontra um recurso de alto desempenho. Como descrito anteriormente.

Finalmente, depois de todas as tarefas finalizarem, a simulação gera um relatório com o resumo da simulação, por meio de cenários, como ilustrado nas Figuras 5.7(a), 5.7(b) e 5.9. Este relatório também descreve as opções para o gerente de projeto. A Figura 5.7(a) é um gráfico que ilustra a relação entre o custo total e os cenários. A Figura 5.7(b) é um gráfico que ilustra a relação entre o total de dias e os cenários. E a Figura 5.9 indica a relação entre a equipe sugerida e cenários.

Até então os gerentes de projeto não tinham informações sobre como

5.7(a): Custos do Projeto x Cenários.



5.7(b): Dias do Projeto x Cenários.

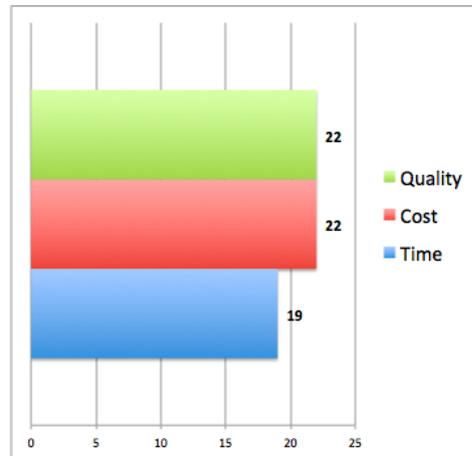


Figura 5.8: Report with the summary of the simulation.

	Team				
Time	Davy	Rafael	Marx	Marx	Davy
Cost	Davy	Davy	Marx	Marx	Davy
Quality	Rafael	Rafael	Marx	Marx	Rafael
Software Requirement Specification					
Software Prototyping					
Software Unit Coding					
Software Unit Debugging					
Unit Test Conduct					

Figura 5.9: Equipes Projeto x Cenários.

a equipe poderia ser escolhida com base em uma determinada estratégia. Para isto, a simulação sugere equipes de acordo com as estratégias, por meio de cenários. Por outro lado, há uma desvantagem, é necessário fornecer as características dos recursos humanos, pois o gerente de projeto só conhece as características dos recursos do qual é responsável. Como sugestão, o LES poderia resolver este problema por meio da criação de uma norma, na qual todos os gerentes de projeto precisassem fornecer tais características do recurso humano por meio de uma ferramenta, que possua uma base de dados que possa ser acessada.

Com esta simulação, existem indícios positivos de que o gerente de projeto é auxiliado no processo de tomada de decisão durante o planejamento do projeto. As Figuras 5.7(a), 5.7(b) e 5.9 ilustram os resultados da simulação. O gerente de projeto pode usar estes resultados para criar uma equipe de acordo com uma estratégia. Por exemplo, se a prioridade é tempo, a equipe sugerida

é Davy, Rafael e Marx com as respectivas tarefas, e o custo total é de \$730 em 19 (dezenove) dias. Se a prioridade é o custo, a equipe sugerida é Davy e Marx com as respectivas tarefas, e custo total é de \$690 em 22 (vinte e dois) dias. Se a prioridade é a qualidade, a equipe sugerida é Rafael e Marx com as respectivas tarefas, e o custo total é de \$800 em 22 (vinte e dois) dias. Assim, o gerente de projeto pode obter sugestões para definir uma equipe. Além disto, os resultados da simulação fornecem uma visão de toda a estrutura do projeto em termos de recursos humanos e tarefas necessárias para concluir o projeto. Com esses dados, o gerente de projeto tem uma ferramenta para auxiliar em suas atividades. Além disso, a simulação só precisa de um modelo para executar vários cenários. Os argumentos apresentados foram de acordo com a análise deste estudo de caso que apresenta evidências de que ProMabs pode ser usado para auxiliar os gerentes de projeto efetivamente em sua tomada de decisão, por meio de cenários.

5.5

Considerações Finais

Neste capítulo, apresentamos e detalhamos como o ProMabs instanciado por um ambiente de simulação, executa cenários por meio de estratégias para auxiliar no processo de tomada de decisão. Para isto, aplicamos a abordagem de representação descrita no capítulo 4. Além disto, utilizamos as diretrizes do PMBOK das áreas de escopo, tempo, custo e recursos humano para criar as estratégias. Este capítulo se baseia no relatório técnico (Baia et al., 2014), que auxiliou na compreensão de como os agentes podem executar cenários baseado em estratégias para simular gestão de projetos e auxiliar a tomada de decisão.

Para instanciar o ProMabs utilizamos a proposta dos componentes filtro de informação, modelo de planejamento, cenários de execução e finalmente o analisador. Esta instanciação permitiu apenas a visualização de cenários ao final da execução. Com isto, auxiliando a verificar como estes cenários podem auxiliar na tomada de decisão, por meio de um estudo de caso.

Entretanto, uma visualização completa, do início ao fim da execução da simulação, pode auxiliar os envolvidos no projeto a analisar o comportamento do projeto durante a simulação. Desta forma, no próximo capítulo, iremos descrever como o ProMabs instanciado em uma ferramenta de simulação, apoia projetos adaptativos de software. Além disto, iremos aplicar uma visualização da simulação do início ao fim da simulação. Para isto, utilizando todos os elementos como agentes. Cada agente incorporando suas características, como

descrita no capítulo 3. Desta forma, tarefas, pacotes de trabalho e projeto, serão agentes também. Permitindo a possibilidade de verificar o comportamento em uma determinada unidade de tempo para todos estes elementos.

6

Suporte ao Gerenciamento de Projetos Adaptativo de Software

Os capítulos anteriores descrevem a identificação dos elementos e seus relacionamentos necessários para representar gestão de projetos de software, no âmbito de gestão de escopo. Execução de cenários são baseados em estratégias, como tempo, custo e qualidade, que auxiliam no processo de tomada de decisão. Respectivamente, instanciando o ProMabs em uma plataforma de programação multiagentes e em um ambiente de simulação multiagentes. Este capítulo apresenta a aplicação do ProMabs para apoiar à gestão de projetos adaptativos de software, por meio de simulações. Projeto de Software ao longo de sua gestão sofrem diversas alterações, seja por motivos de eventuais solicitações dos interessados no projeto ou por imprevisto ao longo da execução do projeto. Desta forma, uma simulação que oferece suporte à projeto adaptativos de software precisa ser robusta o suficiente para adaptar-se a estas alterações. Neste contexto, este capítulo se concentra na terceira questão de pesquisa desta tese:

QP1(iii). Como apoiar ao gerenciamento de projetos adaptativos de software com o uso de SBMA?

Para isto, na seção 6.1 introduzimos a noção de como os elementos que compõem a gestão de projeto de software sofrem adaptações. Na seção 6.2 descrevemos os trabalhos relacionados que utilizam simulação para auxiliar projetos. Na seção 6.3 instanciamos o ProMabs a uma ferramenta de simulação para apoiar projetos adaptativos de software. Na seção 6.4 apresentamos por meio de exemplos ilustrativos, como a instanciação do ProMabs permite a gestão de projetos adaptativos de software. Finalmente, na seção 6.5, apresentamos a conclusão e algumas discussões.

6.1

Projetos Adaptativos de Software

Simulações de forma geral são aplicadas para representar a dinâmica de um modelo que permite avaliar o seu comportamento ao longo da simulação. Porém, ao longo da simulação os seus dados podem sofrer alterações existindo eventuais necessidades da simulação se adaptar a esta mudança. Em particular, simulação de gestão de projetos de software precisa perceber os acontecimentos do projeto no ambiente externo e se adaptar a possíveis alterações que acontecem em seus dados de instanciação. Assim, existe uma maior atenção para que os modelos de simulação de gestão de projeto de software dêem suporte as alterações que ocorrem ao longo do projeto.

Projetos de software, geralmente, estão em constante alteração seja por motivos de mudanças nos requisitos, ou por imprevistos que acontecem ao longo do projeto. Desta forma, chamamos de projeto adaptativo de software os projetos que estão em constante alteração de contexto, ou seja, que alteram seus elementos e seus relacionamentos. Para dar suporte a estas alterações, um modelo de simulação de gestão de projeto de software precisa ser robusto o suficiente para perceber estas alterações que acontecem no ambiente externo e incorpora-las no modelo, sem uma alteração significativa. Segundo Luck e McBurney (Luck e McBurney, 2008), simulações baseada em multiagente fornecem fortes modelos para representar ambientes do mundo real que são complexos e dinâmicos.

Entretanto, o gerenciamento de projeto de software precisa de uma abordagem de modelo que viabilize estas alterações de contexto podendo simular os impactos que estas alterações podem causar. Por exemplo, uma nova função é necessária no software, com isto, uma nova tarefa é criada na Estrutura Analítica do Projeto (EAP). Com esta alteração de contexto, os envolvidos no projeto necessitam verificar o impacto desta mudança. Será que existe recurso para realizar esta tarefa? Ou qual o reflexo desta mudança no tempo ou no custo? Com a simulação desta alteração de contexto os envolvidos na tomada de decisão podem verificar possíveis impactos, desta forma, auxiliando na gestão de projetos. Para isto, utilizamos a abordagem do ProMabs descrita no capítulo 3 para viabilizar estas alterações, aplicadas a uma ferramenta de simulação.

6.2

Trabalhos Relacionados

São duas as principais linhas de investigação que estão relacionadas com este capítulo: (i) simulação que não utiliza multiagentes para auxiliar a gestão de projeto; e (ii) simulação que utiliza multiagentes para auxiliar a gestão de projeto. Na primeira linha de investigação, temos alguns trabalhos que utilizam algoritmos genéticos, porém focados apenas em planejamento (Xiao et al., 2013, Yannibelli e Amandi, 2011, Alba e Chicano, 2007, Hanne e Nickel, 2005). Outros trabalhos aplicam dinâmica de sistema, disparado por eventos introduzidos por humanos, voltado para educação de gestão de projetos de software (Pfahl et al., 2001, Rodríguez et al., 2004). Também existem trabalhos que utilizam mineração de dados (Haapio e Menzies, 2009, Balsera et al., 2012), porém que necessitam um grande volume de dados históricos de gestão de projetos.

Na segunda linha de pesquisa, temos trabalhos que utilizam sistema multiagentes na gestão de projeto de software, porém com objetivos distintos. O objetivo de Wu et al. (Wu et al., 2009) é utilizar agentes para aplicar métricas. Já o de Athavale e Balaraman (Athavale e Balaraman, 2013) foca em verificar o comportamento humano. Araúzo et al., propõem utilização de agentes para a gestão de portfólio, ou seja, a gestão de vários projetos (Araúzo et al., 2010).

Finalmente, nossa abordagem se diferencia das demais supracitadas por focar na simulação de gestão planejamento e execução do projeto, assim, dando suporte as alterações de contexto que acontecem ao longo do projeto. Ao aplicar o ProMabs, os envolvidos na gestão de projeto verificam os comportamentos de cada elemento do projeto e os impactos das alterações destes elementos, com a vantagem de possuir um único modelo de simulação. Assim, focando tanto na gestão do planejamento como na gestão da execução do projeto.

6.3

ProMabs para Ferramenta de Simulação

O objetivo desta seção é aplicar o ProMabs a uma ferramenta de simulação e auxiliar a gestão de projetos adaptativos de software. O ProMabs descrito no capítulo 3 é um modelo conceitual baseado em multiagente com cinco componentes principais. Estes componentes auxiliam na criação de um modelo que aceitem alterações ao longo do projeto.

Para apoiar as alterações de contexto e com isto auxiliar a gestão de projetos de software, instanciamos o ProMabs em uma ferramenta de simulação, chamada Anylogic (Borshchev, 2013). Esta ferramenta está disponível gratuitamente para fins educacionais e é bem utilizada na Europa para fins comerciais. O Anylogic é desenvolvido em Java, com isto facilitando a sua customização e extensão.

Cada componente do ProMabs ilustrado na figura 6.1 possui um objetivo na simulação. O filtro de informação faz a conexão com ambiente externo, assim, fornecendo dados para os componentes modelo de planejamento e execução. A instanciações dos componentes e o ambiente externo, no Anylogic, são descritos em termos de objetivos nas próximas subseções.

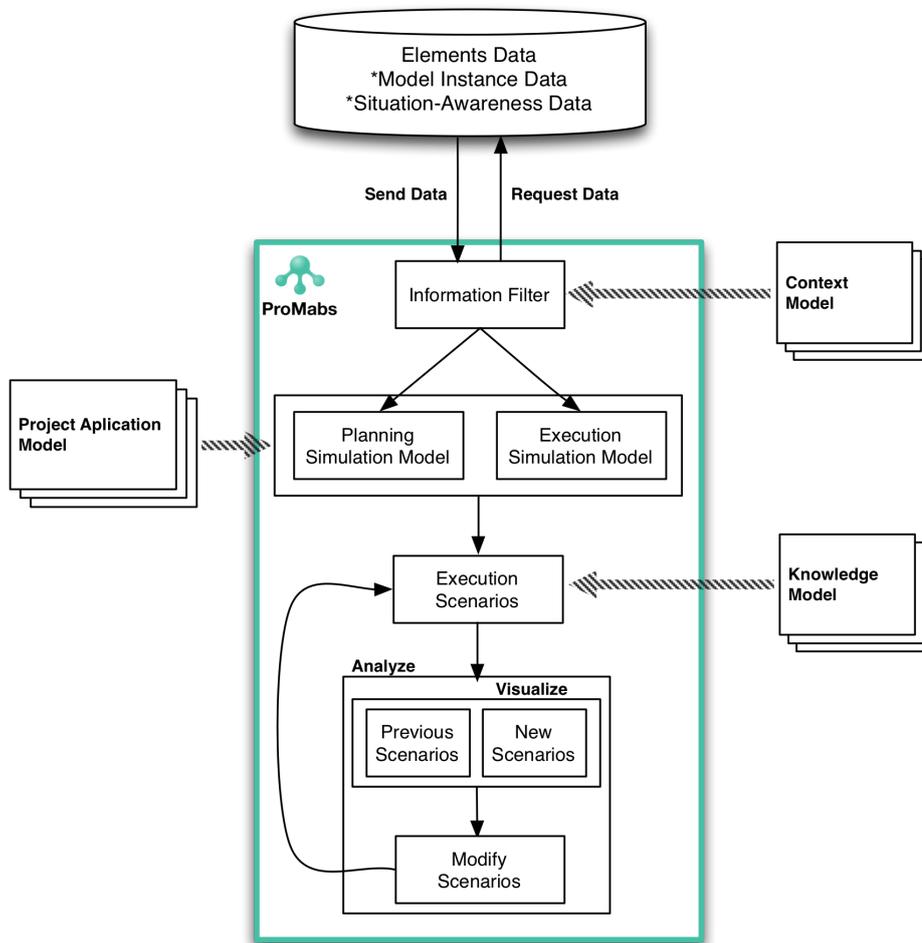


Figura 6.1: Componentes Instanciados do ProMabs no Anylogic.

6.3.1

Ambiente Externo e o Filtro de Informação

O ambiente externo possui os dados de instanciação e dados situacionais que o ProMabs recebe por meio de solicitação do filtro de informação. As tabelas a seguir exemplificam como estes dados são formatados.

A tabela 6.1 apresenta os dados do projeto. Estes dados são usados por exemplo, para simular o modelo de planejamento com a data inicial inserida no campo `dateStart` ou verificar se o projeto irá passar da `dateEnd` estabelecida, sendo isto uma restrição. Por outro lado, se um projeto já iniciou no ambiente externo e os envolvidos querem simular o modelo de execução só poderão utilizar o campo `dateEnd` como restrição, visto que o projeto já foi inicializado no ambiente externo e sua data de início já passou. Os outros dados são informativos e auxiliam na criação do *agente projeto* na EAP do simulador.

Tabela 6.1: Exemplos dos Dados do Projeto.

id_project	name_project	dateStart	dateEnd
1	project1	10/10/15	10/01/16

A tabela 6.2 apresenta os dados do gerente de projeto. Estes dados são usados para verificar onde um gerente de projeto esta alocado. Visto que o ProMabs, permite instanciar mais de um projeto por simulação. Isto acontece quando os envolvidos precisam verificar os recursos compartilhados para um conjunto de projetos. O campo `id_project` identifica qual projeto pertence o gerente de projeto. Tanto no modelo de planejamento quanto no modelo de execução permite a alteração deste contexto. Os outros dados são informativos e auxiliam na criação do *agente gerente de projeto* do simulador.

Tabela 6.2: Exemplos dos Dados do Gerente de Projeto.

id_project_manager	name_project_manager	id_project
1	manager1	1

A tabela 6.3 apresenta os dados do pacote de trabalho. Estes dados são informativo e auxiliam na criação dos *agentes pacotes de trabalhos* da EAP do simulador. Porém, para o modelo de execução estes dados podem se alterar quando existem mudanças nos pacotes de trabalho. Por exemplo, um projeto foi inicializado e um pacote de trabalho ou alterou seu predecessor ou não tem mais predecessor. Então, é no campo `id_super_workpackage` que contem está

informação. Este contexto pode se alterar ao longo do projeto, assim, podendo ser simulada uma nova estruturação dos pacotes de trabalhos e seus impactos com estas alterações.

Tabela 6.3: Exemplos dos Dados do Pacote de Trabalho.

id_workpackage	name_workpackage	id_project	id_super_workpackage
1	workpackage1	1	3

A tabela 6.4 apresenta os dados dos recursos humanos. Estes dados são usados por exemplo, para simular o modelo de planejamento, o qual precisa alocar recursos humanos às tarefas por meio do campo `skill_type`, onde a tarefa exige uma habilidade específica para executá-la. Similarmente, o campo `skill_level` é utilizado, podendo ser um campo aplicado em uma estratégia, na qual encontra os recursos humanos com um maior ou menor níveis de conhecimento. O simulador pode executar vários cenários com combinações de alocações de recursos humanos para tarefas. Descrevemos detalhadamente na subseção 6.3.3. Por outro lado, no modelo de execução, os envolvidos no projeto podem verificar o impacto na alteração do custo de um recurso humano, para o projeto. Outro exemplo seria, por um motivo fortuito o recurso humano que está disponível se tornar indisponível, isto é verificado por meio do campo `available`. Os envolvidos no projeto, precisam analisar o impacto desta alteração de contexto, e também verificar qual o recurso humano deve ser alocado para substituir o recurso humano indisponível. Os outros dados são informativos e auxiliam na criação dos *agentes recurso humano* no simulador.

Tabela 6.4: Exemplos dos Dados do Recurso.

id_h_resource	name_h_resource	skills_type	skill_levels	cost	available
1	humanResource1	1	1	10	1

A tabela 6.5 apresenta os dados da tarefa. Estes dados são usados por exemplo, para simular o modelo de planejamento, o qual precisa verificar as datas de início e fim de cada tarefa por meio da utilização do campo `duration`. Este campo contém a informação de duração estimada da tarefa. Porém, o gerente de projeto pode simular as datas efetivas de início e término da tarefa de acordo com o nível de conhecimento do recurso humano que está executando a tarefa. Neste caso, podendo aplicar informação de performance por nível, por exemplo, o sênior geralmente entrega na data estimada e o júnior geralmente atrasa. Por outro lado, no modelo de execução o campo `state` é atribuído com o valor 1 (um) quando esta tarefa já foi executada no ambiente externo e o simulador precisa seguir exatamente as informações desta tarefa. Caso

o campo `state` possua o valor 0 (zero), o simulador pode fazer alterações, como por exemplo, no recurso humano atribuído para a tarefa, assim, testando outras configurações de recursos humano. Com isto, os envolvidos no projeto podem escolher uma equipe específica de acordo com uma estratégia. Os campos `skill_type` e `skill_level` são usados como informação para estratégia. As estratégias serão descritas na subseção 6.3.3. Já o campo `predecessor_id_task` é a informação na qual diz qual tarefa precisa terminar antes que esta comece, assim, sendo uma restrição. Os outros dados são informativos e auxiliam a criação dos *agentes tarefas* na EAP do simulador.

Tabela 6.5: Exemplos dos Dados da Tarefa.

id_task	name_task	id_workpackage	skills_type	skill_levels	duration
1	task1	1	1	1	5
	predecessor_id_task	id_h_resource	state		
	2	1	0		

Estas configurações de dados fazem parte do modelo do contexto, e os envolvidos na tomada de decisão podem simular alterações destes dados. Os envolvidos no projeto, identificam quais contextos e suas mudanças querem observar.

Estes dados são dados de instanciação e dados situacionais para o simulador. Desta forma, sendo necessários para configuração da EAP no simulador, na qual utiliza ligações entre um campo a outro para montar a sua estrutura dos *agentes projetos*, pacote de trabalho e tarefas. Além disto para criar os agentes gerentes de projetos e recurso humanos. Porém, estes dados passam pelo filtro de informação para identificar para qual é o tipo de dado e contexto, com isto, identificando qual o modelo de simulação apropriado. Para testar o filtro de informação conectado ao ambiente externo, utilizamos o gerador mencionado no capítulo 3 para criar várias e diferentes EAPs. Com esta, verificamos se a instanciação do ProMabs na ferramenta do Anylogic, dá suporte a várias e diferentes EAPs. A instanciação atendeu todas as configurações testadas de EAPs, apresentando um resultado satisfatório, ou seja, montou a sua estrutura tanto para o modelo de planejamento quanto para o modelo de execução.

6.3.2

Modelo de Simulação de Planejamento e Execução

O modelo de simulação para o planejamento e execução do ProMabs, instanciado no Anylogic, é uma Estrutura Analítica de Projeto (EAP) representando o modelo de aplicação de projeto utilizado neste estudo. A EAP é uma ferramenta que geralmente é aplicada para melhorar a compreensão e tangibilidade dos projetos, para isto, divide as entregas do projeto em partes menores, como tarefas e pacote de trabalho. Com os dados e o filtro de informação já estão estruturados em uma EAP então o modelo de simulação precisa montar a estrutura, e para isto, os algoritmos ilustrados nas figuras 6.2, 6.3 são baseados no modelo de aplicação, desta forma, sendo utilizado no componente cenários de execução e no componente analisador para visualizar uma EAP.

A figura 6.2 ilustra o algoritmo utilizado para calcular os espaços na grades dos nós, ou seja, os espaços para os elementos da EAP, que são: projeto, pacotes de trabalho e tarefas. Com as ligações destes elementos, a EAP é representada por uma árvore. O algoritmo percorre a estrutura da árvore fornecida, calculando e com isto acumulando a quantidade de filhos para cada nó.

```

▷ Calculando espaço dos nós da árvore a partir da raiz
function CACULATESPACE
  widthAcc = 0
  ▷ Calculando espaço dos nós filhos e acumulando
  for node in children do
    childWidth = node.CalculateSpace()
    widthAcc = widthAcc + childWidth
  ▷ Guardando espaço em atributo interno
  this.width = max{widthAcc, max{children.size, 1}}
  return this.width

```

Figura 6.2: Pseudocódigo para calcular distância entre nós.

A figura 6.3 ilustra o algoritmo utilizado para calcular as posições na grades dos nós. Para isto, percorre a estrutura da árvore fornecida, calculando as posições dos nós de acordo com os espaços de cada nó calculado anteriormente.

Estes algoritmos são utilizados para montar a EAP. Desta forma, fornece a estrutura a ser seguida nos cenários de execução e conseqüentemente no componente analisador. A figura 6.4 ilustra a EAP resultante destes algoritmos.

```
▷ Computando as posições dos nós filhos a partir da raiz
procedure CALCULATEPOSITION(row, col)
  this.row = row
  this.column = col
  rowAcc = row
  for node in children do
    node.CalculatePosition(rowAcc, this.column + 1)
    ▷ Usa o espaço calculado em CalculateSpace
    rowAcc = rowAcc + node.width
```

Figura 6.3: Pseudocódigo para calcular a posição dos nós.

Utilizamos uma EAP como modelo de aplicação de projeto, por ser flexível o suficiente para ser aplicada em um grande número de tipos de projetos. E também por simplificar a visualização da execução do projeto no momento da simulação. Porém, um outro tipo de modelo de aplicação poderia ter sido usado, como por exemplo, um gráfico de gantt.

Como visto na seção anterior, a identificação do contexto auxilia a diferenciar qual o modelo de simulação será aplicado, planejamento ou execução. No modelo de planejamento os agentes possuem autonomia para executar alterações nas tarefas, por outro lado, no modelo de execução algumas tarefas possuem restrições, no caso será representada e simulada respeitando as informações externas, assim, limitando as alterações nas tarefas. Com isto, os cenários de simulação estão criados permitindo o componente cenários de execução simula-los.

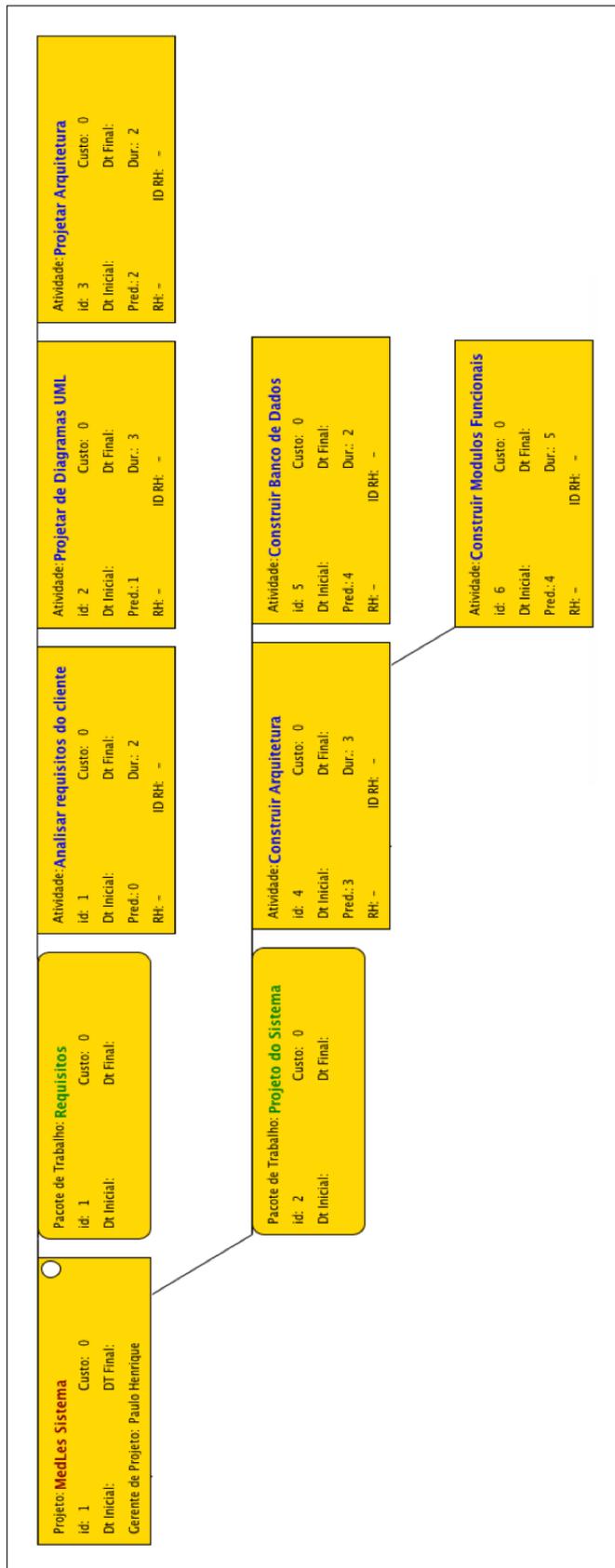


Figura 6.4: Exemplo de EAP para Cenário de Execução e Analisador.

6.3.3

Cenários de Execução

Como descrito na seção 3.3, cada elemento do projeto é um agente que possui uma arquitetura e assume um papel no ProMabs. Assim, temos agentes do tipo projeto, gerente de projeto, pacote de trabalho, tarefa e recurso humano. Cada tipo de agente é criado de acordo com os dados descritos na seção 6.3.1. Para execução da simulação os agentes seguem/obedecem seu diagrama de estado que será apresentados a seguir.

Agente Projeto

O *agente projeto* possui basicamente três estados, o planejado, em execução e finalizado como ilustrado na figura 6.5. Os estados são disparados de acordo com acontecimentos na simulação. Para passar para do estado de planejamento para execução, o *agente projeto* verifica a data de início do projeto e ao chegar a esta data, autoriza os agentes pacotes de trabalhos iniciarem. O *agente projeto* permanece em estado de execução até que todos os agentes pacotes de trabalhos finalizem. Para isto, o *agente projeto* verifica os estados dos agentes pacotes de trabalho. Sendo esta uma das vantagens de utilizar *agente projeto*, permitir a verificação dos estados dos agentes pacotes de trabalho, com isto, sendo também um agente fiscalizador. Além de cada *agente projeto* possuir características próprias, representando um projeto do ambiente externo. Ao final de cada execução do *agente pacote de trabalho*, o atributo custo do *agente pacote de trabalho* é somado no *agente projeto* e ao final de todos agentes pacotes de trabalho é atribuído uma data final para o agente projeto, chegando assim ao estado finalizado.

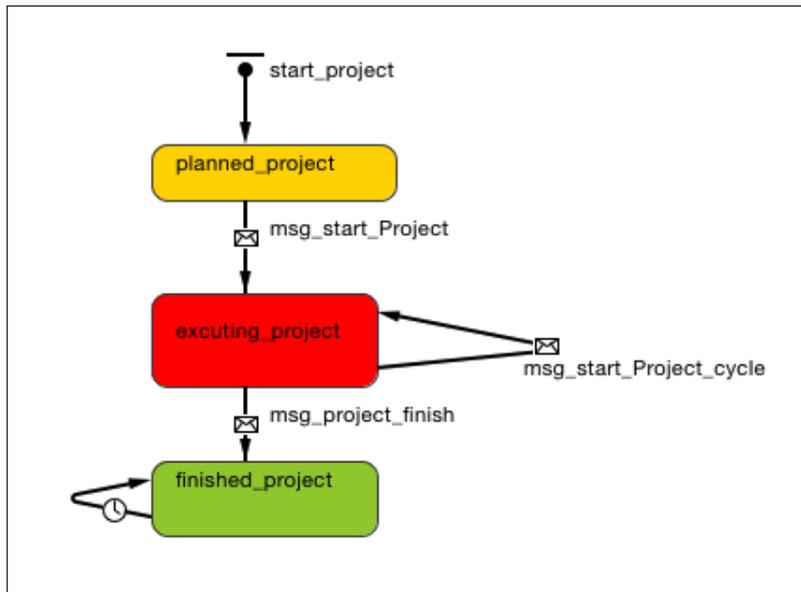


Figura 6.5: Estados do Agente Projeto.

Agente Pacote de Trabalho

O *agente pacote de trabalho* possui basicamente três estados, o planejado, em execução e finalizado como ilustrado na figura 6.6. Os estados são disparados de acordo com acontecimentos na simulação. Para passar do estado de planejamento para o de execução, o agente recebe uma mensagem do *agente projeto*, autorizando seu início. Porém, se este *agente pacote de trabalho* possuir um *agente pacote de trabalho* predecessor, então, só passará para o estado de execução quando o seu predecessor enviar uma mensagem autorizando seu início. Ao passar para o estado de execução, é atribuído uma data de início e seus agentes de tarefas recebem uma mensagem, autorizando seu início. O *agente pacote de trabalho* permanece em estado de execução até que todos os seus agentes tarefas finalizem. Para isto, o *agente pacote de trabalho* verifica os estados dos agentes tarefas. Sendo esta, uma das vantagens de utilizar um *agente pacote de trabalho*, permitir a verificação dos estados dos *agente tarefas*, com isto, sendo também um agente fiscalizador. Além de cada *agente pacote de trabalho* possuir características próprias, representando um pacote de trabalho do ambiente externo. Ao final de cada *agente tarefa*, o atributo custo do *agente tarefa* é somado no *agente pacote de trabalho* e ao final de todos os agentes tarefas é atribuído uma data final para o *agente pacote de trabalho*, chegando assim ao estado finalizado.

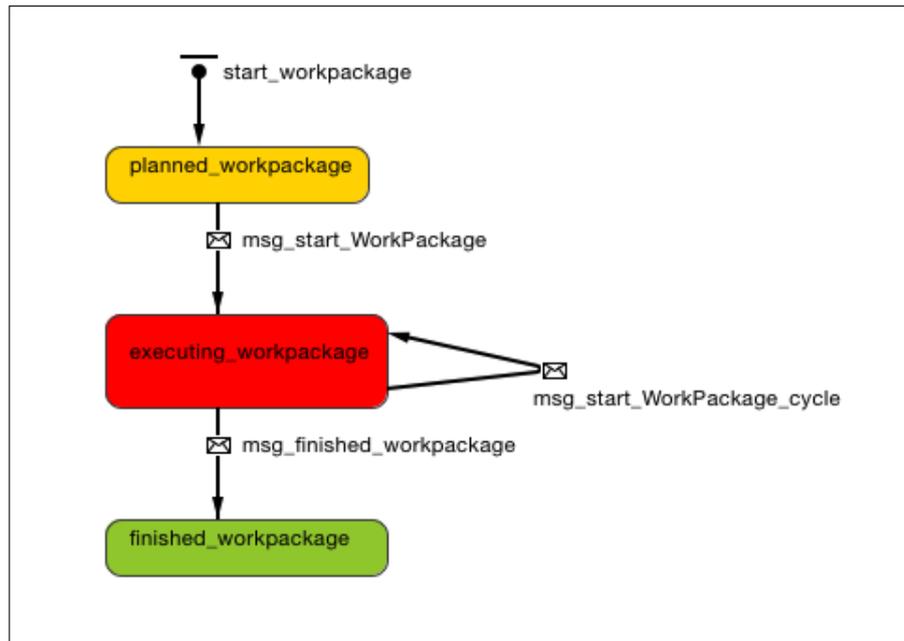


Figura 6.6: Estados do Agente Pacote de Trabalho.

Agente Tarefa

O *agente tarefa* possui cinco estados o planejado, pré-condição, em execução, finalizado e próxima tarefa como ilustrado na figura 6.6. Os estados são disparados de acordo com acontecimento na simulação. Para passar do estado de planejamento para pré-condição, o *agente tarefa* recebe uma mensagem do *agente pacote de trabalho*, autorizando verificar a pré-condição. Porém, se este *agente tarefa* possuir um *agente tarefa* predecessor, então, só passará para o estado de pré-condição quando o seu predecessor enviar uma mensagem autorizando. Ao passar para o estado de pré-condição, o *agente tarefa* verifica se possui um *agente recurso humano* atribuído, caso positivo, envia uma mensagem para o *agente recurso humano* autorizando a execução da tarefa e passa para o estado de execução. Por outro lado, se não tiver um *agente recurso humano* atribuído, é solicitado que o *agente gerente de projeto* intervenha na escolha de um *agente recurso humano*, detalharemos esta atividade na subseção 6.3.3.

Já em estado de execução é atribuído uma data de início para o *agente tarefa*. O *agente tarefa* permanece em estado de execução até que o *agente recurso humano* finalize a execução da tarefa e informe sobre. Uma das vantagens de utilizar um *agente tarefa* é possuir por exemplo, um estado que verifica as pré-condições para a tarefa ser executada, assim, antecipando as alocações e regras necessárias para serem validadas na execução da tarefa.

Além de cada *agente tarefa* possuir características próprias, representando uma tarefa do ambiente externo. Ao ser informado sobre a finalização da execução da tarefa pelo *agente recurso humano*, o custo total e a data final é atribuído para o *agente tarefa*. Após isto, passa-se para o estado que autoriza a inicialização do *agente tarefa* sucessor.

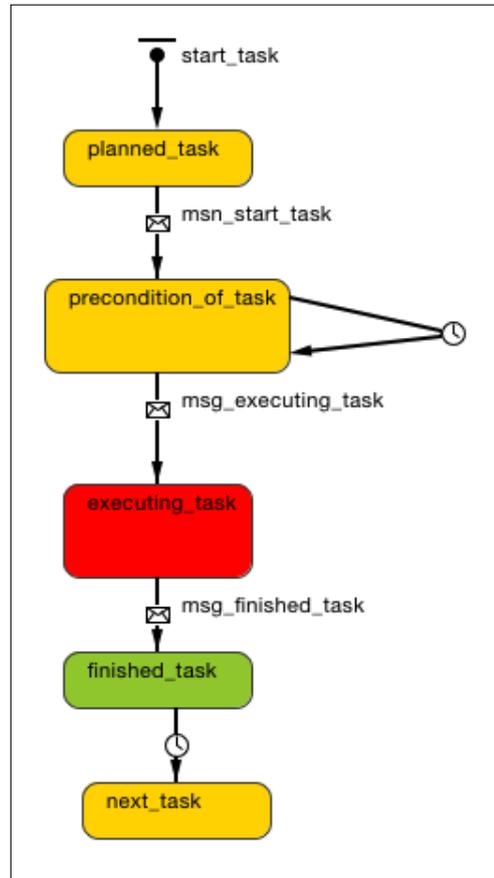


Figura 6.7: Estados do Agente Tarefa.

Agente Recurso Humano

O *agente recurso humano* possui cinco estados o disponível, descanso, indisponível, horário de intervalo e pré-condição como ilustrado na figura 6.8. Os estados são disparados de acordo com acontecimento na simulação. Para passar do estado de disponível para descanso o agente humano recebe uma notificação avisando que na unidade de tempo atual está em descanso. Para voltar para o estado de disponível recebe uma notificação avisando que na unidade de tempo atual está em horário de trabalho. Porém, o *agente recurso humano* só está disponível quando seu campo avaliável está atribuído o valor 1 (um).

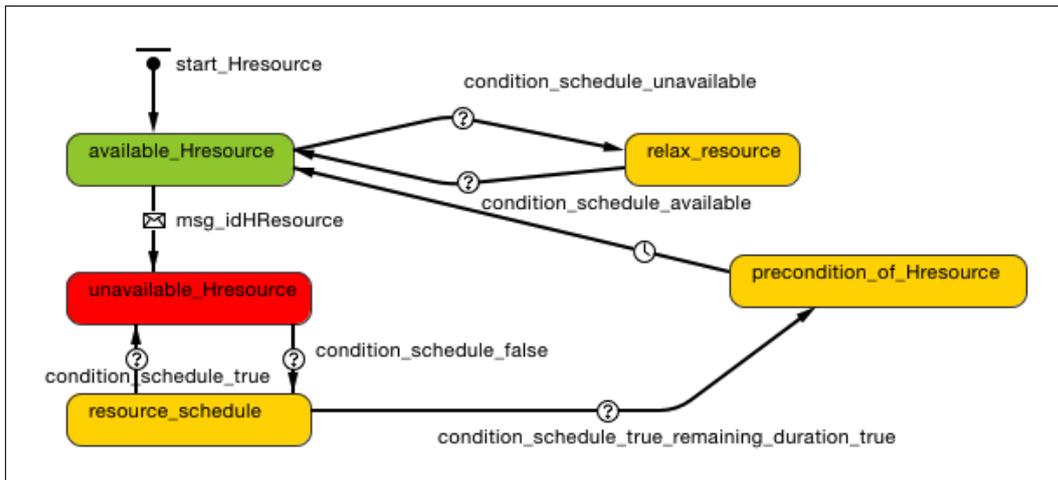


Figura 6.8: Estados do Agente Recurso Humano.

O horário de trabalho do recurso humano segue como descrito na seção 3.3, no caso o sistema atualiza a cada unidade de tempo a informação deste horário. A vantagem de aplicar um objeto horário de trabalho para o recurso é que realisticamente os horários podem variar ao longo do projeto. Assim, os envolvidos na tomada de decisão podem verificar o impacto das mudanças deste horário.

Ao receber uma solicitação de execução de uma tarefa o *agente recurso humano* passa do estado disponível para indisponível, pois neste caso, só pode realizar uma tarefa por vez. Como existe um horário de trabalho atrelada a uma unidade de tempo a cada vez que sai do seu horário de trabalho o agente recuso humano precisa passar para um outro estado, para simular estas unidades de tempo e horário de intervalo de trabalho. Neste caso, o agente passa para o estado horário de intervalo. Para exemplificar, temos um dia de trabalho possuindo 8 hrs, o horário de trabalho é de 8:00 até 18:00 hrs, então ao chegar as 18:00 hrs o *agente recurso humano* entre em seu horário de intervalo. Além disto, vira uma unidade de tempo que neste caso é dia. Com isto, temos um ciclo entre o estado indisponível e horário de intervalo até o *agente recurso humano* completar a tarefa. A vantagem de usar este ciclo é que os envolvidos na tomada de decisão, podem simular eventos como por exemplo, atraso devido a algum imprevisto ou por um nível de conhecimento baixo do agente recurso. Este ciclo também é utilizado para informar o *agente tarefa*, a cada unidade de tempo, o progresso da execução da tarefa.

Após a finalização da tarefa o agente passa para o estado de pré-condição para verificar como terminou a tarefa e voltar para o estado disponível. A vantagem de utilizar um *agente recurso humano* é possuir por exemplo,

características próprias, representando o recurso humano do ambiente externo. Assim, podendo criar simulação diferentes de acordo com as características dos recursos humanos. Por exemplo, um horário diferenciado para um recurso humano ou um recurso humano possuindo várias habilidades ou uma taxa de desempenho que depende da performance dos recursos humanos.

Agente Gerente de Projeto

O *agente gerente de projeto* possui cinco estados conhecendo, definindo tarefa, definindo por nível de habilidade, definindo por custo e atribuindo recurso humano como ilustrado na figura 6.9. No estado conhecendo, o agente de projeto irá se informar de todas as tarefas e recursos humanos. Para este estudo o agente de projeto atua em dois momentos, (i) quando uma tarefa não tem recurso humano alocado; e (ii) quando precisa priorizar tempo e um *agente recurso humano* alocado para tarefa está ocupado realizando outra tarefa, então, o *agente gerente de projeto* é acionado para alocar outro *agente recurso humano* caso tenha o tempo como prioridade. Para essas duas atividades ele usa um modelo de conhecimento baseado em estratégia. São duas estratégias a de (i) nível de conhecimento e de (ii) menor custo. A descrição das duas estratégias são:

Nível de Conhecimento - Para escolher um *agente recurso humano*, não atribuído para uma tarefa, o *agente gerente de projeto* escolhe um recurso humano que tem a mesma habilidade exigida da tarefa, além disso procura um nível de conhecimento igual ao sugerido da tarefa. Porém se não encontrar um nível de conhecimento igual da atividade o *agente gerente de projeto* procura algum nível de conhecimento maior, seguindo uma ordem (júnior, pleno, sênior). Caso não encontre, procura um nível de conhecimento menor seguindo uma ordem (sênior, pleno, júnior). Levando em consideração sempre a disponibilidade do recurso humano no momento da atribuição.

Custo - Para escolher um *agente recurso humano*, não atribuído para uma tarefa, o *agente gerente de projeto* escolhe um *agente recurso humano* que tem a mesma habilidade exigida da tarefa, além disso procura o *agente recurso humano* que tem o menor custo. Levando em consideração sempre a disponibilidade no momento da atribuição.

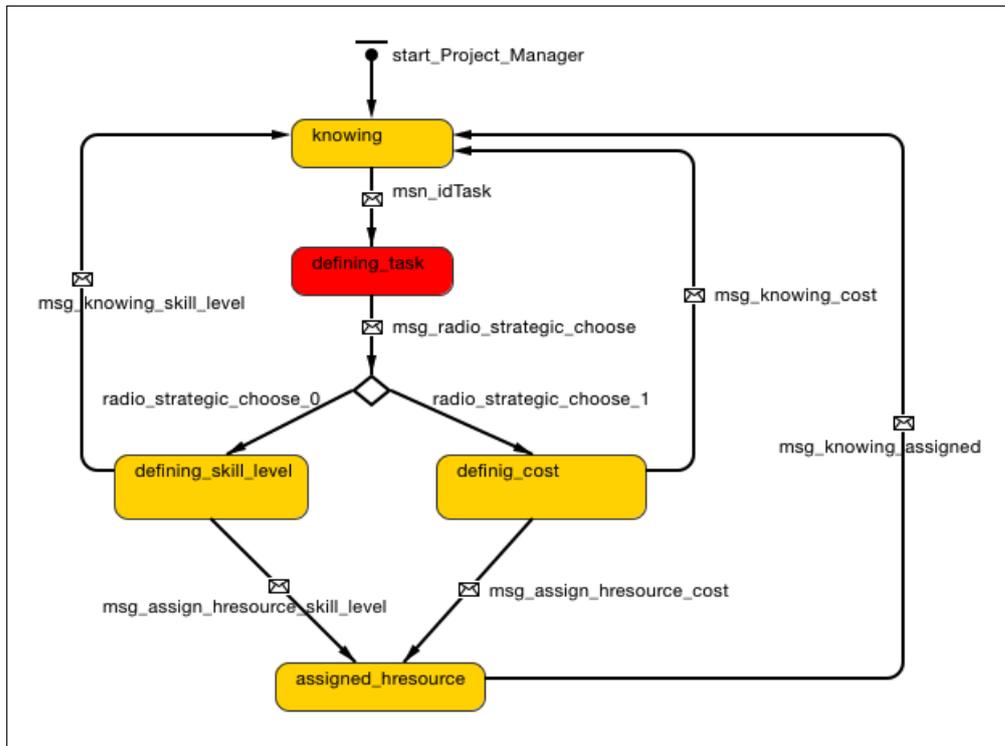


Figura 6.9: Estados do Agente Gerente de Projeto.

Para cada uma dessas estratégias existe um estado respectivamente. A decisão de qual estratégia o agente vai escolher depende de qual estratégia foi configurado na inicialização da simulação, esta escolha é demonstrada na subseção 6.3.4. Ao final o agente envia uma mensagem para o *agente recurso humano* escolhido autorizando a execução da tarefa, isto no estado atribuindo recurso humano.

A vantagem de utilizar um *agente gerente de projeto* é poder executar o mesmo cenário com estratégias diferentes do modelo de conhecimento. Como isto, resultando em simulação diferentes, desta forma, os envolvidos na tomada decisão podem optar por escolher equipes de acordo com execução de estratégia, por exemplo. Além disto, gerente de projeto mais experiente podem alimentar a base de conhecimento com outras estratégias e os gerentes menos experiente podem aplicar na simulação e analisar os resultados.

6.3.4

Analizador

Os agentes apresentados interagem entre si para execução da simulação da gestão do projeto, seguindo a sequencia proposta pelo ProMabs, descrita na seção 3.3. Os agentes são configurados de acordo com suas características

assumindo papéis, após isto, o componente cenários de execução realiza a simulação. Com isto, o componente analisador demonstra a simulação.

Para a visualização da execução dos cenários é criado uma representação visual para os *agentes projeto*, *pacote de trabalho* e *tarefa*, com suas informações, como descrito anteriormente. Além disto, aplicamos cores para representar o estado desses agentes. A cor amarela representa planejado, a vermelhe representa em execução e a verde representa a finalização. As informações são detalhadas por tipos de agente a seguir.

Agente Projeto

O *agente projeto* é representado visualmente como ilustrado na figura 6.10, possuindo informações estáticas e dinâmicas. Informações como nome do projeto e id são estáticas pois não mudam ao longo da execução. Já as informações restantes são dinâmicas e dependem da simulação, como por exemplo, a data final que depende de quando finalizará a ultima tarefa. Já o custo é incrementado de acordo com as finalizações do pacote de trabalho. Desta forma os envolvidos na tomada de decisão podem verificar o custo total do projeto em uma certa data.

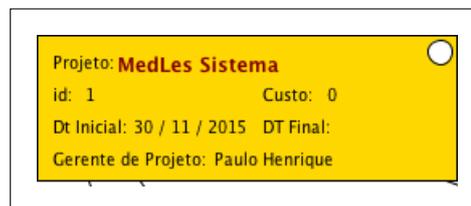


Figura 6.10: Representação Visual do Agente Projeto.

Agente Pacote de Trabalho

O *agente pacote de trabalho* é representado visualmente como ilustrado na figura 6.11, possuindo informações estáticas e dinâmicas. Informações como nome do pacote de trabalho e id são estáticas pois não mudam ao longo da execução. Já as informações restantes são dinâmicas e dependem da simulação, como por exemplo, a data final que depende de quando finalizará a ultima tarefa. Já o custo é incrementado de acordo com as finalizações de cada tarefa. Desta forma os envolvidos na tomada de decisão podem verificar o custo total do pacote de trabalho em uma data específica.

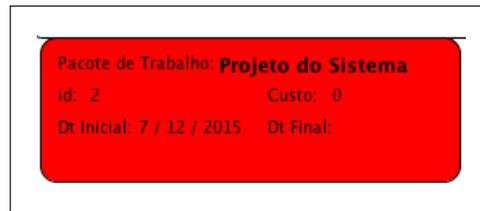


Figura 6.11: Representação Visual do Agente Pacote de Trabalho.

Agente Tarefa

O *agente tarefa* é representado visualmente como ilustrado na figura 6.12, possuindo informações estáticas e dinâmicas. Informações como nome da tarefa e id são estáticas pois não mudam ao longo da execução. Já as informações restantes são dinâmicas e dependem da simulação, como por exemplo, a data final que depende de quando finalizará a tarefa. Já o custo é incrementado de acordo com a unidade de tempo que neste estudo está atribuído para dia, a cada final do horário de trabalho, no caso o dia, é somando o custo atribuído por dia de trabalho do *agente recurso humano* que a está realizando. Desta forma os envolvidos na tomada de decisão podem verificar o custo total da tarefa em uma data específica. Além disto, possui a informação do agente recurso que está realizando a tarefa. Também existe um marcador em formato de um boneco que identifica que o *agente gerente de projeto* atuou nesta tarefa.



Figura 6.12: Representação Visual do Agente Tarefa.

O *agente gerente de projeto* pode atuar de três formas, (i) identificando que um recurso humano diferente do alocado executou a tarefa ou (ii) atribuindo um recurso humano pois não tinha um alocado ou (iii) a tarefa foi executada por um recurso humano que possui um nível de conhecimento diferente da tarefa. Desta forma, os envolvidos na tomada de decisão podem verificar estas atuações do *agente gerente de projeto*, auxiliando a identificar estes acontecimentos.

Cenários em Simulação e Simulados

O componente analisador pode ter dois tipos de cenários, o que está sendo simulado e o cenário já simulado. Os dois apresentam a EAP, porém uma em execução e o outro finalizado, respectivamente. Além disto, o analisador pode instanciar outros elementos de visualização. Neste estudo são aplicados três gráficos que ilustram também o progresso do projeto, assim como a EAP.

O primeiro, é um gráfico de estados o qual identifica quais tarefas estão sendo executadas e finalizadas ao longo dos dias simulados do projeto, como ilustrado na figura 6.13. Para isto, aplicam-se duas cores: a vermelha, a tarefa está em execução, e a verde, a tarefa está finalizada. Este gráfico auxiliar aos envolvidos na tomada de decisão a identificar as tarefas que ainda não foram finalizadas e ainda requerem atenção. Indica-se também quais as tarefas que estão sendo executadas em um momento específico. A medida que o projeto avança as tarefas tendem a ficar verdes e com isto indicar que o projeto está sendo finalizado. O apêndice B possui a figura 6.13 em alta resolução.



Figura 6.13: Gráfico - Estado da Tarefa x Dias.

O segundo, é um gráfico de custo que identifica qual o custo total das tarefas por dia como ilustrado na figura 6.14. Para isto, incrementa-se a cada dia o custo associado do *agente recurso humano* utilizado para executar a tarefa. Este gráfico auxilia os envolvidos na tomada de decisão a identificar o custo total por dia e tarefas, assim, auxiliando na gestão de custo por tempo e tarefas. A medida que o projeto avança as tarefas tendem a ser finalizadas e os custos totais por tarefas serem apresentados. O apêndice B possui a figura 6.14 em alta resolução.

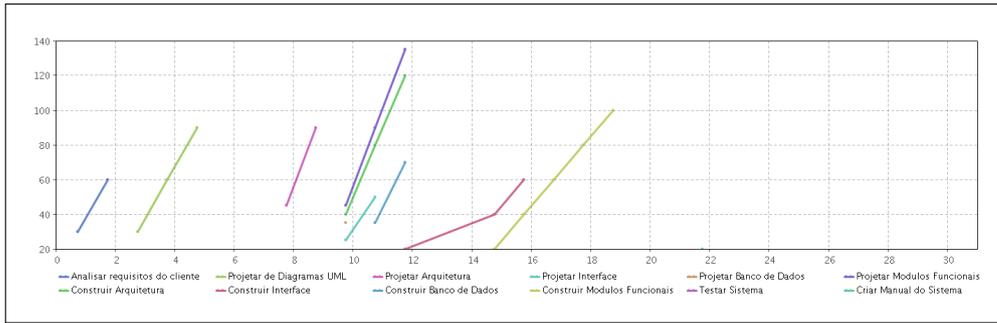


Figura 6.14: Gráfico - Custo da Tarefas x Dias.

Finalmente, o terceiro é um gráfico que identifica os agentes recursos humanos que estão executando tarefas por dia, como ilustrado na figura 6.15. Para isto, incrementa-se um dia de trabalho do *agente recurso humano* associada a tarefa executada. Este gráfico auxilia os envolvidos na tomada de decisão a identificar os recursos humanos e as tarefas que estão sendo executadas, auxiliando na gestão de recurso humano por tempo e tarefas. Também identifica quais os recursos humanos disponíveis e que não executaram nenhuma tarefa à medida que o projeto avança e as tarefas finalizam. O apêndice B possui a figura 6.15 em alta resolução.

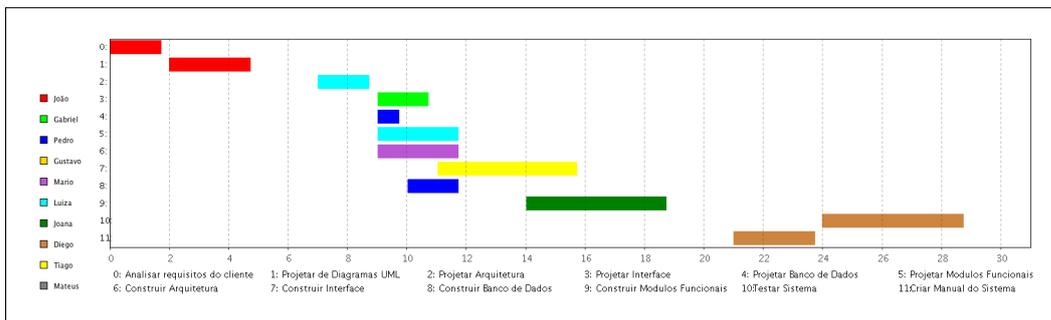


Figura 6.15: Gráfico - Recurso Humano por Tarefas x Dias.

Modificação de Cenário

Para Modificação de Cenário criam-se opções de escolhas de estratégia para o *agente gerente de projeto*. Com as estratégias listadas acima na subseção 6.3.3, os envolvidos na tomada de decisão podem optar por executar um cenário com a estratégia de nível de conhecimento ou por menor custo. Porém, caso necessário, novas opções podem ser implementadas de acordo com implementações de novas estratégias. Como por exemplo, combinar estratégia de menor tempo com uma das duas. Além disto, podendo fazer combinação entre estratégias. Na próxima seção, iremos descrever por meio de um estudo

de caso situações nas quais podemos executar simulações com estratégias diferentes.

6.4

Estudo de Caso

Como descrito anteriormente a instancia do ProMabs permite aos envolvidos na tomada de decisão verificar impactos causado por alterações que acontecem ao longo do projeto, como também, gerar informações para os envolvidos através de execuções de cenários, que ilustra o progresso do projeto. Assim, auxiliando o gerenciamento do projeto. Para isto, adaptamos e utilizamos a estrutura proposta por Runeson e Host (Runeson e Host, 2008) que é aplicada para relatar estudos de casos em engenharia de software. Esta estrutura adaptada, envolve basicamente uma visão geral do estudo de caso, que descreve as características do estudo de caso e como foi sua condução. Além disto, os exemplos ilustrativos que descrevem o que será simulado. E finalmente, as simulações com seus resultados e uma análise qualitativa.

6.4.1

Visão Geral do Estudo de Caso

A instanciação do ProMabs é implementada em uma ferramenta de simulação Anylogic. Os dados utilizados foram criados pelo gerador de Cenários de EAP apresentado no capítulo 3 subseção 3.3.5. Porém, alteramos o nome do projeto e gerente de projeto, assim como, os nomes dos pacotes de trabalhos, das tarefas e dos recursos humano de forma a melhorar a compreensão. Estes dados e o modelo de simulação implementado no Anylogic estão disponíveis em ¹. Após isto, escolhemos dois exemplos ilustrativos realístico para demonstrar como auxiliar a gestão de projetos de software que sofrem alterações, assim, executando cenários para auxiliar os envolvidos na tomada de decisão.

6.4.2

Exemplos Ilustrativos

Os exemplos ilustrativos são utilizados para verificar como os agentes se comportam quando existem alterações no projeto. Para isto, escolhemos um exemplo com alteração envolvendo estratégias e um outro envolvendo alteração

¹<http://www.inf.puc-rio.br/~dbaia/anylogiccpm/cap6>

na performance do recurso humano quando este apresenta atrasos ou avanços. No primeiro, envolvendo a estratégia que inicialmente é focada apenas no menor custo. Porém, modificamos para combinar com a priorização de tempo. Já na segunda, envolvendo alteração de performance, que inicialmente tem o recurso humano executando tarefa de acordo com a sua duração. Porém, modificamos para atribuir um atraso e um avanço de acordo o nível de conhecimento, utilizando uma distribuição uniforme. Após estas modificações, executamos as simulações.

6.4.3

Simulações e Resultados

Para comparar as simulações são demonstrados os resultados da execução sem alteração e depois da alteração, para cada exemplo. Os dois exemplos usam a mesma base de dados e os elementos visuais são os mesmos descritos em seções anteriores. Os dados estão disponível em nosso repositório ², porém também encontra-se no apêndice A.

Combinações de Estratégias

Inicialmente temos a estratégias de custo, como supracitada na seção 6.3.3. Esta estratégia simula um cenário de menor custo, ou seja, procurar um *agente recurso humano* com a habilidade exigida pela tarefa, porém com o menor custo. A tabela 6.6 detalha o resultado da simulação. No resumo do projeto temos a data inicial 30/11/2015 e data final 28/12/2015 com o custo total de 640 como ilustrado na ultima linha do projeto da tabela 6.6.

²<http://www.inf.puc-rio.br/~dbaia/anylogipm/cap6>

Tabela 6.6: Resultado da Simulação com a Estratégia Custo.

Tarefa	R.Humano	Dt Início	Dt Fim	Custo
Analisar requisitos do cliente	Joana	30/11/2015	01/12/2015	40
Projetar Diagramas UML	Diego	02/12/2015	04/12/2015	60
Projetar Arquitetura	Diego	07/12/2015	08/12/2015	40
Projetar Interface	Diego	09/12/2015	10/12/2015	40
Projetar Banco de Dados	Mateus	09/12/2015	09/12/2015	20
Projetar Módulos Funcionais	Diego	09/12/2015	11/12/2015	60
Construir Arquitetura	Joana	09/12/2015	11/12/2015	60
Construir Interface	Tiago	11/12/2015	15/12/2015	60
Construir Banco de Dados	Mateus	10/12/2015	11/12/2015	40
Construir Módulos Funcionais	Joana	14/12/2015	18/12/2015	100
Testar Sistema	Diego	24/12/2015	28/12/2015	60
Criar Manual do Sistema	Diego	21/12/2015	23/12/2015	60
Resumo do Projeto		30/11/2015	28/12/2015	640

Entretanto, após a terceira tarefa ser executada no ambiente externo, os envolvidos na gestão do projetos decidem também priorizar o tempo combinado com o custo. Desta forma ao simular esta combinação de estratégias, o *agente gerente de projeto* é solicitado durante a simulação quando um recurso humano alocado para uma tarefa está ocupado, realizando outra tarefa. Então, o agente de projeto procura um *agente recurso humano* que possua a mesma habilidade e com um menor custo entre os disponíveis e então atribui a tarefa para este *agente recurso humano*. Assim, uma tarefa não fica esperando o *agente recurso humano* alocado inicialmente para ela. Com isto, a execução do projeto simulado continua, porém com modificações no cenário. A tabela 6.7 ilustra o resultado desta modificação.

Tabela 6.7: Resultado da Simulação com a Estratégia Custo e Tempo.

Tarefa	R.Humano	Dt Início	Dt Fim	Custo
Analisar requisitos do cliente	Joana	30/11/2015	01/12/2015	40
Projetar Diagramas UML	Diego	02/12/2015	04/12/2015	60
Projetar Arquitetura	Diego	07/12/2015	08/12/2015	40
Projetar Interface	Diego	09/12/2015	10/12/2015	40
Projetar Banco de Dados	Mateus	09/12/2015	09/12/2015	20
Projetar Módulos Funcionais	Diego	09/12/2015	11/12/2015	60
Construir Arquitetura	Joana	09/12/2015	11/12/2015	60
Construir Interface	Tiago	11/12/2015	15/12/2015	60
Construir Banco de Dados	Mateus	10/12/2015	11/12/2015	40
Construir Módulos Funcionais	Joana	14/12/2015	18/12/2015	100
Testar Sistema	Mario	21/12/2015	23/12/2015	120
Criar Manual do Sistema	Diego	21/12/2015	23/12/2015	60
Resumo do Projeto		30/11/2015	23/12/2015	700

São poucas modificações no cenário, porém o suficiente para diminuir a data final do projeto em cinco dias. Na tarefa *Testar Sistema* antes realizada pelo *Diego*, agora o *Mario* realiza, iniciando no dia 21/12/2015 em vez do dia 24/12/2015, finalizando no dia 23/12/2015 em vez do dia 28/12/2015. Os descansos do *agente recurso humano* são levados em consideração. Estes dias de descanso são sábados e domingos. No resumo do projeto temos então que a data inicial é 30/11/2015 e a data final é 23/12/2015 com o custo total de 700 como ilustrado na ultima linha do projeto da tabela 6.7.

Os envolvidos podem aplicar combinações de estratégias, como tempo e custo ou tempo e nível de conhecimento. Com isto, podem verificar os acontecimentos quando existe uma combinação de interesse. Por exemplo, no caso de combinar tempo e custo, passa a existir um cenário no qual se mudar um dos recursos humanos em uma tarefa o projeto pode ser entregue em uma data anterior à prevista. Antes uma tarefa dependia da outra por causa do recurso humano, porém com a mudança as tarefas podem ser executadas paralelamente.

Atribuindo Atrasos e Avanços para Nível de Conhecimento

Inicialmente temos a estratégias de nível de conhecimento, como supracitada na subseção 6.3.3. Esta estratégia simula um cenário de nível de

conhecimento, ou seja, procura um recurso com a habilidade exigida pela tarefa e que possua o mesmo nível de conhecimento. Porém, se não tiver o mesmo nível de conhecimento procura o mais próximo, primeiro na ordem crescente, caso não encontre, procura em ordem decrescente. A tabela 6.8 detalha o resultado da simulação. No resumo do projeto temos a data inicial 30/11/2015 e data final 28/12/2015 com o custo total de 930, como ilustrado na ultima linha da tabela 6.8.

Tabela 6.8: Resultado da Simulação com a Estratégia Nível de Conhecimento.

Tarefa	R.Humano	Dt Início	Dt Fim	Custo
Analisar requisitos do cliente	João	30/11/2015	01/12/2015	60
Projetar Diagramas UML	João	02/12/2015	04/12/2015	90
Projetar Arquitetura	Luiza	07/12/2015	08/12/2015	90
Projetar Interface	Gabriel	09/12/2015	10/12/2015	50
Projetar Banco de Dados	Pedro	09/12/2015	09/12/2015	35
Projetar Módulos Funcionais	Luiza	09/12/2015	11/12/2015	135
Construir Arquitetura	Mario	09/12/2015	11/12/2015	120
Construir Interface	Tiago	11/12/2015	15/12/2015	60
Construir Banco de Dados	Pedro	10/12/2015	11/12/2015	70
Construir Módulos Funcionais	Joana	14/12/2015	18/12/2015	100
Testar Sistema	Diego	24/12/2015	28/12/2015	60
Criar Manual do Sistema	Diego	21/12/2015	23/12/2015	60
Resumo do Projeto		30/11/2015	28/12/2015	930

Entretanto, quando os envolvidos precisam simular uma performance para um tipo de recurso, atribuímos neste estudo uma unidade de tempo a mais para recursos humanos que possuem um nível de conhecimento júnior. Por outro lado, uma unidade de tempo a menos para recursos humanos que possuem um nível de conhecimento sênior, com uma porcentagem de isto acontecer para ambos. Desta forma, quando o *agente recurso humano* é solicitado para executar uma tarefa, e possui um nível de conhecimento júnior para a habilidade exigida, então, este agente tem 20% (vinte por cento) de chance de executar a tarefa em mais dias que o padrão. Por outro lado, se este *agente recurso humano* tem um nível de conhecimento sênior para a habilidade exigida, então este agente tem 15% (quinze por cento) de chance de executar a tarefa em menos dias que o padrão. Isto acontece quando o *agente recurso humano* entra no ciclo de execução ilustrado no seu diagrama de estado, para cada unidade de tempo ou dia de execução uma distribuição

de probabilidade é usada para determinar se irá atrasar ou avançar de acordo com o nível de conhecimento do *agente recurso humano*. Com isto, a execução do projeto simulado continua, porém com modificações no cenário devido a atrasos e avanços nas execuções das tarefas. A tabela 6.9 ilustra o resultado desta modificações.

Tabela 6.9: Resultado da Simulação com a Estratégia Nível de Conhecimento associada à taxa de performance.

Tarefa	R.Humano	Dt Início	Dt Fim	Custo
Analisar requisitos do cliente	João	30/11/2015	01/12/2015	60
Projetar Diagramas UML	João	02/12/2015	08/12/2015	150
Projetar Arquitetura	Luiza	09/12/2015	10/12/2015	90
Projetar Interface	Gabriel	11/12/2015	14/12/2015	50
Projetar Banco de Dados	Pedro	11/12/2015	11/12/2015	35
Projetar Módulos Funcionais	Luiza	11/12/2015	14/12/2015	90
Construir Arquitetura	Mario	11/12/2015	14/12/2015	80
Construir Interface	Tiago	15/12/2015	17/12/2015	60
Construir Banco de Dados	Pedro	14/12/2015	15/12/2015	70
Construir Módulos Funcionais	Joana	15/12/2015	21/12/2015	100
Testar Sistema	Diego	25/12/2015	29/12/2015	60
Criar Manual do Sistema	Diego	22/12/2015	24/12/2015	60
Resumo do Projeto		30/11/2015	29/12/2015	905

A maioria das datas alteraram, pois houve um atraso em uma das tarefas iniciais e como existem algumas dependências de tarefas, isto reflete em quase todo o projeto. Na tarefa *Projetar Diagrama de UML*, realizada pelo *João* que possui um nível de conhecimento júnior, a tarefa foi atrasada em 2 dias, com isto, finalizando no dia 08/12/2015 em vez do dia 04/12/2015. Os descansos do *agente recurso humano* são levados em consideração. Estes dias de descanso são sábados e domingos. Consequentemente, o custo alterou, era 90 e passou para 150. A tarefa *Projetar Módulos Funcionais*, realizada pela *Luiza* que possui um nível de conhecimento sênior, a entrega avançada em 1 dia, com isto, inicia no dia 11/12/2015 e finaliza no dia 14/12/2015, ressaltando que o trabalho foi realizado no dia 11 e 14 de dezembro de 2015, os dias 12 e 13 são os descansos, sábado e domingo, respectivamente. Consequentemente, o custo alterou. Era 135 e passou a ser 90. A tarefa *Construir Arquitetura* é similar a anteriormente apresentada, pois o seu recurso humano *Mario* possui um nível de conhecimento sênior e avançou a entrega em 1 dia. Com isto refletindo na

entrega antecipada e no custo menor. Ao final, temos a data inicial 30/11/2015 e data final 29/12/2015 com o custo total de 905 como ilustrado na última linha, resumo do projeto da tabela 6.9.

Os envolvidos podem aplicar taxas de performances para o nível de conhecimento júnior e sênior. Com isto, poderão verificar os atrasos e avanços nas tarefas quando um determinado recurso humano está executando a tarefa. Além disto, poderão simular várias vezes e verificar um desvio padrão no tempo do projeto, similarmente no custo. Assim, podendo colocar uma margem na entrega ou no custo do projeto. Como neste caso apresentado anteriormente a taxa foi configurada de forma arbitrária, então, o desvio padrão não foi necessário. Porém, no caso de uma taxa originária de base histórica, o desvio padrão seria aplicado para estipular as margens. Entretanto, este estudo é para exemplificar alterações, e também, para descrever como modelo conceitual ProMabs instaciado, auxilia na geração de informação e visualização de acontecimento por uma simulação de gestão de projetos de software.

6.5

Considerações Finais

Neste capítulo, apresentamos e detalhamos como o ProMabs dá suporte ao gerenciamento de projeto adaptativo de software, ou seja, que sofrem alterações ao longo do progresso do projeto. Para isto, implementamos e instanciamos o ProMabs com uma ferramenta de simulação, chamada Anylogic, e ao final apresentamos exemplos ilustrativos.

Os capítulos 4 e 5, focam na representação para identificar e compreender os elementos pertencentes aos projetos de software e na execuções de cenários para apoiar a tomada de decisão.

Neste capítulo, descreve-se a instanciação e implementação dos componentes do ProMabs, com suas funções, objetivos e como estes apoiam as alterações de Projeto, desde a interação com o ambiente externo até modificações de cenários. Para isto, utilizando uma descrição de linguagem natural dos componentes e do comportamento dos agentes na simulação. Após isto, realizamos um estudo de caso com dois exemplos ilustrativos de alterações no projeto, com o objetivo de descrever como o ProMabs dá suporte a tais alterações. Com estes exemplos, demonstramos como os envolvidos no projeto podem utilizar os resultados das simulações que contém informações para auxiliar na tomada de decisão. Desta forma, são apresentados fatores positivos da aplicação do ProMabs para projetos adaptativos de software.

7

Experimento - Tomada de Decisão

Neste capítulo, detalhamos um estudo de caso realizado com o propósito de avaliar como o ProMabs, instanciado e implementado a uma ferramenta de simulação, auxilia ao envolvidos no projeto em tomar decisões. O estudo é projetado com base na realização de atividades de gestão de projetos de software estando assim fortemente relacionada às atribuições que necessitam de cenários para apoiar em deliberações relevantes no projeto. Para realização do estudo de caso deste capítulo, foram necessárias: (i) A representação dos elementos e seus relacionamentos para simular gestão de projetos de software; (ii) A execução de cenários para auxiliar a gestão de projeto de software em seu processo de tomada de decisão; (iii) O suporte a projeto adaptativos de software. Neste contexto, instanciamos o ProMabs, um modelo conceitual original baseada em multiagentes, que possui componentes necessários para realização deste estudo.

Este capítulo está estruturado da seguinte forma, a seção 7.1 descreve como foi projetado o estudo, em seguida, na seção 7.2 analisamos os dados e apresentamos os resultados, na seção 7.3 descreve-se as possíveis ameaças. E finalmente, a seção 7.4, apresentam as conclusões e discussões.

7.1

Planejamento e Execução do Estudo

Para avaliar como o ProMabs, com os elementos de projetos de software identificados e dando suporte a alterações de contexto que acontecem ao longo do projeto, auxilia na execução de cenários com informações relevantes para a tomada de decisão, projetamos um estudo de caso que é baseado na estrutura proposta por Runeson e Host (Runeson e Host, 2008). Para isto, utilizamos a ferramenta de simulação Anylogic, para implementar o ProMabs como relatado no capítulo 6. Como a simulação precisa de dados de instanciação, utilizamos o gerador descrito no capítulo 3 para gerar estes dados que são o projeto, pacotes de trabalho e tarefas, em uma Estrutura Analítica de Projeto (EAP). Além

disto, o gerente de projeto e recursos humanos. Porém, modificamos os nomes dos elementos para uma exemplificação mais realística. Após isto, elaboramos um questionário com quatro atividades. Aplicamos o questionário a um grupo de alunos e finalmente analisamos os dados.

7.1.1

Dados de Instanciação

Geramos um projeto, contendo quatro pacotes de trabalho e doze tarefas. A figura 7.5 ilustra a EAP deste projeto, na seção 7.2.2. O projeto e suas características estão no quadrado com o topo vermelho. Os pacotes de trabalhos e suas características estão respectivamente em seu quadrado e todos os pacotes de trabalho tem o topo verde. As tarefas e suas características estão respectivamente em seu quadrado e todas as tarefas têm o topo azul. As ligações entre o projeto, pacotes de trabalho e tarefas, são feitos por meio de uma linha continua. As dependências entre tarefa podem ser verificadas por meio do atributo pred. Os restantes das informações sobre a EAP serão descritas a medida que forem utilizadas.

Além da EAP, geramos um gerente de projeto e dez recursos humanos. Os recursos humanos e suas características estão ilustrada na figura 7.4 na seção 7.2.2. Cada recurso humano, possui até duas habilidades com seu respectivo nível de conhecimento e um custo associado. Além disto, temos uma descrição das habilidades e nível de conhecimento, também ilustrada no lado direito da figura 7.4. Todos estes dados, a EAP, gerente de projeto e os recursos humanos, foram criados pelo gerador descrito no capítulo 3 , como mencionado anteriormente. Este dados também estão disponíveis no repositório ¹.

7.1.2

Objetivos Específicos

Especificamente, o objetivo deste estudo, é a avaliação do ProMabs instanciado em três aspectos; (i) se os elementos identificados e representados na simulação possuem informações relevantes para auxiliar na gestão de projeto de software; (ii) se o ProMabs apoia a gestão de projetos adaptativos de software, com um sistema multiagente robusto o suficiente para incorporar alterações, como por exemplo, mudanças de estratégias que acontecem ao longo do projeto; (iii) se os cenários resultantes da simulação auxiliam a tomada de

¹<http://www.inf.puc-rio.br/~dbaia/anylogicpm/>

decisão. Desta forma, por meio deste estudo pretendemos responder à questão principal de pesquisa.

Questão de Pesquisa (QP1) *Como projetar e implementar uma abordagem SBMA para simular gestão de projetos de software?*

- (i) **Representando** *os elementos necessários para simular a gestão de projetos de software.*
- (ii) **Executando cenários** *para auxiliar o gerente de projetos com a tomada de decisão.*
- (iii) **Apoiando** *o gerenciamento de projetos adaptativos de software com o uso de SBMA.*

7.1.3

Questionário e Aplicações do Questionário

O questionário é dividido em duas partes. A primeira parte coleta informações sobre o participante. Para isto, aplicam-se perguntas referentes a formação, conhecimento específicos e finalmente sobre tempo de experiência em projeto de software. Para maiores detalhes sobre estas perguntas, consultar o apêndice A.

A segunda parte do questionário se divide em duas etapas, cada uma com duas atividades de gestão de projetos de software que são fortemente relacionada às atribuições que necessitam de tomada de decisão.

Para aplicação do questionário, dividimos os participantes em dois grupos, A e B. Na primeira etapa do questionário, os participantes do grupo A, realizam as duas atividades utilizando apenas as informações disponibilizadas pelo questionário. E na segunda etapa utilizam o simulador com o ProMabs, para auxiliar a realizar as duas atividades restantes. De forma inversa, na primeira etapa do questionário, os participantes do grupo B, realizam as duas atividades auxiliados com o uso do simulador com o ProMabs. E na segunda etapa utilizam apenas as informações disponibilizadas pelo questionário, para realizar as duas atividades restantes. Com isto, projetamos um *Latin Square*, podendo aplicar métricas para avaliar a relevância do uso do ProMabs por grupos, por etapa e por atividades. Para maiores detalhes sobre o questionário consultar o apêndice A.

Aplicamos os questionários para uma turma em final de período da disciplina de gestão de projeto do curso de bacharelado de sistema de

informação. Esta aplicação aconteceu no mesmo dia e no mesmo horário, porém os grupos se dividiram em duas salas. O grupo A realizava o questionário em uma sala e o Grupo B em outra sala. A figura 7.1 ilustra como ficou a configuração do *Latin Square*.

	A	B
Primeira Etapa	Sem ProMabs	Com ProMabs
Segundo Etapa	Com ProMabs	Sem ProMabs

Figura 7.1: *Latin Square* projetado para o estudo.

7.1.4

Métricas

A fim de avaliar o desempenho dos participantes na execução das atividades de gestão, utilizamos três métricas: tempo de execução das tarefas (medido em minutos), taxa de acerto em relação a cada atividade e nível de dificuldade percebida pelos participantes. Para calcular a métrica de tempo, o participante registra a hora e minuto no início e o fim da execução, em cada atividade. Desta forma, verificamos o tempo total em minutos para uma atividade específica. Além disto, ao final de cada atividade, o participante também registra o nível de dificuldade percebida, em uma escala de 1 a 5. O 1 representa uma menor percepção de dificuldade e 5 representava uma maior percepção de dificuldade.

A fim de avaliar a taxa de corretude das atividades, calculamos o quão ela esta próximo da reposta correta em porcentagem. Assim, não considerando apenas a solução ótima e portanto, adoatnado uma solução que resolva de alguma forma a atividade é considera. Detalharemos os cálculos para a taxa de acerto aplicado em cada atividade, na subseção 7.2.2 em que analisamos os dados coletados e apresentamos os resultado.

7.2

Análise de Dados e Resultados

Como mencionado anteriormente, os questionários contêm duas partes. A primeira parte coleta informações sobre os participantes e a segunda parte

possui as quatro atividades. Para maiores informações sobre os questionários consultar a apêndice A.

Para a análise de dados e apresentação dos resultados, dividimos esta seção em três subseções. Na primeira subseção analisamos o perfil dos participantes. Na segunda subseção analisamos o resultado de cada atividade, com e sem o simulador. Já na terceira subseção analisamos o *Latin Square* projetado, de forma horizontal, por etapas, e de forma vertical, por grupo. Para isto, seguiremos uma abordagem quantitativa e qualitativa.

7.2.1

Perfil dos Participantes

O perfil dos participantes do estudo foi determinado por meio da primeira parte dos questionários, que é comum para ambos. O estudo contou com a participação de 10 alunos do curso de sistemas de informação. Destes 10 alunos, apenas 1 tinha uma primeira formação, que é em processamento de dados. E nenhum, possui curso de pós-graduação. A metade dos participantes cursam o sexto período do curso, o restante se divide em 3 no oitavo período, 1 do nono período e 1 no decimo período. Estes dez alunos foram divididos em dois grupos, A e B.

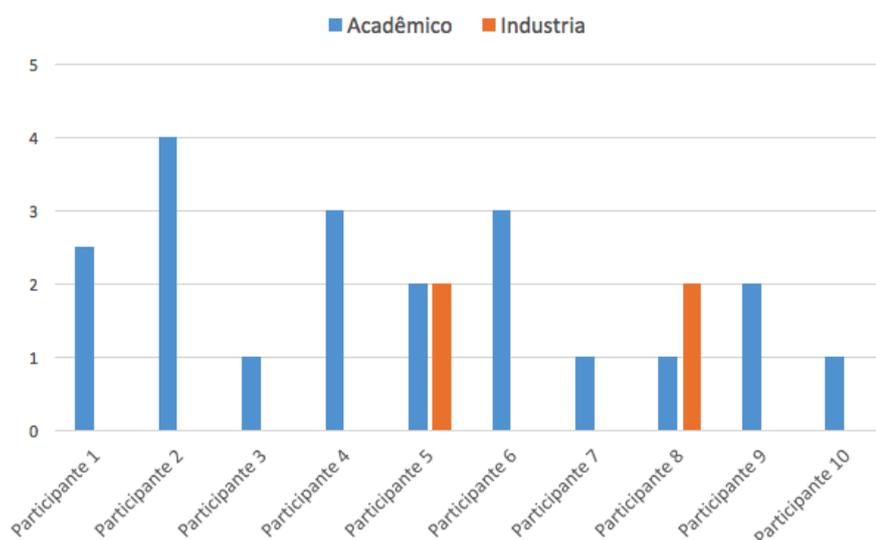


Figura 7.2: Anos de Experiência dos participantes em Projetos de Software.

O principal objetivo da primeira parte, foi determinar a experiência dos participantes em projetos de software e seus conhecimentos em gestão de projetos de software, EAP e sistemas multiagentes. Em relação a experiência em projetos de software, tanto acadêmico como na indústria, todos os

participantes possuem no mínimo 1 ano de experiência. Especificamente, em projetos acadêmicos, 4 participantes possuem 1 ano de experiência, 2 participantes possuem 2 anos, 1 participante possui 2,5 anos, 2 participantes possuem 3 anos e 1 participante possui 4 anos de experiência. Em projeto na indústria apenas 2 informaram 2 anos de experiência. O gráfico da figura 7.2 ilustra esta distribuição em anos de experiências por participantes.

Sobre o conhecimento de sistemas multiagentes a maioria dos participantes informaram nenhum conhecimento, mais especificamente 8, um participante informou um conhecimento de nível baixo, e um outro participante informou o nível bom. Já para o conhecimento sobre EAP, os níveis de conhecimento se concentraram em 4 participantes para pouco, 4 participantes para razoável, e 2 participantes para bom. E finalmente, para o conhecimento sobre gestão de projeto, os níveis de conhecimento se concentraram em 3 participantes para pouco, 3 participantes para razoável, e 4 participantes para bom, de forma mais equilibrada. Provavelmente, isto acontece pelo fato de todos os participantes estarem no final da disciplina de gestão de projeto, mais precisamente na ultima semana do curso.

PUC-Rio - Certificação Digital Nº 1212395/CA

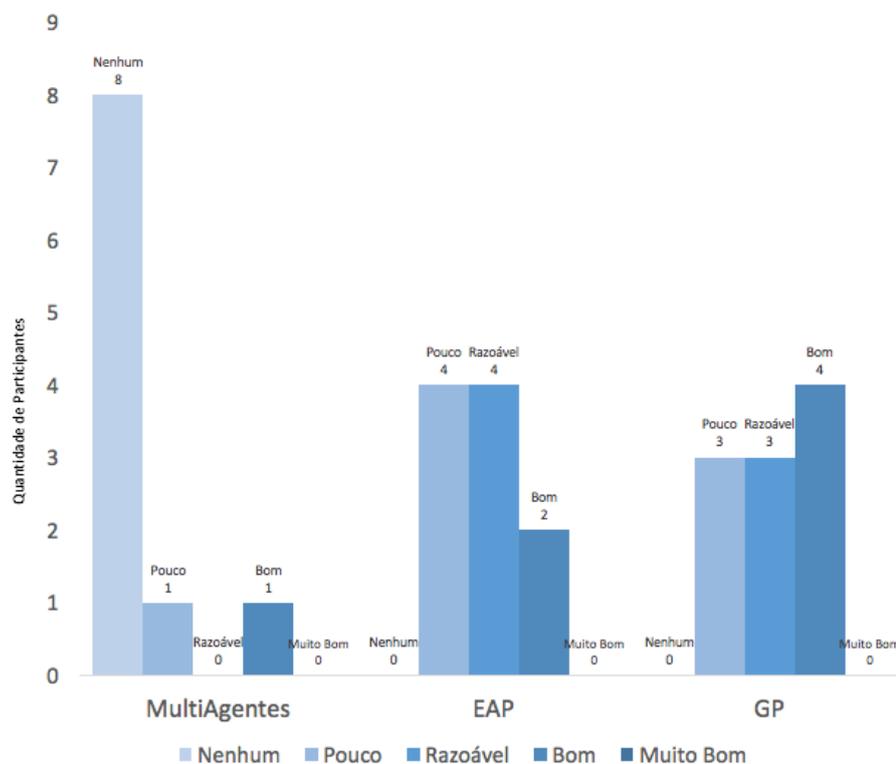


Figura 7.3: Perfil dos participantes do Estudo.

7.2.2

Analisando os Dados e Resultados das Atividades

Como mencionado anteriormente os participantes foram divididos em dois grupos, A e B. Para estes grupos, foram aplicados o mesmo questionário que possui 4 atividades. Porém, quando um grupo fazia uma atividade sem o auxílio do ProMabs, o outro grupo fazia a mesma atividade com o auxílio do ProMabs. Desta forma, proporcionando a comparação dos resultados de cada atividade quando é realizada com e sem o auxílio da simulação. Para isto, é analisado e apresentado os resultados para cada uma das atividades.

As informações comuns para todas as atividades são a EAP, ilustrada na figura 7.5, e uma tabela com as características dos recursos humanos, com habilidades, custos e níveis de conhecimento, ilustradas na figura 7.4. A figura 7.4 contém informações sobre os recursos humanos que poderão ser utilizados para execução das tarefas da EAP. Já a figura 7.5 é a EAP que foi elaborada para o desenvolvimento de um software. Quando existirem atividade para serem realizadas com o auxílio do simulador com o modelo baseado no ProMabs, são utilizados os mesmos dados da EAP e recursos humanos ilustrados nas figuras 7.5 e 7.4.

	Recurso Humano	Nível de		Custo
		Habilidade (Skill)	Conhecimento (Level)	
1	João	1	2	30
		2	1	30
2	Gabriel	3	3	25
3	Pedro	4	3	35
4	Gustavo	5	2	25
5	Mario	6	2	40
		5	3	40
6	Luiza	2	3	45
		5	3	45
7	Joana	1	1	20
		5	1	20
8	Diego	6	1	20
		2	2	20
9	Tiago	3	1	20
10	Mateus	4	1	20

ID	Habilidade (Skill)
1	Analista de Requisitos
2	Engenheiro de Software
3	Analista de Design
4	Analista de Banco de Dados
5	Programador
6	Analista de Teste

ID	Nível de Conhecimento (Level)
1	Júnior
2	Pleno
3	Sênior

Figura 7.4: Recursos Humanos, Habilidade (Skill) e Nível de Conhecimento (Level).

Atividade 1

Na atividade 1, o participante precisa atribuir recursos humanos para as tarefas da EAP da figura 7.5. Porém, este projeto precisa ser executado com um **baixo custo** e com um recurso humano apropriado para a tarefa, ou seja, levando em consideração a habilidade **exigida** pela tarefa e a recomendação do nível de conhecimento. Para isso, solicita a utilização da estratégia a seguir.

Escolher um recurso humano que possua a mesma habilidade exigida da tarefa, além disso procurar o recurso humano que tem o menor custo.

Com isto, o participante descreve a sua escolha através de uma tabela, preenchendo o recurso humano para cada tarefa e ao final somam o custo total. A tabela pode ser visualizada com detalhes no apêndice A.

Ao utilizar apenas as informações das figuras 7.4 e 7.5, o participante precisa verificar quais os recursos humanos possuem a mesma habilidade exigida pela tarefa. Após isto, atribuir o recurso humano com o menor custo, para calcular o custo da tarefa, foi necessário multiplicar o custo do recurso humano pela quantidade de dias da tarefa. Ao final, o participante soma todos os custos das tarefas, assim, calculando o custo total do projeto. Por outro lado, ao utilizar também a simulação, o participante precisa verificar o cenário simulado para cada tarefa. Este cenário foi executado com a estratégia custo configurado. Com isto, analisa-se o recurso humano alocado para cada tarefa possui o menor custo. Após isto, atribui-se o recurso humano para a tarefa com seu custo envolvido, e ao final verifica-se e atribui-se o custo total.

Para calcular a métrica tempo é utilizada a hora de início e fim registrado por cada participante. Desta forma, verificamos o tempo total em minutos por participante na atividade. Além disto, ao final da atividade o participante também registra o nível de dificuldade da atividade. A métrica taxa de acerto é representada em porcentagem. O cálculo da porcentagem é baseado nos acertos das atribuições dos recursos humanos com o menor custo para as 12 tarefas existentes nesta atividade. O acerto ocorre quando um recurso humano alocado possui a habilidade exigida para executar a tarefa, e possui o menor custo dentro dos recursos humanos com a mesma habilidade. Uma atribuição errônea, ocorre quando um recurso humano é alocado para uma tarefa a qual ele não possui habilidade para executar. Além disto, também consideramos uma atribuição errônea, o recurso humano alocado que possui a habilidade exigida para execução da tarefa, porém existe um recurso humano de menor valor com mesma habilidade.

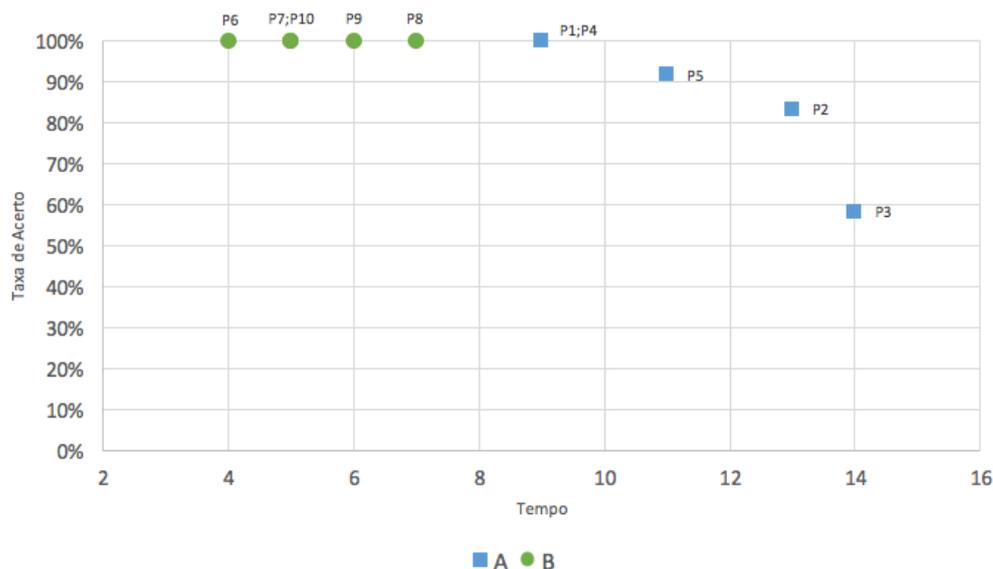


Figura 7.6: Taxa de acerto por tempo da Atividade 1.

A figura 7.6 ilustra o gráfico da atividade 1 que relaciona a taxa de acerto com o tempo. O grupo A se refere aos participantes que realizaram a atividade 1 sem o simulador, representados no gráfico por meio dos quadrados azuis. Já o grupo B se refere aos participantes que realizaram a atividade 1 com o simulador, representados no gráfico por meio das bolinhas verdes. Os participantes do grupo A apresentam o tempo entre 9 a 14 minutos, com uma média de 11,2 minutos. Com isto, podemos observar que esses participantes se concentram no lado direito do gráfico, ou seja, apresentam maior tempo para executar a tarefa. Além disso, podemos notar que apenas dois participantes tiveram taxas de acerto máxima (100%), enquanto o restante dos participantes apresentou taxas de acertos de 58%, 83% e 92%, com uma média de 87%. Por outro lado, os participantes do grupo B apresentam o tempo entre 4 a 7 minutos, com uma média de 5,4 minutos. Com isto, podemos observar que esses participantes se concentram no lado esquerdo do gráfico, ou seja, apresentam menor tempo para executar a tarefa. Além disso, todos os participantes tiveram 100% de taxa de acerto, possuindo assim, a maior taxa de acerto média possível. Então, concluímos que para a atividade 1, o Grupo B, que utilizou o simulador, obteve melhor desempenho se comparado ao Grupo A, que realizou a atividade sem o simulador.

Atividade 2

Na atividade 2, o participante precisa identificar quais as tarefas possuem inconformidade com o recurso humano alocado, ou seja, habilidade diferente

da **exigida** pela tarefa ou nível de conhecimento diferente da recomendação da tarefa. Esta atividade é baseada nas alocações realizadas na atividade 1. Para isto, o participante utilizava uma tabela para assinalar em quais tarefas existiam inconformidade. A tabela pode ser visualizada com detalhes no apêndice A.

Ao utilizar apenas as informações da figuras 7.4 e 7.5, o participante precisa analisar em cada tarefa o recurso humano alocado. Com isto, verificar a inconformidade entre a habilidade exigida pela tarefa e a habilidade do recurso humano alocado. Também, verificar a inconformidade entre o nível de conhecimento recomendado pela tarefa e o nível de conhecimento do recurso humano alocado. Por outro lado, ao utilizar também a simulação, o participante precisa verificar o marcador (em formato de boneco) aplicado pelo simulador, nas tarefas que apresentam inconformidades com o recurso humano alocado. Além disto, analisar e relatar a inconformidade.

Para calcular a métrica tempo é utilizada a hora de início e fim registrado por cada participante. Desta forma, verificamos o tempo total em minutos por participante na atividade. Além disto, ao final da atividade o participante também registra o nível de dificuldade da atividade. A métrica taxa de acerto é representada em porcentagem. O cálculo da porcentagem é baseado nos acertos da identificação das inconformidades dentre as 12 tarefas existentes nesta atividade. O acerto ocorre quando uma tarefa que possui inconformidade está assinalada. Além disto, também consideramos acerto uma tarefa sem inconformidade, e com isto não assinalada. Uma identificação errônea ocorre quando uma tarefa é assinalada com inconformidade, porém o recurso humano alocado possui habilidade exigida pela tarefa e o nível de conhecimento recomendado pela tarefa.

A figura 7.7 ilustra o gráfico da atividade 2 que relaciona a taxa de acerto com o tempo. O grupo A se refere aos participantes que realizaram a atividade 2 sem o simulador, representados no gráfico por meio dos quadrados azuis. Já o grupo B se refere aos participantes que realizaram a atividade 2 com o simulador, representados no gráfico por meio das bolas verdes. Os participantes do grupo A, apresentam o tempo entre 6 a 9 minutos, com uma média de 7,6 minutos. Com isto, podemos observar que esses participantes se concentram no lado direito do gráfico, ou seja, apresentam maior tempo para executar a tarefa. Além disso, podemos notar que apenas dois participantes tiveram taxas de acerto máxima (100%), enquanto o restante dos participantes apresentou taxas de acertos de 42%, 83% e 92%, com uma média de 83%. Por outro lado, os participantes do grupo B, apresentam o tempo entre 2 a 4 minutos, com

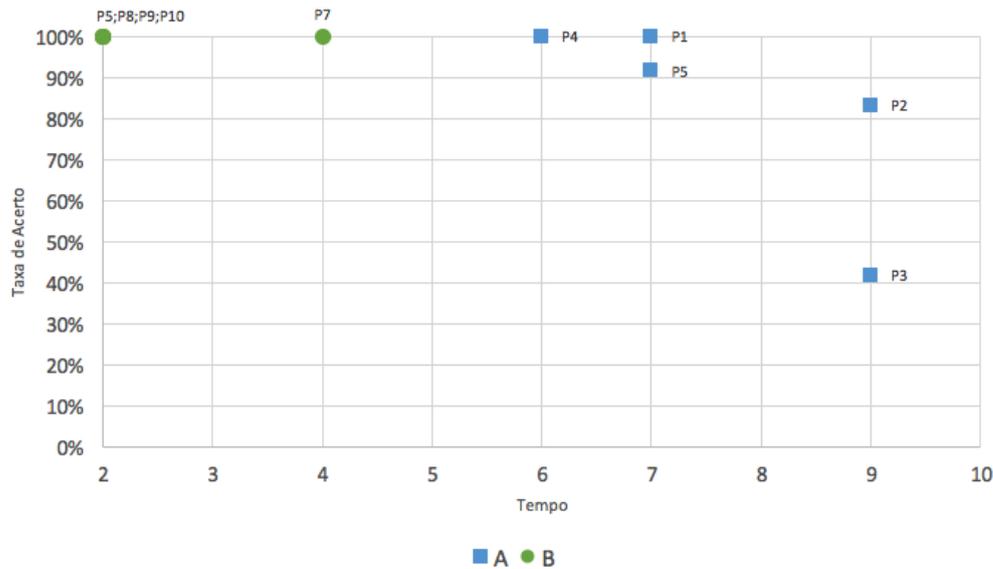


Figura 7.7: Taxa de acerto por tempo da Atividade 2.

uma média de 2,4 minutos. Com isto, podemos observar que esses participantes se concentram no lado esquerdo do gráfico, ou seja, apresentam menor tempo para executar a tarefa. Além disto, todos os participantes tiveram 100% de taxa de acerto, possuindo assim, a maior taxa de acerto média possível. Então, concluímos que para a atividade 2, o Grupo B, que utilizou o simulador, obteve melhor desempenho se comparado ao Grupo A, que realizou a atividade sem o simulador.

Atividade 3

Na atividade 3, o participante precisa atribuir recursos humanos para as tarefas da EAP da figura 7.5. Porém, este projeto precisa ser executado com um recurso humano apropriado para a tarefa, ou seja, levando em consideração a habilidade **exigida** pela tarefa e a recomendação do nível de conhecimento. Porém, este projeto precisa ser executado com uma equipe que possua um maior nível de conhecimento possível. Para isso, solicita a utilização da estratégia a seguir.

Escolher um recurso humano que possua a mesma habilidade exigida da tarefa. Além disto, procurar um nível de conhecimento igual ao sugerido da tarefa. Porém, se não encontrar um nível de conhecimento igual da tarefa, procure algum nível de conhecimento maior, seguindo uma ordem (júnior, pleno, sênior). Caso não encontre, procure um nível de conhecimento menor seguindo uma ordem (sênior, pleno, júnior).

Com isto, o participante descreve a sua escolha através de uma tabela, preenchendo o recurso humano para cada tarefa e ao final soma o custo total. A tabela pode ser visualizada com detalhes no apêndice A.

Ao utilizar apenas as informações da figuras 7.4 e 7.5, o participante precisa verificar quais os recursos humanos possuem a mesma habilidade exigida pela tarefa. Após isto, verifica quais destes recursos humanos possuem o mesmo nível de conhecimento, caso exista, escolhe um. Se não existe, procura-se um recurso humano com um nível de conhecimento maior, seguindo a ordem de júnior, pleno e sênior. Caso não encontre, procurava um recurso humano com o menor nível, seguindo a ordem sênior, pleno e júnior. Após isto, atribui-se o recurso humano de acordo com estes passos. Para calcular o custo da tarefa, era necessário multiplicar o custo do recurso humano pela quantidade de dias da tarefa. Ao final, o participante soma todos os custos das tarefas, assim, calculando o custo total do projeto. Por outro lado, ao utilizar também a simulação, o participante precisa verificar o cenário simulado para cada tarefa. Este cenário foi executado com a estratégia nível de conhecimento configurada. Com isto, analisa-se o recurso humano alocado para tarefa possui o nível de conhecimento mais próximo e maior recomendado pela tarefa. Após isto, atribui o recurso humano para a tarefa com seu custo envolvido, e ao final verifica e atribui o custo total.

Para calcular a métrica tempo é utilizada a hora de início e fim registrado por cada participante. Desta forma, verificamos o tempo total em minutos por participante na atividade. Além disto, ao final da atividade o participante também registra o nível de dificuldade da atividade. A métrica taxa de acerto é representada em porcentagem. O cálculo da porcentagem é baseado nos acertos das atribuições dos recursos humanos com o nível de conhecimento apropriado para as 12 tarefas existentes nesta atividade. O acerto ocorre quando um recurso humano alocado possui a habilidade exigida para executar a tarefa, e possui o nível de conhecimento mais próximo e maior dentro dos recursos humanos com a mesma habilidade. Uma atribuição errônea ocorre quando um recurso humano é alocado para uma tarefa na qual ele não possui habilidade para executar. Além disto, também consideramos uma atribuição errônea o recurso humano alocado que possui a habilidade exigida para execução da tarefa, porém existe um recurso humano com nível de conhecimento mais próximo e maior recomendando pela tarefa e com a mesma habilidade.

A figura 7.8 ilustra o gráfico da atividade 3 que relaciona a taxa de acerto com o tempo. O grupo A se refere aos participantes que realizaram a atividade 3 com o simulador, representados no gráfico por meio dos quadrados

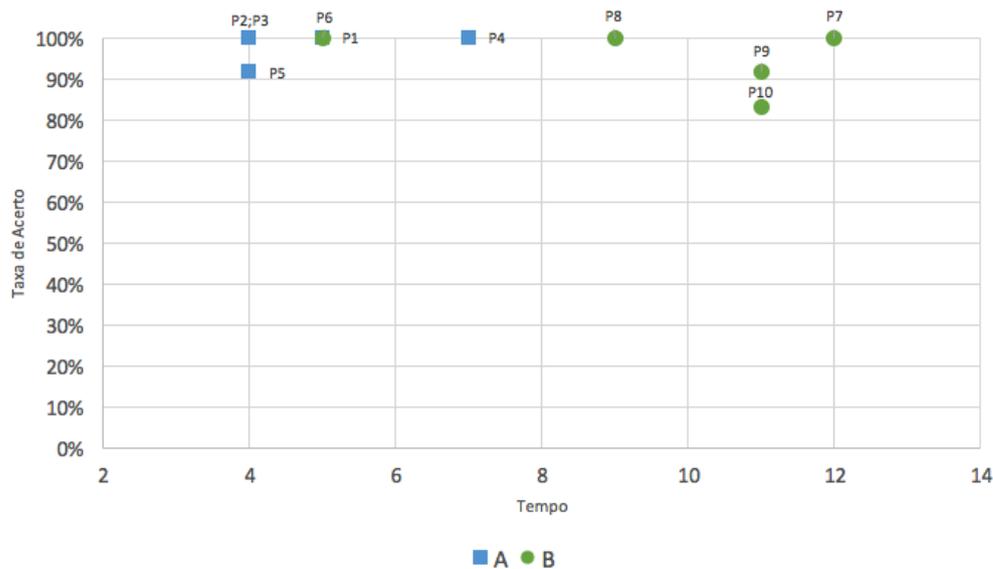


Figura 7.8: Taxa de acerto por tempo da Atividade 3.

azuis. Já o grupo B se refere aos participantes que realizaram a atividade 3 sem o simulador, representados no gráfico por meio das bolas verdes. Os participantes do grupo A, apresentam o tempo entre 4 a 7 minutos, com uma média de 4,8 minutos. Com isto, podemos observar que 4 dos participantes se concentram no lado esquerdo do gráfico, ou seja, apresentam menor tempo para executar a tarefa. Porém, um dos participantes apresenta o tempo de 7 minutos. Além disso, podemos notar que 4 dos participantes tiveram taxas de acerto máxima (100%), enquanto apenas um participante apresentou taxa de acertos de 92%, com uma média de 98%. Por outro lado, os participantes do grupo B, apresentam o tempo entre 5 a 12 minutos, com uma média de 9,6 minutos. Com isto, podemos observar que 4 dos participantes se concentram no lado esquerdo do gráfico, ou seja, apresentam maior tempo para executar a tarefa. Porém, um dos participantes apresenta o tempo de 5 minutos. Além disso, podemos notar que 3 dos participantes tiveram taxas de acerto máxima (100%), enquanto os 2 restantes dos participantes apresentam taxas de acertos de 92%, 83%, com uma média de 95%. Então, concluímos que para a atividade 3, o Grupo A, que utilizou o simulador, obteve melhor desempenho se comparado ao Grupo B, que realizou a atividade sem o simulador. Porém na média de taxa de acerto a diferença é de apenas 3%, e o grupo A executa a tarefa na metade do tempo do Grupo B, isto na média da métrica tempo.

Atividade 4

Na atividade 4, o participante precisa realizar alterações na EAP da Figura 7.9 para diminuir o tempo do projeto. Inicialmente esta EAP prioriza o custo do projeto, porém foram solicitadas mudanças para entregar o projeto o mais breve possível. Desta forma, é necessário que o participante verifique quais alterações de recursos humanos podem ser realizadas na EAP para atender esta solicitação. Entretanto, respeitando a exigência para realizar as tarefas. O recurso humano precisa possuir a habilidade igual à **exigida** pela tarefa. Além disto, existe uma recomendação do nível de conhecimento da tarefa. Outro ponto, é que um recurso não pode realizar mais de uma tarefa ao mesmo tempo. Por fim, sábado e domingo são dias de descanso, ou seja, ninguém realiza atividade. Como suporte para realizar a atividade, o anunciado incorporava uma sugestão. O participante precisa identificar as tarefas que podem ser realizadas paralelamente. Além disto, identificar as tarefas que inicialmente estão sendo realizadas em sequencia, devido a restrição do recurso humano realizar apenas uma tarefa por vez e por motivos de custo.

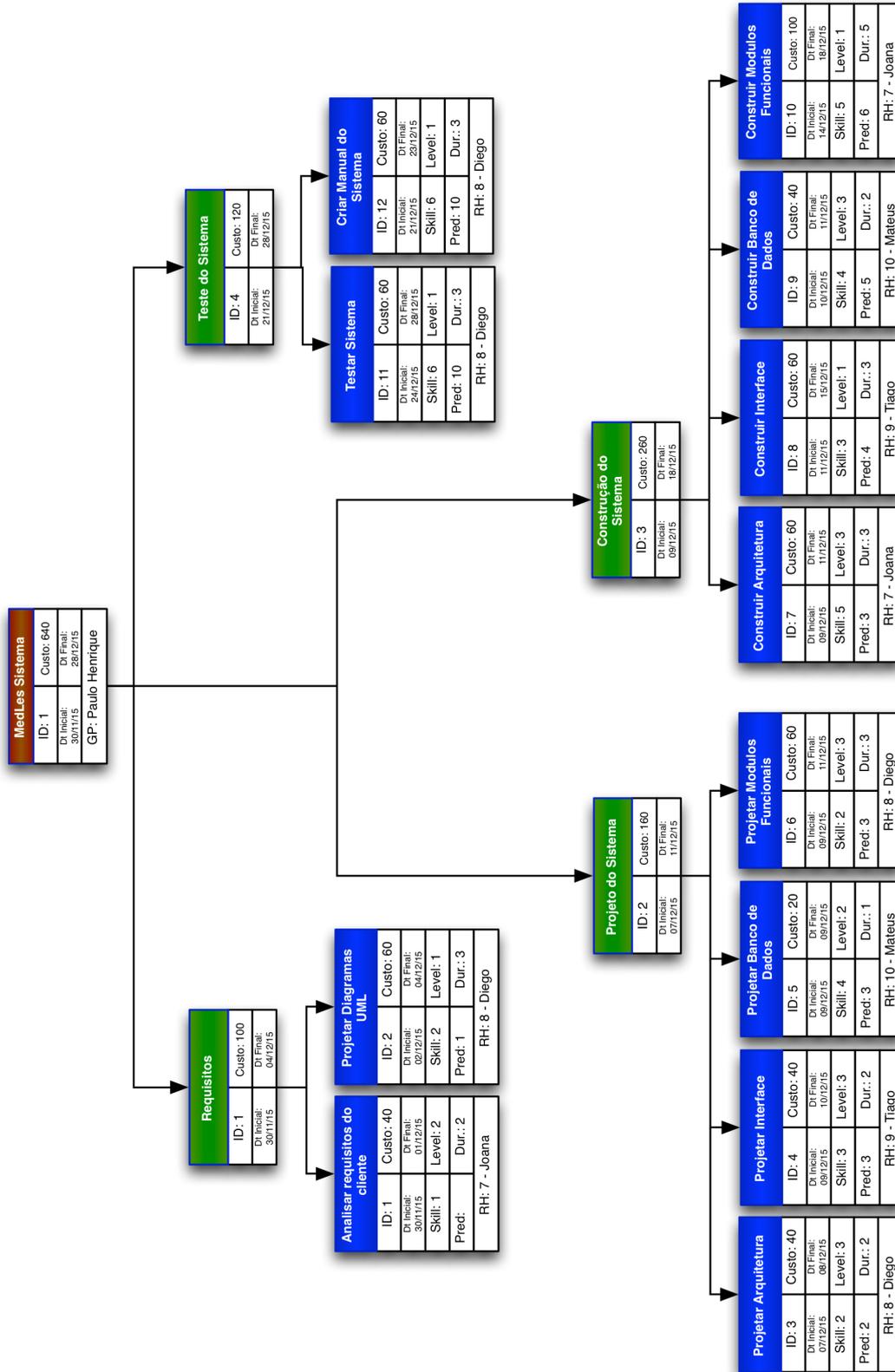


Figura 7.9: EAP para a Atividade 4.

Neste contexto, o participante descreve suas alterações através de uma

tabela, preenchendo alterações em tarefas para atender a solicitação de reduzir a data final do projeto. Como o nome da tarefa, o recurso humano, data inicial e final da respectiva tarefa. Ao final preenchia a data inicial e final do projeto. A tabela pode ser visualizada com detalhes no apêndice A.

Ao utilizar apenas as informações da figuras 7.9 e 7.4, o participante precisa verificar quais as tarefas que estão sendo executadas em sequencia e com o mesmo recurso humano. Além disto, deve verificar se estas tarefas não são dependentes. Após isto, verificar se existe um recurso humano apropriado para realizar uma das tarefas, ou seja, um recurso humano que possui a mesma habilidade exigida pela tarefa. Desta forma, encontrando um recurso humano e atribuindo a tarefa, com isto, analisar o impacto no projeto. Esta mudança de recurso humano pode ser em uma tarefa específica, porém, sendo refletida em outras tarefas. Neste caso, reduzindo a data final do projeto, caso as tarefas possam ser executadas de forma paralela, com recursos diferentes. Finalmente, descreve a mudança e o impacto nas tarefas, assim como, as datas inicial e final do projeto. Por outro lado, ao utilizar também a simulação, o participante precisa verificar o cenário simulado para cada tarefa. Este cenário era executado com a combinação das estratégias de custo e tempo configurada. Com isto, analisa-se a execução da simulação para encontrar em qual momento tarefas que foram executadas inicialmente em sequencia, são executadas em paralelo quando a estratégia de tempo também é configurada. Além disto, existe um gráfico que ilustra informações sobre qual tarefa está sendo executada e qual o recurso humano que executa a tarefa. Assim, auxiliando a identificar tarefas que podem alterar o recurso humano. Desta forma, impactando na data final de entrega. Finalmente, descreve na tabela a mudança e o impacto nas tarefas, assim como, as datas inicial e final do projeto.

Para calcular a métrica tempo é utilizada a hora de inicio e fim registrado por cada participante. Desta forma, verificamos o tempo total em minutos por participante na atividade. Além disto, ao final da atividade o participante também registrava o nível de dificuldade da atividade. A métrica taxa de acerto é representada em porcentagem. O cálculo da porcentagem é baseado nos acertos das alterações nas tarefas apropriadas para redução de tempo, dentre as 12 tarefas existentes nesta atividade. O acerto ocorre quando uma tarefa possui alteração devido as mudanças realizadas, para atender a solicitação de diminuir a data de entrega do projeto. Porém, o recurso humano não pode estar realizando outra tarefa ao mesmo tempo. Além disto, é necessário obediência aos dias de descanso dos recursos humanos, assim como, as dependências das tarefas. Também, a compatibilidade da habilidade do recurso humano, com a

habilidade exigida para executar a tarefa. Além disto, também consideramos acerto uma tarefa sem alterações, porém não impactando na solicitação de diminuir a data de entrega do projeto. Uma alteração errônea ocorre quando uma das exigências da atividade não são respeitadas. Uma exigência é que um recurso humano não pode executar tarefa em seu dia de descanso. Uma outra exigência é que o recurso humano precisa possuir a habilidade exigida para executar a tarefa. E também é necessária uma conformidade com as dependências das tarefas.

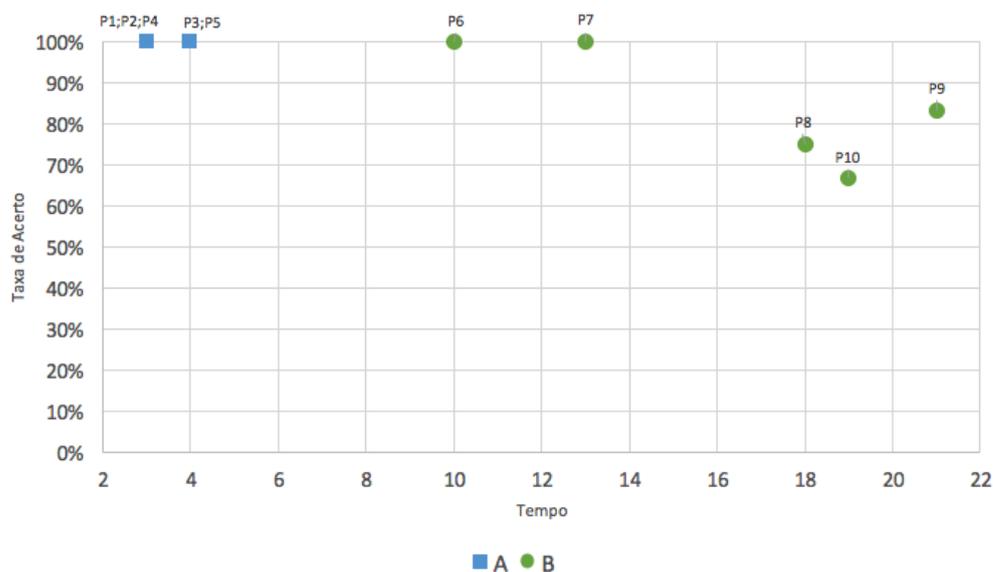


Figura 7.10: Taxa de acerto por tempo da Atividade 4.

A figura 7.10 ilustra o gráfico da atividade 4 que relaciona a taxa de acerto com o tempo. O grupo A se refere aos participantes que realizaram a atividade 4 com o simulador, representados no gráfico por meio dos quadrados azuis. Já o grupo B se refere aos participantes que realizaram a atividade 4 sem o simulador, representados no gráfico por meio das bolinhas verdes. Os participantes do grupo A, apresentam o tempo entre 3 a 4 minutos, com uma média de 3,4 minutos. Com isto, podemos observar que esses participantes se concentram no lado esquerdo do gráfico, ou seja, apresentam menor tempo para executar a tarefa. Além disto, todos os participantes tiveram 100% de taxa de acerto, possuindo assim, a maior taxa de acerto média possível. Por outro lado, os participantes do grupo B, apresentam o tempo entre 10 a 21 minutos, com uma média de 16,2 minutos. Com isto, podemos observar que esses participantes se concentram no lado direito do gráfico, ou seja, apresentam maior tempo para executar a tarefa. Além disso, podemos notar que apenas dois participantes tiveram taxas de acerto máxima (100%), enquanto o restante dos participantes apresentou taxas de acertos de 67%, 75% e 83%, com uma média de 85%.

Neste contexto, concluímos que para a atividade 4, o Grupo A, que utilizou o simulador, obteve melhor desempenho se comparado ao Grupo A, que realizou a atividade sem o simulador. Acreditamos, que isto ocorre devido o fato do participante precisar analisar vários elementos de forma manual, quando não usa o simulado. Consequentemente, o tempo para analisar aumenta e também de realizar uma alteração errônea. Pois a cada alteração em uma tarefa, o participante precisa propagar para as demais, assim, verificando os impactos de cada alteração. Podemos observar o ganho significativo, por meio da média de tempo, do Grupo A em relação ao Grupo B.

7.2.3

Analisando os Dados e Resultados por Etapas e Grupos

Como mencionado anteriormente os participantes foram divididos em dois grupos, A e B. Para estes grupos, foram aplicados o mesmo questionário que possui duas etapas, cada etapa possui duas atividades. Porém, quando um grupo fazia uma etapa sem o auxílio do ProMabs, o outro grupo fazia a mesma etapa com o auxílio do ProMabs. Desta forma, proporcionando a comparação dos resultados da mesma etapa quando é realizada com e sem o auxílio da simulação, ou seja, uma comparação horizontal dos grupos, A e B, na primeira etapa e outra na segunda etapa. Assim, como comparação de etapas diferentes para cada um dos grupos, ou seja, uma comparação vertical das etapas, primeira e segunda, no Grupo A e outra no Grupo B. Desta forma, utilizando o *Latin Square* apresentado anteriormente para mitigar viés da utilização do simulador tanto para o grupo quanto para etapas.

Neste contexto, para auxiliar na avaliação estatística utilizamos a linguagem de programação R ². Com isto, calculamos inicialmente as médias por métrica, ou seja, para métrica de taxa de acerto, tempo e dificuldade, ilustrada na tabela 7.1. Após isto, verificamos qual o teste de hipótese aplicar, com este, calculamos o p-value para a hipótese. Definimos um valor de intervalo de confiança de 95% como geralmente é utilizado.

²<https://www.r-project.org>

Tabela 7.1: Médias das Métricas.

	A			B		
	T. de Acerto	Tempo	N. Dificuldade	T. de Acerto	Tempo	N. Dificuldade
Primeira Etapa	85%	9,4	2,8	100%	3,9	1,4
Segunda Etapa	99%	4,1	1,4	90%	12,9	3,1

Para adotar um teste estatístico, primeiro verificamos por meio de um teste de Shapiro-Wilk se alguma das amostras segue uma distribuição normal. Com o resultado, não conseguimos afirmar que nenhuma das amostras seguem uma distribuição normal. Além disto, as duas amostras de dados são de grupos independentes com relação ao atribuído variável. Para a análise horizontal, o atributo variável é o grupo. Já para a análise vertical, o atributo variável é a etapa. Com isto, mitigando a interferências entre conjuntos de amostras sendo comparados. Neste contexto, aplicamos o teste estatístico de Mann-Whitney-Wilcoxon, para verificar a hipótese nula de que duas amostras são idênticas para cada métrica. Geralmente o teste estatístico de Mann-Whitney-Wilcoxon é aplicado quando as amostras são diferentes e quando não assumimos que as distribuições das amostras seguem uma distribuição normal.

Análise Horizontal – Primeira Etapa

A primeira análise horizontal é realizada entre os grupos A e B, na primeira etapa. O resultado do teste estatístico de Mann-Whitney-Wilcoxon para taxa de acerto, retornou o *p-value* de 0,005. Com isto, refutando que A e B são iguais em relação a taxa de acerto. Neste contexto, comparamos as médias das taxas de acertos do grupo A e B, apresentada na tabela 7.1. Desta forma, obtemos indícios para duas suposições; (i) o grupo A possui uma taxa de acerto menor que o grupo B; (ii) o uso do simulador auxilia os participantes a alcançar uma maior taxa de acerto.

Para o tempo de execução, o resultado do teste estatístico de Mann-Whitney-Wilcoxon, retornou o *p-value* de 0,0003. Com isto, refutando que A e B são iguais em relação a tempo. Neste contexto, comparamos as

médias de tempo do grupo A e B, apresentada na tabela 7.1. Desta forma, obtemos indícios para duas suposições; (i) o grupo A executa em um tempo maior que o grupo B; (ii) o uso do simulador auxilia os participantes a alcançar um menor tempo.

Já para o nível de dificuldade, o resultado do teste estatístico de Mann-Whitney-Wilcoxon, retornou o *p-value* de 0,005. Com isto, refutando que A e B são iguais em relação a nível de dificuldade. Neste contexto, comparamos as médias de nível de dificuldade percebida do grupo A e B, apresentada na tabela 7.1. Desta forma, obtemos indícios para duas suposições; (i) o grupo A percebe um nível de dificuldade maior que o grupo B; (ii) o uso do simulador auxilia a diminuir o nível de dificuldade percebida pelo candidato.

Análise Horizontal – Segunda Etapa

A segunda análise horizontal é realizada entre os grupos A e B, na segunda etapa. O resultado do teste estatístico de Mann-Whitney-Wilcoxon para taxa de acerto, retornou o *p-value* de 0.04485. Com isto, refutando que A e B são iguais em relação a taxa de acerto. Neste contexto, comparamos as médias das taxas de acertos do grupo A e B, apresentada na tabela 7.1. Desta forma, obtemos indícios para duas suposições; (i) o grupo A possui uma taxa de acerto maior que o grupo B; (ii) o uso do simulador auxilia os participantes a alcançar uma maior taxa de acerto.

Para o tempo de execução, o resultado do teste estatístico de Mann-Whitney-Wilcoxon, retornou o *p-value* de 0,0002. Com isto, refutando que A e B são iguais em relação a tempo. Neste contexto, comparamos as médias de tempo do grupo A e B, apresentada na tabela 7.1. Desta forma, obtemos indícios para duas suposições; (i) o grupo A executa em um tempo menor que o grupo B; (ii) o uso do simulador auxilia os participantes a alcançar um menor tempo.

Já para o nível de dificuldade, o resultado do teste estatístico de Mann-Whitney-Wilcoxon, retornou o *p-value* de 0,0003. Com isto, refutando que A e B são iguais em relação a nível de dificuldade. Neste contexto, comparamos as médias de nível de dificuldade do grupo A e B, apresentada na tabela 7.1. Desta forma, obtemos indícios para duas suposições; (i) o grupo A percebe um nível de dificuldade menor que o grupo B; (ii) o uso do simulador auxilia a diminuir o nível de dificuldade percebida pelo candidato.

Análise Vertical – Grupo A

A primeira análise vertical é realizada entre a primeira e segunda etapas, no Grupo A. O resultado do teste estatístico de Mann-Whitney-Wilcoxon para taxa de acerto, retornou o *p-value* de 0,018. Com isto, refutando que a primeira e segunda etapas são iguais em relação a taxa de acerto. Neste contexto, comparamos as médias das taxas de acertos da primeira e segunda etapas, apresentada na tabela 7.1. Desta forma, obtemos indícios para duas suposições; (i) os participantes possuem uma taxa de acerto menor na primeira etapa do que na segunda etapa; (ii) o uso do simulador auxilia os participantes a alcançar uma maior taxa de acerto.

Para o tempo de execução, o resultado do teste estatístico de Mann-Whitney-Wilcoxon, retornou o *p-value* de 0,0002. Com isto, refutando que a primeira e segunda etapas são iguais em relação a tempo. Neste contexto, comparamos as médias de tempo da primeira e segunda etapas, apresentada na tabela 7.1. Desta forma, obtemos indícios para duas suposições; (i) os participantes executam em um tempo maior a primeira etapa do que a segunda etapa; (ii) o uso do simulador auxilia os participantes a alcançar um menor tempo.

Já para o nível de dificuldade, o resultado do teste estatístico de Mann-Whitney-Wilcoxon, retornou o *p-value* de 0,00579. Com isto, refutando que a primeira e segunda etapas são iguais em relação a nível de dificuldade. Neste contexto, comparamos as médias de nível de dificuldade da primeira e segunda etapas, apresentada na tabela 7.1. Desta forma, obtemos indícios para duas suposições; (i) os participantes percebem um nível de dificuldade maior na primeira etapa do que na segunda etapa; (ii) o uso do simulador auxilia a diminuir o nível de dificuldade percebida pelo candidato.

Análise Vertical – Grupo B

A segunda análise vertical é realizada entre a primeira e segunda etapas, no Grupo B. O resultado do teste estatístico de Mann-Whitney-Wilcoxon para taxa de acerto, retornou o *p-value* de 0,014. Com isto, refutando que a primeira e segunda etapas são iguais em relação a taxa de acerto. Neste contexto, comparamos as médias das taxas de acertos da primeira e segunda etapas, apresentada na tabela 7.1. Desta forma, obtemos indícios para duas suposições; (i) os participantes possuem uma taxa de acerto maior na primeira etapa do que na segunda etapa; (ii) o uso do simulador auxilia os participantes a alcançar uma maior taxa de acerto.

Para o tempo de execução, o resultado do teste estatístico de Mann-Whitney-Wilcoxon, retornou o *p-value* de 0,0004. Com isto, refutando que a primeira e segunda etapas são iguais em relação a tempo. Neste contexto, comparamos as médias de tempo da primeira e segunda etapas, apresentada na tabela 7.1. Desta forma, obtemos indícios para duas suposições; (i) os participantes executam em um tempo menor a primeira etapa do que a segunda etapa; (ii) o uso do simulador auxilia os participantes a alcançar um menor tempo.

Já para o nível de dificuldade, o resultado do teste estatístico de Mann-Whitney-Wilcoxon, retornou o *p-value* de 0,0003. Com isto, refutando que a primeira e segunda etapas são iguais em relação a nível de dificuldade. Neste contexto, comparamos as médias de nível de dificuldade da primeira e segunda etapas, apresentada na tabela 7.1. Desta forma, obtemos indícios para duas suposições; (i) os participantes percebem um nível de dificuldade menor na primeira etapa do que na segunda etapa; (ii) o uso do simulador auxilia a diminuir o nível de dificuldade percebida pelo candidato.

Análise do Resultados do Latin Square

A partir das análises horizontal e vertical do *Latin Square* obtemos indícios para algumas suposições. A tabela 7.2 lista as suposições para a métrica de taxa de acerto na análise horizontal. Com isto, podemos observar que a suposição PE-TA(I) é refutada pela suposição SE-TA(I). Desta forma, restando apenas as suposições PE-TA(II) e SE-TA(II), que são equivalentes, ou seja, o uso do simulador auxilia os participantes a alcançar uma maior taxa de acerto. Porém, ainda temos que concluir sobre o viés de taxa de acerto nas etapas.

Tabela 7.2: Suposições de Taxa de Acerto - Horizontal.

Primeira	PE-TA(I)	O grupo A possui uma taxa de acerto menor que o grupo B.	$TA(A) < TA(B)$
	Etapa	PE-TA(II)	O uso do simulador auxilia os participantes a alcançar uma maior taxa de acerto.
Segunda	SE-TA(I)	O grupo A possui uma taxa de acerto maior que o grupo B.	$TA(A) > TA(B)$
	Etapa	SE-TA(II)	O uso do simulador auxilia os participantes a alcançar uma maior taxa de acerto.

A tabela 7.3 lista as suposições para a métrica de tempo na análise horizontal. Com isto, podemos observar que a suposição PE-T(I) é refutada pela suposição SE-T(I). Desta forma, restando apenas as suposições PE-T(II) e SE-T(II), que são equivalentes, ou seja, o uso do simulador auxilia os participantes a executar em um menor tempo. Porém, ainda temos que concluir sobre o viés tempo de execução nas etapas.

Tabela 7.3: Suposições de Tempo - Horizontal.

Primeira Etapa	PE-T(I)	O grupo A executa em um tempo maior que o grupo B.	$T(A) > T(B)$
	PE-T(II)	O uso do simulador auxilia os participantes a executar em um menor tempo.	$T(SPM) > T(CPM)$
Segunda Etapa	SE-T(I)	O grupo A executa em um tempo menor que o grupo B.	$T(A) < T(B)$
	SE-T(II)	O uso do simulador auxilia os participantes a executar em um menor tempo.	$T(CPM) < T(SPM)$

A tabela 7.4 lista as suposições para a métrica de nível de dificuldade na análise horizontal. Com isto, podemos observar que a suposição PE-D(I) é refutada pela suposição SE-D(I). Desta forma, restando apenas as suposições PE-D(II) e SE-D(II), que são equivalentes, ou seja, o uso do simulador auxilia a diminuir o nível de dificuldade percebida pelo participante. Porém, ainda temos que concluir sobre o viés percepção de dificuldade nas etapas.

Tabela 7.4: Suposições de Nível de Dificuldade - Horizontal.

Primeira Etapa	PE-D(I)	O grupo A percebe um nível de dificuldade maior que o grupo B.	$D(A) > D(B)$
	PE-D(II)	O uso do simulador auxilia a diminuir o nível de dificuldade percebida pelo participante.	$D(SPM) > D(CPM)$
Segunda Etapa	SE-D(I)	O grupo A percebe um nível de dificuldade menor que o grupo B.	$D(A) < D(B)$
	SE-D(II)	O uso do simulador auxilia a diminuir o nível de dificuldade percebida pelo participante.	$D(CPM) < D(SPM)$

Neste contexto, analisamos de forma horizontal as suposições restantes, mitigando o viés dos grupos, na taxa de acerto, no tempo de execução e na percepção de nível de dificuldade. Porém, resta verificar as suposições geradas na análise vertical. A tabela 7.5 lista as suposições para a métrica de taxa de acerto na análise vertical. Com isto, podemos observar que a suposição A-TA(I) é refutada pela suposição B-TA(I). Desta forma, restando apenas as suposições A-TA(II) e B-TA(II), que são equivalentes, ou seja, o uso do simulador auxilia os participantes a alcançar uma maior taxa de acerto. Com isto, mitigando o viés de taxa de acerto nas etapas.

Tabela 7.5: Suposições de Taxa de Acerto - Vertical.

Grupo	A-TA(I)	Os participantes possuem uma taxa de acerto menor na primeira etapa do que na segunda etapa.	$TA(PE) < TA(SE)$
	A	A-TA(II)	O uso do simulador auxilia os participantes a alcançar uma maior taxa de acerto.
Grupo	B-TA(I)	Os participantes possuem uma taxa de acerto maior na primeira etapa do que na segunda etapa.	$TA(PE) > TA(SE)$
	B	B-TA(II)	O uso do simulador auxilia os participantes a alcançar uma maior taxa de acerto.

A tabela 7.6 lista as suposições para a métrica de tempo na análise vertical. Com isto, podemos observar que a suposição A-T(I) é refutada pela suposição B-T(I). Desta forma, restando apenas as suposições A-T(II) e B-T(II), que são equivalentes, ou seja o uso do simulador auxilia os participantes a executar em um menor tempo. Com isto, mitigando o viés de tempo de execução nas etapas.

Tabela 7.6: Suposições de Tempo - Vertical.

Grupo	A-T(I)	Os participantes executam em um tempo maior a primeira etapa do que a segunda etapa.	$T(PE) > T(SE)$
	A	A-T(II)	O uso do simulador auxilia os participantes a executar em um menor tempo.
Grupo	B-T(I)	Os participantes executam em um tempo menor a primeira etapa do que a segunda etapa.	$T(PE) < T(SE)$
	B	B-T(II)	O uso do simulador auxilia os participantes a executar em um menor tempo.

A tabela 7.7 lista as suposições para a métrica de nível de dificuldade na análise vertical. Com isto, podemos observar que a suposição A-D(I) é refutada pela suposição B-D(I). Desta forma, restando apenas as suposições A-D(II) e B-D(II), que são equivalentes, ou seja, o uso do simulador auxilia a diminuir o nível de dificuldade percebida pelo participante. Com isto, mitigando o viés de percepção de dificuldade dos nas etapas.

Tabela 7.7: Suposições de nível de Dificuldade - Vertical.

Grupo	A-D(I)	Os participantes percebem um nível de dificuldade maior na primeira etapa do que na segunda etapa.	$D(PE) > D(SE)$
	A	A-D(II)	O uso do simulador auxilia a diminuir o nível de dificuldade percebida pelo participante.
Grupo	B-D(I)	Os participantes percebem um nível de dificuldade menor na primeira etapa do que na segunda etapa.	$D(PE) < D(SE)$
	B	B-D(II)	O uso do simulador auxilia a diminuir o nível de dificuldade percebida pelo participante.

Neste contexto, com a mitigação do viés dos grupos e etapas, podemos analisar as suposições restantes. Com estas chegamos a três conclusões. Para a métrica de taxa de acerto restaram PE-TA(II), SE-TA(II), A-TA(II) e B-TA(II), podemos observar que estas são equivalentes. Desta forma, obtemos a nossa primeira conclusão de que o uso do simulado auxilia os participantes a alcançar uma maior taxa de acerto. Para a métrica de tempo de execução restaram PE-T(II), SE-T(II), A-T(II) e B-T(II), podemos observar que estas são equivalentes. Desta forma, obtemos nossa segunda conclusão de que o uso do simulador auxilia os participantes a executar em um menor tempo. E finalmente, para a métrica de percepção de nível de dificuldade restaram PE-D(II), SE-D(II), A-D(II) e B-D(II), podemos observar que estas são equivalentes. Desta forma, obtemos nossa terceira conclusão de que o uso do simulador auxilia a diminuir o nível de dificuldade percebida pelo participante.

7.3

Possíveis Ameaças

Para evitar imprevisto na realização do estudo, aplicamos previamente o questionário para um grupo de três pessoas. Estas pessoas não fazem parte dos grupos que participaram do estudo. Com isto, verificamos os possíveis imprevisto, sendo um teste para nosso estudo.

Uma das possíveis ameaças ao estudo seria os participantes não possuírem um conhecimento prévio sobre gestão de projetos e EAP. Mitigamos esta ameaça com aplicação do questionário no fim do curso da disciplina de gestão de projetos. Além disto, o conhecimento necessário para realização do estudo foi discutido no início do curso, em conjunto com o professor. A ementa do curso foi aprimorada para contemplar o conhecimento necessário para realização do estudo. Podemos observar por meio da análise de perfil que os participantes possuíam um conhecimento prévio sobre Gestão de Projetos e EAP.

Outra ameaça seria o tempo muito longo para realizar as tarefas. Por exemplo, um participante poderia investir um tempo muito grande em uma atividade. Desta forma, restando pouco tempo para outra atividade. Configuramos um tempo máximo para todas as atividades. Com isto, o participante não poderia ultrapassar este tempo. Entretanto, nenhum participante alcançou o tempo limite.

A falta de interesses dos participantes poderia ser uma ameaça a realização do estudo. Junto com o professor da disciplina, decidimos incentivar o aluno, pontuando os participantes de acordo com os acertos nas atividades. Com isto, mitigando a ameaça de falta de interesse.

Um outra ameaça seria o ganho de informação de uma atividade para outra, pois os dados iniciais do estudo eram comuns para todas as atividades. Acreditamos que como cada tarefa tinha objetivos diferentes, a informação inicial não teve uma grande influência à medida que o participante progredia na execução do estudo. A análise entre atividades auxiliou a verificar que o dado inicial comum para atividades não teve uma grande influência no resultado.

O nível de complexidade das etapas poderia influenciar no resultado final, uma etapa poderia ser mais complexa que a outra, sendo assim, outra ameaça. Porém, com o auxílio da análise do *Latin Square*, verificamos que as etapas não tiveram uma grande influência em nosso estudo.

Por fim, a ameaça de grupos desbalanceados, ou seja, um grupo poderia possuir um conhecimento prévio melhor do que o outro grupo. Porém, com

o auxiliou do *Latin Square* verificamos que esta possibilidade não teve uma grande influência em nossas conclusões.

7.4

Considerações Finais

Neste capítulo, apresentamos um estudo realizado com o objetivo de avaliar uma simulação, com o modelo conceitual ProMabs instanciado, para auxiliar os envolvidos na gestão de projetos de software com a tomada de decisão. O estudo realizado foi baseado na execução de atividades de gestão de projeto. Cada uma dessas atividades possuem um objetivo específico. Além disto, estas atividades foram divididas em duas etapas. Configuramos um *Latin Square* para mitigar os vies da realização das etapas e também o viés do grupos. O estudo foi aplicado em alunos de graduação de sistema de informação. Estes alunos estavam na ultima semana da disciplina de Gestão de Projetos.

Neste contexto, realizamos uma avaliação qualitativa e quantitativa. Primeiro avaliamos os resultados por atividades, com e sem simulador. Após isto, realizamos uma avaliação de forma horizontal e vertical no *Latin Square*, da utilização do simulador. Com estas avaliações, podemos observar que o simulador auxilia os participantes nas atividades de gestão em três aspectos; (i) A alcançar uma maior taxa de acerto; (ii) A executar em um menor tempo; E finalmente, (iii) a diminuir o nível de dificuldade percebida pelo participante. O resultado do estudo apontou uma melhoria no desempenho das atividades de gestão de projetos de software, quando é utilizado o simulador. Desta forma, este resultado permitiu concluir que a instanciação do ProMabs contribuiu positivamente para representar elementos e seus relacionamento em simulação de gestão de projetos, executar de cenários para auxiliar na tomada de decisão, e finalmente para apoiar projetos adaptativos de software.

Conclusão e Trabalhos Futuros

A complexidade que envolve gestão de projetos de software dificulta a criação de um modelo robusto o suficiente para representar e instanciar elementos e suas relações que fazem parte deste domínio. Estes elementos e suas relações, identificados e representados facilitam a execução de cenários que podem auxiliar os gestores de projetos. Entretanto, além da complexidade destes elementos e suas relações, ao longo do projeto, acontecem alterações nestes elementos, refletindo em suas relações. Desta forma, além de representar e executar cenários, existe a necessidade de apoiar as alterações destes elementos que acontecem ao longo do projeto.

O modelo conceitual ProMabs contém componentes básico para auxiliar a criar simulações que incorporem elementos e seus relacionamento. Com isto, executar cenários com esta incorporação, para auxiliar a tomada de decisão. Além disto, seus componentes permitem ao modelo capturar possíveis alterações nestes elementos e suas relações e simular seu impacto. Por meio de três instanciações do ProMabs, podemos observar: (i) Como estes elementos são representados e como os agentes se comportam ao simular uma gestão de projeto; (ii) Como estes elementos representados executam cenários para auxiliar o processo de tomada de decisão; (iii) Como apoiar alterações ao longo do projeto com multiagentes.

Neste contexto, o ProMabs e suas implementações apresentam indícios positivos de que pode ser aplicado para auxiliar no gerenciamento para estratégias, planejamento, controle e operação, no âmbito de gestão de projetos de software. Desta forma, apoiando o processo de tomada de decisão, por meio de uma análise de diferentes cenários que podem ser aplicadas tanto para o planejamento como para a execução do projeto. Além disto, o ProMabs instanciado permite a análise do impacto de possíveis alterações ao longo do projeto. Isto é resultado da composição de componentes essenciais para simular projetos de software complexos e dinâmicos.

8.1

Contribuições do trabalho

No capítulo 3, são definidos os componentes que permitem a simulação de gestão de projeto de software por meio de simulação multiagentes. Para isto, o ProMabs foi descrito em termos de componentes básicos para ser aplicado a partir de dados externos, utilizando-os para simular eventos para o planejamento e execução. Além disto, descrevemos o ProMabs utilizando o MAS-ML, uma linguagem de modelagem para sistemas multiagentes, permitindo a extensão do ProMabs para outros objetivos, por meio da compreensão da modelagem. Como por exemplo, criar agentes para coletar métricas do ambiente externo e comparar com a simulação. Além disto, criamos um gerador de EAP para ser utilizado em estudos que precisam gerar diferentes e diversas EAPs, ou seja, mais de uma EAP e mais de uma configuração.

A partir do ProMabs, no capítulo 4, obtemos uma representação de elementos e seus relacionamento para simular gestão de projeto de software, na área de gestão de escopo. Aplicando a baixo nível como estes elementos e seus relacionamentos podem ser representados e simulados. Para isto, utilizamos uma plataforma de programação multiagentes, baseada em noções de crenças, metas e planos, como atitudes mentais que orientam a realização de comportamentos dos agentes. Utilizamos também um mecanismo de organização do ambiente e social. Por fim, por meio de um exemplo ilustrativo descrevemos como a utilização do JaCaMoPM, uma instância do ProMabs, pode auxiliar nos processos de gestão de escopo. Para isto, representamos os elementos de projeto no âmbito de escopo.

Para o capítulo 5, obtemos cenários baseado em estratégias. Para isto, utilizamos a instanciação do ProMabs em um ambiente de simulação. Estes cenários são executados simultaneamente por meio de um único modelo. Além disto, por meio de um estudo de caso ilustramos como estes cenários podem auxiliar a tomada de decisão, no contexto de escopo, tempo, recurso humano e custo.

Para apoiar simulação de gestão de projetos adaptativos de software, instanciamos o ProMabs em uma ferramenta de simulação, no capítulo 6. Para isto, aplicamos os conhecimentos adquiridos para representar elementos e seus relacionamentos, assim como, a execução de cenários para auxiliar no processo de tomada de decisão. Com isto, utilizamos uma ferramenta de simulação, para instanciar exemplos ilustrativos e descrever como a simulação de gestão projetos adaptativos de software auxilia a gestão.

Neste contexto, o ProMabs é instanciado em uma plataforma de programação multiagentes, em um ambiente de simulação e em uma ferramenta de simulação. Com isto, criamos uma perspectiva em termos de implementação de como o modelo conceitual ProMabs pode ser instanciado em tecnologias diferentes. Auxiliando na compreensão dos agentes e implementação dos componentes do ProMabs, em diferentes níveis de abstração de tecnologias.

Podemos observar por meio das instanciações apresentadas que, em termos de nível de compreensão dos agentes e a nível de implementação dos componentes do ProMabs, cada uma das instanciações tem suas vantagens e desvantagens. O JaCaMoPM possui recursos para uma compreensão dos comportamentos dos agentes pois utiliza o Jason, uma plataforma para programação de agentes, que permite uma maior especificação. Similarmente, isto acontece para a organização do ambiente e organização social. A organização do ambiente com o Cartago, uma plataforma utilizada pelo JaCaMoPM, permite uma maior especificação de como os artefatos serão observados e operados pelos agentes. A organização social com o Moise, uma plataforma utilizada pelo JaCaMoPM, permite uma maior especificação de como é o esquema social dos agentes por meio de normas, missões e metas. Por outro lado, como é baseada nestas três plataformas a nível de implementação isto dificulta pois existe a necessidade de conhecer as três plataformas e seus recursos, aumentando significativamente a curva de aprendizado. Ainda em termos de nível de implementação do ProMabs, o JaCaMoPM dificulta a implementação do componente Filtro de Informação com seus objetivos, pois não existe um suporte que conecte a simulação com o ambiente externo. Com isto, a inserção de novos contextos é realizada de forma manual, dificultando a utilização.

O Cormas possui recursos para a compreensão dos comportamentos dos agentes quando utilizados para executar simulações simultâneas. Porém as metas, missões e normas não são expressas de forma sistemática, ou seja, não possui os recursos de especificação. Já a nível de implementação do ProMabs, o componente modelo de simulação de execução não é implementado, pois o Cormas não oferece os recursos para visualizar a simulação de gestão de projetos em execução. Desta forma, dificulta a implementação por completo dos componentes filtro de informação e analisador. Por outro lado, o Anylogic, a nível de implementação do ProMabs, permite a instanciação de todos os componentes por completo por meio dos recursos oferecido pela ferramenta. Também, permite por meio de seu ambiente de desenvolvimento uma interação trivial com o desenvolvedor e com o usuário final da simulação. Desta forma,

facilitando o desenvolvimento da simulação, como também, seu uso. Por outro lado, a nível de compreensão dos agentes, a ferramenta não possui recursos para especificar de forma sistemática a organização do ambiente e esquema social dos agentes.

Finalmente no capítulo 7, apresentamos um estudo de caso com análise qualitativa e quantitativa, utilizando a instanciação do ProMabs em um ferramenta de simulação. Este estudo aplicou atividades de gestão com e sem o uso do simulador. De forma mais geral, o objetivo deste estudo é verificar se o simulador auxilia no processo de tomada de decisão para estas atividades. Como resultado do estudo, podemos observar que a instanciação do ProMabs auxiliou nas atividades em termos de alcançar uma maior taxa de acerto, diminuir o tempo de execução da atividade, e finalmente diminuir o nível de dificuldade percebida pelos participantes.

8.2

Limitações da Tese

Uma das limitações deste trabalho é o esforço para instanciar o ProMabs. Cada um dos componentes do ProMabs tem um objetivo específico para a simulação da gestão de projeto de software. Esta limitação ficou evidenciada na implementação do ProMabs nas diversas tecnologias, apesar de não apresentarmos nenhuma avaliação mais profunda neste sentido. Entretanto, estudos apontam (GROUP, 2013), (Rubinstein, 2007) que projetos na área de tecnologia de informação possuem uma elevada taxa de insucesso. Com evidências crescentes que as atribuições do gerente de projeto podem ser cruciais para o desempenho eficiente e eficaz da equipe do projeto em melhorar os resultados da equipe, conseqüentemente do projeto (Flyvbjerg e Budzier, 2011), (Taylor e Woelfer, 2011). Neste contexto, acreditamos que o esforço inicial para instanciar o ProMabs, compensaria na diminuição de erros humanos no processo de tomada de decisão, auxiliando assim, os gerentes de projetos.

Outra limitação deste trabalho, é a utilização apenas de EAP para instanciação e visualização da simulação. Escolhemos a EAP, com o objetivo de utilizar a simulação de gestão de projeto em outros domínios. Visto que, uma EAP é a base para o planejamento e acompanhamento em vários outros domínios de projeto. Apesar de utilizarmos apenas a EAP para instanciar e visualizar os resultados da simulação, acreditamos que é possível aplicar outra técnica, como por exemplo, um diagrama de flecha para as tarefas.

Finalmente, a limitação da falta de um estudo de caso, no qual poderíamos utilizar uma instanciação do ProMabs para acompanhar um projeto real do começo ao fim. Como um projeto de software geralmente ocupa uma grande fatia de tempo, este tipo de estudo não pôde ser realizado. Porém, com a disponibilidade deste trabalho, assim como das instanciações do ProMabs, acreditamos que no futuro pesquisadores poderão realizar um estudo como este, aproveitando estes artefatos.

8.3

Trabalho Futuros

Embora esta tese possua várias contribuições supracitadas, existem ainda algumas direções para trabalhos futuros. Uma das direções, concentra-se em aplicar o ProMabs em outro domínio de projeto. Acreditamos que o ProMabs pode ser aplicado no domínio de construção civil. Na qual instanciaríamos outros sensores para o ambiente externo além do banco de dados. Por exemplo, poderíamos usar a internet das coisas para analisar o andamento da obra de acordo com as tarefas e metas estabelecidas. Caso existisse alguma anomalia nas tarefas de execução, o simulador verificaria o impacto. Por exemplo, se para uma tarefa de construção de parede, foi estabelecido um tempo para execução, um sensor poderia estar informando ao ProMabs a evolução por unidade de tempo. Com isto, executando e analisando se, de fato, estava de acordo com o esperado, por meio de simulação.

Outra direção é aplicar o ProMabs para acompanhar um projeto real do início ao fim, nas atividades que envolvem alguma tomada de decisão. Porém, a comparação com um projeto que não o utilize, seria um esforço enorme. Visto que, existem várias variáveis que podem influenciar nesta comparação. Entretanto, poderíamos realizar uma análise qualitativa do uso do ProMabs em um projeto real, sem realizar comparações com outro projeto sem o uso do ProMabs.

Poderíamos aplicar o experimento apresentado no capítulo 7 em um grupo de participantes com um maior tempo de vivência em projetos. Podemos observar no perfil dos participantes, que o tempo em projetos de software é baixo. Com esta outra aplicação, poderíamos comparar o quanto o ProMabs auxilia na tomada de decisão, entre participantes com tempos de vivência diferentes, ou seja, mais experientes com os menos experientes.

Por meio dos estudos realizados, verificamos que uma das dificuldades da instanciação do ProMabs, está na obtenção dos dados com as características

dos seres humanos que são representados por agentes. Poderíamos mitigar esta dificuldade por meio da utilização de uma base histórica de projeto, para extrair as características dos seres humanos, aplicando uma técnica de aprendizagem. Desta forma, os agentes são instanciados por meio desta extração e a simulação pode mudar suas características caso a extração forneça novos dados em tempo real, durante a execução do projeto simulado.

Referências Bibliográficas

ADAMATTI, D. F.; SICHMAN, J. S.; RABAK, C. Jogoman: A Prototype using Multi-Agent-Based Simulation and Role-Playing Games in Water Management. **Conference On Multi-Agent Modelling For Environmental Management**, 2005.

AGARWAL, R.; UMPHRESS, D. A Flexible Model for Simulation of Software Development Process. **Proceedings of the 48th Annual Southeast Regional Conference**, 2010.

ALBA, E.; CHICANO, J. F. Software Project Management with GAs. **Information Sciences**, v. 177, n. 11, p. 2380–2401, jun. 2007.

ALI, N. B.; PETERSEN, K. A Consolidated Process for Software Process Simulation: State of the Art and Industry Experience. In: **SEAA '12: Proceedings of the 2012 38th Euromicro Conference on Software Engineering and Advanced Applications**. [S.l.]: IEEE Computer Society, 2012.

ARAÚZO, J. A.; PAJARES, J.; LOPEZ-PAREDES, A. Simulating the Dynamic Scheduling of Project Portfolios. **Simulation Modelling Practice and Theory**, Elsevier, v. 18, n. 10, p. 1428–1441, 2010.

ATHAVALE, S.; BALARAMAN, V. Human Behavioral Modeling for Enhanced Software Project Management. In: **7th International Conference on Software Engineering**. [S.l.: s.n.], 2013. p. 15–17.

BAIA, D. et al. **MultiAgent-Based Simulation in Software Project Management: Scope Management Representantion and Visualization**. [S.l.], Nov 2014. Disponível em: <<https://cs.uwaterloo.ca/sites/ca.computer-science/files/uploads/files/CS-2014-22.pdf>>.

BAIA, D. de M. An Integrated Multi-Agent-Based Simulation Approach to Support Software Project Management. In: IEEE PRESS. **Proceedings of**

the 37th International Conference on Software Engineering-Volume 2. [S.l.], 2015. p. 911–914.

BAIA, D. de M. et al. **A MultiAgent-Based Simulation Model to Support Management Decision Making in Software Development.** [S.l.], Mar 2014. Disponível em: <<https://cs.uwaterloo.ca/sites/ca.computer-science/files/uploads/files/CS-2014-04.pdf>>.

BALSERA, J. V.; GONZÁLEZ-FANJUL, C. A.; FERNANDEZ, F. O. **Data Mining Applied to the Improvement of Project Management.** 2012.

BOISSIER, O. et al. Multi-Agent Oriented Programming with JaCaMo. **Science of Computer Programming**, v. 78, n. 6, p. 747–761, jun. 2013.

BORDINI, R. H.; HÜBNER, J. F.; WOOLDRIDGE, M. **Programming Multi-Agent Systems in AgentSpeak using Jason.** [S.l.]: John Wiley & Sons, 2007.

BORSHCHEV, A. **The Big Book of Simulation Modeling: Multimethod Modeling with AnyLogic 6.** [S.l.]: AnyLogic North America, 2013.

BOUSQUET, F. et al. Cormas: Common-Pool Resources and Multi-Agent Systems. In: **Tasks and Methods in Applied Artificial Intelligence.** [S.l.]: Springer, 1998. p. 826–837.

BRAZIER, F. M.; JONKER, C. M.; TREUR, J. Modelling Project Coordination in a Multi-Agent Framework. In: IEEE. **Enabling Technologies: Infrastructure for Collaborative Enterprises, 1996. Proceedings of the 5th Workshop on.** [S.l.], 1996. p. 148–155.

BRIOT, J.-P.; GUYOT, P.; IRVING, M. Participatory Simulation for Collective Management of Protected Areas for Biodiversity Conservation and Social Inclusion. **AIS-CMS**, v. 7, p. 183–188, 2007.

CHERIF, R.; DAVIDSSON, P. Software Development Pprocess Simulation: Multi Agent-Based Simulation versus System Dynamics. In: **Multi-Agent-Based Simulation X.** [S.l.]: Berlin, Springer, 2010. v. 5683, n. 7, p. 73–85. ISBN 3642135528.

DRAPPA, A.; LUDEWIG, J. Simulation in Software Engineering Training. **International Conference on Software Engineering**, 2000.

DROGOUL, A.; VANBERGUE, D.; MEURISSE, T. Multi-Agent Based Simulation: Where are the Agents? **Multi-agent-based Simulation II**, Springer Berlin Heidelberg, Berlin, Heidelberg, p. 1–15, 2003.

FLYVBJERG, B.; BUDZIER, A. Why Your IT Project may be Riskier than You Think. **Harvard Business Review**, v. 89, n. 9, p. 601–603, 2011.

GAETA, M.; RITROVATO, P. Generalized Environment for Process Management in Cooperative Software Engineering. **Proceedings of the 26th International Computer Software and Applications Conference on Prolonging Software Life: Development and Redevelopment**, p. 1049–1053, 2002.

GROUP, S. CHAOS MANIFESTO 2013. **The Standish Group International**, 2013.

HAAPIO, T.; MENZIES, T. Data Mining with Software Industry Project Data: A Case Study. **IADIS AC**, p. 33–38, 2009.

HANNE, T.; NICKEL, S. A Multiobjective Evolutionary Algorithm for Scheduling and Inspection Planning in Software Development Projects. **European Journal of Operational Research**, Elsevier, v. 167, n. 3, p. 663–678, 2005.

HUBNER, J. F.; SICHMAN, J. S.; BOISSIER, O. Developing Organised Multiagent Systems using the noise+ Model: Programming Issues at the System and Agent Levels. **International Journal of Agent-Oriented Software Engineering**, Inderscience Publishers, v. 1, n. 3-4, p. 370–395, 2007.

JENNINGS, N.; WOOLDRIDGE, M. Applications of Intelligent Agents. **Agent Technology: Foundations, Applications and Markets**, p. 3–28, 1998.

JOSLIN, D.; POOLE, W. Agent-Based Simulations for Software Project Planning. **Winter Simulation Conference**, p. 1059–1066, 2005.

KAMEAS, A. Towards the Next-Generation of Ambient Intelligent Environments. **Enabling Technologies: Infrastructures for Collaborative Enterprises (WETICE), 2010 19th IEEE International Workshop on**, p. 1–6, 2010.

KELLNER, M. I.; MADACHY, R. J.; RAFFO, D. M. Software Process Simulation Modeling: Why? What? How? **Journal of Systems and Software**, v. 46, n. 2-3, p. 91–105, abr. 1999.

KORB, W. et al. Risk Analysis for a Reliable and Safe Surgical Robot System. **International Congress Series**, p. 766–770, 2003.

KRASNER, G. E.; POPE, S. T. A Cookbook for Using the Model-view Controller User Interface Paradigm in Smalltalk-80. **J. Object Oriented Program.**, SIGS Publications, Denville, NJ, USA, v. 1, n. 3, p. 26–49, ago. 1988. ISSN 0896-8438.

KRAVARI, K.; BASSILIADES, N. A Survey of Agent Platforms. **Journal of Artificial Societies and Social Simulation**, p. 1–21, jan. 2015.

LAKEY, P. B. A Hybrid Software Process Simulation Model for Project Management. In: **Proceedings of the 2003 International Workshop on Software Process Simulation and Modeling (ProSim-03)**. [S.l.: s.n.], 2003.

LEUNG, H.; POON, C. Multi-Agent Environment for Software Quality Assurance, autoq. **EUROMICRO Conference, 1999. Proceedings. 25th**, 1999.

LUCK, M.; MCBURNEY, P. Computing as Interaction: Agent and Agreement Technologies. 2008.

MAURER, R. Project Coordination in Design Processes. **IEEE International 5th Workshop for Enabling Technologies (WET) : Infrastructure for Collaborative Enterprises**, p. 191–198, 1996.

MI, P.; SCACCHI, W. A Knowledge-Based Environment for Modeling and Simulating Software Engineering Processes. **Knowledge and Data Engineering, IEEE Transactions on**, IEEE, v. 2, n. 3, p. 283–289, 1990.

MI, P.; SCACCHI, W. Articulation: An Integrated Approach to the Diagnosis, Replanning, and Rescheduling of Software Process Failures. **KBSE**, p. 77–84, 1993.

MITTERMEIR, R. et al. AMEISE - A Media Education Initiative for Software Engineering. **Proceedings International Workshop ICL - Interactive Computer Aided Learning**, 2003.

MULLER, J.; BAUER, B.; FRIESE, T. Programming Software Agents as Designing Executable Business Processes: A Model Driven Perspective. **Programming Multi-Agent Systems (LNAI)**, v. 28, p. 49–71, 2004.

NIENABER, R.; BARNARD, A. A Generic Agent Technology Framework to Support the Various Software Project Management Processes. **Proceedings of the Informing Science and Information Technology Conference (INSITE)**, p. 108–118, 2007.

NIENABER, R.; CLOETE, E. A Software Agent Framework for the Support of Software Project Management. In: **SAICSIT 03: Proceedings of the 2003 annual research conference of the South African institute of computer scientists and information technologists on Enablement through technology**. [S.l.]: South African Institute for Computer Scientists and Information Technologists, 2003.

O'DONNELL, M. Context in Dynamic Modelling. **AMSTERDAM STUDIES IN THE THEORY AND HISTORY OF LINGUISTIC SCIENCE SERIES 4**, JOHN BENJAMINS BV, p. 63–100, 1999.

PAGE, C. L. et al. Participatory Agent-Based Simulation for Renewable Resource Management: The Role of the Cormas Simulation Platform to Nurture a Community of Practice. **Journal of Artificial Societies and Social Simulation**, JASSS, 2012.

PARUNAK, H.; BAKER, A.; CLARK, S. The AARIA Agent Architecture: An Example of Requirements-Driven Agent-Based Systems Design. **Proceedings of the First International Conference on Autonomous Agents**, p. 482–483, 1997.

PERSSON, C.; PICARD, G.; RAMPARANY, F. A Multi-Agent Organization for the Governance of Machine-to-Machine Systems. In: **Proceedings of the IEEE International Conference on Intelligent Agent Technology**. [S.l.: s.n.], 2011.

PFAHL, D.; RUHE, G.; KOVAL, N. An Experiment for Evaluating the Effectiveness of Using a System Dynamics Simulation Model in Software Project Management Education. **IEEE METRICS**, p. 97–109, 2001.

PMI. **A Guide to The Project Management Body of Knowledge - PMBOK Guides - Fifth Edition**. [S.l.]: Project Management Institute, 2013. ISBN 193069945X, 9781933890517.

RAO, A. S.; GEORGEFF, M. P. Modeling Rational Agents within a BDI-architecture. **KR**, v. 91, p. 473–484, 1991.

RICCI, A. et al. Environment Programming in CArtAgO. In: **Multi-Agent Programming**. [S.l.]: Springer, 2009. p. 259–288.

RIGAUD, E.; GUARNIERI, F. Towards an Agent Oriented Virtual Organization Dedicated to Risk Prevention in Small and Medium Sized Companies. **Proceedings of the 13th International Workshop on Database and Expert Systems (DEXA)**, 2002.

RODRÍGUEZ, D.; SATPATHY, M.; PFAHL, D. Effective Software Project Management Education through Simulation Models: An Externally Replicated Experiment. **PROFES**, v. 3009, n. Chapter 21, p. 287–301, 2004.

RUBINSTEIN, D. Standish Group Report: There's Less Development Chaos Today. **Software Development Times**, 2007.

RUNESON, P.; HOST, M. Guidelines for Conducting and Reporting Case Study Research in Software Engineering. **Empirical Software Engineering**, v. 14, n. 2, p. 131–164, dez. 2008.

RUS, I.; COLLOFELLO, J.; LAKEY, P. Software Process Simulation for Reliability Management. **Journal of Systems and Software**, v. 46, p. 173–182, 1999.

SAUER, J.; APPLERATH, H. Scheduling the Supply Chain in Teams of Agents. **Proceedings of the 36th Hawaii International Conference on System Sciences**, 2003.

SILVA, V. T. D.; CHOREN, R.; LUCENA, C. J. D. MAS-ML: a Multiagent System Modelling Language. **International Journal of Agent-Oriented Software Engineering**, Inderscience Publishers, v. 2, n. 4, p. 382–421, 2008.

SILVA, V. T. da. **Uma Linguagem de Modelagem para Sistemas Multi-agentes Baseada em um Framework Conceitual para Agentes e Objetos**. Tese (Doutorado) — Pontifícia Universidade Católica do Rio de Janeiro, 2004.

SILVA, V. T. da; LUCENA, C. J. P. de. From a Conceptual Framework for Agents and Objects to a Multi-Agent System Modeling Language. **Autonomous Agents and Multi-Agent Systems**, v. 9, n. 1/2, p. 145–189, 2004.

SPASIC, B.; ONGGO, B. S. S. Agent-Based Simulation of the Software Development Process: A Case Study at AVL. **Winter Simulation Conference :1-400:11**, p. 400–11, 2012.

SZYMCZAK, M. et al. Resource Management in an Agent-based Virtual Organization — Introducing a Task Into the System. **International**

Conferences on Web Intelligence and Intelligent Agent Technology, p. 458–462, 2007.

TAUSWORTHE, R. The work breakdown structure in software project management. **Journal of Systems and Software**, v. 1, p. 181–186, 1980.

TAYLOR, H.; WOELFER, J. P. Leadership Behaviors in Information Technology Project Management: An Exploratory Study. In: **Proceedings of the 2011 44th Hawaii International Conference on System Sciences**. [S.l.: s.n.], 2011. ISBN 978-0-7695-4282-9.

TOLEDO, C. et al. Developing a Knowledge Management Multi-Agent System using the Jacamo Platform. In: **Proceedings of ProMAS (AAMAS)**. [S.l.: s.n.], 2011.

WICKENBERG, T.; DAVIDSSON, P. On Multi Agent Based Simulation of Software Development Processes. **MABS**, v. 2581, n. Chapter 12, p. 171–180, 2002.

WICKENBERG, T.; DAVIDSSON, P. On Multi Agent Based Simulation of Software Development Processes. **Multi-Agent-Based Simulation II**, v. 2581, p. 171–180, 2003. ISSN 0302-9743.

WU, C.; SIMMONS, D. Plan Tracking Knowledge Base. **Proceedings of the 24th Annual International Computer Software and Applications Confernece (COMPSAC)**, p. 299–304, 2000.

WU, C.-S.; CHANG, W.-c.; SETHI, I. K. A Metric-Based Multi-Agent System for Software Project Management. In: **ICIS '09: Proceedings of the 2009 Eighth IEEE/ACIS International Conference on Computer and Information Science**. [S.l.]: IEEE Computer Society, 2009.

XIAO, J.; AO, X.-T.; TANG, Y. Solving Software Project Scheduling Problems with Ant Colony Optimization. **Computers and Operation Research**, v. 40, n. 1, p. 33–46, jan. 2013.

YANNIBELLI, V.; AMANDI, A. A Knowledge-Based Evolutionary Assistant to Software Development Project Scheduling. **Expert Systems With Applications**, v. 38, n. 7, p. 8403–8413, jul. 2011.

ZHANG, H.; KITCHENHAM, B.; PFAHL, D. Software Process Simulation Modeling: Facts, Trends and Directions. In: **APSEC '08: Proceedings of the 2008 15th Asia-Pacific Software Engineering Conference**. [S.l.]: IEEE Computer Society, 2008.

ZHANG, H.; KITCHENHAM, B. A.; PFAHL, D. Reflections on 10 Years of Software Process Simulation Modeling: A Systematic Review. **ICSP**, v. 5007, n. Chapter 30, p. 345–356, 2008.

ZHANG, H.; KITCHENHAM, B. A.; PFAHL, D. Software Process Simulation Modeling: An Extended Systematic Review. **ICSP**, v. 6195, n. Chapter 27, p. 309–320, 2010.

A

Questionários Utilizados no Estudo

A.1

Questionário para Grupo A

A.1.1

Formação Acadêmica, Conhecimentos Específicos e Participação/Experiência

Gostaríamos de *saber qual* sua formação, conhecimentos específicos e participação/experiência em projetos de software.

Sobre sua formação acadêmica, responda as perguntas a seguir.

Tabela A.1: Formação Acadêmica

Possui curso superior ou está cursando?		
Sim <input type="checkbox"/>	Não <input type="checkbox"/>	Qual? _____
Concluiu?		
Sim <input type="checkbox"/>	Não <input type="checkbox"/>	Qual período ou ano? _____
Possui pós-graduação ou está cursando?		
Especialização <input type="checkbox"/>	Mestrado <input type="checkbox"/>	Doutorado <input type="checkbox"/>
Qual? _____		
Concluiu?		
Sim <input type="checkbox"/>	Não <input type="checkbox"/>	Qual período ou ano? _____

Sobre seus conhecimentos específicos, responda as perguntas a seguir.

Tabela A.2: Conhecimentos Específicos

Qual seu conhecimento sobre simulação baseada em multi-agentes?				
Nenhum <input type="checkbox"/>	Pouco <input type="checkbox"/>	Razoável <input type="checkbox"/>	Bom <input type="checkbox"/>	Muito Bom <input type="checkbox"/>
Qual seu conhecimento sobre Estrutura Analítica de Projeto (EAP)?				
Nenhum <input type="checkbox"/>	Pouco <input type="checkbox"/>	Razoável <input type="checkbox"/>	Bom <input type="checkbox"/>	Muito Bom <input type="checkbox"/>
Qual seu conhecimento sobre Gestão (Gerência) de Projetos de Software?				
Nenhum <input type="checkbox"/>	Pouco <input type="checkbox"/>	Razoável <input type="checkbox"/>	Bom <input type="checkbox"/>	Muito Bom <input type="checkbox"/>

Sobre sua participação em projetos, informe para cada tópico quantos anos de experiência você possui na área.

Tabela A.3: Participação em Projetos de Software

	Experiência	
Projeto de software acadêmico (faculdade, trabalho, escola)		anos
Projeto de software na industria		anos

A próxima parte do questionário será composta por duas etapas, cada uma com duas atividades. A primeira etapa possui duas atividades que serão realizadas utilizando apenas as informações ilustradas nas figuras A.1 e A.2. A figura A.1 contém informações sobre os recursos humanos que poderão ser utilizados para execução das tarefas da Estrutura Analítica de Projeto (EAP). Já a figura A.2 é a EAP que foi elaborada para o desenvolvimento de um sistema (software). Já na segunda etapa do questionário serão mais duas atividades. Porém, estas atividades serão realizadas utilizando simulações do mesmo projeto. O simulador foi criado por meio da instanciação do modelo conceitual ProMabs. As simulações irão utilizar os mesmos dados da EAP e recursos humanos ilustrados nas figuras A.2 e A.1.

ID	Recurso Humano	Nível de		Custo
		Habilidade (Skill)	Conhecimento (Level)	
1	João	1	2	30
		2	1	30
2	Gabriel	3	3	25
3	Pedro	4	3	35
4	Gustavo	5	2	25
5	Mario	6	2	40
		5	3	40
6	Luiza	2	3	45
		5	3	45
7	Joana	1	1	20
		5	1	20
8	Diego	6	1	20
		2	2	20
9	Tiago	3	1	20
10	Mateus	4	1	20

ID	Habilidade (Skill)
1	Analista de Requisitos
2	Engenheiro de Software
3	Analista de Design
4	Analista de Banco de Dados
5	Programador
6	Analista de Teste

ID	Nível de Conhecimento (Level)
1	Júnior
2	Pleno
3	Sênior

Figura A.1: Recursos Humanos, Habilidade (Skill) e Nível de Conhecimento (Level).

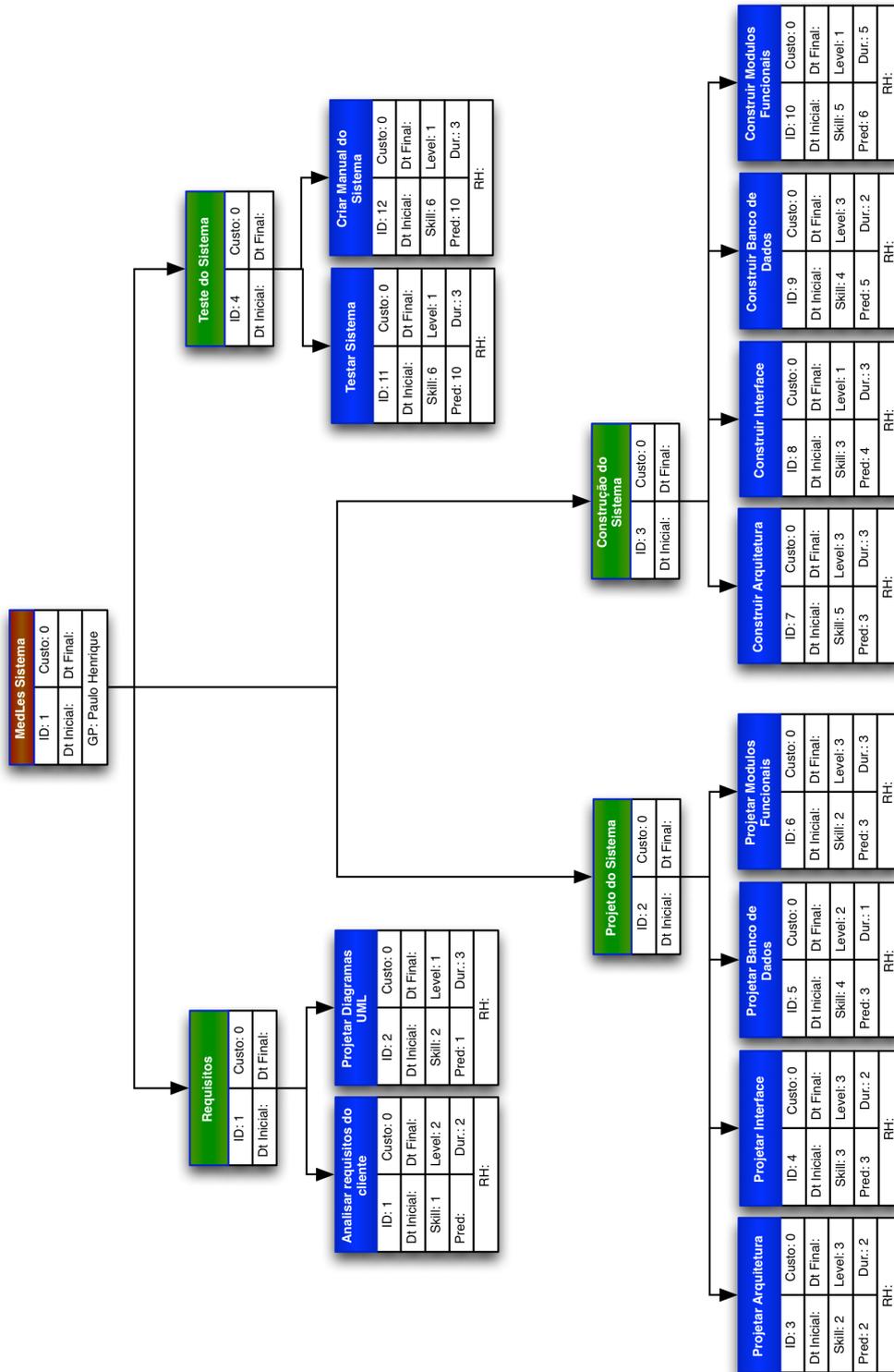


Figura A.2: EAP para o desenvolvimento do Sistema

A.1.2**Primeira Etapa**

Esta primeira etapa será realizada apenas com o auxílio das figuras A.1 e A.2.

Atividade 1

Você precisa atribuir recursos humanos para tarefas na EAP da figura A.2. Este projeto precisa ser executado com um **baixo custo** e com um recurso humano apropriado para a tarefa, ou seja, levando em consideração a habilidade (skill) **exigida** pela tarefa e a recomendação do nível de conhecimento (level). Para isso, utilize a estratégia a seguir.

Escolher um recurso humano que possua a mesma habilidade exigida da tarefa, além disso procurar o recurso humano que tem o menor custo.

Obs.: Um recurso humano só pode realizar uma tarefa por vez.

Utilize a tabela abaixo para preencher o recurso humano para cada tarefa e ao final some o custo total. **(25 minutos)**

Início: __:__:__

Tarefas	Recurso Humano	Duração	Custo RH	Valor
Analisar requisitos do cliente		2		
Projetar Diagramas UML		3		
Projetar Arquitetura		2		
Projetar Interface		2		
Projetar Banco de Dados		1		
Projetar Modulos Funcionais		3		
Construir Arquitetura		3		
Construir Interface		3		
Construir Banco de Dados		2		
Construir Modulos Funcionais		5		
Testar Sistema		3		
Criar Manual do Sistema		3		
			Total	

Fim: __:__:__

Dificuldade: ++ + +- - - -

Atividade 2

Baseado na Atividade 1 (anterior) identifique quais as tarefas tem inconformidade com seu recurso humano, ou seja, habilidade (skill) diferente da **exigida** pela tarefa ou nível de conhecimento (level) diferente da recomendação da tarefa. **Atenção:** Preencher apenas as tarefas que contém inconformidade. **(10 minutos)**

Início: __:__:__

Tarefas	Recurso Humano	Incoformidade	
Analisar requisitos do cliente		Skill <input type="checkbox"/>	Level <input type="checkbox"/>
Projetar Diagramas UML		Skill <input type="checkbox"/>	Level <input type="checkbox"/>
Projetar Arquitetura		Skill <input type="checkbox"/>	Level <input type="checkbox"/>
Projetar Interface		Skill <input type="checkbox"/>	Level <input type="checkbox"/>
Projetar Banco de Dados		Skill <input type="checkbox"/>	Level <input type="checkbox"/>
Projetar Modulos Funcionais		Skill <input type="checkbox"/>	Level <input type="checkbox"/>
Construir Arquitetura		Skill <input type="checkbox"/>	Level <input type="checkbox"/>
Construir Interface		Skill <input type="checkbox"/>	Level <input type="checkbox"/>
Construir Banco de Dados		Skill <input type="checkbox"/>	Level <input type="checkbox"/>
Construir Modulos Funcionais		Skill <input type="checkbox"/>	Level <input type="checkbox"/>
Testar Sistema		Skill <input type="checkbox"/>	Level <input type="checkbox"/>
Criar Manual do Sistema		Skill <input type="checkbox"/>	Level <input type="checkbox"/>

Fim: __:__:__

Dificuldade: ++ + +- - - -

A.1.3

Segunda Etapa

Esta segunda etapa será realizada utilizando um simulador que instanciou o modelo conceitual ProMabs. Este simulador irá utilizar os dados da EAP e recursos humanos ilustrados nas figuras A.2 e A.1.

Atividade 3

Você precisa atribuir recursos humanos para tarefas na EAP. Este projeto precisa ser executado com um recurso humano apropriado para a tarefa, ou seja, levando em consideração a habilidade (skill) **exigida** pela tarefa e a recomendação do nível de conhecimento (level). Para isso, utilize a estratégia a seguir.

Escolher um recurso humano que tem a mesma habilidade exigida da tarefa, além disso procurar um nível de conhecimento (level) igual ao sugerido da tarefa. Porém, se não encontrar um nível de conhecimento (level) igual da tarefa, procure algum nível de conhecimento (level) maior, seguindo uma ordem (júnior, pleno, sênior). Caso não encontre, procure um nível de conhecimento (level) menor, seguindo uma ordem (sênior, pleno, júnior).

Obs.: Um recurso humano só pode realizar uma tarefa por vez.

Utilize a tabela abaixo para preencher o recurso humano para cada tarefa e ao final some o custo total. **(15 minutos)**

Início: __:__:__

Tarefas	Recurso Humano	Duração	Custo RH	Valor
Analisar requisitos do cliente		2		
Projetar Diagramas UML		3		
Projetar Arquitetura		2		
Projetar Interface		2		
Projetar Banco de Dados		1		
Projetar Modulos Funcionais		3		
Construir Arquitetura		3		
Construir Interface		3		
Construir Banco de Dados		2		
Construir Modulos Funcionais		5		
Testar Sistema		3		
Criar Manual do Sistema		3		
			Total	

Fim: __:__:__

Dificuldade: ++ + +- - - -

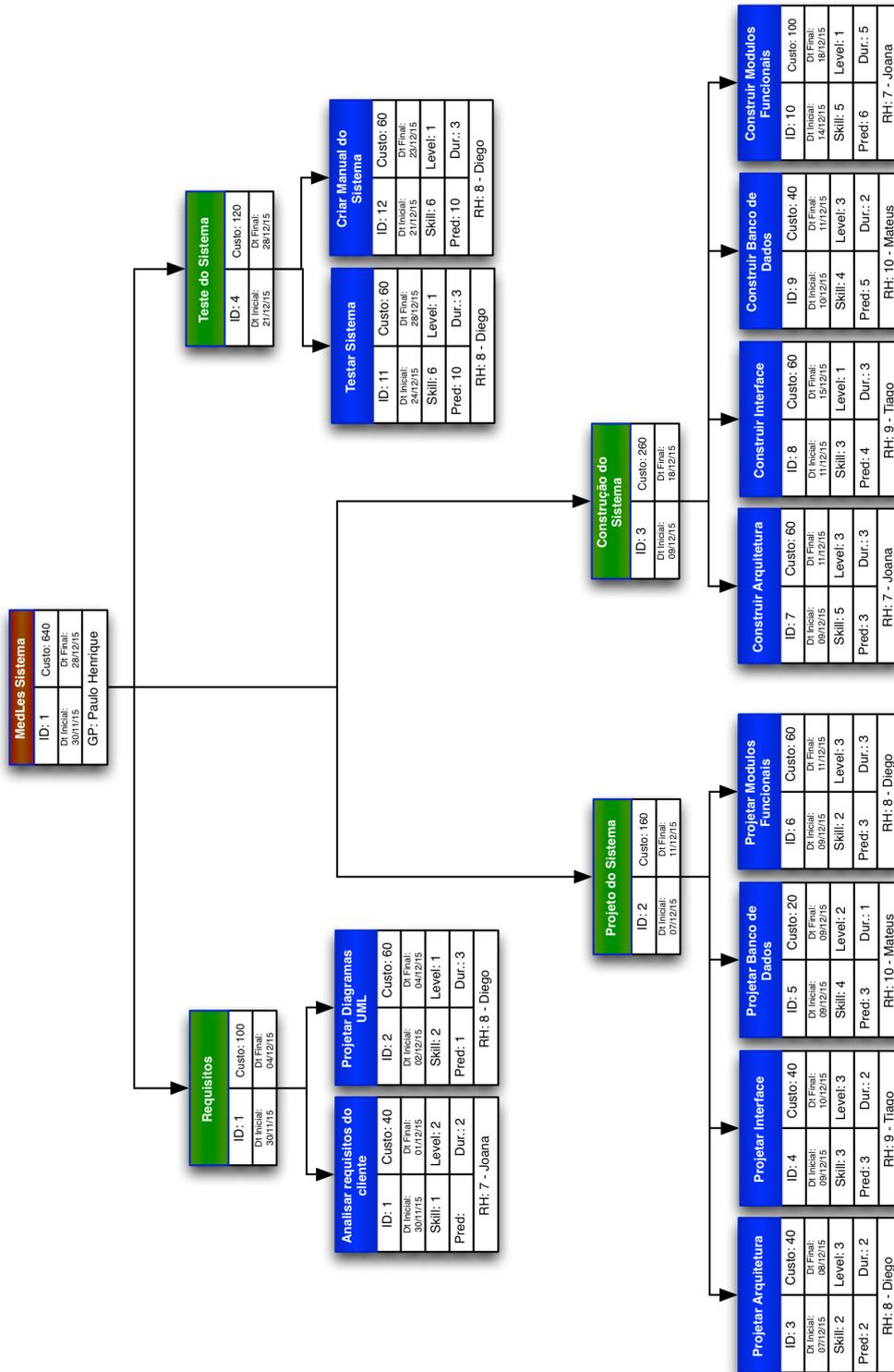


Figura A.3: EAP para redução tempo

Atividade 4

No projeto da figura A.3 o cliente inicialmente priorizou o custo do projeto, agora solicitou a entrega do projeto o mais rápido possível. Verifique quais alterações de recursos humanos podem ser realizadas na EAP para atender esta solicitação.

Obs.: A exigência para realizar uma tarefa é que o recurso humano precisa possuir a habilidade (skill) igual a **exigida** pela tarefa. Além disto, existe uma recomendação do nível de conhecimento (level) da tarefa. Outro ponto, é que um recurso não pode realizar mais de uma tarefa ao mesmo tempo. Por fim, sábado e domingo são dias de descanso, ou seja, ninguém realiza tarefa. Utilize o calendário que foi entregue em anexo, caso necessário.

Dica: Identifique as tarefas que podem ser realizadas paralelamente e que por motivo de custo estão sendo realizadas em sequencia devido a restrição do recurso fazer uma tarefa por vez.

Utilize a tabela abaixo para preencher alterações de recursos humanos nas tarefas e informar o impacto dessa mudança na data de início e fim das tarefas. Ao final informe a data final prevista para o projeto.

Atenção: Preencher apenas as tarefas que possuem alterações.

(25 minutos)

Início: __:__:__

Tarefas	Recurso Humano	Dt Início	Dt Fim
Analisar requisitos do cliente		__/__/__	__/__/__
Projetar Diagramas UML		__/__/__	__/__/__
Projetar Arquitetura		__/__/__	__/__/__
Projetar Interface		__/__/__	__/__/__
Projetar Banco de Dados		__/__/__	__/__/__
Projetar Modulos Funcionais		__/__/__	__/__/__
Construir Arquitetura		__/__/__	__/__/__
Construir Interface		__/__/__	__/__/__
Construir Banco de Dados		__/__/__	__/__/__
Construir Modulos Funcionais		__/__/__	__/__/__
Testar Sistema		__/__/__	__/__/__
Criar Manual do Sistema		__/__/__	__/__/__
Datas do Projeto		__/__/__	__/__/__

Fim: __:__:__

Dificuldade: ++ + +- - - -

A.2**Questionário para Grupo B****A.2.1****Formação Acadêmica, Conhecimentos Específicos e Participação/Experiência**

Gostaríamos de *saber qual* sua formação, conhecimentos específicos e participação/experiência em projetos de software.

Sobre sua formação acadêmica, responda as perguntas a seguir.

Tabela A.4: Formação Acadêmica

Possui curso superior ou está cursando?		
Sim <input type="checkbox"/>	Não <input type="checkbox"/>	Qual? _____
Concluiu?		
Sim <input type="checkbox"/>	Não <input type="checkbox"/>	Qual período ou ano? _____
Possui pós-graduação ou está cursando?		
Especialização <input type="checkbox"/>	Mestrado <input type="checkbox"/>	Doutorado <input type="checkbox"/>
Qual? _____		
Concluiu?		
Sim <input type="checkbox"/>	Não <input type="checkbox"/>	Qual período ou ano ? _____

Sobre seus conhecimentos específicos, responda as perguntas a seguir.

Tabela A.5: Conhecimentos Específicos

Qual seu conhecimento sobre simulação baseada em multi-agentes?				
Nenhum <input type="checkbox"/>	Pouco <input type="checkbox"/>	Razoável <input type="checkbox"/>	Bom <input type="checkbox"/>	Muito Bom <input type="checkbox"/>
Qual seu conhecimento sobre Estrutura Analítica de Projeto (EAP)?				
Nenhum <input type="checkbox"/>	Pouco <input type="checkbox"/>	Razoável <input type="checkbox"/>	Bom <input type="checkbox"/>	Muito Bom <input type="checkbox"/>
Qual seu conhecimento sobre Gestão (Gerência) de Projetos de Software?				
Nenhum <input type="checkbox"/>	Pouco <input type="checkbox"/>	Razoável <input type="checkbox"/>	Bom <input type="checkbox"/>	Muito Bom <input type="checkbox"/>

Sobre sua participação em projetos, informe para cada tópico quantos anos de experiência você possui na área.

Tabela A.6: Participação em Projetos de Software

	Experiência	
Projeto de software acadêmico (faculdade, trabalho, escola)		anos
Projeto de software na industria		anos

A próxima parte do questionário será composta por duas etapas, cada uma com duas atividades. A primeira etapa possui duas atividades que serão realizadas utilizando simulações de um projeto. O simulador foi criado por meio da instanciação do modelo conceitual chamado ProMabs. As simulações irão utilizar os mesmo dados da Estrutura Analítica de Projeto (EAP) e recursos humanos ilustrados nas figuras A.5 e A.4. A figura A.4 contém informações sobre os recursos humanos que poderão ser utilizados para execução das tarefas da EAP. Já a figura A.5 é a EAP que foi elaborada para o desenvolvimento de um sistema (software). Já na segunda etapa do questionário serão mais duas atividades. Porém, estas atividades serão realizadas utilizando apenas as informações ilustradas nas figuras A.4 e A.5.

PUC-Rio - Certificação Digital Nº 1212395/CA

	Recurso Humano	Habilidade (Skill)	Nível de Conhecimento (Level)	Custo
1	João	1	2	30
		2	1	30
2	Gabriel	3	3	25
3	Pedro	4	3	35
4	Gustavo	5	2	25
5	Mario	6	2	40
		5	3	40
6	Luiza	2	3	45
		5	3	45
7	Joana	1	1	20
		5	1	20
8	Diego	6	1	20
		2	2	20
9	Tiago	3	1	20
10	Mateus	4	1	20

ID	Habilidade (Skill)
1	Analista de Requisitos
2	Engenheiro de Software
3	Analista de Design
4	Analista de Banco de Dados
5	Programador
6	Analista de Teste

ID	Nível de Conhecimento (Level)
1	Júnior
2	Pleno
3	Sênior

Figura A.4: Recursos Humanos, Habilidade (Skill) e Nível de Conhecimento (Level).

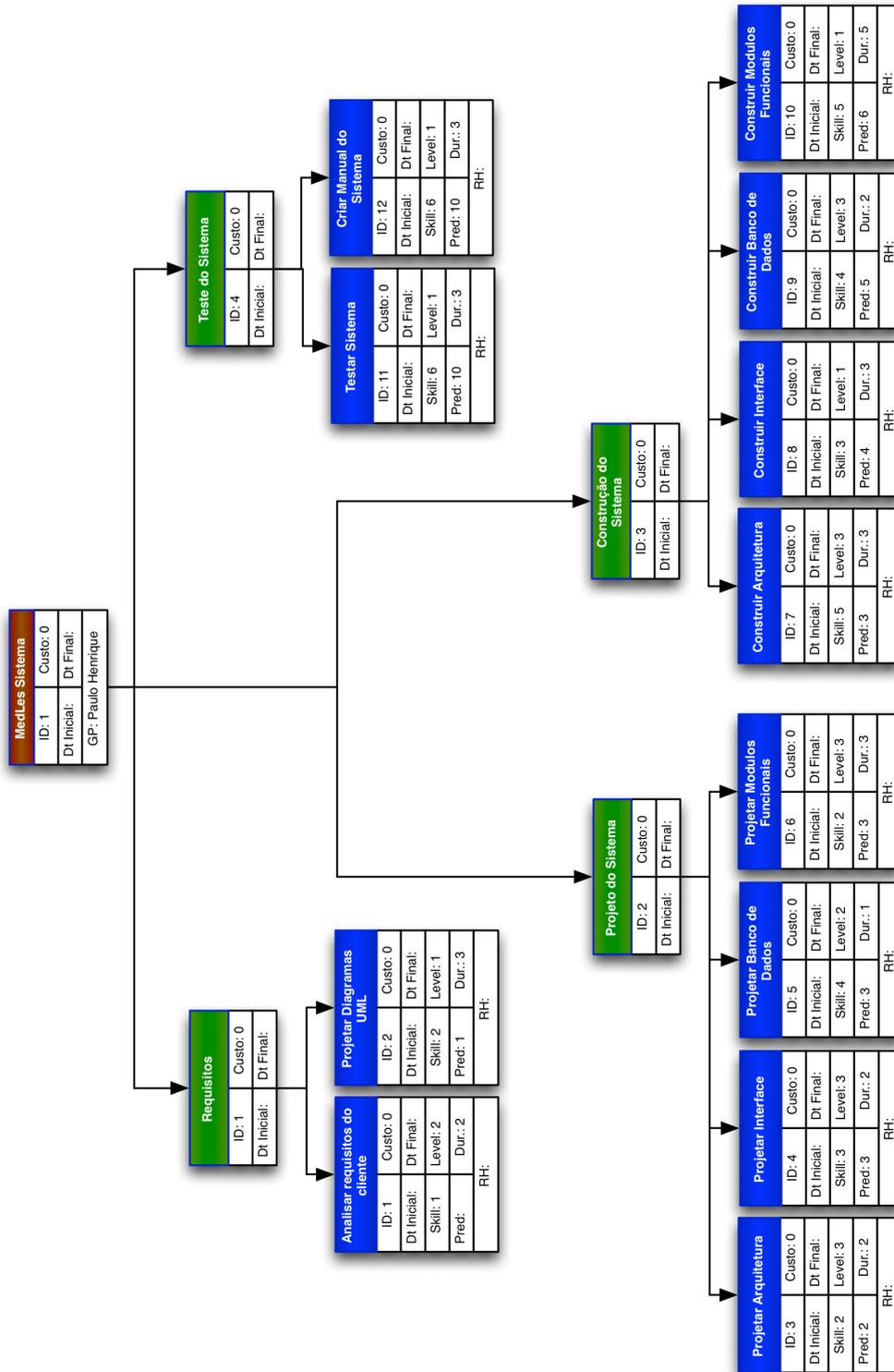


Figura A.5: EAP para o desenvolvimento do Sistema

A.2.2

Primeira Etapa

Esta segunda etapa será realizada utilizando um simulador que instanciou o modelo conceitual ProMabs. Este simulador irá utilizar os dados da EAP e recursos humanos ilustrados nas figuras A.5 e A.4.

Atividade 1

Você precisa atribuir recursos humanos para tarefas na EAP da figura A.5. Este projeto precisa ser executado com um **baixo custo** e com um recurso humano apropriado para a tarefa, ou seja, levando em consideração a habilidade (skill) **exigida** pela tarefa e a recomendação do nível de conhecimento (level). Para isso, utilize a estratégia a seguir.

Escolher um recurso humano que possua a mesma habilidade exigida da tarefa, além disso procurar o recurso humano que tem o menor custo.

Obs.: Um recurso humano só pode realizar uma tarefa por vez.

Utilize a tabela abaixo para preencher o recurso humano para cada tarefa e ao final some o custo total. **(25 minutos)**

Início: __:__:__

Tarefas	Recurso Humano	Duração	Custo RH	Valor
Analisar requisitos do cliente		2		
Projetar Diagramas UML		3		
Projetar Arquitetura		2		
Projetar Interface		2		
Projetar Banco de Dados		1		
Projetar Modulos Funcionais		3		
Construir Arquitetura		3		
Construir Interface		3		
Construir Banco de Dados		2		
Construir Modulos Funcionais		5		
Testar Sistema		3		
Criar Manual do Sistema		3		
Total				

Fim: __:__:__

Dificuldade: ++ + +- - - -

Atividade 2

Baseado na Atividade 1 (anterior) identifique quais as tarefas tem inconformidade com seu recurso humano, ou seja, habilidade (skill) diferente da **exigida** pela tarefa ou nível de conhecimento (level) diferente da recomendação da tarefa. **Atenção:** Preencher apenas as tarefas que contém inconformidade. **(10 minutos)**

Início: __:__:__

Tarefas	Recurso Humano	Incoformidade	
Analisar requisitos do cliente		Skill <input type="checkbox"/>	Level <input type="checkbox"/>
Projetar Diagramas UML		Skill <input type="checkbox"/>	Level <input type="checkbox"/>
Projetar Arquitetura		Skill <input type="checkbox"/>	Level <input type="checkbox"/>
Projetar Interface		Skill <input type="checkbox"/>	Level <input type="checkbox"/>
Projetar Banco de Dados		Skill <input type="checkbox"/>	Level <input type="checkbox"/>
Projetar Modulos Funcionais		Skill <input type="checkbox"/>	Level <input type="checkbox"/>
Construir Arquitetura		Skill <input type="checkbox"/>	Level <input type="checkbox"/>
Construir Interface		Skill <input type="checkbox"/>	Level <input type="checkbox"/>
Construir Banco de Dados		Skill <input type="checkbox"/>	Level <input type="checkbox"/>
Construir Modulos Funcionais		Skill <input type="checkbox"/>	Level <input type="checkbox"/>
Testar Sistema		Skill <input type="checkbox"/>	Level <input type="checkbox"/>
Criar Manual do Sistema		Skill <input type="checkbox"/>	Level <input type="checkbox"/>

Fim: __:__:__

Dificuldade: ++ + +- - --

A.2.3

Segunda Etapa

Esta segunda etapa será realizada apenas com o auxílio das figuras A.4 e A.5.

Atividade 3

Você precisa atribuir recursos humanos para tarefas na EAP. Este projeto precisa ser executado com um recurso humano apropriado para a tarefa, ou seja, levando em consideração a habilidade (skill) **exigida** pela tarefa e a recomendação do nível de conhecimento (level). Para isso, utilize a estratégia a seguir.

Escolher um recurso humano que tem a mesma habilidade exigida da tarefa, além disso procurar um nível de conhecimento

(level) igual ao sugerido da tarefa. Porém, se não encontrar um nível de conhecimento (level) igual da tarefa, procure algum nível de conhecimento (level) maior, seguindo uma ordem (júnior, pleno, sênior). Caso não encontre, procure um nível de conhecimento (level) menor, seguindo uma ordem (sênior, pleno, júnior).

Obs.: Um recurso humano só pode realizar uma tarefa por vez.

Utilize a tabela abaixo para preencher o recurso humano para cada tarefa e ao final some o custo total. **(15 minutos)**

Início: __:__:__

Tarefas	Recurso Humano	Duração	Custo RH	Valor
Analisar requisitos do cliente		2		
Projetar Diagramas UML		3		
Projetar Arquitetura		2		
Projetar Interface		2		
Projetar Banco de Dados		1		
Projetar Modulos Funcionais		3		
Construir Arquitetura		3		
Construir Interface		3		
Construir Banco de Dados		2		
Construir Modulos Funcionais		5		
Testar Sistema		3		
Criar Manual do Sistema		3		
			Total	

Fim: __:__:__

Dificuldade: ++ + +- - - -

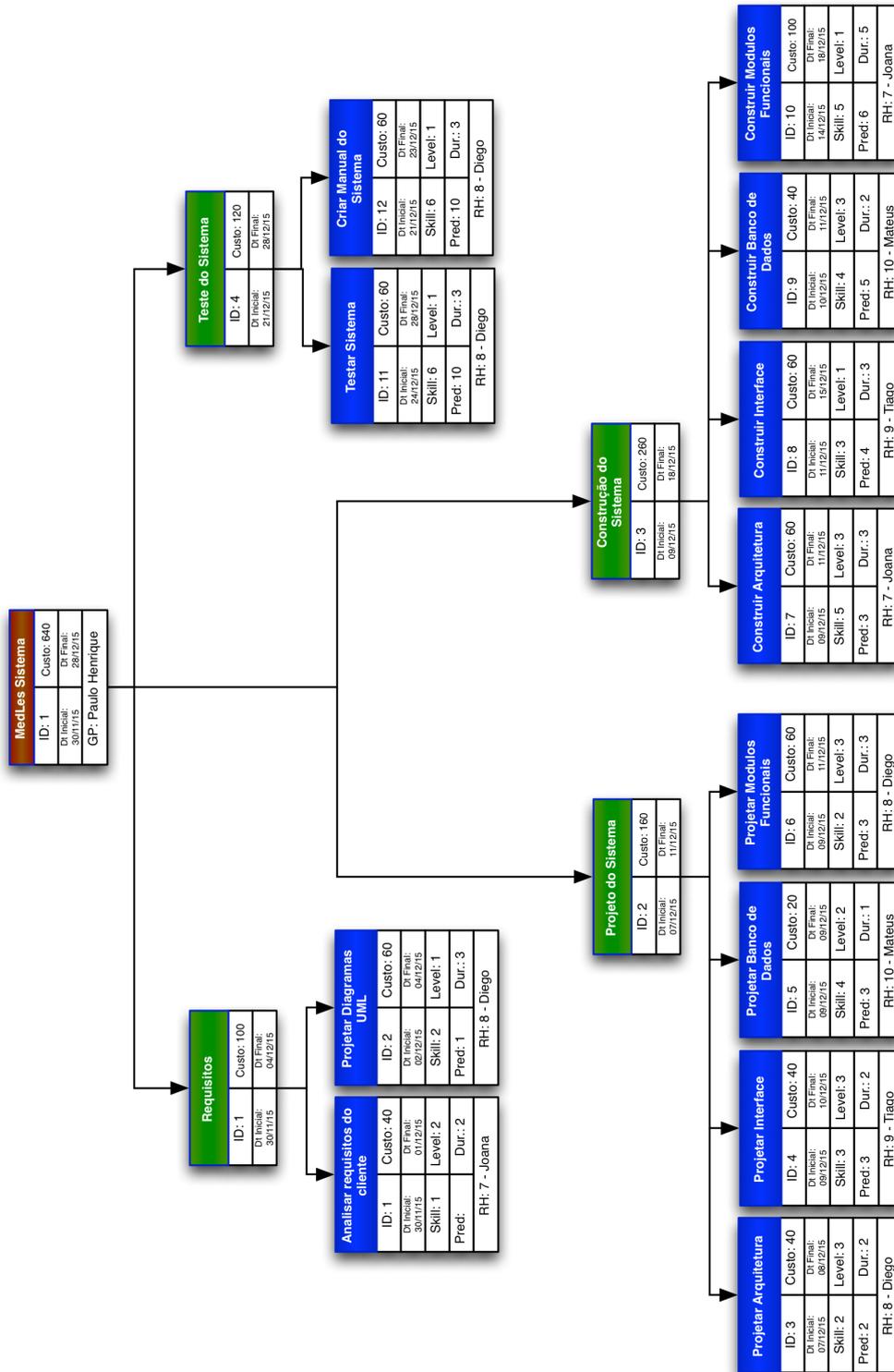


Figura A.6: EAP para redução tempo

Atividade 4

No projeto da figura A.6 o cliente inicialmente priorizou o custo do projeto, agora solicitou a entrega do projeto o mais rápido possível. Verifique quais alterações de recursos humanos podem ser realizadas na EAP para atender esta solicitação.

Obs.: A exigência para realizar uma tarefa é que o recurso humano precisa possuir a habilidade (skill) igual a **exigida** pela tarefa. Além disto, existe uma recomendação do nível de conhecimento (level) da tarefa. Outro ponto, é que um recurso não pode realizar mais de uma tarefa ao mesmo tempo. Por fim, sábado e domingo são dias de descanso, ou seja, ninguém realiza tarefa. Utilize o calendário que foi entregue em anexo, caso necessário.

Dica: Identifique as tarefas que podem ser realizadas paralelamente e que por motivo de custo estão sendo realizadas em sequencia devido a restrição do recurso fazer uma tarefa por vez.

Utilize a tabela abaixo para preencher alterações de recursos humanos nas tarefas e informar o impacto dessa mudança na data de início e fim das tarefas. Ao final informe a data final prevista para o projeto.

Atenção: Preencher apenas as tarefas que possuem alterações.

(25 minutos)

Início: __:__:__

Tarefas	Recurso Humano	Dt Início	Dt Fim
Analisar requisitos do cliente		__/__/__	__/__/__
Projetar Diagramas UML		__/__/__	__/__/__
Projetar Arquitetura		__/__/__	__/__/__
Projetar Interface		__/__/__	__/__/__
Projetar Banco de Dados		__/__/__	__/__/__
Projetar Modulos Funcionais		__/__/__	__/__/__
Construir Arquitetura		__/__/__	__/__/__
Construir Interface		__/__/__	__/__/__
Construir Banco de Dados		__/__/__	__/__/__
Construir Modulos Funcionais		__/__/__	__/__/__
Testar Sistema		__/__/__	__/__/__
Criar Manual do Sistema		__/__/__	__/__/__
Datas do Projeto		__/__/__	__/__/__

Fim: __:__:__

Dificuldade: ++ + +- - - -

B

Cenários em Simulação e Simulados

Seguem os três gráficos em alta resolução.

PUC-Rio - Certificação Digital Nº 1212395/CA

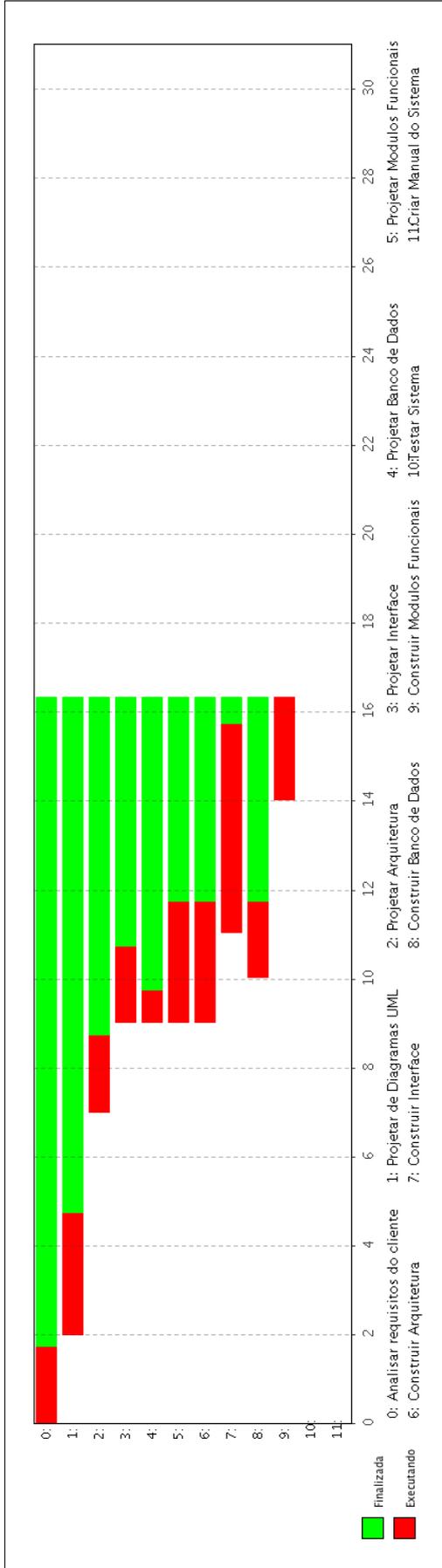


Figura B.1: Gráfico - Estado da Tarefa x Dias. (Alta Resolução)

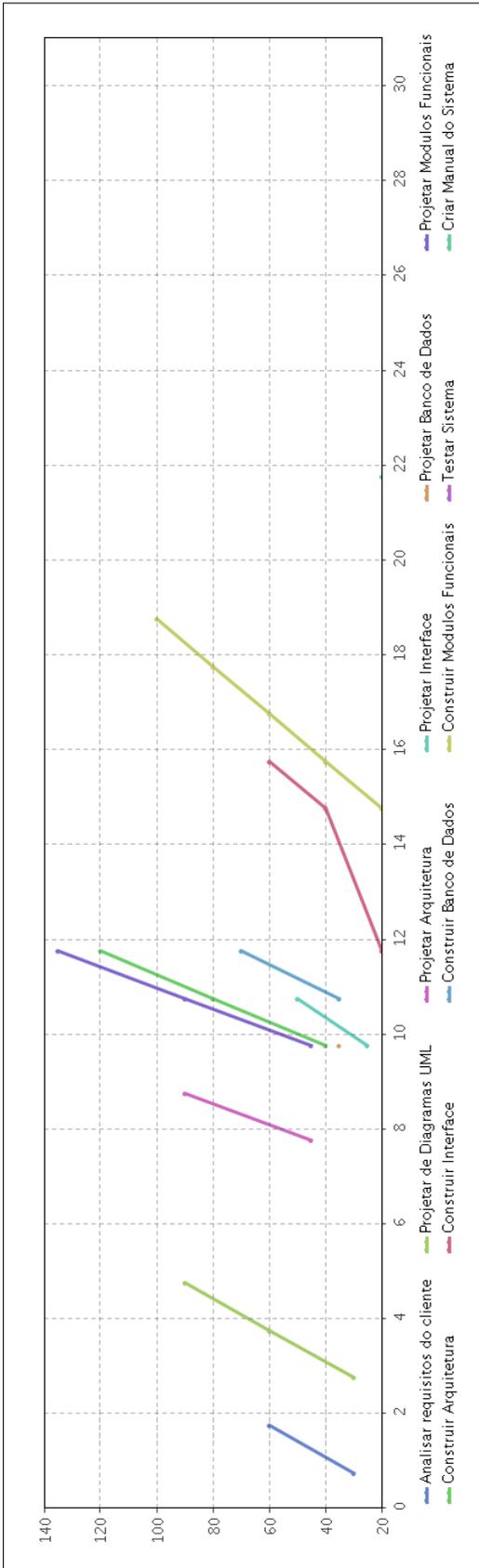


Figura B.2: Gráfico - Custo da Tarefas x Dias. (Alta Resolução)

PUC-Rio - Certificação Digital Nº 1212395/CA

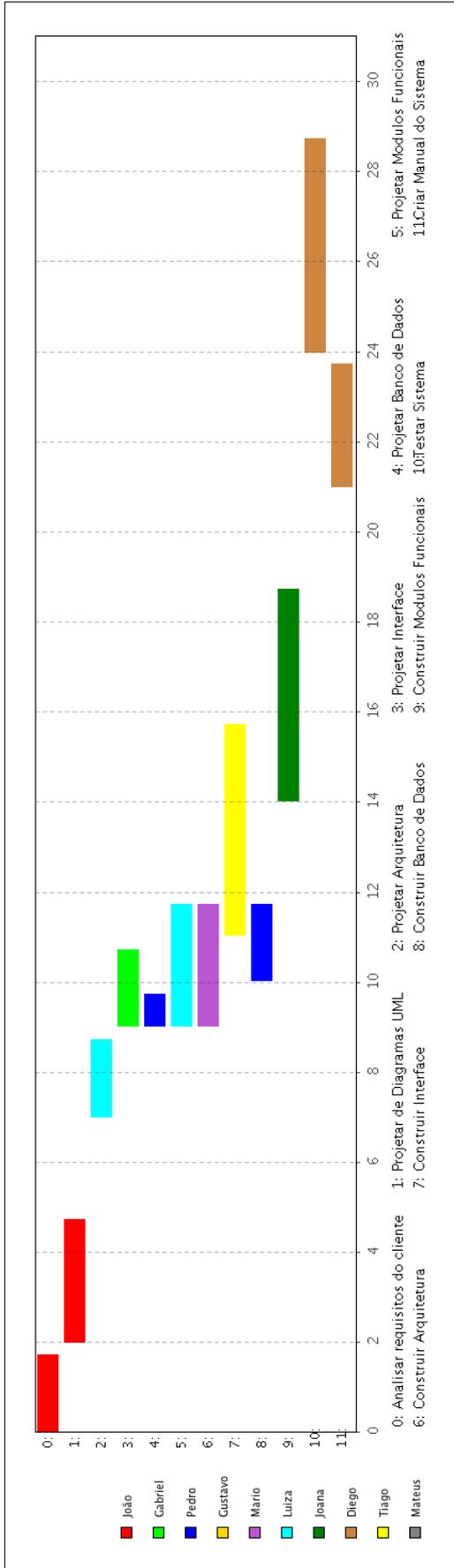


Figura B.3: Gráfico - Recurso Humano por Tarefas x Dias. (Alta Resolução)