CAIO CESAR FATTORI

COLABORAÇÃO E COOPERAÇÃO ENTRE SISTEMAS PRODUTIVOS
DISPERSOS DE EMPRESAS VIRTUAIS

São Paulo
2015

CAIO CESAR FATTORI

COLABORAÇÃO E COOPERAÇÃO ENTRE SISTEMAS PRODUTIVOS
DISPERSOS DE EMPRESAS VIRTUAIS

Tese apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do título de doutor em Ciências

São Paulo
2015

CAIO CESAR FATTORI

COLABORAÇÃO E COOPERAÇÃO ENTRE SISTEMAS PRODUTIVOS
DISPERSOS DE EMPRESAS VIRTUAIS

Tese apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do título de doutor em Ciências

Área de concentração:
Engenharia de Controle e Automação Mecânica

Orientador:
Paulo Eigi Miyagi

São Paulo
2015

Este exemplar foi revisado e alterado em relação à versão original, sob responsabilidade única do autor e com a anuência de seu orientador.

São Paulo, de de 2015

Assinatura do autor

Assinatura do orientador

Catálogo-na-publicação

Fattori, Caio Cesar

Colaboração e cooperação entre sistemas produtivos dispersos de empresas virtuais / Fattori, C. C. --ed.rev.--São Paulo, 2015.

214 p.

Tese (Doutorado) - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos.

1. Empresas virtuais 2. Redes de Petri 3. Interoperabilidade 4. Negócios (Processos) I. Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos II. t.

DEDICATÓRIA

Dedico essa tese ao meu sobrinho Matias de Melo Fattori Abbade Miguel e ao sobrinho de coração Davi Lucca Carrelli Homem de Menezes, que nasceram durante a realização desse estudo e trouxeram alegria para nós.

AGRADECIMENTOS

Ao professor Paulo Eigi Miyagi, pela orientação e pelo constante estímulo transmitido durante todo o trabalho.

O presente trabalho foi realizado com apoio da CAPES, Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil.

Aos professores Diolino José dos Santos Filho e Fabrício Junqueira, pelo apoio técnico e acadêmico constantes ao trabalho.

Aos amigos Marcosiris Amorim de Oliveira Pessoa, Reinaldo Squillante Jr., André Cavalheiros, Osvaldo Asato, Maurício Blos, Edson Hiroshi Watanabe, Jeferson Afonso Lopes de Souza e Robson Marinho da Silva que apoiaram o desenvolvimento do trabalho no LSA.

À amiga e aos amigos Camila Lainetti de Moraes, Gabriel de Souza Pontes e Gustavo Lopes Duarte Oliveira pelo trabalho conjunto em parte da tese.

À minha mãe e meu pai Nielse Cristina de Melo Fattori e Cesar André Fattori; minhas irmãs e meu irmão Glaucia Cristina de Melo Fattori Abbade Miguel, Diogo Cesar Fattori e Giovana Nara Barbosa Fattori; meu cunhado André de Melo Fattori Abbade Miguel; às minhas avós e ao meu avô Celso Aparecido Fattori, Irene Caseiro Fattori e Nancy Baracat de Melo.

Às minhas amigas e amigos Diana Barlem dos Santos, Camila Luisa Cremoneze, Caroline Simões Lima, Isabel Akemi, Roberto Guimarães Costa, Mariana Castelli, Milena Correia Cafruni e Bruno Monti de Oliveira pela convivência e companhia indispensáveis para esse período.

E a todas as minhas amigas, todos os meus amigos e todos meus parentes pela paciência durante esse tempo em que não pude estar presente.

E a todos que colaboraram direta ou indiretamente na execução deste trabalho.

"Autogestão! Sem catraca e sem patrão!"

(Povo nas ruas em 2015)

RESUMO

No mercado global e digital, as empresas são desafiadas a encontrar caminhos inovadores para atender o aumento da pressão competitiva. A competição é uma das formas de interação das organizações, além da colaboração e da cooperação. A cooperação e a colaboração apresentam formas de produzir conjuntamente aumentando o potencial de atendimento das empresas. Os desafios mais encontrados no mercado são: reduzir os custos, sempre assegurar qualidade e personalizar os produtos e serviços. Um fenômeno de negócios comum hoje é a terceirização da manufatura e da logística para fornecedores domésticos e estrangeiros e provedores de serviços. Essa terceirização provoca, intrinsecamente, um espalhamento geográfico da produção em novos centros que oferecem vantagens nos recursos energéticos, matérias-primas e centros de produção de conhecimento. Essa terceirização pode ser realizada também nas formas de colaboração e cooperação. Para isso, as empresas necessitam estabelecer uma forma de confiança entre si. No conceito de empresa virtual, a confiança é amplamente discutida para atingir uma colaboração e/ou cooperação entre empresas. O objetivo deste trabalho é propor e modelar uma ferramenta que atenda as necessidades das empresas para colaboração e/ou cooperação entre elas, considerando suas necessidades de confiança. As empresas aqui são vistas como sistemas produtivos, com suas camadas de gerenciamento de negócios, de acordo com o padrão ANSI/ISA 95. Além disso, um tipo de interpretação da rede de Petri, chamada de rede de Petri produtiva é introduzida como ferramenta para descrever o processo produtivo realizado pelas empresas na forma de workflow. A modelagem dessa arquitetura do sistema produtivo utiliza técnicas de sistemas distribuídos, como a arquitetura orientada a serviços. Além disso, um dos enfoques é das necessidades para o desenvolvimento de novos produtos, que envolve o desafio de personalização. Testes foram realizados para avaliar a proposta de workflow com pessoas de diferentes níveis de conhecimento sobre os processos, sejam de manufatura, sejam de outras áreas. Já a arquitetura proposta foi submetida a um estudo analítico das hipóteses levantadas no ambiente colaborativo.

Palavras-chave: Workflow. Rede de Petri. Empresas virtuais. Colaboração. Cooperação.

ABSTRACT

In global and digital market, companies are challenged to find innovative ways to meet the increased competitive pressure. Competition is one way of interacting for organizations, as well as collaboration and cooperation. Cooperation and collaboration have ways to produce together increasing the potential of companies of meeting demands. Challenges most commonly found on the market are reducing costs, always ensuring quality and customizing products and services. A common business phenomenon today is the outsourcing of manufacturing and logistics for domestic and foreign suppliers and service providers. This outsourcing causes, intrinsically, a geographical spread of production in new centers which offer advantages in energy resources, raw materials and knowledge production centers. This outsourcing can be done also in the ways of collaboration and cooperation. For this, companies need to establish a trust between them. In the concept of virtual enterprise, trust is widely discussed to achieve the collaboration and/or cooperation between companies. The goal of this work is proposing and modeling a tool that meets the needs of companies for collaboration among them, considering their trust needs. Companies are understood as productive systems, in here, with their business process management layers, according to the standard ANSI/ISA 95. In addition, a kind of interpretation of Petri net, called productive Petri net is introduced as a tool to describe the production process performed by companies as workflow. The modeling of this architecture of productive system uses techniques of distributed systems, such as service-oriented architecture. Furthermore, one of the approaches is the need for developing new products, which involves the customization challenge. Tests were conducted to evaluate the proposal of workflow with people from different levels of knowledge about the processes, both manufacturing and other areas. And the proposed architecture was studied analytically with the hypotheses elaborated from the collaborative environment.

Keywords: Workflow. Petri net. Virtual enterprises. Collaboration. Cooperation

LISTA DE ILUSTRAÇÕES

Figura 1	Representação da demanda, do ambiente de trabalho, de suas necessidades e de suas capacidades	21
Figura 2	Ciclo de vida para desenvolvimento da pesquisa	26
Figura 3	Ciclo de desenvolvimento considerado nesta pesquisa	26
Figura 4	Modelo hierárquico de controle para empresas de manufatura	29
Figura 5	Estrutura tradicional de negócio	32
Figura 6	Representação de uma EV	34
Figura 7	Etapas de criação de uma EV	37
Figura 8	Arquitetura de informação do NIIP	40
Figura 9	Arquitetura de informação do E-DREAM	44
Figura 10	Estrutura da VECCF	45
Figura 11	Arquitetura distribuída SODA	49
Grafo 1	Rede de Petri	51
Grafo 2	Portas habilitadoras e inibidoras	53
Grafo 3	Arco de sinal de saída	53
Grafo 4	Elementos da SIPN	53
Grafo 5	Elementos específicos da Wf-net	55
Figura 12	Estrutura das informações no arquivo PNML	57
Grafo 6	Elementos do PFS	58
Grafo 7	Exemplo de fluxo principal de informação em PFS	59
Grafo 8	Exemplo de fluxo secundário de informação em PFS	59
Figura 13	Analogia entre a interoperabilidade da informação e a comunicação entre pessoas	61
Figura 14	Um modelo para projeto colaborativo	68
Figura 15	Arquitetura global proposta	73
Figura 16	Arquitetura do ponto de vista das empresas parceiras potencial	74
Quadro 1	Exemplo de valores dos pesos para cálculo da confiança	79
Quadro 2	Exemplo de valores dos indicadores da avaliação dos clientes e da empresa consultada	79
Quadro 3	Exemplo de valores da confiança de empresas cadastradas	79
Figura 17	Exemplo de resumo dos cálculos da confiança da empresa consultada sobre a empresa avaliada	80
Figura 18	Diagrama de casos de uso para o cliente- <i>framework</i>	81

Figura 19	Diagrama de atividades da função “Consultar empresas cadastradas” . . .	82
Figura 20	Diagrama de atividades da função “Consultar andamento da demanda” . . .	82
Figura 21	Diagrama de atividades da função “Avaliar atendimento da demanda” . . .	83
Figura 22	Diagrama de casos de uso para o empresa- <i>framework</i>	86
Figura 23	Diagrama de atividades da função “Cadastrar empresa”	86
Figura 24	Diagrama de atividades da função “Alterar cadastro de empresa” . . .	87
Figura 25	Diagrama de atividades da função “Desativar cadastro de empresa” . . .	87
Figura 26	Diagrama de atividades da função “Consultar confiança da empresa” . .	88
Figura 27	Diagrama de atividades da função “Consultar histórico de processos concluídos”	88
Figura 28	Diagrama de atividades da função “Cadastrar processo concluído” . . .	89
Figura 29	Diagrama de atividades da função “Avaliar processo concluído”	89
Figura 30	Diagrama de atividades da função “Atualizar status do processo em andamento”	89
Figura 31	Diagrama de atividades da função “Alterar avaliação de processo con- cluído”	90
Figura 32	Diagrama de atividades da função “Cadastrar projeto”	90
Figura 33	Diagrama de atividades da função “Alterar projeto”	91
Figura 34	Diagrama de atividades da função “Consultar projeto”	91
Figura 35	Diagrama de casos de uso para o cliente-empresa	93
Figura 36	Diagrama de atividades da função “Consultar proposta”	93
Figura 37	Diagrama de atividades da função “Negociar proposta acordada” . . .	94
Figura 38	Diagrama de atividades da função “Confirmar proposta acordada” . . .	94
Figura 39	Diagrama de atividades da função “Negar proposta acordada”	95
Figura 40	Diagrama de atividades da função “Solicitar P/S”	95
Figura 41	Diagrama de atividades da função “Alterar solicitação de P/S”	96
Figura 42	Diagrama de atividades da função “Remover solicitação de P/S” . . .	96
Figura 43	Diagrama de casos de uso para o empresa-empresa	98
Figura 44	Diagrama de atividades da função “Consultar andamento da atividade” .	98
Figura 45	Representação das camadas de controle de um processo produtivo de manufatura	102
Grafo 9	Modelo de comando atômico	103
Grafo 10	Modelo de instrução que pode ser executada	105
Grafo 11	Modelo simplificado de instrução que pode ser executada	105
Grafo 12	Modelo de caixa de instruções	106

Grafo 13	Modelo simplificado da caixa de instruções	107
Grafo 14	Representação da característica de processos de <i>workflow</i> em sequência em RdPP	111
Grafo 15	Outras representações da característica de processos de <i>workflow</i> em sequência	111
Grafo 16	Representação da característica de processos de <i>workflow</i> em paralelo em RdPP	112
Grafo 17	Outras representações da característica de processos de <i>workflow</i> em paralelo	112
Grafo 18	Representação da característica de processos de <i>workflow</i> em conflito em RdPP	114
Grafo 19	Outras representações da característica de processos de <i>workflow</i> em conflito	114
Grafo 20	Representação da característica de processos de <i>workflow</i> em sequência arbitrária em RdPP	115
Grafo 21	Outras representações da característica de processos de <i>workflow</i> em sequência arbitrária	116
Figura 46	Estrutura das informações no arquivo PNML adotada para a RdPP . .	117
Figura 47	Peça padrão para o teste em manufatura	121
Figura 48	Instrução proposta para torneamento externo	121
Figura 49	Instrução proposta para furo	121
Figura 50	Instrução proposta para corte de canal	122
Figura 51	Instrução proposta para torneamento cônico pela direita	122
Figura 52	Instrução proposta para torneamento cônico pela esquerda	122
Figura 53	Peça proposta para os técnicos de máquinas avaliarem a RdPP	123
Figura 54	Traço externo da peça	123
Figura 55	Cotas do traço externo para a peça com $x \in [0, L0]$	124
Figura 56	Primeiro corte de torneamento externo da peça	124
Figura 57	Segundo corte de torneamento externo da peça	125
Figura 58	Primeiro corte de canal da peça	125
Figura 59	Segundo corte de canal da peça	126
Figura 60	Traço interno da peça	127
Figura 61	Peças inicial e final com indicações de usinagem	127
Figura 62	Usinagem da peça inicial separada em partes	127
Grafo 22	Modelo do <i>workflow</i> em RdPP para a peça proposta	128

Figura 63	Esboço da peça gerado pelos operadores de máquina	129
Quadro 4	Gerador do código G	131
Grafo 23	Modelo em Wf-net para o <i>workflow</i> utilizado no teste	134
Figura 64	Sequência de tarefas da inspeção diária de poços com BM ou BCP . .	136
Grafo 24	Primeira representação em PFS da geração do <i>workflow</i> em RdPP a partir do modelo da Figura 64	137
Grafo 25	Representação em PFS da geração do <i>workflow</i> em RdPP a partir do modelo da Figura 64, adaptada aos requisitos de <i>workflow</i>	137
Grafo 26	Representação em PFS da geração do <i>workflow</i> em RdPP a partir do modelo da Figura 64, fusão de atividades	138
Grafo 27	Representação em PFS da geração do <i>workflow</i> em RdPP a partir do modelo da Figura 64, fusão de trechos	138
Grafo 28	Representação em PFS da geração do <i>workflow</i> em RdPP a partir do modelo da Figura 64, divisão do arco	139
Grafo 29	Representação em PFS da geração do <i>workflow</i> em RdPP a partir do modelo da Figura 64, nova aplicação da divisão de arco, fusão de atividade e fusão de trechos	139
Grafo 30	Representação final do <i>workflow</i> do modelo da Figura 64 em RdPP . .	142
Figura 65	Sequência de tarefas da inspeção diária de poços com BCS	143
Grafo 31	Representação final do <i>workflow</i> do modelo da Figura 65 em RdPP . .	144
Grafo 32	Workflow unificado para a atividade de inspeção de poços produtores de petróleo independente do equipamento utilizado	145
Quadro 5	Comparação entre arquiteturas de informação de EV existentes e a arquitetura proposta	147

LISTA DE LISTAGENS

Listagem 1	Código G para a representação sequencial do processo produtivo τ_1 . .	132
Listagem 2	Código G para a representação sequencial do processo produtivo τ_2 . .	133
Listagem 3	Código PNML para o teste com o torno	135

LISTA DE ABREVIATURAS E SIGLAS

CA	Comando Atômico
EDI	<i>Electronic Data Interchange</i>
EV	Empresa Virtual
HTTP	<i>Hyper-text Transfer Protocol</i>
MFG	<i>Mark Flow Graph</i>
NIIP	<i>National Industrial Information Infrastructure Protocols</i>
PFS	<i>Production Flow Schema</i>
PNML	<i>Petri net Markup Language</i>
QoS	<i>Quality of service</i>
RdPP	Rede de Petri Produtiva
RdP	Rede de Petri
SED	Sistema a Eventos Discretos
SI	Sistema de Informação
SIPN	<i>Signal Interpreted Petri net</i>
SOA	<i>Service-Oriented Architecture</i>
SODA	<i>Service-Oriented Database Architecture</i>
SP	Sistema Produtivo
STEP	<i>Standard for the Exchange Product model data</i>
Wf-net	<i>Workflow net</i>
WS	<i>Web Services</i>
WS-CDL	<i>Web Services Choreography Description Language</i>

LISTA DE SÍMBOLOS

PN	Rede de Petri lugar-transição
P	Conjunto de lugares
T	Conjunto de transições
F	Conjunto de arcos orientados
M	Vetor de marcações
C	Vetor de capacidades
σ	Sequência de disparos
R	Função de alcançabilidade
$SIPN$	Rede de Petri de Sinal Interpretado
IS	Conjunto de sinais de entrada
OS	Conjunto de sinais de saída
φ	Função de transição-sinal de entrada
ω	Função de lugar-sinal de saída
ϵ	Lugar fornecedor
θ	Lugar sorvedor
CPN	Rede de Petri colorida
CS	Conjunto de cores
V	Conjunto de variáveis
CF	Função de cores-lugar
G	Função de guarda
A	Função de expressão dos arcos orientados
S	Função de inicialização dos lugares
CA	Conjunto de comandos atômicos
I	Conjunto de instruções
B	Conjunto de caixas de instruções
PPN	Rede de Petri produtiva
τ	Representação sequencial do processo

SUMÁRIO

1 INTRODUÇÃO	18
2 CONCEITOS FUNDAMENTAIS	28
2.1 CAMADAS DE CONTROLE DE EMPRESAS ANSI/ISA 95	28
2.2 EMPRESAS VIRTUAIS E ARQUITETURA ORIENTADA A SERVIÇOS	31
2.2.1 Ciclo de vida da EV	35
2.2.2 Arquiteturas de informação de EVs	39
2.2.2.1 Projetos recentes	44
2.2.3 Arquitetura orientada a serviços	46
2.3 REDE DE PETRI E WORKFLOW	49
2.3.1 Formalização	50
2.3.2 Portas e arcos de sinais de saída	52
2.3.3 Workflow net	54
2.3.4 Rede de Petri Colorida	55
2.3.5 Petri net Markup Language (PNML)	56
2.3.6 Production Flow Schema (PFS)	58
2.4 SÍNTESE DO CAPÍTULO	59
3 ATRIBUIÇÕES DE EMPRESAS NO AMBIENTE COLABORATIVO	60
3.1 REQUISITOS DO AMBIENTE COLABORATIVO	60
3.1.1 Universalização da informação	60
3.1.2 Distribuição da informação	64
3.2 REQUISITO DO CONTROLE DE SISTEMAS	65
3.2.1 Sub-contratação em tempo real	65
3.3 ATRIBUIÇÕES DE SISTEMAS COLABORATIVOS	66
3.3.1 Confiança entre participantes	66
3.3.2 Desenvolvimento de produtos personalizados	67
3.4 ANÁLISE DAS HIPÓTESES NAS FERRAMENTAS DE EVS	68
3.5 SÍNTESE DO CAPÍTULO	69
4 MODELAGEM DA ARQUITETURA DE INFORMAÇÃO PARA EVS	71
4.1 ARQUITETURA DE INFORMAÇÃO PROPOSTA DO SISTEMA	72
4.2 CÁLCULO DA CONFIANÇA	75
4.3 CLIENTE-FRAMEWORK	80
4.4 EMPRESA-FRAMEWORK	83

4.5	CLIENTE-EMPRESA	92
4.6	EMPRESA-EMPRESA	97
4.7	ANÁLISE DAS HIPÓTESES NA ARQUITETURA	99
4.8	SÍNTESE DO CAPÍTULO	99
5	REDE DE PETRI PRODUTIVA	101
5.1	ABORDAGEM CONCEITUAL	101
5.2	CAMADA DE EQUIPAMENTOS E CONTROLADORES	102
5.3	CAMADA DE CONTROLE SUPERVISÓRIO	103
5.4	CAMADA DE CONTROLE COLABORATIVO	105
5.5	CARACTERÍSTICAS DE PROCESSOS DE WORKFLOW	109
5.5.1	Representação sequencial do processo	109
5.5.2	Sequência	110
5.5.3	Paralelismo	111
5.5.4	Conflito	113
5.5.5	Sequência Arbitrária	114
5.6	ABORDAGEM DE INFORMAÇÃO	116
5.7	SÍNTESE DO CAPÍTULO	117
6	RESULTADOS EXPERIMENTAIS	119
6.1	TESTE SOBRE A HIPÓTESE DE SIMPLICIDADE DO WORKFLOW COM USUÁ- RIOS DE MÉDIO CONHECIMENTO	119
6.2	TESTE DE APLICAÇÃO DA FERRAMENTA DE WORKFLOW EM UMA EM- PRESA REAL	135
6.3	ANÁLISE DOS RESULTADOS	141
7	CONCLUSÕES	146
	Referências	149
	Anexo A – PPNet Tool	157
	Anexo B – Percepção do ambiente colaborativo em um nicho de mercado	194
	Anexo C – Comparação entre a rede de Petri produtiva e a rede de Petri colorida para workflow	200

1 INTRODUÇÃO

Uma das mais comuns disputas ou contradições nas sociedades envolve o valor do trabalho. O valor do trabalho é definido como o ganho que se pode obter (material ou não) pelo empenho de energia e conhecimento para realização de tarefas, em geral associadas a um emprego. O valor do trabalho não depende apenas da quantidade de energia consumida e conhecimento pelo trabalhador para realizá-lo, mas também uma série de elementos históricos que levam a execução desse trabalho. Por exemplo, segundo Kropotkin (2011), apesar de construir uma edificação, o trabalhador não pode se dizer o único responsável por esse trabalho e lucrar com o uso dessa edificação, uma vez que outros trabalhadores já construíram estradas para acessar essa edificação, ou máquinas que ajudaram a levantá-la, ou mesmo operaram no mesmo solo dando a ele a resistência que ele possui hoje. Com isso, Kropotkin (2011) tenta dizer que o valor do trabalho não é algo que possa ser isolado apenas de sua função, pois por trás desse mesmo há um conjunto de outros trabalhos que já foram realizados e que o permitiram de ser executado. Para organizações de trabalho, isto é, conjunto de trabalhadores com equipamentos e outros recursos, assim como as empresas, o conceito do valor do trabalho é o mesmo. Em cada sistema econômico e social, há uma forma de se medir, então, esse valor do trabalho. No socialismo comunista de Karl Marx (STEIN, 2009) o valor do trabalho seria medido de acordo com o histórico de trabalho. No comunismo anarquista de Nestor Makhno (MHAKNO et al., 1926) o valor do trabalho seria medido de acordo com as necessidades específicas do trabalhador. Já no regime capitalista atual, o valor do trabalho é medido de acordo com as regras livres do mercado, composto por empresas. Com a globalização e o avanço das tecnologias de comunicação, esse mercado tornou-se global e digital (CAO; HOFFMAN, 2011).

No mercado global e digital, as empresas são desafiadas a encontrar caminhos inovadores para atender o aumento da pressão competitiva (CAO; HOFFMAN, 2011; HUSPENINOVÁ, 2013; JASTI; KODALI, 2015). Essa pressão competitiva surge no mercado, dentro do capitalismo, como uma forma de se autorregular o valor do trabalho das organizações (ou empresas) utilizando a competição como critério. A competição é uma das formas de interação das organizações, além da colaboração e da cooperação. A competição é a interação em que há uma disputa entre as organizações para a produção. A cooperação é a interação de mútuo acordo para trabalho conjunto, mas não passa disso. A colaboração é a interação em que todo o desenvolvimento e execução são feitos em conjunto (POLENSKE, 2004).

A escolha da forma de interação entre as empresas afeta as estratégias de mudanças em suas plantas produtivas, pois ao optarem por cooperar ou colaborar uma com as outras, as empresas podem reduzir a necessidade de novos equipamentos e/ou pessoal especializado para

atender uma parte de uma dada demanda que essas não sejam capazes de atender sozinhas. As mudanças das plantas produtivas que incluem compra de equipamentos e/ou contratação de pessoal especializado, na maioria das vezes, não são interessantes por acarretarem em maior custo para empresa e tornarem as oportunidades de negócios, na forma de demandas, menos interessantes para as empresas de pequeno e médio porte (WANG; YUNG; IP, 2001).

Apesar dos desafios crescerem constantemente, pode-se dizer que as três formas de interação acima citadas são exploradas para atender os objetivos principais que continuam os mesmos: reduzir os custos, aumentar as vendas e sempre assegurar (ou melhorar quando possível) a qualidade dos produtos e dos serviços (LEITÃO, 2009). Um fenômeno de negócios comum hoje é a subcontratação da manufatura e da logística para fornecedores domésticos e estrangeiros e provedores de serviços. Essa subcontratação provoca, intrinsecamente, um espalhamento geográfico da produção em novos centros que oferecem vantagens nos recursos energéticos, matérias-primas e centros de produção de conhecimento (CHEN; CHIN; WANG, 2011). Além disso, há uma grande demanda de mercado por produtos personalizados (ZENG; LI; ZHU, 2006). A própria inovação tecnológica inspira mudanças na demanda a cada ciclo produtivo em que um novo conhecimento ou funcionalidade pode ser agregado ao produto ou ao serviço (PIRES; PUTNIK; AVILA, 2012). Essa característica de mudanças inesperadas nas demandas do mercado o torna mais dinâmico e exige das empresas novas estratégias para continuarem competitivas (WANG; WONG; WANG, 2012).

Segundo Chen, Chin e Wang (2011), no começo da década de 1990, a manufatura ágil foi proposta para atender, na rápida globalização, a diversidade de demanda dos consumidores. Porém o espalhamento geográfico dos sistemas produtivos dificultou o aprimoramento do conceito, que demandava uma concentração da produção. As melhorias nos processos de manufatura também requeriam novos níveis de flexibilidade. A produção com estrutura tradicional, centralizada em uma empresa, não era mais compatível com as inovações exigidas pela demanda (HORSTMANN et al., 1997).

Em Jasti e Kodali (2015), é feita uma revisão de literatura completa sobre os conceitos de produção enxuta e de manufatura ágil. Os conceitos de redução de custos passam desde o *just-in-time*, pela redução do estoque e produção no tempo certo, até a própria colaboração.

Nesse cenário, as organizações estão inclinadas a focar-se em suas principais competências (que é uma adaptação das empresas pós-produção enxuta e também pode ser chamada de empresa enxuta) e dependerem de outras para complementar suas habilidades e serviços por cooperação, colaboração e compartilhamento de conhecimento umas com as outras (WANG; WONG; WANG, 2012). Segundo Jarimo e Salo (2009), quando empresas buscam, individualmente, ganho de eficiência se focando nas suas principais competências e subcontratando outras para as operações não principais, o número e a complexidade das transações inter firmas

crecem. Essa mudança não impede que as organizações conservem sua estrutura tradicional de produção, ou seja, a negociação direta entre empresa e cliente, sem necessidade de colaboração com outras organizações, tornando a colaboração para a organização como uma alternativa para aumentar a diversidade de sua produção.

Em Huspeninová (2013), a colaboração e cooperação são tratadas como redes de negócios e uma aplicação direta desse conceito é a empresa virtual (EV). No trabalho citado, as redes de negócios com diferentes níveis de relacionamentos podem compartilhar recursos, atividades e reduzir os custos de risco.

Segundo Pires, Putnik e Avila (2012), EV é uma rede de trabalho (ou de negócios) temporária entre empresas independentes, diversas e, possivelmente, dispersas geograficamente, em que uma empresa recebe uma oportunidade de negócios que ela não pode atender unicamente por suas competências e busca as competências que lhe faltam nessa oportunidade em outras empresas, procurando assim empresas colaboradoras. A característica temporária da EV envolve o tempo de duração da produção do produto ou execução do serviço, uma vez que essa é um acordo entre empresas para atender uma oportunidade de negócio específica. As empresas que formam a EV compartilham suas competências, que podem ser suas habilidades, seus planejamentos, seus equipamentos, seus recursos humanos, suas matérias primas, suas informações, seus custos e seus benefícios devidamente combinados para atender a oportunidade de negócios surgida e para agregar valor de mercado para os produtos e serviços, em um acordo de duração temporária, tornando as relações entre empresas mais dinâmicas (IP et al., 2003; WANG; WONG; WANG, 2012). Dessa forma, as empresas podem não possuir as competências necessárias para atender uma demanda, mas a soma das competências das empresas é capaz de atendê-la (ZENG; LI; ZHU, 2006).

As EVs utilizam ferramentas baseadas em rede de comunicação, principalmente via internet, para trocar informações e fazer sua negociação, como a arquitetura orientada a serviços (SOA – *Service-Oriented Architecture*) que aborda as competências de uma empresa como serviços prestados por essa e a composição de serviços é equivalente a um processo produtivo para atender uma oportunidade de negócios. Nas pesquisas de EVs, são propostas diferentes ferramentas que atendem problemas específicos e um conjunto estruturado dessas ferramentas compõem a arquitetura de informação das EVs. As ferramentas de EVs auxiliam as fases do desenvolvimento do ciclo de vida dela em quatro etapas: criação, operação, evolução e dissolução (TAO et al., 2012; WANG; WONG; WANG, 2012).

Nessa divisão do trabalho das organizações em competências toma-se o cuidado de garantir que a todas essas organizações juntas, formando um ambiente de trabalho, possuam capacidade de atender toda a demanda do mercado. Caso contrário, faz-se necessário adicionar outras organizações com as competências necessárias para complementar as já existentes. A

Figura 1 ilustra essa necessidade de adicionar empresas para a colaboração.

Figura 1 - Representação da demanda, do ambiente de trabalho, de suas necessidades e de suas capacidades

Em grande parte da literatura, as EVs são acordos entre pequenas e médias empresas. Mas existem exemplos de grandes empresas que adotaram, em alguns casos, a EV para melhorar a qualidade de seus produtos, como: a Apple Computer que junto com a Sony estabeleceram algumas cooperações para a manufatura dos notebooks PowerBook; a IBM, a Apple Computer e a Motorola que fizeram uma parceria para desenvolver um sistema operacional e um microprocessador para a nova geração de computadores; a Dell Computer que junto com os parceiros de sua cadeia de suprimentos se aproximou de seus clientes pelo comércio eletrônico (CAO; HOFFMAN, 2011).

Essa divisão do trabalho das organizações com objetivo na colaboração e cooperação com competências específicas para atender as demandas de mercado global e digital aqui é chamado de ambiente colaborativo.

A colaboração entre empresas se dá principalmente em duas formas: por sub-contratação ou por terceirização. A sub-contratação é uma forma de colaboração em que cada empresa não tem domínio total das capacidades para atender a demanda, o processo não é necessariamente de responsabilidade da empresa e uma empresa escolhe outra empresa para atender parte do processo como forma de colaboração. A terceirização é uma forma de colaboração em que o processo está nas capacidades da empresa, o processo é de responsabilidade da empresa e a empresa escolhe outra empresa para atender parte do processo por estratégia de redução de custo (INVESTOPEDIA, s.d.). Algumas empresas utilizam da terceirização como forma de reduzir encargos em legislações trabalhistas. Pela complexidade do tema, tanto do ponto de vista social como do ponto de vista econômico, essa tipo de estratégia não é considerada no escopo dessa pesquisa.

Por oferecer maior diversidade nas combinações de competências para a produção, entende-

se que o ambiente colaborativo cria novos tipos de cooperação e de colaboração entre as empresas, variado entre os níveis de cooperação e colaboração e de duração dessas (PIRES; PUTNIK; AVILA, 2012). A maior diversidade de combinações de competências gera também maior diversidade de produtos e serviços oferecidos pelas empresas, atendendo as condições de produtos personalizados pelos clientes. Ao focarem em suas principais competências, essas empresas são responsáveis por aprimorar os processos que estão dentro de suas competências: baixando custos, elevando qualidade, reduzindo tempo de entrega e outros (ZENG; LI; ZHU, 2006).

Nesse ambiente, também, a logística e o transporte têm papel fundamental para permitir que os processos produtivos realizados por empresas geograficamente dispersas sejam viáveis do ponto de vista econômico. Outro ponto fundamental é a confiança entre as empresas, uma vez que essas não são necessariamente parceiras prévias e ampliar a possibilidade de parceiras em colaboração e cooperação aumenta a capacidade de atender diferentes demanda. Essa confiança entre empresas é uma das condições necessárias para garantir tanto a colaboração quanto a cooperação (POLENSKE, 2004). Pelas mudanças constantes do mercado, é necessário que os acordos entre as empresas sejam mais dinâmicos (SARI; SEN; KILIC, 2008). Para atingir uma colaboração efetiva, é necessário que as empresas incluídas no ambiente colaborativo troquem mensagens em linguagens que podem ser inequivocadamente compreendidas por todas (LIU; ZHANG, 2011). A necessidade dessa compreensão inequívoca das informações é fundamental para integrar as empresas, com suas capacidades específicas, nessa colaboração.

Outras abordagens encontradas abstraem o conceito da empresa e abordam a colaboração no plano do meio físico, como o paradigma de *internet of things*, que junta a internet com os equipamentos, e o de manufatura nas nuvens, que aborda a manufatura como orientada a serviço (CHO; SWAMI; CHEN, 2011). Essas abordagens dependem de uma automação plena dos processos de negócios, que não é uma realidade tão tangível para empresas de pequeno e médio porte. Outro problema é que a definição do processo de negócio depende da disponibilidade de serviços oferecidos e nessas abordagens, as atuações específicas dos equipamentos (no chão de fábrica) ampliam a diversidade desses serviços. Com isso, o projeto de novos produtos torna-se uma etapa de maior complexidade. Por fim, outra necessidade na colaboração (que será discutida mais adiante no trabalho) é a confiança. Nesse tema, quanto maior a heterogeneidade (ou a distância entre as atividades realizadas por cada entidade), maior a dificuldade de dimensionar a confiança, uma vez que uma das formas de determinar a confiança das entidades é conhecer a atividade e como esta é realizada.

O ambiente colaborativo aqui descrito tem uma forte presença da heterogeneidade de processos, seja pela presença de vários ramos de atividade compostos em uma execução, seja pela presença de várias empresas aplicadas aos mesmos ramos de atividade, implementando técnicas

e ferramentas diferentes para as mesmas aplicações. Esse ambiente pode ser também considerado um ambiente heterogêneo, em que se tem equipamentos e recursos muito diversificados em funções, e um dos problemas que surge em ambientes com essa diversidade é necessidade de definir linguagens padronizadas para descrever precisamente e detalhadamente os processos produtivos que são exigidos das empresas, para que essas possam cooperar e/ou colaborar entre si. Ao mesmo tempo, devido a diferentes equipamentos serem capazes de executar as mesmas atividades, as linguagens que descrevem processos produtivos devem também representá-los em diferentes níveis (do conceitual ao detalhado), para que independente do equipamento, o resultado seja o mesmo para a mesma representação (CALABRESE; CELENTANO, 2007).

Dessa forma, o conceito de processos ou *workflows* recebe atenção específica, tanto do ponto de vista de interpretação (LIU et al., 2002), quanto do ponto de vista de projeto (CHOI; PARK; LEE, 2002). Os *workflows* podem orientar parcerias nas colaborações por descreverem as capacidades necessárias para as empresas atenderem uma certa demanda.

A abordagem de interpretar uma empresa como um sistema produtivo (SP) e prestador de um serviço específico, procurando outros SPs (ou empresas) para se integrarem em uma colaboração foi feita em pesquisas, como Souit et al. (2013) e Melo, Morales e Miyagi (2008). Os SPs são definidos como sendo aqueles que realizam processos produtivos utilizando recursos materiais, equipamentos, recursos humanos, informações e outras entidades para a produção de bens ou realização de serviços (VILLANI; MIYAGI; VALETTE, 2007). Esses sistemas representam uma grande parcela do mercado. Esse tipo de sistema pode ser inserido em uma classe de sistemas cujo comportamento pode ser modelado via estados discretos em que a mudança abrupta de estados é regida pela ocorrência de eventos considerados instantâneos. Sistemas dessa classe são chamados de sistemas a eventos discretos (SEDs), que também podem envolver outros sistemas físicos como sistemas logísticos e sistemas de controle de tráfego (ZHANG et al., 2011). Os *workflows* também podem ser abordados utilizando SEDs, uma vez que uma mudança entre uma atividade em execução e outra a ser executada depende da ocorrência de um evento de observação da conclusão da primeira atividade. Segundo Villani, Miyagi e Valette (2007), cada SP é composto por um sistema físico (que é chamado de planta) e um sistema de controle. Os sistemas físicos e de controle comunicam-se pela troca de informações em que o estado da planta é identificado pelo sistema de controle por sinais derivados de dispositivos de detecção, e que este sistema de controle, concomitantemente, transmite comandos aos dispositivos de atuação da planta para que as metas do SP sejam atendidas. Para esse trabalho, a abordagem de integração de SPs utilizando SEDs será aplicada à formação de EVs, considerando que as empresas dispostas a colaborar e/ou cooperar umas com as outras são entendidas como SPs.

Para formação das EVs, em que se utiliza ferramentas baseadas em rede de comunicação,

como a internet, para trocar informações e fazer sua negociação, envolve-se processos com elevado nível de automação associada, uma vez que as oportunidades de negócios são dinâmicas no ambiente heterogêneo. Uma forma de automatizar esse processo é entendendo a colaboração entre empresas no contexto de prestação de serviços, que na internet pode utilizar tecnologias como a SOA que aborda as competências de uma empresa como serviços prestados por essa e a composição de serviços é equivalente a um processo produtivo para atender uma oportunidade de negócios (TAO et al., 2012; WANG; WONG; WANG, 2012).

Outra vantagem de utilizar as competências das empresas como prestação de serviços em SOA é a presença de técnicas de composição de serviços (que equivale a formação da EV) formalizadas utilizando ferramentas de SEDs. Uma importante ferramenta na área de SEDs para a modelagem e controle é a rede de Petri (RdP). A RdP utiliza grafos que permitem análise em diferentes níveis de abstração (VILLANI; MIYAGI; VALETTE, 2007). A RdP é também amplamente utilizada para modelar sistemas distribuídos, o que a torna também adequada para o espelhamento geográfico das empresas.

As dificuldades encontradas para as empresas nas possibilidades de colaboração estão principalmente ligadas à confiança entre empresas e à disponibilidade da informação, tanto do ponto de vista de direitos de acesso a ela, quanto do ponto de vista de compreensão das informações.

As propostas existentes hoje na base de dados do *ISI Web of Science*, *Scopus* e *Scielo* ainda não contemplam todas as necessidades originadas das dificuldades anteriores, priorizando a disponibilidade da informação no direito de acesso. Do levantamento realizado, as soluções propostas foram na forma de arquiteturas de informação de EVs e incluem: o consórcio *NIIIP* () nos Estados Unidos da América, com uma arquitetura com estrutura aberta (por utilizar ferramentas abertas em CORBA e em camadas para as trocas de informações entre empresas) (HORSTMANN et al., 1997); a arquitetura *PRODNET II* na União Europeia, com funcionalidades no projeto do produto, utilizando o *STEP* () e o *EDI* () para trocas de dados técnicos de produtos e diferentes níveis de acesso às informações entre as empresas envolvidas (GARITA; AFSARMANESH; HERTZBERGER, 2001); e o *E-DREAM*, da China, que é proposto como uma arquitetura para trabalho virtual, para implementar a manufatura ágil. Os trabalhos mais recentes de EVs se focam em partes do ciclo de vida delas, principalmente na criação delas (TANG et al., 2002).

Essas soluções apresentadas oferecem certo grau de autonomia para as empresas, mas ainda centralizam muitas das funções necessárias para a colaboração em serviços web específicos para coordenação das negociações. Dessa forma, as empresas não possuem total independência para elaborar seus contratos. Essa centralização é percebida ao observar que nas arquiteturas de informação deve invariavelmente transitar por dentro de *frameworks* propostos pelas próprias

arquiteturas.

Ao mesmo tempo, a inexistência de um conjunto de ferramentas que possa ser utilizado pelas empresas, força-as a serem possuidoras de seus próprios SIs (algo que possui um custo elevado e pode ser desvantajoso para algumas empresas de pequeno e médio portes).

Uma solução para garantir autonomia das empresas e oferecer serviços que, normalmente, estão contidos em SIs, é utilizar, na arquitetura, um *framework* de apoio, em que as funções disponíveis nesse *framework* sejam as mesmas que possam ser encontradas nos SIs das empresas que não utilizam o *framework*. Ou seja, o *framework* de apoio não possui nenhuma funcionalidade que não possa estar presente nos SIs das empresas.

OBJETIVOS

Objetivo geral

Propor uma ferramenta de apoio aos SPs para permitir a colaboração e cooperação utilizando a internet no ambiente colaborativo pelo conceito de empresa virtual.

Objetivos específicos

- Propor uma ferramenta de representação de *workflow* que suporte padronização de sintaxe e semântica;
- A representação de *workflow* deve ser simples para a compreensão de operadores humanos;
- Propor uma técnica para determinar confiança das empresas.

METODOLOGIA CIENTÍFICA

A Figura 2 sintetiza o método de desenvolvimento de pesquisas, citada em Jensen (1992) e que também foi adotada no presente trabalho.

Esta abordagem evidencia a forte relação entre teoria, ferramentas e aplicações, estando os três aspectos integrados no processo de desenvolvimento do presente trabalho e contribuindo sinergeticamente nos processos associados a cada um deles.

É uma abordagem de engenharia que, considerando os aspectos formais associados à definição do modelo e dos métodos de análise aplicáveis, utiliza as aplicações como motor de desenvolvimento para ferramentas de suporte à aplicação prática considerada.

Figura 2 - Ciclo de vida para desenvolvimento da pesquisa

Fonte: (JENSEN, 1992)

Uma caracterização tripartida, como a proposta em (JENSEN, 1992), é também seguida nas metodologias de projeto de sistemas.

Também neste trabalho, os três aspectos identificados por Jensen (1992) foram conduzidos simultaneamente, condicionando-se mutuamente. Os desenvolvimentos nas três áreas identificadas beneficiaram-se das sinergias resultantes das atividades em cada uma das outras duas áreas.

A Figura 3 sintetiza as principais referências do presente trabalho em termos da trilogia apresentada.

Figura 3 - Ciclo de desenvolvimento considerado nesta pesquisa

ORGANIZAÇÃO DO TRABALHO

A organização adotada para o presente texto é a seguinte:

- No capítulo 2, os conceitos fundamentais são apresentados. Alguns dos conceitos são considerados na formulação das hipóteses (como os conceitos de desenvolvimento de no-

vos produtos e de gerenciamento de produção no nível de empresa). Outros conceitos são considerados na definição do escopo do trabalho (como o conceito de empresa virtual) e para modelagem e análise dos processos (como os conceitos de rede de Petri e suas extensões) e para o desenvolvimento de ferramentas. Também são apresentadas as ferramentas de outros trabalhos que se aplicam ao mesmo escopo;

- No capítulo 3, são apresentadas as necessidades, requisitos ou atribuições aplicadas aos elementos que fazem parte do escopo do trabalho (as empresas, as empresas virtuais, a percepção das empresas sobre o mercado). Com base nesse levantamento, é que são definidas as hipóteses que são aqui consideradas a tese;
- No capítulo 4, é descrita a modelagem da ferramenta de arquitetura orientada a serviços, com técnicas de modelagem de sistemas, como os diagramas de casos de uso e de atividades do UML. As hipóteses são avaliadas de maneira analítica na ferramenta modelada;
- No capítulo 5, a partir do conceito de rede de Petri, a proposta de descrição de workflow é introduzida;
- No capítulo 6, são mostrados os testes realizados sobre as hipóteses com base nos conceitos adotados e nas ferramentas desenvolvidas;
- No capítulo 7, são apresentadas as conclusões desta tese;
- No anexo 7, é apresentada a modelagem em UML da ferramenta de *software* desenvolvida para projeto de *workflow* utilizando o conceito de rede de Petri;
- No anexo 7, é apresentada uma pesquisa realizada com empresas de um nicho de mercado para identificar a percepção de seus administradores sobre as características aqui levantadas para o ambiente colaborativo;
- No anexo 7, é apresentado um teste de comparação entre modelos de workflow em rede de Petri.

2 CONCEITOS FUNDAMENTAIS

Os conceitos aqui apresentados são fundamentais para descrição do escopo em que se aplica o trabalho, formulação das hipóteses, modelagem e análises das hipóteses e testes, desenvolvimento das ferramentas aplicadas ao teste, verificação analítica de algumas hipóteses e comparação da ferramenta proposta com outras ferramentas existentes aplicadas ao mesmo escopo de trabalho.

Na seção 2.1, é apresentado, primeiramente, o conceito de organização das empresas em suas camadas de negócios dentro da ANSI/ISA 95. Ainda nessa seção, é apresentado o conceito de desenvolvimento de novos produtos. Esses dois conceitos são apresentados conjuntamente por serem complementares nas atividades realizadas nas camadas de negócios das empresas.

Na seção 2.2, é apresentado o conceito de empresa virtual, aplicado em colaboração e cooperação entre empresas. Destes conceitos, são extraídas as soluções de ferramentas aplicadas ao trabalho. Ainda na mesma seção é apresentado o conceito de arquitetura orientada a serviços que é utilizada para modelagem e análise das hipóteses e verificação analítica.

Na seção 2.3, são apresentados os conceitos de rede de Petri e suas extensões que são aqui exploradas para modelagem e análise dos processos e para desenvolvimento das ferramentas.

2.1 CAMADAS DE CONTROLE DE EMPRESAS ANSI/ISA 95

No final do século passado iniciou uma preocupação dos sistemas de manufatura em se integrar com os sistemas de controle das empresas (AMERICAN NATIONAL STANDARD, 2010), visando:

- reduzir os tempos para os consumidores alcançarem novos produtos;
- permitir aos vendedores suprir ferramentas apropriadas para implementar a integração do sistema de controle ao sistema da empresa;
- permitir aos consumidores identificar melhor suas necessidades;
- reduzir os custos de automatizar os processos de manufatura;
- otimizar as cadeias de suprimento;
- reduzir o ciclo de vida do desenvolvimento.

Assim, um modelo hierárquico de controle foi proposto em 5 níveis, como na Figura 4.

Na Figura 4, o nível 4 é reservado para funções relacionadas a projeto e planejamento geral da produção na planta produtiva, também chamado de nível de negócios. As funções desse

Figura 4 - Modelo hierárquico de controle para empresas de manufatura

Fonte: (AMERICAN NATIONAL STANDARD, 2010)

nível têm maior preocupação com uso de material, entregas, níveis de inventários e outras similares. O nível 3 gerencia produções específicas e recebe do nível 4 o planejamento geral da planta para alocar especificamente os recursos para produção. O nível 3 tem, como função principal, a otimização dos processos produtivos. O nível 2 tem funções de monitoração, controle supervisorio e controle automático dos processos produtivos. O nível 1 tem funções de sensoramento e manipulação de atuadores dos processos produtivos. O nível 0 tem as funções de máquinas.

O principal foco da integração de controles está nas funções de planejamento geral (Nível 4) e de otimização dos processos produtivos (Nível 3). Os demais níveis são específicos para cada produção. Dentro do ambiente colaborativo, o funcionamento interno da planta produtiva, que em geral tem relação com as funções dos níveis 3, 2, 1 e 0, são de responsabilidade exclusiva da empresa e suas funções têm pouco ou nenhum relacionamento com funções de outras empresas. Já o nível 4, possui funções que, se adaptadas ao ambiente colaborativo, podem integrar a produção local da empresa com outros processos produtivos de forma global.

Segundo a American National Standard (2010), as atividades típicas do nível 4 são:

- coletar e manter inventário de uso e disponibilidade de matéria-prima e peças de reposição,

e prover dados para compra de matéria-prima e peças de reposição;

- coletar e manter inventário de uso e disponibilidade de energia global e prover dados para compra de fontes energéticas;
- coletar e manter arquivos de inventário com bens globais em processo e produção;
- coletar e manter arquivos de controle de qualidade e de como eles se relacionam com os requisitos dos clientes;
- coletar e manter arquivos de uso e histórico de maquinário e equipamentos necessários para planejamento de manutenção preventiva e preditiva;
- coletar e manter dados de uso de mão de obra para os departamentos de recursos humanos e de contabilidade;
- estabelecer o agendamento básico de produção da planta;
- modificar o agendamento básico de produção da planta em função de pedidos recebidos, baseado nas mudanças de disponibilidade dos recursos, fontes energéticas, níveis de demanda de potência e requisitos de manutenção;
- desenvolver a agenda de manutenção preventiva e de renovação de equipamentos otimizada em coordenação com o agendamento básico de produção da planta;
- determinar o inventário ótimo de níveis de matéria-prima, fontes energéticas, peças de reposição e bens em processo a cada ponto de armazenagem. Estas funções também incluem planejamento de requisitos de materiais e aquisição de peças de reposição;
- modificar o agendamento básico de produção da planta tanto quanto necessário sempre que ocorram interrupções na produção principal.

Algumas das funções anteriores afetam unicamente o funcionamento interno da empresa, seja ela em uma planta produtiva ou em múltiplas plantas produtivas, como previsto pela American National Standard (2010). As funções de coletar e manter inventários, arquivos e dados, implicam em funções que, do ponto de vista de EVs, definem as capacidades das empresas participantes. Com essas informações, uma empresa é capaz de cadastrar em um banco de dados as capacidades das empresas, isto é, as suas capacidades oferecidas às demais empresas do ambiente colaborativo. As capacidades das empresas são, então, uma compilação das informações dos inventários e são compartilhadas dessa forma (compilada) para outras empresas.

O agendamento básico de produção da planta recebe como entrada os múltiplos pedidos feitos à empresa e nesse é definido a ordem de atendimento dos pedidos para garantir que todos sejam feitos nos prazos determinados pelo solicitante do pedido. No caso do ambiente colaborativo, o solicitante tanto pode ser um cliente da empresa por um produto que essa oferece, quanto pode ser uma outra empresa, no modelo de EV, por uma capacidade que essa oferece. A função de agendamento básico é capaz de determinar se a empresa é capaz de concluir o pedido e qual é a sua proposta para o mesmo. Em sistemas de agendamento, as empresas definem uma faixa de tempo em que o produto ou serviço pode ser concluído e essa faixa de tempo, no ambiente colaborativo, é considerada responsabilidade da empresa atender, ou seja, se a empresa recebe a matéria-prima ou produto parcialmente pronto antes ou exatamente no momento em que se inicia a sua faixa de tempo, ela deve concluir toda a operação determinada a ela antes ou exatamente no momento em que se finaliza a sua faixa de tempo. Essa faixa de tempo também é chamada de janela de tempo.

A mudança no agendamento básico de produção da planta devido a interrupções é uma função que tem o objetivo de remediar falhas ocorridas no processo produtivo. As falhas, normalmente, ocorrem por razões inesperadas ou não preditivas, como o mal funcionamento de um equipamento.

Entre os níveis 3 e 4 do modelo hierárquico, algumas informações precisam ser trocadas entre as funções presentes em cada nível. As principais informações trocadas são:

- **Informação da definição do produto:** que define como deve ser feito o produto;
- **Informação da capacidade de produção:** que define quais recursos estão disponíveis;
- **Informação de agendamento da produção:** que define qual produção será executada em cada instante;
- **Informação do desempenho da produção:** que define a condição atual da produção em andamento.

2.2 EMPRESAS VIRTUAIS E ARQUITETURA ORIENTADA A SERVIÇOS

Duas variáveis importantes da economia global são a oferta e a demanda, sendo uma atribuída a produção e outra a requisição, respectivamente. A disparidade entre esses dois componentes pode gerar diferentes cenários: subprodução, em que a demanda é maior que a oferta, portanto, não há atendimento das necessidades do mercado; superprodução, em que a oferta é maior que a demanda, portanto, há uso excessivo de recursos e força de trabalho. Ambos cenários são prejudiciais, um por promover desigualdades e, outro por utilizar sem

necessidade recursos escassos (STEIN, 2009). A solicitação de um cliente é feita por um pedido a uma empresa, na estrutura de negócio, que retorna ao cliente uma proposta para atender seu pedido, como na Figura 5.

Figura 5 - Estrutura tradicional de negócio

Quando a solicitação de um cliente não é um bem ou serviço anteriormente produzido pela empresa ou por outras empresas no mercado, a solicitação é aqui considerada personalizada ou uma demanda personalizada. A demanda personalizada exige que as empresas reorganizem sua cadeia produtiva (ou *workflow*) para atender a solicitação, por serem essas interessadas em participar da oportunidade de negócio gerada (PIRES; PUTNIK; AVILA, 2012).

Algumas reorganizações da cadeia produtiva de uma empresa podem incluir a compra de novos equipamentos, o que também pode implicar em contratação de pessoal especializado. Outras podem exigir compra de matérias-primas que não faziam parte do escopo de trabalho da empresa. Tais reorganizações não são interessantes para essa pesquisa por acarretarem em maior custo para empresa e tornarem as oportunidades de negócios de solicitações personalizadas menos interessantes para as mesmas, principalmente as empresas de pequeno e médio porte (WANG; YUNG; IP, 2001).

Em algumas condições das empresas, como as anteriormente expostas, o atendimento a solicitação do cliente pode ser transformada em um conjunto de processos específicos que podem ser executados por entidades próprias (que possuem especialidade) para executá-los. Neste caso, torna-se uma estratégia interessante para a empresa que quer atender a solicitação contratar uma ou mais empresas diferentes que possuam especialidades necessárias do conjunto de processos específicos. Assim, a empresa passa a ser responsável por apenas parte do processo produtivo e outras empresas são também responsáveis por partes específicas do processo. Esse é o entendimento sobre a especialização das empresas em suas principais

capacidades (WANG; WONG; WANG, 2012).

A contratação de outras empresas, nesse caso, tem duração determinada pela oportunidade de negócio, pois trata-se de uma solicitação personalizada. Assim que finalizada a oportunidade de negócio, o acordo pode ser rompido e ambas partes se consideram satisfeitas por terem, cada uma, obtido algum tipo de vantagem para realizar operações de sua especialidade (TAO et al., 2012).

O mercado global possui um potencial de clientes com solicitações personalizadas que cresce, constantemente, devido à abertura e desenvolvimento de novos mercados. Além disso, o crescimento das demandas também ocorre com a evolução tecnológica (KROPOTKIN, 2011). Um exemplo disso é o aparelho de telefone celular que não era considerado um item essencial no começo da década de 90 (1,4 milhões de aparelhos em uso no país em 1995 (AGÊNCIA NACIONAL DE TELECOMUNICAÇÕES, 2012)) para se tornar essencial no momento atual da sociedade (261 milhões de aparelhos em uso no país em 2012 (AGÊNCIA NACIONAL DE TELECOMUNICAÇÕES, 2012)). Com isso, as empresas fornecedoras de produtos e serviços associadas a este mercado necessitam atender as solicitações de forma mais eficiente e, para isso, necessitam reduzir os tempos gastos em processos de negociação.

No cenário anterior e baseado no conceito de manufatura ágil, iniciou-se no começo da década de 90 a prospecção da ideia de empresa virtual (EV). Uma EV é entendida como uma empresa comum do ponto de vista de um cliente que consome seus produtos e/ou serviços. Entretanto, internamente, uma EV é um conjunto de empresas com objetivo comum do ponto de vista de mercado.

Uma EV é formada para atender uma solicitação de um cliente, que pode ser uma pessoa física ou uma pessoa jurídica (como empresas e organizações). A solicitação de um cliente é entendida pelo mercado como uma oportunidade de negócio, por ser resultante dessa um bem ou um serviço prestado. A Figura 6 ilustra como pode ser entendida uma EV.

Na Figura 6, as setas vermelhas indicam que a empresa na extremidade da linha sem seta está fazendo uma solicitação de colaboração ou cooperação para a empresa que está na outra extremidade da linha, ou seja, a empresa ABC age como cliente da empresa CAT e da empresa FFF. As setas azuis indicam que a empresa na extremidade da linha sem seta está fazendo uma proposta baseada na solicitação da empresa que está na outra extremidade da linha, ou seja, a empresa CAT e a empresa FFF são prestadoras (ou fornecedoras) para a empresa ABC.

Conceitualmente, uma EV é uma aliança dinâmica temporária de empresas autônomas criada para atender uma oportunidade específica de mercado (que pode ser uma solicitação personalizada de um cliente), ou uma rede de serviços temporária de empresas autônomas em que uma empresa iniciadora de uma EV interage com suas empresas parceiras (tal como uma empresa que contrata outras empresas por suas especialidades) para compartilhar competências

Figura 6 - Representação de uma EV

(ou habilidades), recursos (ou equipamentos e matéria prima), custos e benefícios para atender uma oportunidade de mercado e agregar maior valor nos produtos e serviços. A EV, em princípio, não tem recursos humanos ou inventário de equipamentos. Cada empresa provê suas principais competências em áreas como comércio, engenharia e manufatura para a EV (IP et al., 2003; TAO et al., 2012; WANG; WONG; WANG, 2012).

As EVs não existem fisicamente ou na forma de pessoa jurídica, pois não são efetivamente empresas, entretanto existem na forma de sua colaboração ou cooperação por contratos entre as empresas. Sua existência pode ser explicada em um ciclo de vida que vai desde uma demanda de cliente até a entrega de um produto ou serviço. O ciclo de vida de uma EV é dividido em quatro etapas: criação, operação, evolução e dissolução (CHEN; CHIN; WANG, 2011; TAO et al., 2012). Em alguns trabalhos, como em Sari, Sen e Kilic (2008), o ciclo de vida das EVs não considera a etapa de evolução. Para cada etapa dessas, algumas responsabilidades são atribuídas às empresas envolvidas para garantir que a EV seja eficiente. Uma EV é, então, uma integração de empresas autônomas com competências complementares.

Como as empresas envolvidas na EV são empresas autônomas, diferentes ferramentas podem ser utilizadas por cada empresa para participar de cada etapa do ciclo de vida da EV. Assim, integrar diferentes empresas autônomas também envolve a integração das ferramentas que essas utilizam, pois a integração dessas facilita a coleta e o compartilhamento dos recursos produtivos ou competências, que pode ser realizada mais rapidamente e amplamente, entre

empresas parceiras (TAO et al., 2012). A incompatibilidade entre as ferramentas ou sistemas de processamento de informações e as tecnologias, ainda usuais, de manufatura tem sido um dos maiores inibidores dessa integração. Os avanços na tecnologia da informação permitiram progressos no desenvolvimento automatizado de produtos. Entretanto, a heterogeneidade dos ambientes de computação inibem o progresso da automação, por restringir o compartilhamento de dados dos processos umas com as outras (HORSTMANN et al., 1997).

Desta forma, por causa da natureza dinâmica e flexível de uma EV, são necessários: (i) uma tecnologia de informação adequada para a habilitação de serviços para fazer o gerenciamento da EV; (ii) a integração e inter operação dos processos de negócio e toda cooperação das empresas sustentados por redes de computadores (CHEN; CHIN; WANG, 2011; WANG; WONG; WANG, 2012).

As EVs são vantajosas devido a suas respostas rápidas e competitivas ao mercado em um ambiente de negócios (SARI; SEN; KILIC, 2008).

Algumas pesquisas atuais se preocupam com o ciclo de vida das EVs, em etapas específicas desse, e definem técnicas e atribuições necessárias para garantir e otimizar o funcionamento da EV. Assim o ciclo de vida da EV é uma área indispensável de pesquisa (WANG; WONG; WANG, 2012). A subseção 2.2.1 define o ciclo de vida de uma EV com suas etapas. Outras pesquisas focam em condições para inter operação eficiente com uma infraestrutura de informação, como forma de integrar os sistemas de controle das empresas para a produção numa estrutura de EV, formando um conjunto estruturado de ferramentas em arquiteturas de informação (SARI; SEN; KILIC, 2008). Assim, a subseção 2.2.2 apresenta as arquiteturas existentes na literatura com as especificações do funcionamento dessas.

2.2.1 Ciclo de vida da EV

O ciclo de vida de uma EV pode ser analisado em quatro etapas. A primeira etapa se inicia com o surgimento de uma oportunidade de negócio (CHEN; CHIN; WANG, 2011).

O surgimento de uma oportunidade de negócio é quando um cliente tem uma solicitação. Ao enviar sua solicitação para uma empresa, inicia-se a etapa de criação da EV. A empresa que recebe a solicitação do cliente é dita a empresa principal e é ela que comanda todo o processo para garantir os prazos combinados com o cliente (CHEN; CHIN; WANG, 2011).

A empresa principal deve transformar a solicitação do cliente em uma especificação dos processos de acordo com os tipos de operações necessárias para atender a solicitação (como em processos de manufatura, processos logísticos, processos químicos e outros). A própria empresa principal pode fazer essa transformação ou pode contratar uma empresa especialista no tipo de operação da solicitação para obter a especificação dos processos do produto ou serviço

solicitado pelo cliente. Esta transformação da solicitação em especificação dos processos é uma subetapa da criação da EV chamada de projeto do produto/serviço.

Após obter o a especificação dos processos, a empresa principal define os processos que estão na especificação e que ela se considera competente e tem interesse de atender e procura, num repositório de informações, os contatos de empresas que tenham competência para atender os processos, chamadas de empresas parceiras potenciais (CHEN; CHIN; WANG, 2011).

Com os contatos das empresas parceiras potenciais, a empresa principal inicia uma etapa de negociação entre empresas para obter a melhor proposta para o cliente. A empresa principal utiliza a rede de computadores para fazer a negociação com as empresas parceiras potenciais. Usa-se, para determinar a melhor proposta para o cliente, critérios de custo, prazo de entrega, qualidade e outros em uma subetapa da criação da EV chamada de seleção de parceiros (CHEN; CHIN; WANG, 2011; WANG; YUNG; IP, 2001). Essa etapa é representada pela Figura 7, em que (1) representa a solicitação do cliente em que a informação transmitida é o pedido, (2) inicia a subetapa de transformar a solicitação do cliente em especificação dos processos e a informação transmitida é o pedido sem dados de prazos ou quantidades, (3) finaliza a subetapa de transformar a solicitação do cliente em especificação dos processos e a informação transmitida é a especificação dos processos, (4) envia a especificação de processos para as empresas parceiras potenciais, (5) recebe a proposta das empresas parceiras potenciais incluindo as etapas que essa poderá executar, (6) é a subetapa de seleção de parceiros para enviar uma proposta final para o cliente.

Após a seleção de parceiros e “aceite” da proposta pelo cliente, a empresa principal assina um acordo com as empresas parceiras potenciais selecionadas, também chamadas de empresas parceiras, e assim a EV é finalmente criada para atender a solicitação do cliente (CHEN; CHIN; WANG, 2011).

Após criada a EV, inicia-se a etapa de operação da EV. Nessa etapa, as empresas principal e parceiras, isto é, as empresas envolvidas na EV, devem executar os processos que foram definidos como de sua responsabilidade no acordo entre as empresas que inclui os prazos máximos para conclusão de seus processos. A monitoração da produção é definida no acordo das empresas e é uma subetapa da operação da EV que, no caso de ocorrência de falhas ou imprevistos no processo, gera correções de curso da execução. O sucesso da EV depende de todas as empresas cooperarem como uma unidade sinérgica, pois cada empresa parceira oferece suas principais competências. Isso quer dizer que a vantagem competitiva que pode ser alcançada pela EV depende em quão bem as empresas se completam e suas competências se integram (CHEN; CHIN; WANG, 2011; ZENG; LI; ZHU, 2006).

Quando todos os processos são concluídos e o produto ou serviço é entregue ao cliente, o acordo entre as empresas principal e parceiras não tem mais validade. Então, após a etapa de

Figura 7 - Etapa de criação de uma EV

operação da EV, duas diferentes etapas podem ocorrer. Caso haja uma nova solicitação de um novo cliente, ou do mesmo cliente anterior, as empresas podem rever o acordo assinado anteriormente e refazê-lo para atender a nova solicitação. Tal etapa é chamada de evolução da EV (CHEN; CHIN; WANG, 2011).

A evolução da EV é uma etapa que surgiu em pesquisas recentes para facilitar a subetapa de seleção de parceiros, pois uma informação usada nessa subetapa é a confiança entre as empresas. Caso essa etapa seja realizada, as empresas envolvidas na EV reconfiguram seus requisitos e recursos para atender a nova solicitação (CHEN; CHIN; WANG, 2011; WANG; YUNG; IP, 2001).

Caso não haja uma nova solicitação, ou haja uma solicitação em que as empresas envolvidas na EV não considerem a parceria estabelecida como a ideal para atendê-la, a etapa que se segue é a dissolução da EV. Nesta etapa, as empresas envolvidas na EV realimentam uma base de dados de processos por elas realizados (que auxilia a subetapa de projeto do produto/serviço) e rompem o acordo assinado sem prejuízo para as partes.

Como as EVs surgem para atender demandas específicas e se desfazem quando não há novas

solicitações que possam atender ou quando terminam o atendimento da primeira demanda, o ambiente de mercado em que se desenvolvem as EVs é caracterizado por frequentes reconfigurações não previstas e, relativamente, drásticas. A etapa de criação influencia a estabilidade da aliança. Muitas pesquisas descrevem a etapa de criação baseada somente na seleção de parceiros. Na etapa de criação da EV, um possível conflito entre as competências da empresa principal e das empresas parceiras (ou seja, caso um ou mais processos que podem ser atendido pela empresa principal podem também ser atendido por uma ou mais empresas parceiras potenciais) as empresas podem optar pelo custo ou prazo para escolher qual empresa deve atender o processo (CHEN; CHIN; WANG, 2011; SARI; SEN; KILIC, 2008). Segundo Sari, Sen e Kilic (2008), encontrar os parceiros corretos e estabelecer as condições necessárias para iniciar o processo de colaboração pode consumir recursos e tempo.

A criação de uma EV é uma etapa chave que influencia as etapas seguintes de seu ciclo de vida. Em Talluri e Baker (1996) foi proposta uma abordagem de programação matemática para resolver o problema de solução de parceiros na criação de uma EV, em que fatores de custo, tempo e distância foram considerados sem, necessariamente, analisar todas as combinações de empresas que atendem os processos da solicitação. Wang, Yung e Ip (2001) desenvolveram uma técnica de decisão *fuzzy* com algoritmo genético embarcado para obter a solução do problema de seleção de parceiros com restrição de data de entrega em uma EV, em que subprojetos formam uma rede de precedência. Ip et al. (2003) descreveram e modelaram um problema de seleção de parceiros baseada em risco, e um algoritmo genético baseado em regras foi desenvolvido para solucionar o problema de seleção de parceiros. Zeng, Li e Zhu (2006) provaram que a seleção de parceiros com restrição de data de entrega é um problema NP-completo e um modelo de programação inteira não linear foi estabelecido para este problema de solução de parceiros. Sari, Sen e Kilic (2008) propuseram um método hierárquico analítico para contribuir com a seleção de parceiros, baseado em confiança, em EVs. Jarimo e Salo (2009) estudaram como a seleção de parceiros em uma EV pode ser auxiliada por modelos de programação inteira linear, com fatores de custo de transporte, riscos de déficit de capacidade e dependências inter organizacionais.

A maioria dos trabalhos em EVs fala de alianças entre empresas que possuem diferentes competências. Entretanto essa não é a única possibilidade de aliança. Em Seok e Nof (2014), a colaboração entre empresas abordada é a dita “colaboração interna”, ou seja, entre empresas da mesma área de atuação.

Para Esposito e Evangelista (2014), as EVs devem possuir duas características indispensáveis:

- flexibilidade: as estruturas das empresas precisam ser flexíveis e adaptáveis;

- uso intensivo de tecnologias de informação e comunicação (ICTs) para gerenciamento de conhecimentos: as empresas precisam de ferramentas de ICT para gerenciamento de informação e conhecimento para explorar as relações colaborativas de modo mais eficiente.

Poucas pesquisas em EVs mostram claramente como tratar dessas duas características (ESPOSITO; EVANGELISTA, 2014).

2.2.2 Arquiteturas de informação de EVs

As arquiteturas de informação de EVs apresentam um conjunto de ferramentas e de infraestruturas de informação em ambientes de rede de computadores que auxiliam as empresas nas etapas do ciclo de vida de uma EV.

Essas arquiteturas são as encontradas na literatura (base de dados do *ISI Web of Science*, *Scopus* e *Scielo*) que se aplicam ao escopo de trabalho da tese e serão utilizadas para comparação com a ferramenta proposta quanto às hipóteses que serão levantadas.

As arquiteturas de informação de EVs encontrados na literatura e aqui apresentados são o NIIIP dos Estados Unidos da América, o PRODNET da União Europeia e o E-DREAM da China. Outras duas arquiteturas também serão apresentadas, mas devido o baixo grau de maturidade de suas propostas, essas não serão usadas para efeito de comparação.

National Industrial Information Infrastructure Protocols (NIIIP)

Neste caso, uma das principais preocupações é com a infraestrutura de informação industrial. De acordo com Horstmann et al. (1997), a incompatibilidade entre os sistemas de processamento de informações e as tecnologias usuais de manufatura foi um dos maiores inibidores da manufatura ágil. Os avanços na tecnologia da informação permitiram avanços no desenvolvimento automatizado de produtos. Entretanto, a heterogeneidade dos ambientes de computação inibiu o progresso da automação de processos envolvendo diferentes empresas, por restringir a habilidade das empresas de manufatura de compartilhar os dados dos processos umas com as outras.

O consórcio NIIIP consiste de fornecedores de tecnologia da informação e usuários com um objetivo comum de desenvolver uma arquitetura de informação e prover tecnologias para permitir a evolução das EVs.

O consórcio NIIIP adotou uma arquitetura baseado em padrões industriais emergentes, existentes e de facto (HORSTMANN et al., 1997), como:

- Internet (HTTP, TCP/IP) – para comunicação;

- *Object Management Group CORBA* (OMG-CORBA) – para interoperabilidade de aplicações (programas computacionais);
- *International Standard Organization Standard for the Exchange of Product Model Data* (ISO-STEP) – para compartilhamento de informação; e
- *Workflow* – para gerenciamento de processo e trabalho.

O NIIP é representado na Figura 8. Esta arquitetura define um conjunto de componentes ou serviços que permitem a definição e operação de uma EV. O NIIP foi desenvolvido especificamente para orientar as implementações usando aplicações comerciais de prateleira (HORSTMANN et al., 1997).

Figura 8 - Arquitetura de informação do NIIP

Fonte: Traduzido de (HORSTMANN et al., 1997)

Na Figura 8, “Cliente Humano” representa uma interface oferecida para o cliente realizar seu pedido via *web*. “Cliente *Software*” é uma interface de recebimento e acompanhamento de pedidos por aplicações remotas. Nesta arquitetura, o CORBA é usado para integrar serviços de diferentes naturezas na internet em “Serviços de Objetos” e os pedidos dos clientes, sejam *softwares* ou humanos são descritos pela empresa em linguagem própria utilizando um banco de dados históricos de outros pedidos já atendidos e disponibilizados na internet pelo “Servidor de Objeto”. O “Monitor EV” monitora a operação da EV para os pedidos dos diferentes clientes. Dentro de “Serviços da Empresa Virtual (EV)” há um conjunto de serviços para atender

necessidades específicas como: “Internet”, “Aplicação”, “Negociador” e outros. Quando um pedido chega, os serviços do “Serviços da Empresa Virtual (EV)” traduzem o pedido com base nos bancos de dados históricos recebidos em “Servidor de Objeto”, mas para que a informação seja interpretada independente do modelo do repositório de dados (como modelos relacionais, transacionais, orientados a objetos, etc) usa-se uma ferramenta de tradução chamada “Tradutor de Esquema & Dados”. Após este processo, as empresas principal e parceiras registram as informações em um banco de dados “Base de Conhecimento de EV”, que alimenta também os bancos de dados históricos (HORSTMANN et al., 1997).

PRODNET II

A arquitetura de informação PRODNET II visa principalmente o gerenciamento de informações nas EVs. De acordo com Garita, Afsarmanesh e Hertzberger (2001), as principais necessidades para uma implantação de EV são: ampla variedade de tipos de informações compartilhadas e distribuídas na rede da EV, diferentes níveis de acesso e visibilidade da informação definida pelos nós (que são as empresas parceiras), incorporação e compartilhamento parcial de informações preexistentes, gerenciamento de informação distribuída associada à funcionalidade da EV, e suporte à monitoração e coordenação das atividades da EV.

No PRODNET II, todas as empresas que são parceiras potenciais da EV, são consideradas nós, em que cada nó pode executar atividades diferentes como fornecedor de material, produtor, ou consumidor final de bens ou serviços. Cada nó é composto de um módulo interno, uma camada de cooperação PRODNET (CCP) e um módulo avançado de coordenação de funcionalidades de EV (GARITA; AFSARMANESH; HERTZBERGER, 2001).

O módulo interno do nó consiste dos sistemas internos da empresa, seus processos internos de tomada de decisão (chamado de sistema de controle de planejamento da produção) e outros sistemas de gerenciamento de informação e de engenharia necessários para atender as tarefas locais. O módulo interno é estendido com um módulo de mapeamento, com suporte para todas as interações entre os módulos internos de uma empresa em sua CCP (GARITA; AFSARMANESH; HERTZBERGER, 2001).

A CCP possui funcionalidades necessárias para a interação entre um dado nó e outros nós da EV. Todos compartilhamentos e trocas entre diferentes nós são conduzidos pela CCP (GARITA; AFSARMANESH; HERTZBERGER, 2001).

O módulo de coordenação local executa e controla atividades da CCP (definida como fluxo de planos), de acordo com as políticas da empresa e o comportamento de cooperação desejado. O módulo de coordenação local de uma empresa conduz todos os eventos de cooperação desta empresa com outras empresas, de acordo com as regras específicas definidas por esta empresa

(GARITA; AFSARMANESH; HERTZBERGER, 2001).

Os componentes STEP e EDI inicialmente suportam a troca de dados técnicos de produtos e os relacionados a operação comercial, respectivamente. O módulo de interface com usuários disponibiliza a interação dos usuários finais da empresa com a CCP. A configuração do componente define os elementos necessários e as funcionalidades da empresa dentro da EV em que esta está envolvida (GARITA; AFSARMANESH; HERTZBERGER, 2001).

O sistema de gerenciamento de informação distribuída atende todos os requisitos de gerenciamento de informações da EV e provê uma plataforma de interoperabilidade para trocar informações e cooperativamente gerenciar uma grande quantidade de dados entre os membros da EV (GARITA; AFSARMANESH; HERTZBERGER, 2001).

As funcionalidades de coordenação avançada provêm algumas características adicionais para estender o escopo da CCP, incluindo a monitoração avançada e a coordenação de atividades relacionadas à EV, e comporta ferramentas para busca de parceiros e operações logísticas. O módulo de gerenciamento de processos de negócios distribuídos para monitoração e controle de um processo de negócio distribuído em uma EV é uma das funcionalidades de coordenação avançada (GARITA; AFSARMANESH; HERTZBERGER, 2001).

A interface de comunicação da PRODNET representa o componente que disponibiliza o canal de comunicação de um nó com outros nós da EV. Os protocolos de comunicação implementados permitem a troca de parâmetros de segurança e autenticidade entre os nós (GARITA; AFSARMANESH; HERTZBERGER, 2001).

E-DREAM

O E-DREAM é uma arquitetura de informação que foca principalmente em oferecer uma plataforma reconfigurável para implantação de EVs. De acordo com Tang et al. (2002), o E-DREAM concentra-se no ambiente de trabalho do parceiros da EV com os seguintes objetivos:

- Oferecer plataformas de trabalho flexíveis e reconfiguráveis em que as empresas parceiras podem selecionar seus recursos necessários. Em empresas tradicionais, há um conjunto de recursos no escopo da empresa. Na EV, as empresas parceiras podem configurar dinamicamente sua plataforma de trabalho e os recursos para atender as mudanças das solicitações de mercado. Assim, as empresas parceiras em uma EV são dependentes e independentes. São dependentes por cooperarem umas com as outras para desenvolver o produto e são independentes por poderem praticar de forma completamente autônoma suas políticas.
- Prover esquemas flexíveis e estrutura aberta para criar EVs. As empresas parceiras em

uma EV têm diferentes regras de negócios e infraestruturas de informação. Para criar uma EV, um parceiro deve adaptar sua infraestrutura de informação a arquitetura da EV, assim como compartilhar e trocar dados com outros parceiros. Isso requer que a estrutura seja aberta e fácil de ser adaptada por novas infraestruturas.

- Prover um mecanismo para compartilhamento e gerenciamento de informações distribuídas. Em uma EV, a informação é segmentada e distribuída por lugares físicos distribuídos de empresas parceiras que precisam acessar a informação frequentemente e dinamicamente. O compartilhamento de informação e a comunicação distribuída são essenciais para oferecer o desenvolvimento de produtos ágeis. As empresas parceiras com diferentes níveis de direitos tem diferentes domínios da informação comum como os dados e conhecimento dos produtos. Assim, o E-DREAM oferece o controle de acesso à informação e a segurança dos dados, domínio e responsabilidade.
- Prover métodos de atribuição de recursos aos parceiros. As atividade distribuídas são executadas por diferentes atores usando diferentes recursos. Os recursos atribuídos a uma empresa parceira podem estar dentro de uma única empresa ou entre empresas. A otimização das requisições de recursos e das atribuições necessita de um método de decisão efetivo. As interações e cooperações entre as diversas funções, como empresa principal da EV e empresas parceiras da EV, também necessitam de um mecanismo de coordenação flexível.
- Explorar uma tecnologia de interface para implementar a interoperabilidade das ferramentas de *software* distribuídas. É possível utilizar as soluções baseadas em componentes seguindo a tecnologia de objetos distribuídos. Um componente é um objeto reusável que pode ser integrado em uma aplicação sem precisar saber a forma como esse foi implementado. O CORBA é um das mais conhecidas tecnologias de objetos distribuídos. Ele é um modelo de linguagem neutra que permite a inter operação de objetos, tipos de máquinas e sistemas operacionais. A interface de um objeto CORBA é completamente separada da sua implementação.

A arquitetura de informação do E-DREAM pode ser visto na Figura 9.

A arquitetura do E-DREAM se inicia com uma oportunidade de mercado, ou um pedido, em “Necessidades de Mercado”. Gera-se, pela oportunidade de mercado, uma especificação dos processos com o conhecimento da empresa que recebeu esse pedido. Neste caso, a especificação dos processos é representada por um conjunto de atividades ligadas entre nós. Os nós ligam duas ou mais atividades e a ligação entre os nós define a ordenação das atividades. Dos nós na especificação dos processo, define-se quais são as empresas parceiras que podem atender

Figura 9 - Arquitetura de informação do E-DREAM

Fonte: Traduzido de (TANG et al., 2002)

as capacidades requeridas e são representadas como “Parceiro”. Um serviço de coordenação da produção é feito pela empresa principal e essa rege as atividades das empresas parceiras por “Coordenação”. Para definir se a empresa parceira possui capacidade de atender uma etapa, é necessário verificar os recursos demandados por essa empresa em “Recursos envolvidos”, que se comunica com uma ferramenta mais complexa para controlar a produção chamada de “Ambiente de Trabalho Virtual de Compartilhamento de Infraestruturas” utilizando o CORBA. Essa ferramenta para controlar a produção também recebe como entrada a especificação dos processo e as empresas parceiras potenciais. Dentro do “Ambiente de Trabalho Virtual de Compartilhamento de Infraestruturas” existem serviços para gerenciamento e coordenação de tarefas, gerenciamento de informações de engenharia distribuídas, gerenciamento de regras de produção, configuração do espaço de trabalho e gerenciamento e registro de recursos (TANG et al., 2002).

2.2.2.1 Projetos recentes

Outros projetos de arquiteturas de informação aplicadas a viabilização de EVs também estão presentes na literatura. Entretanto, esses projetos ainda não apresentaram resultados para compará-los com os projetos já implementados no final do último século. Ao mesmo tempo, os projetos já implementados no final do último século já não apresentam novos trabalhos na literatura, alguns desses tendo seus domínios virtuais removidos (como o caso do NIIP).

O projeto *Virtual Enterprise Chain Collaboration Framework* (VECCF) (CHOI et al., 2008) se propõe como um *framework* para guiar empresas para a colaboração. A arquitetura apresentada no projeto se foca na informação, dividida em: meta-modelos; modelos de referências; modelos; e instâncias. A estrutura da informação para o VECCF é apresentada na Figura 10.

Figura 10 - Estrutura da VECCF

Fonte: Traduzido de (CHOI et al., 2008)

Como a arquitetura do VECCF não foi aplicada a um caso real na literatura, nem apresenta as tecnologias de *middleware* utilizadas, essa arquitetura não foi utilizada para comparação com a proposta no trabalho.

Outro projeto é o Virtual Enterprise Management Platform (VEMP) (FERREIRA et al., 2014) que é uma plataforma para simplificar o estabelecimento, o gerenciamento, a adoção e o monitoramento de processos de manufaturas dinâmicos em EVs.

Esse último projeto não apresenta uma arquitetura na literatura, mas é importante de ser apresentado por ser elaborado em cooparticipação de uma universidade brasileira.

2.2.3 Arquitetura orientada a serviços

A abordagem de serviços aplicada a EV aumenta a capacidade de flexibilidade estrutural do processo de formação da mesma.

A SOA é um paradigma na área de TI para organizar e utilizar funcionalidades computacionais (como processamento, armazenamento, filtragem e outras) distribuídas que podem estar em diferentes domínios físicos (plataformas de *hardware*). Em geral, entidades possuem funcionalidades específicas para resolver ou suportar uma solução para os problemas que eles encontram no curso de seus negócios. É natural imaginar que as necessidades de uma pessoa também são atendidas pelas funcionalidades oferecidas por outra pessoa, ou em computação, os requisitos de um agente computacional de um certo proprietário sendo atendidos por outro agente computacional pertencente a outro proprietário. Não há necessariamente uma correlação de um para um entre necessidades e funcionalidades, ou seja, uma funcionalidade que atende uma necessidade, também pode atender outras necessidades. A granularidade das necessidades e funcionalidades varia da fundamental à complexa, já que dependem da quantidade de operações computacionais feitas dentro de uma funcionalidade ou da quantidade de operações computacionais requeridas dentro de uma necessidade. Mesmo uma necessidade pode requerer a combinação de várias funcionalidades enquanto uma simples funcionalidade pode atender mais do que uma necessidade (MACKENZIE et al., 2006; WANG; SONG; XIONG, 2009). Com o conceito de serviço, que é a implementação de uma funcionalidade computacional, um sistema de informação pode ser concebido para um novo nível e ser mais adequado a ambientes interoperáveis e heterogêneos (WANG; HU, 2009).

A SOA pode ser entendida como a composição estruturada de seus elementos. Ela provê uma estrutura eficiente para identificar necessidades e funcionalidades e para combinar funcionalidades para atender uma necessidade (MACKENZIE et al., 2006; ZHU; MELO, 2009). SOA é utilizada como elemento de integração universal alavancando funcionalidades de *software* existentes. A lógica de integração de orientação a processo é separada da interação do usuário, ou seja, a interface com o usuário é feita de uma forma independente da forma como é feito o controle de atividades. A conectividade é assegurada por adaptadores, ou seja, um serviço comunica-se com outro por um adaptador. Um *message broker* (corretor de mensagens) fornece um mecanismo para validação, transformação e filtro de dados para a SOA (MAYERL; VOGEL; ABECK, 2005).

Com as características acima, a SOA adota um tipo de acoplamento considerado "fraco", ou seja, a comunicação entre serviços independe da forma como esses foram criados, mas sim de adaptadores (MAYERL; VOGEL; ABECK, 2005; WANG; HU, 2009; ZHU; MELO, 2009) e usa coleções de serviços reusáveis que podem ser recombinaados para definir novos

processos ou refinar um já existente. A ideia fundamental é similar a de modelos baseados em agentes: um serviço de baixo nível pode ser parte de alguns serviços de alto nível ou de processos de negócios colaborando com outros serviços enquanto têm um certo grau de autonomia (MEDINA et al., 2009; WANG; HU, 2009; ZHU; MELO, 2009). Ambientes fracamente acoplados apresentam uma redução de dependências entre aplicações com interação de estilo de mensagem, semântica e sintaxe. Como a SOA é, em geral, concebida sobre uma plataforma existente, a implementação e as tecnologias específicas do detalhamento do serviço são ocultados por interfaces abstratas, por causa do fraco acoplamento, dos serviços a serem integrados (MAYERL; VOGEL; ABECK, 2005). Os serviços na SOA devem ser auto contidos, ou seja, devem executar tarefas pré-determinadas (WANG; HU, 2009; ZHU; MELO, 2009). A SOA provê um padrão de desenvolvimento flexível independente da complexidade da granularidade do serviço.

Alguns conceitos descritos em MacKenzie et al. (2006) relacionados com o paradigma SOA são:

- **Visibilidade:** refere-se ao poder das necessidades e das funcionalidades de uma entidade ver as outras. É feita pela descrição de aspectos, como funcionais e requisitos técnicos; restrições e políticas relacionadas; e mecanismos de acesso e resposta.
- **Semântica:** refere-se à forma de como uma necessidade é descrita. A semântica e a sintaxe de uma necessidade devem ser amplamente acessíveis e entendíveis.
- **Interação:** é a atividade de usar uma funcionalidade. Tipicamente mediada pela troca da mensagens, uma interação prossegue por uma série de trocas de informações e chamadas de ação.
- **Contexto de execução:** é um conjunto de elementos técnicos e de negócios que fica entre as necessidades e funcionalidades. Este permite ao provedor (que fornece uma funcionalidade) e usuário (que tem uma necessidade de serviço) interagirem e estabelecerem um ponto de decisão para qualquer política e contrato que podem estar em vigor.

A ferramenta mais comum para implementar a SOA são os *WS* (). Os *WS* foram concebidos para permitir o acesso consistente a recursos computacionais em múltiplas plataformas heterogêneas, suportando certo grau de colaboração inter organizacional no gerenciamento de processos. *WS* estabelece meios padrões de inter operação entre diferentes aplicações de *software* que podem ser executados em uma variedade de plataformas. Sistemas heterogêneos podem ser facilmente integrados por *WS* (KIM et al., 2006). Os *WS* possuem um conjunto de padrões que são utilizados para sua especificação (FARAHBOD, 2001).

Para implementações de serviços que reutilizam outros serviços já oferecidos, por outros fornecedores, os WSs utilizam o conceito de composição de serviços (SANTOS; MADEIRA, 2006). Há dois métodos mais comuns de composição de serviços:

- **Orquestração:** composição de serviços em que todas as interações de um processo de negócios devem ser descritos, como em um sistema tradicional. Esta descrição é executada por uma ferramenta de orquestração que tem controle sobre a composição global (SANTOS; MADEIRA, 2006). Neste método, os WSs provêm uma interface abstrata capaz de encapsular as funcionalidades e as capacidades de processos de equipamentos. A ferramenta de orquestração implementa a lógica de aplicação necessária para prover interfaces de alto nível para processos compostos a partir de serviços mais especializados (DELAMER; LASTRA, 2006). Um dos padrões de composição de serviços que adotam a orquestração é o *Business Process Execution Language for Web Services* (BPEL4WS);
- **Coreografia:** é mais colaborativa e menos centralizada que a orquestração. Apenas as trocas de mensagens públicas são consideradas relevantes e cada serviço apenas conhece suas próprias interações e comportamentos (SANTOS; MADEIRA, 2006). Enquanto a orquestração se preocupa com a execução das atividades e o sequenciamento dos serviços, a coreografia preocupa-se com as regras que definem as mensagens e as sequências de interações que devem ocorrer para executar um dado processo em uma dada instância de serviço (DELAMER; LASTRA, 2006). Um dos padrões de composição de serviços que adotam a coreografia é o *WS-CDL* ().

Uma implementação de WS que contribui para a distribuição das informações é o *SODA* (), que é implementado a partir do conceito de plataforma para ambientes de banco de dados distribuídos. A ideia é distribuir o processamento da consulta dentro de um sistema aberto baseado em normas comuns, permitindo a todos juntar e prover suas próprias implementações de algoritmos em uma base WS para estender a capacidade de processamento de consultas, como na Figura 11 (BERAN; HABEL; SCHIKUTA, 2008).

A integração de dados pode ser ágil e configurável, assim como novos serviços podem ser adicionados ao sistema sem necessidade de interromper a execução de aplicações e novas aplicações podem ser compostas a partir de serviços existentes alterando a especificação das mesmas, sem mudar a estrutura do sistema de gerenciamento de banco de dados em cada banco de dados. Uma SODA pode ser adotada para um sistema que possui a capacidade de suportar serviços de busca, consulta, armazenamento e remoção de dados (CHOW et al., 2009).

Figura 11 - Arquitetura distribuída SODA

Fonte: (BERAN; HABEL; SCHIKUTA, 2008)

2.3 REDE DE PETRI E WORKFLOW

A rede de Petri (RdP), é uma técnica de representação gráfica e matemática SEDs aplicada para modelagem, análise formal e projeto destes sistemas (ADAM; ATLURI; HUANG, 1998; NASSAR et al., 2008; ZURAWSKI; ZHOU, 1994). Ela é efetiva como técnica de descrição e especificação de processos (HAMADI; BENATALLAH, 2003; MORALES; MELO; MIYAGI, 2007; YOO; JEONG; CHO, 2010). A RdP é uma representação que pode ser usada tanto no nível conceitual quanto no nível funcional, em que os sistemas ou processos podem ser analisados e validados antes de prosseguir com o projeto detalhado e a implementação. A RdP pode ser vista como uma ferramenta de comunicação entre pessoas relacionadas ao projeto, permitindo uma fácil interpretação e clara identificação dos estados e ações. Possui a vantagem de um mesmo modelo poder ser usado para análise das propriedades comportamentais e para avaliação de desempenho, assim como para a especificação da solução de controle. Pode ser usada para identificar propriedades de sistemas como sincronização de processos, eventos assíncronos, operações concorrentes, conflitos ou compartilhamento de recursos, etc. Matematicamente, a RdP pode ser descrita como um conjunto de equações algébricas lineares e, por isso, pode ser usada para a verificação formal de relações de precedência entre eventos e condições.

Desde sua apresentação por Carl Adam Petri (BRAUER; REISIG, 2006), a RdP tem sido usada na modelagem e análise de diferentes tipos de sistemas e aplicações tais como: protocolos distribuídos (KANESHIRO et al., 2008), aplicações industriais (ZURAWSKI; ZHOU, 1994), fluxo de processos (KIEPUSZEWSKI; HOFSTED; AALST, 2003), controle supervisor

(MELO; MORALES; MIYAGI, 2008; LEE; ZHOU; HSU, 2005), entre outras. Essas aplicações da RdP estão diretamente associadas a SPs e são partes que compõem um ambiente colaborativo.

Na seção 2.3.1, a RdP lugar-transição é apresentada com o formalismo matemático que é também adotado pela ferramenta proposta de descrição de *workflow*.

Na seção 2.3.2, as extensões da RdP que permitem a modelagem da comunicação com o meio físico são apresentadas. Essas extensões também são usadas pela ferramenta proposta de descrição de *workflow*.

Na seção 2.3.3, é apresentada a *Workflow net*, que é uma ferramenta baseada na RdP originalmente proposta para descrição de *workflows*.

Na seção 2.3.4, a RdPC é apresentada, pois ela é utilizada nos testes de validação do trabalho e para comparação com a ferramenta de descrição de *workflow*.

Na seção 2.3.5, é apresentada uma estrutura de dados para troca de informações da RdP e de suas extensões, chamada de *Petri net Markup Language*, que é usada pela ferramenta proposta para as trocas de informações de *workflow*.

Na seção 2.3.6, é apresentada a *Production Flow Schema*, que é uma extensão da RdP usada para sistematização do processo de modelagem de *workflows*.

2.3.1 Formalização

Inicialmente tem-se a RdP lugar-transição que matematicamente, é descrita como uma 5-tupla $PN = (P, T, F, M, C)$ em que:

P é o conjunto de elementos passivos chamados de lugares,

T é o conjunto de elementos ativos chamados de transições,

F é o conjunto de relacionamentos entre os elementos passivos e os elementos ativos chamados de arcos orientados,

M é um vetor de recursos chamado de marcação, isto é, um vetor de números naturais em que cada elemento m_i indica o número de marcas no lugar p_i , isto é $m_i = M(p_i)$ e, $\forall p \in P, M(p) \in \mathbb{N}$,

C também é um vetor de marcações que define a quantidade máxima de marcas que cada lugar pode possuir, chamado de capacidade, isto é $M(p_i) \leq C(p_i)$, $\forall p_i \in P$, e onde,

P e T são conjuntos finitos, não nulos ($P \cup T \neq \emptyset$) e disjuntos ($P \cap T = \emptyset$), e

$F \subseteq (P \times T) \cup (T \times P)$ (CHEN; LI; ZHOU, 2012).

A representação gráfica dos lugares da RdP são circunferências; a representação gráfica das transições são retângulos; a representação gráfica dos arcos orientados são setas apontando

do elemento de origem para o elemento de destino ($f = (origem, destino)$ com $f \in F$); e a representação gráfica das marcas são pontos dentro dos lugares, assim como na Grafo 1 (MURATA, 1989).

Grafo 1 - Rede de Petri

O pré-conjunto de um elemento $x \in P \cup T$ é definido como $\bullet x = y | (y, x) \in F$, ou seja, o pré-conjunto de uma transição é todo lugar tal que exista um arco orientado com origem nele e destino na transição. Da mesma forma, pós-conjunto de um elemento $x \in P \cup T$ é definido como $x \bullet = y | (x, y) \in F$, ou seja, o pós-conjunto de uma transição é todo lugar tal que exista um arco orientado com origem na transição e destino nele (MURATA, 1989).

Além da estrutura, a RdP é um modelo dinâmico que segue o conceito de mudança de estados pela ocorrência de eventos. O estado é representado na RdP por dois conjuntos de elementos, os lugares e as marcas (ou o vetor de marcação), ou seja, para cada estado i do modelo, $\exists m_i \in M$ que representa esse estado. Ocorrência de um evento na RdP é representada pelo “disparo” de uma transição associada às regras para a mudança de estado da RdP que considera os arcos orientados (MURATA, 1989).

Pelas regras para a mudança de estados, as marcas “fluem” de alguns lugares para outros lugares após um “disparo” de transição.

Para que uma transição dispare, ela deve estar em uma condição dita “habilitada para o disparo”, que significa ter todas suas pré-condições e pós-condições atendidas.

Assim, uma transição “habilitada para o disparo” é uma transição $t \in T$ em que $\forall p \in \bullet t | M(p) > 0$ (possui as pré-condições atendidas) e $\forall p \in t \bullet | M(p) < C(p)$ (possui as pós-condições atendidas). Quando essa transição t dispara, ocorre a mudança de estado, entre os estados i e j , e com isso, o vetor de marcação altera da seguinte maneira:

$$\diamond \forall p \in \bullet t \ M^j(p) = M^i(p) - 1, \text{ e}$$

◇ $\forall p \in t^\bullet \ M^j(p) = M^i(p) + 1$ (CHIOLA et al., 1993; MURATA, 1989).

Essa mudança de estados entre o estado i e o estado j pela ocorrência do disparo da transição t é representada como $M^i[t > M^j$. Para mudanças de estado pelo disparo de mais de uma transição tem-se o conceito de sequência de disparos de transições. Sendo t_a uma transição cujo disparo leve o estado de v para w e sendo t_b uma transição cujo disparo leve o estado de w para z , tem-se a seguinte representação: $M^v[t_a > M^w[t_b > M^z$, ou de forma simplificada, considerando a sequência de disparos $\sigma = t_a t_b$, $M^v[\sigma > M^z$.

Os estados que um modelo com uma marcação inicial podem alcançar podem ser representados em um grafo de alcançabilidade do modelo. Esse grafo utiliza todas as marcações alcançáveis a partir da marcação inicial, determinadas pela função de alcançabilidade $R : M \rightarrow M$. Neste caso, $R(M_i)$ é um conjunto de estados alcançáveis a partir do estado M_i , de forma que, se $M_j \in R(M_i) \Rightarrow \exists \sigma_{ij} | M_i[\sigma_{ij} > M_j$.

2.3.2 Portas e arcos de sinais de saída

A RdP também é utilizada como técnica de controle que envolve sinais de entrada (vindo de sensores) e de saída (indo para atuadores). Com essa técnica, é possível utilizar a RdP para programar controladores de equipamentos, utilizando entradas de sinais de sensores e/ou saídas de sinais para atuadores dos mesmos. Uma extensão da RdP que permite esse tipo de modelagem são as portas, como as utilizadas no MFG () (TAKAHASHI; HASEGAWA; MIYAGI, 1999). As portas podem ser classificadas entre habilitadoras ou inibidoras, e internas ou externas. As portas são associadas aos elementos ativos, isto é as transições na RdP ou no MFG. Quando uma porta habilitadora está associada a uma transição t , a porta habilita o disparo dessa transição no caso de receber o sinal “1”, equivalente a um lugar p marcado. Quando uma porta inibidora se associa a uma transição t , a porta inibe o disparo dessa transição no caso de receber o sinal “1”, equivalente a um lugar p marcado.

As portas internas indicam os lugares da RdP ou as *boxes* do MFG que definem o estado da porta. As portas externas indicam que o estado da porta deriva do estado de um elemento/sinal externo ao grafo.

Graficamente, as portas habilitadoras são representadas por arcos com círculos sólidos na ponta em contato com a transição a que ele se associa, assim como na Grafo 2. As portas inibidoras são representadas por arcos com círculos ocos na ponta em contato com a transição a que ele se associa, assim como na Grafo 2.

Os arcos de sinais de saída levam uma informação de um elemento passivo para uma entidade externa a RdP como uma máquina ou um atuador (TAKAHASHI; HASEGAWA; MIYAGI, 1999). Os arcos de sinais de saída levam informação booleana e, em geral, representam a

Grafo 2 - Portas habilitadoras e inibidoras

ativação ou não de um equipamento. Se $M(p) = 0$ o arco de sinal de saída leva a informação “0” para o equipamento externo. Se $M(p) > 0$ o arco de sinal de saída leva a informação “1” para o equipamento externo.

A representação gráfica de um arco de sinal de saída é um arco ligado a um objeto externo ao modelo, como na Grafo 3.

Grafo 3 - Arco de sinal de saída

Com base na experiência de modelagem de sistemas reais, algumas extensões da RdP foram introduzidas: o MFG foi desenvolvido no Japão e na década de 80 já fazia parte do conjunto de normas do JIS (*Japan Industrial Standard*) como uma forma padronizada de representação de controle. O GRAFCET tem um histórico análogo na França. Uma abordagem mais conceitual é a do SIPN () (IVANOVA; FREY; BATCHKOVA, 2008), definida como $SIPN = (P, T, F, M, IS, OS, \varphi, \omega)$, em que: P , T , F , e M são, respectivamente, os conjuntos de lugares, transições e arcos orientados e o vetor de marcação da RdP; IS é o conjunto de sinais de entrada (sinais gerados em um sensor que são transmitidos ao controlador); OS é o conjunto de sinais de saída (sinais gerados no controlador que são transmitidos para um atuador); $\varphi : T \rightarrow IS$ é a função que indica para cada transição, quais sinais de entrada são esperados; e $\omega : P \rightarrow OS$ é a função que indica para cada lugar, quais sinais de saída são esperados. Os elementos da SIPN são mostrados na Grafo 4.

Grafo 4 - Elementos da SIPN

Na Grafo 4, $P = \{p\}$, $T = \{t\}$, $F = \{(p, t)\}$, $M = (0)$, $IS = \{in\}$, $OS = \{out\}$,

$$\varphi(t) = in, \text{ e } \omega(p) = out.$$

2.3.3 Workflow net

Um modelo *workflow* é uma representação formal de um processo produtivo ou processo de negócios que permita a execução automatizada do processo (LI; FAN; ZHOU, 2004). Pode também ser compreendido como a especificação dos processos associados a solicitação de um cliente, no caso de uma EV. A execução automatizada é uma característica desejável em uma produção industrial por aumentar seu desempenho e a qualidade dos produtos.

Um *workflow* é entendido em 3 dimensões: de fluxo de controle, de recursos e de casos. A dimensão de fluxo de controle é sobre a ordenação das tarefas. A dimensão de recursos trata de como se executam essas tarefas. Na terceira dimensão está o entendimento dos processos individuais que são executados de acordo com a definição do processo (primeira dimensão) e com os recursos adequados (segunda dimensão) (AALST, 2000).

Na RdP, a *Wf-net* () aplica algumas condições para um modelo ser considerado um modelo *workflow* (que significa ser capaz de modelar na dimensão do fluxo de controle), sem limitar as propriedades dos processos produtivos capazes de serem modelados (LI; FAN; ZHOU, 2004).

As condições para um modelo ser uma *Wf-net* são (LI; FAN; ZHOU, 2004):

- A *Wf-net* é uma $PNet = (P, T, F)$ em que há dois lugares especiais ϵ e θ , em que ϵ é um lugar “fornecedor” com $\bullet\epsilon = \emptyset$ e θ é um lugar “sorvedor” com $\theta\bullet = \emptyset$;
- Quando adicionar uma transição t' para a $PNet$ que conecta os lugares θ e ϵ , com $\bullet t' = \{\theta\}$ e $t'\bullet = \{\epsilon\}$, o modelo resultante $P\bar{N}et = (\bar{P}, \bar{T}, \bar{F})$, em que $\bar{P} = P$, $\bar{T} = T \cup \{t'\}$, $\bar{F} = F \cup \{(\theta, t'), (t', \epsilon)\}$ deve ser fortemente conectado.

Pela definição em Aalst (2000), um modelo fortemente conectado é um modelo tal que dados x e y dois nós do modelo com $(x \cup y) \in (P \cup T)$, há um caminho levando x até y . Ou seja, o modelo é fortemente conectado se $\exists Path = \{F(n_0, n_1), F(n_1, n_2), \dots, F(n_{m-1}, n_m)\}$ com $\forall F(n_i, n_{i+1}) \in F$ e $n_0 = x$, $n_m = y$ e $\forall (x \cup y) \in (P \cup T)$.

A *Wf-net* inclui 4 representações específicas de elementos para descrever algumas características de processos de *workflow*, que são: AND-split para indicar o início de um paralelismo, OR-split para indicar o início de um conflito, AND-join para indicar o fim de um paralelismo e OR-join para indicar o fim de um conflito (AALST; HEE, 2004). Estes elementos são mostrados na Grafo 5.

Além das condições anteriormente citadas, segundo Aalst (2000), para que o modelo seja um *workflow* é preciso analisar a propriedade robustez do modelo.

Grafo 5 - Elementos específicos da Wf-net

A propriedade de robustez do modelo é descrita como “Para qualquer caso, o procedimento irá sempre terminar e o momento em que o procedimento terminar, haverá um marca no lugar θ e todos demais lugares estão vazios” (AALST, 2000).

Segundo a definição, dado $\exists M_\epsilon | M_\epsilon(p) = 0, \forall p \neq \epsilon \in P$ e $M_\epsilon(\epsilon) = 1$, assim como, $\exists M_\theta | M_\theta(p) = 0, \forall p \neq \theta \in P$ e $M_\theta(\theta) = 1$, um modelo é robusto se:

1. para todo estado M alcançável do estado M_ϵ , há uma sequência de disparos que leva M ao estado M_θ . Ou seja, $\forall M \in R(M_\epsilon) \Rightarrow M_\theta \in R(M)$;
2. o estado M_θ é o único estado alcançável do estado M_ϵ com ao menos uma marcação no lugar θ . Ou seja, para $M_k \in R(M_\epsilon)$ se $M_k(\theta) > 0 \Rightarrow M_k = M_\theta$;
3. não há transições mortas em $(PNet, M_\epsilon)$. Ou seja, $\forall t \in T, \exists M, M' \in R(M_\epsilon) | M[t > M'$.

2.3.4 Rede de Petri Colorida

A RdPC foi proposta inicialmente por Kurt Jensen na década de 80 (JENSEN, 1981), como uma forma de simplificar a representação gráfica dos modelos em RdP utilizando lógicas computacionais ligadas aos arcos e às transições. A RdPC é uma RdP de alto nível que atingiu grande sucesso na representação formal de *workflows* por utilizar o mesmo arcabouço teórico da RdP e por existir trabalhos que demonstram como fazer análise de propriedades importantes para os *workflows*, como a robustez.

Muitas representações formais de RdPC são utilizadas na literatura. A representação formal utilizada nesse trabalho é $CPN = (P_C, T_C, F_C, CS_C, V_C, CF_C, G_C, A_C, S_C)$ (HE et al., 2012), em que:

P_C é um conjunto finito de lugares,
 T_C é um conjunto finito de transições,
 $F_C \subseteq P_C \times T_C \cup T_C \times P_C$ é um conjunto de arcos orientados,
 CS_C é um conjunto finito e não vazio de conjuntos de cores,
 V_C é um conjunto finito de variáveis com tipos, tal que $Type[v] \in CS_C$ para todas as variáveis $v \in V_C$,

$CF_C : P_C \rightarrow CS_C$ é uma função de conjunto de cores que atribui um conjunto de cores para cada lugar,

$G_C : T_C \rightarrow EXPR_V$ é uma função guarda que atribui uma guarda para cada transição t , tal que $Type[G_C(t)] = Bool$, em que $EXPR_V$ denota o conjunto de expressões sobre o conjunto de variáveis V_C . Ou seja, a função guarda é um conjunto de expressões booleanas usando as variáveis do conjunto de variáveis V_C associada a cada transição,

$A_C : F_C \rightarrow EXPR_V$ é a função de expressão dos arcos orientados que atribui uma expressão para cada f , tal que $Type[A_C(f)] = CF_C(p)_{MS}$, em que p é um lugar conectado ao arco f e $CF_C(p)_{MS}$ é um dos possíveis conjuntos de cores, definidos pela função de conjunto de cores, associado ao lugar p . Ou seja, a função de expressão dos arcos orientados é uma expressão que utiliza as variáveis de V_C , tal que os conjuntos de cores a que pertencem as variáveis utilizadas na expressão, apenas podem estar no conjunto de conjuntos de cores do lugar a que conecta o arco,

$S_C : P_C \rightarrow EXPR_\Phi$ é uma função de inicialização que atribui uma expressão de inicialização para cada lugar p , tal que $Type[S_C(p)] = CF_C(p)_{MS}$.

Na dinâmica dos modelos em RdPC, além das regras de disparos já presentes na RdP, que é ter marcações nos lugares dos pré-conjuntos das transições, é preciso também verificar as expressões booleanas das funções de guarda das transições (HE et al., 2012).

2.3.5 Petri net Markup Language (PNML)

O *PNML* () é um formato de troca de informações baseadas em XML para representar as várias extensões da RdP (WEBER; KINDLER, 2003). Este formato possui uma estrutura de representação de redes e sub-redes, que permite a inclusão de redes de alto nível e outras representações. A Figura 12 mostra a estrutura geral para os arquivos PNML.

Na Figura 12, o arquivo PNML inicia com o elemento “PetriNetFile” que pode conter um ou mais elementos do tipo “PetriNet”. Cada elemento “PetriNet” contém objetos (“Objects”), ferramentas específicas (“ToolInfo”) e textos (“Label”) e representam a estrutura de redes de um conjunto. Os elementos “Objects” podem ser de 3 tipos: nós (“Node”) que se dividem em lugares (“Place” ou “RefPlace”) ou transições (“Transition” ou “RefTransition”); páginas

Figura 12 - Estrutura das informações no arquivo PNML

Fonte: (WEBER; KINDLER, 2003)

("Page") que podem ser sub-redes de Petri; e arcos ("Arc") que estabelecem a relação entre os nós ("Node"). Os elementos "ToolInfo" são elementos específicos das diferentes aplicações e, como podem ser ligados aos objetos, permitem a inclusão de novos elementos das diferentes interpretações das RdPs. Os elementos "Label" são elementos textuais que são apresentados na tela para auxiliar a compreensão da rede. Para algumas aplicações que envolvem a separação de modelos, existem os elementos do tipo "RefNode" que são diretamente relacionados aos elementos de mesma natureza por um identificador.

2.3.6 Production Flow Schema (PFS)

O PFS () é uma interpretação da RdP do tipo Canal-Agente própria para sistematizar a construção de modelos na área de SPs. O PFS mostra, ao nível conceitual, como os itens envolvidos num processo produtivo são objetos da execução de funções que são necessárias para obter o produto ou serviço desejado. Um importante conceito apresentado pelo PFS é o fluxo de itens (materiais ou informações), que é associado a alocação e desalocação de recursos. O PFS indica que as atividades ou etapas de um processo envolvem a interação entre itens, seus fluxos e outras atividades (HASEGAWA et al., 1999).

O grafo de PFS apresenta os elementos do sistema organizados hierarquicamente, utilizando uma abordagem *top-down* explorando o conceito de macro-eventos. A técnica foi desenvolvida para sistematizar e facilitar a modelagem por RdP (HASEGAWA et al., 1999).

O PFS possui 3 elementos estruturais básicos (HASEGAWA et al., 1999):

- **Atividades:** Representam os macro-eventos, elementos que podem ser detalhados internamente obtendo um novo PFS ou alguma outra RdP interpretada, graficamente representado como no Grafo 6.a;
- **Distribuidores:** Similares ao lugares na RdP, esses elementos são capazes de representar o acúmulo de itens e são responsáveis pela habilitação de uma atividade, representado como no Grafo 6.b;
- **Arcos:** Elementos que fornecem as relações entre os distribuidores e as atividades, representado no Grafo 6.c.

Grafo 6 - Elementos do PFS

Os distribuidores em PFS são elementos passivos e são responsáveis apenas por indicar os itens que fluem pelo grafo. Já as atividades são elementos ativos e são responsáveis por indicar a ocorrência de um evento como produção, transporte e modificação de itens. Os arcos são os elementos que indicam o fluxo dos itens dentro do grafo, em que estes podem ir de um distribuidor para uma atividade ou de uma atividade para um distribuidor (HASEGAWA et al., 1999). O PFS pode representar o fluxo principal de itens (Grafo 7) ou representar relações de comunicação assíncrona (Grafo 8) que explora o conceito de fluxo secundário de informação (GUSTIN, 2000).

Grafo 7 - Exemplo de fluxo principal de informação em PFS

Grafo 8 - Exemplo de fluxo secundário de informação em PFS

A metodologia PFS é baseada em um procedimento de refinamento gradativo (refinamentos sucessivos), substituindo um nó (atividade ou distribuidor) por um modelo de PFS ou RdP, garantidas as condições de contorno em que os elementos obtidos do refinamento se liguem da mesma forma que o elemento refinado estava ligado ao modelo. Um refinamento terá sido executado corretamente se a interpretação dos sub-grafos que substituem os nós mantiverem as funcionalidades/comportamentos do grafo original (HASEGAWA et al., 1999).

2.4 SÍNTESE DO CAPÍTULO

O escopo desse trabalho é baseado na colaboração entre empresas dispostas a colaborar umas com as outras utilizando a internet na forma de EVs. Com a descrição de EVs fica claro como se entende o ambiente colaborativo (uma das características do escopo do trabalho) somado à definição da heterogeneidade também desse ambiente (outra característica fundamental do escopo).

Os problemas apontados na pesquisa de EVs também orientam a elaboração das hipóteses do trabalho, tal como a formação de uma EV.

As ferramentas que são aplicadas ao escopo do trabalho (as arquiteturas de informação NIIP, PRODNET II, e E-DREAM) tomam como base o conceito de EV e utilizam CORBA, ou prestação de serviços, como solução tecnológica para colaboração entre empresas. Por essa razão, a utilização de serviços é amplamente pesquisado no escopo deste trabalho.

A rede de Petri e suas extensões são aplicadas para o projeto e descrição de *workflow*, que é uma etapa do anterior a execução de uma produção, no ciclo de vida das EVs. Essa etapa é parte da formação das EVs e a ferramenta que descreve o *workflow* deve atender as necessidades das empresas que desejam participar de uma colaboração.

3 ATRIBUIÇÕES DE EMPRESAS NO AMBIENTE COLABORATIVO

Este capítulo apresenta as características encontradas na literatura que definem as parcerias de empresas para estabelecer o que é compreendido como ambiente colaborativo. Para isso, os conceitos fundamentais de colaboração serão utilizados.

Na seção 3.1, são descritos os requisitos para a colaboração nesse ambiente.

Na seção 3.2, é apresentado um requisito que tanto se aplica a colaboração entre empresas, quanto se aplica ao controle individual da empresa.

Na seção 3.3, são descritas as atribuições das empresas em uma colaboração (ou no ambiente colaborativo).

Na seção 3.4, as arquiteturas de informação de EVs, apresentados anteriormente, serão analisadas.

3.1 REQUISITOS DO AMBIENTE COLABORATIVO

O ambiente colaborativo é entendido como um ambiente heterogêneo composto de algumas empresas com capacidades semelhantes umas as outras. Pela colaboração, é importante para as empresas comunicação, como um requisito para elaboração de um acordo entre elas. Esse requisito é aqui chamado de universalização da informação.

Por se tratar de uma grande quantidade de informações, em ambientes com tais características, implica em grande quantidade de dados registrados em bancos de dados. Essa grande quantidade de dados pode gerar problemas quanto a dependência dos SGBDs e quanto à consulta e ao processamento desses dados. Esse tema é abordado no requisito de distribuição da informação.

3.1.1 Universalização da informação

A universalização da informação é o nome dado ao requisito de como deve ser a informação que é trocada entre as empresas dispostas a colaborar umas com as outras no ambiente colaborativo. Nas áreas de comunicação de sistemas, principalmente voltadas a sistema de informação (SI), esse requisito é abordado no conceito de interoperabilidade da informação.

Segundo Di e Fan (2010), as empresas necessitam trocar informações eletronicamente nos processos de colaboração. Ou seja, colaborar e/ou cooperar não apenas se resume a compartilhar as capacidades que a empresa possui, mas dentro dos processos de negócio, inclui a troca de informações. As empresas há tempos já aproveitam as tecnologias presentes na forma de SIs para controlar seus processos de negócio (JIANG et al., 2012).

Segundo o IEEE (GUO, 2013), a interoperabilidade da informação é definida como a habilidade de dois ou mais sistemas ou componentes de trocar informações e usar as informações trocadas. Isso significa que a troca de informações entre diferentes SIs não pode se dar apenas pelo acesso à informação, mas às outras formas de manipulação desta informação também, permitindo aos diferentes participantes dessa troca de informação usá-la da maneira que lhe convir.

Como as empresas se aproveitam de uma grande quantidade de SIs muito variados entre si, um dos problemas que surge para essa interoperabilidade envolve a compreensão da informação trocada (DI; FAN, 2010).

Figura 13 - Analogia entre a interoperabilidade da informação e a comunicação entre pessoas

A Figura 13 é uma analogia ao problema encontrado na compreensão da informação entre sistemas, com casos reais de comunicação. Nesta figura, “Pessoa A” e “Pessoa B” são análogos a SIs diferentes implementados com suas próprias linguagens de programação e “Mensagem C” e “Mensagem D” são análogos às informações enviadas pelos SIs análogos às “Pessoa A” e “Pessoa B”, respectivamente.

Assim como “Mensagem C” e “Mensagem D” (da Figura 13) estão em diferentes idiomas, as informações enviadas pelos diferentes SIs possuem diferentes padrões de sintaxe e semântica (ESTÉVEZ; MARCOS, 2012). Existem duas condições para que a comunicação, da forma de interoperabilidade, entre dois SIs ocorra: (1) a informação deve ser capaz de ser trocada entre os sistemas conectados, do ponto de vista físico; (2) a informação deve ser usável nos receptores dos sistemas, do ponto de vista lógico (GUO, 2013). Analogamente, a primeira condição é atendida entre “Pessoa A” e “Pessoa B” se considerarmos que a distância entre essas é suficiente para que o som se propague entre elas e seja perceptível ao ouvido dessas, entretanto, considerando que o idioma falado por cada uma é diferente e cada uma desconhece qualquer padrão do outro idioma, essa comunicação não atende a segunda condição.

A internet é uma ferramenta que os SIs podem utilizar para se comunicarem entre si e sua utilização é adequada às características do ambiente colaborativo. Para completar a interoperabilidade da informação, é necessário atender a segunda condição, da informação ser

usável nos sistemas.

Algumas áreas importantes já utilizam o conceito de interoperabilidade da informação, como: comércio eletrônico, integração de empresa, gerenciamento de cadeia de suprimento, composição de serviços, projeto de sistemas distribuídos, qualidade de serviços, projeto colaborativo. A razão desses sistemas utilizarem a interoperabilidade da informação é o fato de que em todas essas áreas, as empresas/serviços participantes necessitam de uma perspectiva global na troca e uso da informação (GUO, 2013).

Quando tratados de forma global, ou seja, não apenas como a comunicação entre dois SIs, mas envolvendo várias e diferentes entidades, essas condições necessárias para a interoperabilidade ganham também novas formas: (3) interoperabilidade de dados estáticos, que é a habilidade de comunicação e troca de dados em certo formato de dados e protocolos de comunicação entre os sistemas conectados; (4) interoperabilidade de informação semântica, que é a habilidade de interpretar automaticamente e precisamente os dados trocados como informações significativas compreensíveis pelos sistemas conectados e os usuários finais dos sistemas conectados (GUO, 2013).

Considerando que as diferentes áreas de produção (como manufatura, logística, transporte, etc.) possuem grandes diferenças entre si quanto às especificações de suas necessidades, é compreensível que as empresas que possuem capacidades nas diferentes áreas possam ter em seus SIs, parcial ou completamente, implementações diferentes utilizando semânticas e sintaxes específicas nas informações tratadas. Assim, é importante atender essas duas novas condições (3 e 4) da interoperabilidade da informação.

Com a própria internet, a condição (3) já possui um amplo conjunto de soluções disponível, como os protocolos de comunicação *HTTP* (), as linguagens de marcação *HTML* e *XML* entre outras (GUO, 2013).

Para as condições (2 e 4) as soluções se dividem em dois conceitos diferentes: padronização comum das sintaxes e semânticas; tradução a partir das diferentes ontologias (DI; FAN, 2010). Cada um dos conceitos traz consigo alguns problemas a serem superados.

Na padronização, há uma redução da flexibilidade da informação e também obrigação dos SIs a se adaptarem a uma tecnologia única (JIANG et al., 2012). O problema da redução da informação reside no fato de que assim como há uma grande variedade de processos possíveis de serem realizados no mercado, alguns desses processos podem não serem similares aos processos já padronizados. Desta maneira, é necessário uma representação para a informação que nunca foi realizada nem é similar às já padronizadas, mas que ainda se adeque ao padrão definido.

Na tradução, o problema mais recorrente encontrado é que é necessário pessoal técnico especializado em cada SI nas empresas para traduzir a informação de um sistema para o outro

(DI; FAN, 2010).

Uma outra alternativa óbvia seria a que cada SI possua, entre as suas especialidades, todas as diferentes sintaxes e semânticas existentes para o ambiente colaborativo, o que, na prática, é inviável, pois exige um grande consumo de recursos computacionais, além de transferir para dentro da empresa o problema da tradução entre essas sintaxes e semânticas.

Utilizando novamente a analogia da Figura 13, o conceito de padronização seria o mesmo de criar um novo idioma comum que ambas “Pessoa A” e “Pessoa B” aprendam e saibam como traduzir para seu próprio idioma. Do ponto de vista teórico, esse novo idioma comum seria formado pelas palavras que cada pessoa se lembra e sabe expressar. Expandindo essa ideia para muitas pessoas com diferentes idiomas, o problema desse conceito seria o caso de um objeto ou sentimento ainda não descrito no novo idioma comum necessitar de uma descrição. Já o conceito de tradução, seria o mesmo que a “Pessoa A” criasse um dicionário para si do que pode estar presente nas mensagens enviadas pela “Pessoa B” e vice-versa. Quando expandido para muitas pessoas com diferentes idiomas, o problema desse conceito seria o caso de muitos dicionários para cada pessoa possuir e, para cada nova pessoa que entrar na conversa, um novo dicionário tem que ser criado em cada uma das pessoas. Já o último conceito, seria o caso de “Pessoa A” aprender corretamente o idioma de “Pessoa B” e vice-versa. Da mesma forma, aumentando a disponibilidade de diferentes idiomas, o problema associado a esse conceito é que no cérebro humano, assim como nos sistemas computacionais, a velocidade de interpretação (processamento) da informação diminui com o aumento da quantidade de idiomas utilizados.

Pela analogia acima, é possível entender porque não há soluções propostas utilizando o conceito de implementação de todas as sintaxes e semânticas.

Do ponto de vista teórico, as soluções dos conceitos de padronização e tradução se destacam de acordo com a variabilidade da informação. Para a padronização, quanto menos variada a informação num mesmo idioma (ou quanto menor a quantidade de palavras dentro desse idioma), mais fácil é a formulação de um padrão de linguagem. Já para a tradução, a pequena variabilidade de informação num mesmo idioma (ou a menor quantidade de informações em um mesmo idioma) faz com que o custo de informação de criar um novo dicionário (que envolve definir a relação entre as palavras do novo idioma e as dos idiomas já existentes) seja maior do que o de traduzir a informação propriamente dita (que seria o caso da padronização).

Uma das características do ambiente colaborativo aqui tratado é que não há uma grande variação dentro das classes em que estão os processos, de modo que o requisito de universalização da informação pode ser atendido via padronização para as classes de processos com a possibilidade de interoperabilidade da informação.

A hipótese aqui definida na forma de um requisito é que para o ambiente colaborativo, a informação deve ser padronizada. Essa padronização aqui é tanto da sintaxe quanto da

semântica.

Nas execuções de produção outra necessidade das informações é compreender nelas o fluxo de atividades. Esse fluxo de atividades é o *workflow* ou o processo realizado pelas empresas. Disso tem-se a hipótese de que o *workflow* deve ser descrito em linguagem simples e acessível para as empresas, da mesma forma que a sintaxe e a semântica da informação o são.

3.1.2 Distribuição da informação

A distribuição da informação é um requisito que versa pela menor dependência de cada empresa envolvida num processo global quanto as informações que ela possui. O intuito desse requisito é que múltiplos servidores de bancos de dados contenham os dados necessários para auxiliar a produção.

A etapa de projeto na formulação de um acordo colaborativo entre empresas, é uma etapa que depende de informações históricas de outras produções. A necessidade dessas informações fica mais clara quando se estuda o processo de desenvolvimento de novos produtos (PDNP), que inclui a categoria de produtos personalizados. Como mencionado anteriormente, entende-se que para o PDNP, as empresas não possuem, necessariamente, o domínio completo dos conhecimentos necessários para elaborar todo o projeto conceitual ou executivo do produto ou serviço solicitado. Para isso, ela reutiliza estratégias de outras produções.

Em toda a literatura sobre colaboração entre empresas, um dos principais tópicos é o compartilhamento da informação (JASTI; KODALI, 2015).

Segundo Chen, Chen e Hsu (2014), uma necessidade para a distribuição das informações está no fato de que as empresas que têm suas próprias implementações em TI, possuem gastos excessivos para manter um sistema para suas aplicações. Com isso, as pequenas e médias empresas encontram dificuldades para manter sistemas de dados de grande porte.

Uma das abordagens para a distribuição da informação é a SODA, descrita na seção 2.2.3, em que as empresas oferecem os dados na forma de serviços e reaproveitam as implementações de *web services* para desenvolver um sistema de gerenciamento de bancos de dados de alto nível.

A hipótese aqui definida também na forma de um requisito é que para o ambiente colaborativo, as informações que devem ser compartilhadas pelas empresas se encontram de forma distribuídas para diminuir a dependência de implementações complexas de TI nas empresas.

3.2 REQUISITO DO CONTROLE DE SISTEMAS

O requisito do controle de sistemas das empresa envolve compromissos dessas com o ambiente colaborativo, não apenas estratégias de produção. Esses compromissos são necessários para que, mesmo considerando empresas independentes, a produção seja executada com igual responsabilidade de todas as empresas que estipularam parceria.

3.2.1 Sub-contratação em tempo real

O ambiente colaborativo considera não só que há uma grande possibilidade de oferta, mas que há uma grande presença de demanda. Com isso, novos compromissos são assumidos pelas empresas que optam por participar desse modelo de colaboração e cooperação de forma muito mais rápida e que exige respostas mais dinâmicas.

Esse requisito aborda falhas e intercorrências do processo que foram contratados de uma empresa EV que podem ser descumpridas, quanto a prazos ou qualidade.

Em vários casos, as empresas oferecem seus serviços em propostas na forma de orçamentos, com a presença de prazo de validade da proposta. Com isso, a empresa assume o compromisso para o prazo adotado em seu agendamento.

Esses casos em que os prazos não podem ser cumpridos pela empresa, são aqui chamados de agenda impossível. No sistema colaborativo, há alternativas para os casos de agenda impossível, como a sub-contratação de outra empresa para realizar a atividade. Essa sub-contratação pode ser dita entre empresas concorrentes, uma vez que a atividade demandada e que não pode ser cumprida pela empresa contratada terá que ser cumprida pela empresa sub-contratada, portanto, ambas têm as mesmas capacidades para essa atividade.

Essa sub-contratação só pode ocorrer, sem prejuízo para as partes, se a empresa que sub-contrata assumir todas as responsabilidades do serviço prestado pela empresa sub-contratada, uma vez que por contrato, essa seria a responsável pela etapa da produção sub-contratada. Nessa condição, o contrato de EV deve possibilitar a existência de contratos individuais entre empresas para que essas possam sub-contratar empresas parceiras em casos de agenda impossível.

Além disso, do ponto de vista da concorrência, tanto a empresa que sub-contrata, quanto a empresa sub-contratada devem ter a percepção de ganho na parceria assumida. Isso significa que essas não devem entender que a concorrência existente leva à competição, pois a disponibilidade de demandas faz com que ambas possam trabalhar em parceria.

Então o requisito de sub-contratação em tempo real para agenda impossível considera que as empresas devem entender o mercado como tendo demanda suficiente para manutenção da

existência de empresas concorrentes. Também exige que o sistema que viabiliza os acordos de empresas em EVs permita a existência de contratos individuais entre empresas.

O requisito para o ambiente colaborativo, então, envolve a viabilidade da sub-contratação de empresas.

3.3 ATRIBUIÇÕES DE SISTEMAS COLABORATIVOS

3.3.1 Confiança entre participantes

Numa condição em que as partes de um sistema optam por cooperar e/ou colaborar, como é um sistema colaborativo, a confiança entre as partes faz com que não seja necessário o estabelecimento de hierarquias nem elaboração de regras de alta complexidade para o estabelecimento da parceria.

Um conceito que estuda essa relação de confiança é o de qualidade de serviço (QoS – *Quality of service*). Em Cho, Swami e Chen (2011) é feito um levantamento sobre gerenciamento de confiança em redes de comunicação ad hoc. A confiança é parte importante da tomada de decisão, tal como a realizada na etapa de formação das EVs.

Sociologicamente, a confiança pode ser dita subjetiva. Existem trabalhos na área de ciência da computação que quantificam a confiança como uma variável contínua dependente da percepção do contexto ou da aceitação de riscos (CHO; SWAMI; CHEN, 2011).

Para economia, existe uma distinção entre a confiança pessoal e a impessoal. A primeira depende de uma amizade direta entre as entidades envolvidas (como o caso de pessoas vizinhas) e a segunda é institucionalizada. Da mesma forma que para a ciência da computação, ambas definições da confiança para economia envolvem a aceitação de riscos para investimentos (CHO; SWAMI; CHEN, 2011). Outros problemas são levantados em trabalhos sobre confiança, como o egocentrismo e o altruísmo, no sentido de não envolverem uma relação direta entre a confiança e a aceitação de riscos.

Em gerenciamento organizacional, a confiança tem o sentido da vontade de contar com alguém ou alguma coisa, com o sentimento de relativa segurança. Nesse campo, a confiança não é necessariamente mútua ou recíproca (CHO; SWAMI; CHEN, 2011).

Numa condição de colaboração de empresas, como a proposta do ambiente heterogêneo, é considerado que as empresas não assumem riscos como forma de obtenção de maiores rendimentos, mas na vontade e necessidade de trabalhar com outras empresas, pela característica de focarem-se em suas competências. Dessa forma, a abordagem mais próxima é a do gerenciamento organizacional.

A abordagem para medição da confiança entre empresas feita em Griffiths (2006) é por taxas

de sucesso e fracasso de desenvolvimentos da empresa com a própria empresa interessada ou com outras empresas.

A hipótese aqui levantada envolve o compromisso das empresas em oferecer seus conhecimentos umas sobre as outras para definir a confiança entre empresas pra estabelecimento de parcerias.

3.3.2 Desenvolvimento de produtos personalizados

O desenvolvimento de produtos personalizados envolve as técnicas de projetos de novos produtos. Além do reaproveitamento de informações, esse tipo de projetos pode utilizar também da colaboração entre projetistas das mesmas ou de diferentes áreas (GRIFFITHS, 2006).

Em Griffiths (2006), apresenta-se o conceito de projeto cooperativo, descrevendo o *workflow* para desenvolvimento de produtos utilizando múltiplos participantes da etapa de projeto.

Segundo o portfólio da Kendall (2015), o projeto colaborativo significa aprender como resolver, criativamente, problemas e se comprometer com atividades de projeto, gerenciamento e coordenação em organizações públicas e privadas, com tecnologias avançadas de comunicação para trabalho em equipe. Nesse contexto, os projetistas devem ter conhecimentos das várias áreas que podem afetar o trabalho de projeto.

Klein et al. (2003) fala que desde o começo do século, os grandes projetos de aviões, *softwares* corporativos e outros já utilizavam grandes equipes com milhares de participantes trabalhando em diferentes elementos do projeto. Uma das dificuldades apontadas para tal está na tomada de decisões de projeto, devido a sua interdependência com a convergência para um único projeto. Esse trabalho aborda o problema por agentes elaborando acordos entre as partes para a tomada de decisão.

Um projeto completo de um produto ou serviço depende de levantar os requisitos, as especificações, o processo de criação e o ciclo de vida do produto ou serviço (KLEIN et al., 2003).

Os participantes de um projeto colaborativo são pessoas ou organizações que potencialmente podem acrescentar valor ao levantamento dos problemas do projeto e/ou à resolução dos mesmos nas escolhas das soluções sob a ótica de sua área de conhecimento (KLEIN et al., 2003). Essa representação pode ser vista na Figura 14.

Na Figura 14, os participantes do projeto, ou projetistas (representados como rostos) são as pessoas ou organizações com algum domínio que possa acrescentar valor às etapas do projeto. Os problemas do projeto (representados com círculos preenchidos de cor preta) são os motivadores para tomadas de decisões, pois estão entre as especificações e os requisitos do projeto. As interdependências dos problemas (representadas pelas linhas ligando os problemas

Figura 14 - Um modelo para projeto colaborativo

Fonte: (KLEIN et al., 2003)

de projeto) são as limitações que uma solução escolhida para um problema deve garantir para que não prejudique as soluções para outro problema. E os sub-espacos do projeto (representados pelas elipses sem preenchimento) são as áreas de domínios diferentes. Com isso, vê-se que os problemas de projetos podem estar em mais de uma área de domínio, como um todo. E ao mesmo tempo um mesmo problema tange mais de uma área de domínio. Essa é uma configuração comum em projetos colaborativos.

A hipótese aqui definida a partir da atribuição é que, na forma de uma pergunta, é:

- O ambiente colaborativo oferece a possibilidade da realização da etapa de projeto também de forma colaborativa?.

3.4 ANÁLISE DAS HIPÓTESES NAS FERRAMENTAS DE EVS

Estudando a universalização da informação como a padronização de sintaxe e semântica e a acessibilidade/facilidade do *workflow* para o ambiente colaborativo, os arquiteturas de EVs tratam esse tema como:

- O NIIP suporta o ISO-STEP, que padroniza as sintaxes e semânticas. Sobre o *workflow*, há citação da existência de uma ferramenta de gerenciamento de *workflow*, mas não há nenhuma explicação mais funcional sobre ela;
- O PRODNET II suporta o STEP e o EDI como padrão para sintaxes e semânticas. Na arquitetura não é apresentada nenhum item para gerenciamento de *workflow*;
- O E-DREAM, por procurar uma ontologia mais aberta, não busca a padronização de

sintaxe ou semântica. Não há descrição também da forma de gerenciamento de *workflow*.

Sobre a hipótese de bancos de dados distribuídos, as três arquiteturas utilizam o CORBA para integração de serviços e sistemas de bancos de dados distribuídos.

Sobre a hipótese da sub-contratação de empresas:

- O NIIIP dispõe de controle cognitivo centralizado para monitoração e resolução de conflitos para negociação, fazendo com que as empresas precisem que suas negociações passem por um *framework* (incluído na arquitetura) para serem efetivadas. Tal característica não impede a sub-contratação, mas coloca barreiras (como *firewalls*) para que essas sejam resolvidas totalmente entre os pares;
- O PRODNET II possui módulos de coordenação local, deixando os nós (ou empresas) comunicarem independentes de elementos centralizadores;
- O E-DREAM não mostra como ocorre a negociação para formação da EV.

Sobre a hipótese de confiança entre as empresas:

- O NIIIP não cita a confiança nas funções da arquitetura;
- O PRODNET II diz que de acordo com a necessidade de confiança das relações das empresas é possível implementar ferramentas para auxiliar o cálculo de confiança;
- O E-DREAM não cita a confiança nas funções da arquitetura.

Sobre a hipótese de projeto colaborativo:

- O NIIIP possui implementação de projeto colaborativo na arquitetura;
- O PRODNET II diz que essa ferramenta está em implementação;
- O E-DREAM possui implementação de projeto colaborativo na arquitetura.

3.5 SÍNTESE DO CAPÍTULO

Os requisitos identificados no capítulo fundamentam a proposta de arquitetura deste trabalho, quando se trata de viabilizar a colaboração entre as empresas.

As atribuições das empresas no ambiente colaborativo fundamentam as hipóteses sobre o seu funcionamento.

De forma analítica, as hipóteses levantadas são as seguintes:

- “Há uma padronização de sintaxe e semântica da informação?”;

- “Há uma padronização de *workflow*?”;
- “Há bancos de dados distribuídos?”;
- “Há a possibilidade de subcontratação de empresas?”;
- “Há funções para o compartilhamento de informações de confiança entre empresas?”;
- “Há funções para permitir o projeto colaborativo?”.

4 MODELAGEM DA ARQUITETURA DE INFORMAÇÃO PARA EVS

Com as atribuições e requisitos levantados no capítulo 3, é possível gerar diagramas representativos de como deve ser um sistema de informação para atender as necessidades de colaboração de empresas num processo produtivo no esquema de EV.

Para atender as necessidades apontadas, é proposta uma arquitetura de informação que contém um *framework* de apoio com premissa de não centralizar as atividades realizadas na colaboração. Essa premissa assegura as empresas a uma independência maior na colaboração e cooperação.

A manutenção de um *framework* na arquitetura de informação se justifica principalmente nas empresas que não possuem os complexos SIs implementados em suas estruturas e que ainda assim podem oferecer suas capacidades em uma colaboração.

Os diagramas representativos são diagramas de linguagens semi-formais (como o UML) que indicam as funcionalidades que podem ser implementadas, isto é, programadas em qualquer linguagem computacional para compor o sistema de informação desejado, que neste caso faz parte do *framework* de apoio a ambientes colaborativos.

Inicialmente, é apresentada a arquitetura de informação proposta para o sistema, considerando todos os elementos que a compõe, na seção 4.1. Essa arquitetura reutiliza tecnologias de comunicação que possuem implementações avançadas, como o SOA e os sistemas de gerenciamento de banco de dados disponíveis. Nessa arquitetura 3 elementos nominais são descrito: os clientes, as empresas e o *framework*.

Após isso, é apresentada algumas partes componentes da arquitetura que relacionam os elementos. Primeiramente, são apresentados os casos de uso (do UML) que relacionam o ator “Cliente” com o sistema “*Framework*”, na seção 4.3.

Em seguida, é apresentada as funções que relacionam as empresas com o *framework*, também na forma de diagrama de casos de uso, em que o ator é a “Empresa” e o sistema é o “*Framework*”, na seção 4.4.

Depois, é apresentada as funções que relacionam os clientes com as empresas, em que o ator é o “Cliente” e o sistema é a “Empresa”, do diagrama de casos de uso, na seção 4.5.

Por fim, é apresentada as funções que relacionam as empresas com outras empresas, em que o ator é uma “Empresa” e o sistema é outra “Empresa”, do diagrama de casos de uso, na seção 4.6.

Como uma necessidade na colaboração e cooperação é o estabelecimento de confiança entre as empresas, a seção 4.2 mostra um cálculo usado para mensurar a confiança na arquitetura proposta.

4.1 ARQUITETURA DE INFORMAÇÃO PROPOSTA DO SISTEMA

A arquitetura proposta para o sistema é uma forma de atender as necessidades do ambiente heterogêneo e colaborativo. Entende-se aqui a presença básica dos atores do sistema real (a economia atual) como dividida entre clientes e empresas, em uma dicotomia equivalente à “quem tem uma necessidade” e “quem pode atender uma necessidade”. Nessa divisão, a arquitetura proposta prevê a posição da empresa (ou de quem pode atender uma necessidade) como uma prestação de serviços e a utilização de meios eletrônicos de comunicação difundem ainda mais a possibilidade de que uma necessidade seja atendida. Assim, a arquitetura proposta se baseia no SOA. Para a empresa, a *framework* se comunica com as funções próprias de planejamento e logística (camada 4 da ANSI-ISA 95 (AMERICAN NATIONAL STANDARD, 2010)).

Os elementos reaproveitados do SOA na arquitetura são, a estrutura física (internet), os protocolos de comunicação (TCP/IP, UDP), os protocolos de apresentação (HTTP, FTP etc), os *middlewares* de integração de serviços (SOA, WS-CDL etc). Dessa maneira, as empresas se comportam como provedores de serviços e os clientes como requisitantes de serviços. Na realidade de ambiente colaborativo, uma requisição de cliente necessita da integração (ou composição) de mais de um serviços de diferentes empresas, gerando a necessidade de compor uma EV (ou, de forma equivalente, um serviço de maior complexidade).

Também é proposta, na arquitetura, a presença de um *framework* de apoio como um conjunto de ferramentas de serviços virtuais. Esse *framework* é utilizado pelo cliente ou contratante de um produto ou serviço. Por ser um *framework* de apoio, ele não se encontra centralizado nas comunicações, mas provê serviços para superar as dificuldades e custos para implantação de SIs em todas as pequenas e médias empresas dispostas a participar dessa arquitetura. Esse *framework* é mantido pelas próprias empresas participantes (ou grupos abertos) cujo interesse seja o de ampliar a quantidade de elementos dispostos a colaborar e cooperar. Os serviços presentes no *framework* devem estar também presentes, e de forma equivalente, em empresas que participem da arquitetura, para que não haja dependência total de seus serviços.

Ainda, por estar apenas no mundo lógico, o *framework* não possui equipamentos de produção ou recursos materiais, dentre outros requisitos para considerá-lo um SP. Assim sendo, o *framework* de apoio não é um prestador de serviço que participa diretamente da execução da produção. Então, a verdade oposta que é “o resultado que pode ser obtido sem a utilização do *framework* também pode ser obtido pela utilização do *framework*” não pode ser dita em todos os casos.

Globalmente, a arquitetura proposta pode ser vista como na Figura 15.

Figura 15 - Arquitetura global proposta

Na Figura 15, os clientes foram representados de 4 tipos: “Cliente humano”, “Cliente Empresa”, “Cliente governo”, “Outros clientes”. Entretanto, seus comportamentos em relação à arquitetura são os mesmos, mudando apenas sua origem do ponto de vista jurídico. Esses clientes tem o papel de oferecer uma requisição (que é uma demanda ou uma oportunidade de negócio) e utilizam os mecanismos dispostos no SOA e internet para encontrar serviços que sejam prestados para atender suas requisições.

As “EPPs” da Figura 15 (“EPP 1”, “EPP i” e “EPP n”) são as empresas parceiras potencial, que são as empresas que oferecem suas capacidades em forma de serviços. Outros serviços também podem ser oferecidos por elas para auxiliar a negociação entre empresas ou entre clientes e empresas.

Outro elemento dessa arquitetura é o *framework* de apoio que está descrito como “*Framework*”, na Figura 15. O *framework* de apoio conecta-se a um banco de dados próprio de sua aplicação (BD) e à nuvem. O *framework* é capaz de oferecer apenas alguns dos serviços que podem ser oferecidos pelas “EPPs”.

A descrição da arquitetura sob o ponto de vista das empresas na forma de “EPPs” pode ser vista na Figura 16.

Pela Figura 16, os componentes internos da arquitetura para as empresas na forma de EPPs são: “*Workflow*”, “*Confiança*”, “*Serviços para EV*”, “*SGBD*”, “*Serviços das capacidades*”. O

Figura 16 - Arquitetura do ponto de vista das empresas parceiras potencial

workflow é um componente que a empresa utiliza para descrever as etapas e a ordenação de uma execução, seja de um produto, seja de um serviço. Esse *workflow* depende, na arquitetura, de alguns elementos padronizados que são uma ferramenta matemática para formalização desse *workflow* e um conjunto de classes específicas padronizadas que possam detalhar as atividades realizadas nas etapas do *workflow*. A descrição da ferramenta matemática proposta para o *framework* (a RdPP) está no capítulo 5. Alguns exemplos da utilização do *workflow* estão no capítulo 6.

A “Confiança”, da Figura 16, é um componente das EPPs que auxilia na tomada de decisão (que é a atribuição de confiança entre participantes) quando se trata do problema de escolha de parceiros, da formação de uma EV. Esse componente consulta bases de dados para escolher as melhores EPPs segundo critérios que a própria empresa define, que são seus critérios de confiança para empresas. Neste caso, as empresas se dividem em três tipos:

- Empresa consultadora: uma empresa que busca outras empresas para obter avaliações de confiança sobre uma empresa avaliada;
- Empresa consultada: cada uma das empresas buscadas pela empresa consultadora para obter avaliação da confiança;
- Empresa avaliada: uma empresa sobre a qual se deseja saber os resultados de avaliações de confiança.

A descrição matemática dos cálculos de confiança propostos para a arquitetura podem ser

encontradas na seção 4.2.

O componente “Serviços para a EV”, da Figura 16, é composto por um conjunto de funções (ou serviços) de uma empresa, quando essa o possui, oferecidos para as outras empresas, para os clientes ou para o *framework*, para propor, negociar, formar, consultar ou outras atividades das etapas de formação de EVs. O conjunto de funções desse componente está descrito nas seções 4.3, 4.4, 4.5 e 4.6.

O componente “SGBD”, da Figura 16, é um sistema de gerenciamento de banco de dados implementado na empresa, quando essa o possui, que armazena os dados de suas produções ou de outras empresas que utilizam sua implementação de SI. Esse “SGBD” liga-se a uma “Base de Dados própria”, que é um banco de dados presente na estrutura do SI da empresa, e é responsável por coletar, armazenar e gerenciar os dados presentes no banco de dados. A tecnologia de sistemas de gerenciamento de banco de dados é reaproveitada nas diversas implementações, sendo importante para a arquitetura apenas as permissões de acesso a informação (que é o requisito da distribuição da informação) e a apresentação dessas informações (que é o requisito da universalização da informação).

O último componente é o “Serviços das capacidades”, pela Figura 16, que é o conjunto de capacidades da empresa oferecido na forma de serviços para execução de uma produção. Por se tratar de um SP, essa empresa pode se utilizar dos recursos materiais, financeiros, tecnológicos, humanos e energéticos, como representado na Figura 16. Outra necessidade para oferecer suas capacidades é informação que pode vir de duas origens diferentes: do seu próprio banco de dados pelo “SGBD”; dos bancos de dados distribuídos na internet entre empresas que participam ou têm interesse de participar de colaboração ou cooperação na arquitetura pelo “SODA”. O “SODA” é uma forma de representar bancos de dados distribuídos como serviços, compatíveis com a arquitetura proposta, como descrito na seção 2.2.3.

4.2 CÁLCULO DA CONFIANÇA

O cálculo da confiança entre empresas auxilia estas na tomada de decisão da escolha de parceiros para uma EV. Em Leina e Tiejun (2012), a confiança foi calculada usando o método QFD. Devido sua estrutura, esse método pode ser uma solução atrativa para cálculos mais complexos de confiança, mas também pode ser menos atrativo para áreas que necessitem de menos complexidade na determinação da confiança e mais agilidade na etapa de negociação. Outro ponto importante sobre o método proposto é que ele utiliza apenas as informações que uma empresa detém sobre a outra (é proposta a confiança para parcerias constantes e de longa duração) (LEINA; TIEJUN, 2012) e uma das necessidades para a confiança em EVs é conseguir avaliar a confiança de uma empresa sobre a qual não se tem informações (incluindo parcerias

pontuais e de curta duração). Em Mun, Shin e Jung (2011), é proposto um serviço central para cálculo de confiança com modelo orientado a objetivos em que inicialmente é gerada uma lista de parceiros potenciais para uma oportunidade de negócios utilizando fractais e após determinados os objetivos da empresa que recebeu a oportunidade de negócios, é feito um cálculo, usando lógica fuzzy, da confiança das empresas que foram inicialmente listadas pelo serviço. Esse cálculo de confiança é capaz de oferecer a confiança de uma empresa para outra sem que essas tenham conhecimento prévio uma da outra, mas é feita de forma centralizada em um serviço, o que reduz a flexibilidade da distribuição na rede.

A confiança aqui adotada é quantitativa e se baseia nas quantidades de sucessos e fracassos das produções que as empresas já trabalharam juntas. Além disso, as empresas interessadas em conhecer a confiança de outras empresas, podem questionar terceiras empresas sobre seus cálculos de confiança.

Para definir o cálculo será usada a seguinte nomenclatura:

- Empresa consultadora: uma empresa que busca outras empresas para obter avaliações de confiança sobre uma empresa avaliada;
- Empresa consultada: cada uma das empresas buscadas pela empresa consultadora para obter avaliação da confiança, somado a um conjunto de avaliações feitas da empresa avaliada feita pelas empresas cadastradas;
- Empresa avaliada: uma empresa sobre a qual se deseja saber os resultados de avaliações de confiança;
- Empresas cadastradas: empresas que estão no banco de dados de cadastro de empresas da empresa consultada.

O cálculo é feito em uma composição de medidas de confianças com avaliações de clientes e de empresas que estabeleceram parceria em algum projeto.

Em particular, a avaliação do cliente sobre um produto ou serviço pode aparecer de duas formas:

- Geral - em que o cliente avalia o produto ou serviço recebido sem especificar qual parte dele é problematizada;
- Específica - em que o cliente avalia o produto ou serviço recebido em suas partes.

A avaliação geral é registrada para todas as empresas que participaram das atividades do processo que levaram ao acordo com o cliente. Já a específica é registrada exclusivamente para as empresas que realizaram a etapa do processo que houve questionamento (positivo ou

negativo) por parte do cliente. A primeira possibilidade de avaliação se aplica a clientes que desconhecem o conjunto do produto ou serviço para dizer qual parte do processo causou sua satisfação ou insatisfação. Já a segunda possibilidade de avaliação pressupõe que o cliente tem um nível de conhecimento do produto ou serviço que o torne capaz de avaliá-lo parte a parte.

Para simplificação nessa proposta, a avaliação será feita em dois níveis para três critérios. Os níveis são satisfatório e insatisfatório. Os critérios aqui adotados são preço, tempo para entrega e qualidade do produto/serviço.

Essas avaliações são colocadas como índice nos indicadores: S_P para avaliações satisfatórias de preço; S_T para avaliações satisfatórias de tempo de entrega; S_Q para avaliações satisfatórias de qualidade do produto/serviço; I_P para avaliações insatisfatórias de preço; I_T para avaliações insatisfatórias de tempo de entrega; I_Q para avaliações insatisfatórias de qualidade do produto/serviço.

O cálculo da confiança para a empresa avaliada feita pela empresa consultada com base nas avaliações dos clientes será então feito como:

$$Conf_{cliente} = \frac{\omega_S \times (S_P \times \omega_P + S_T \times \omega_T + S_Q \times \omega_Q) - \omega_I \times (I_P \times \omega_P + I_T \times \omega_T + I_Q \times \omega_Q)}{N_{clientes}} \quad (1)$$

Na equação 1, ω_S , ω_I , ω_P , ω_T e ω_Q são os pesos dos níveis (satisfatório e insatisfatório, respectivamente) e dos critérios (preço, tempo de entrega e qualidade, respectivamente) que cada empresa consultada adota para calcular a confiança e o $N_{clientes}$ é o número de avaliações de clientes para a empresa avaliada. Os pesos são escolhidos pela empresa consultada, mas seguem as seguintes regras: $\omega_S + \omega_I = 1$ e $\omega_P + \omega_T + \omega_Q = 1$. Como as avaliações se apresentam em apenas dois níveis, pode-se concluir que $S_{VAR} + I_{VAR} = N_{clientes}$ para $VAR \in [C, T, Q]$. Sendo assim, para $\omega_S = 1$, a avaliação apenas se importa com os casos de satisfação dos clientes e para $\omega_S < 1$ aumenta-se a punição para os casos de falha.

A apresentação da avaliação da empresa avaliada pelos clientes deve ser feita apresentando também os valores dos pesos dos critérios e dos níveis adotados pela empresa consultada e dos valores específicos, para que a empresa consultadora também possa fazer o cálculo utilizando pesos diferentes. Outra informação que deve ser fornecida é a identificação dos projetos executados avaliados para evitar redundância de avaliações e para que seja possível rastrear as atividades problemáticas específicas dentro do rol de capacidades da empresa avaliada.

Os dados de entradas de avaliações dos clientes também podem ser restringidos. Essa restrição pode ocorrer por razões de tempo passado ou região do cliente (como verificar a avaliação do último ano apenas, ou apenas para clientes de um certo país).

Para a proposta, aqui, a avaliação de empresas sobre outras empresas é sempre específica, uma vez que essas tem o domínio do mesmo projeto e podem acompanhar todas as etapas.

Da mesma forma que para o cliente, a avaliação é feita com os mesmos dois níveis e três critérios. Dessa forma, a avaliação específica da empresa consultada sobre uma empresa avaliada fica como:

$$Conf_{Bemp} = \frac{\omega_S \times (S_P \times \omega_P + S_T \times \omega_T + S_Q \times \omega_Q) - \omega_I \times (I_P \times \omega_P + I_T \times \omega_T + I_Q \times \omega_Q)}{N_{projetos}} \quad (2)$$

A equação 2 tem a mesma linha da avaliação feita por clientes, exceto que a razão é feita por $N_{projetos}$ que é o número de avaliações da empresa avaliada em projetos com a empresa consultada.

Para avaliação final, a empresa consulta entre suas empresas cadastradas, as suas avaliações para a empresa avaliada. As respostas para essas consultas são registradas em $Conf_{Oempresa}$. O cálculo da confiança pelas avaliações de empresas cadastradas, então, tem a forma:

$$Conf_{empresa} = \omega_B \times Conf_{Bemp} + \omega_O \times \frac{\sum \omega_{Ai} Conf_{Oempresai}}{\sum \omega_{Ai}} \quad (3)$$

Na equação 3, ω_B e ω_O são os pesos entre a própria avaliação da empresa consultada sobre a empresa avaliada e a avaliação das empresas cadastradas sobre a empresa avaliada ($\omega_P + \omega_O = 1$), ou seja, medem a relação entre a opinião da empresa consultada em relação à opinião das empresas cadastradas. Os pesos ω_{Ai} são pesos que a empresa consultada aplica à opinião de cada uma de suas empresas cadastradas. Esse valor é uma forma de dizer qual a credibilidade que a empresa consultada dá à opinião de uma certa empresa cadastrada, por motivos de parceiras históricas ou razões sociais, como amizade, fama e outras.

Da mesma forma, os valores dos pesos e dos índices devem ser apresentados pela empresa consultada quando apresentada a confiança para que a empresa consultadora possa fazer seus cálculos utilizando outros pesos. Para esse caso, também é importante apresentar a identificação da empresa cadastrada que emitiu um certo valor de confiança $Conf_{Oempresai}$ para que os pesos de credibilidade possam ser apresentados. Os Quadros 1, 2 e 3 mostram um exemplo de relatório de confiança da empresa consultada (Empresa A) sobre a empresa avaliada (Empresa C) para a empresa consultadora (Empresa B).

Ao final, a confiança da empresa consultada sobre a empresa avaliada do exemplo se apresenta com o relatório da confiança composto pelo resumo de cálculos, presente na Figura 17, e os Quadros 1, 2 e 3. O primeiro fornece a informação do cálculo total da confiança e os demais fornecem os dados detalhados que permitem chegar a esse cálculo.

Essa proposta foi realizada levando em conta apenas os três critérios citados (preço, tempo

Quadro 1 - Exemplo de valores dos pesos para cálculo da confiança

Tipo de peso	Valor do peso
Satisfatório	$\omega_S = 0,9$
Insatisfatório	$\omega_I = 0,1$
Preço	$\omega_P = 0,6$
Tempo de entrega	$\omega_T = 0,1$
Qualidade	$\omega_Q = 0,3$
Própria	$\omega_B = 0,8$
Cadastradas	$\omega_O = 0,2$

Quadro 2 - Exemplo de valores dos indicadores da avaliação dos clientes e da empresa consultada

Tipo de indicador	Valor do indicador
Satisfatório de preço para o cliente	$S_P = 6$
Satisfatório de tempo de entrega para o cliente	$S_T = 4$
Satisfatório de qualidade para o cliente	$S_Q = 4$
Número de projetos avaliados por clientes	$N_{clientes} = 7$
Satisfatório de preço para a empresa	$S_P = 5$
Satisfatório de tempo de entrega para a empresa	$S_T = 3$
Satisfatório de qualidade para a empresa	$S_Q = 4$
Número de projetos avaliados pela empresa	$N_{projetos} = 8$

Quadro 3 - Exemplo de valores da confiança de empresas cadastradas

Empresa parceira	Credibilidade	Valor da confiança
Empresa D	$\omega_A = 0,5$	$Conf_{Oempresa} = 0,79$
Empresa E	$\omega_A = 0,6$	$Conf_{Oempresa} = 0,82$
Empresa F	$\omega_A = 1,0$	$Conf_{Oempresa} = 0,61$
Empresa G	$\omega_A = 0,6$	$Conf_{Oempresa} = 0,65$
Empresa H	$\omega_A = 1,0$	$Conf_{Oempresa} = 0,72$

de entrega e qualidade). Algumas áreas específicas do ambiente colaborativo podem ter outros critérios para ser adotados no cálculo, então, os critérios devem ser definidos pelas empresas da área para o cálculo da confiança.

Figura 17 - Exemplo de resumo dos cálculos da confiança da empresa consultada sobre a empresa avaliada

```

1 Relatório da confiança da Empresa A sobre a Empresa C.
2 Remetente: Empresa A
3 Destinatário: Empresa B
4 Empresa avaliada: Empresa C
5 Projetos avaliados: Manuf-5528.2003.109; Manuf-5528.2005.005; Manuf-5528.2005.007; Manuf-5528.2007.082
6 Manuf-5528.2010.015; Logist-4322.2010.002; Manuf-5528.2012.054; Logist-5528.2013.001
7 Confiança(cliente)=(0,9*(0,6*6+0,1*4+0,3*4)-0,1*(0,6*(7-6)+0,1*(7-4)+0,3*(7-4)))/7=0,64
8 Confiança(empresa)=(0,9*(0,6*5+0,1*3+0,3*4)-0,1*(0,6*(8-5)+0,1*(8-3)+0,3*(8-4)))/8=0,46
9 Confiança(completa de empresas)=0,8*0,46+0,2*(0,5*0,79+0,6*0,82+1,0*0,61+0,6*0,65+1,0*0,72)/(0,5+0,6+1,0+0,6+1,0)=0,51

```

4.3 CLIENTE-FRAMEWORK

Como descrito anteriormente, o cliente é um usuário do sistema que possui uma demanda a ser executada, na forma de um processo de negócio.

O *framework* oferece três funções para o cliente:

- Consultar empresas cadastradas;
- Consultar andamento da demanda;
- Avaliar atendimento da demanda.

A primeira função é uma possibilidade para o cliente de consultar entre as empresas que oferecem suas capacidades para o processo de negócio. Essa função utiliza o conceito de banco de dados distribuídos, pois as empresas podem se cadastrar em vários sistemas de gerenciamentos de bancos de dados diferentes e a consulta da função deve apresentar um resultado padronizado para o cliente.

A função “Consultar andamento da demanda” é uma função que o cliente pode usar para acompanhar seu pedido para empresas caso essas optem por um sistema *on-line* de monitoração. Esses sistema significa, para a empresa, reportar qual porcentagem da atividade de responsabilidade da empresa que já foi finalizada. Com as informações das empresas da porcentagem da atividades concluídas e a informação da visão geral do processo, essa função é capaz de calcular a porcentagem concluída do processo, estimar o tempo para conclusão do mesmo e reportar se os prazos estabelecidos estão sendo cumpridos.

A função “Avaliar atendimento da demanda” é uma função em que o cliente aponta suas satisfações e/ou insatisfações com o produto ou serviço recebido. Os dados dessa função ajudam nas medidas de confiança da empresa. A parte da avaliação do cliente para a medida de confiança será adotada aqui nos quesitos:

- Preço;

- Tempo para entrega;
- Qualidade do P/S.

Os cálculos da confiança serão apresentados na seção 4.2.

O diagrama de caso de uso para o cliente é apresentado na Figura 18.

Figura 18 - Diagrama de casos de uso para o cliente-*framework*

Os casos de uso da Figura 18 são detalhados em diagramas de atividades para elaborar etapas para alcançar o resultado final.

A função “Consultar empresas cadastradas” é detalhada pelo diagrama de atividades da Figura 19.

Na Figura 19, a função inicia consultando a própria base de dados da empresa ou do *framework* das empresas que estão cadastradas nela. Essas empresas cadastradas, obtidas nessa consulta, vão compor também o resultado da função. Além disso, a empresa pode consultar as empresas cadastradas, sobre suas próprias empresas cadastradas, para aumentar o banco de dados de empresas. Caso seja necessário fazer a consulta às empresas cadastradas, é preciso remover as redundâncias dos dados, que seriam as empresas que estão cadastradas em mais de uma base de dados. Após o conjunto de dados composto pelas empresas tanto da empresa consultadas, quanto das parceiras (caso necessário) é possível filtrar os dados segundo critérios estabelecidos pelo solicitante ou pela própria empresa consultada. Os filtros podem ser baseados em confiança, disponibilidade e outras informações que não são necessariamente incluídas no banco de dados. Por fim, os dados são retornados para a empresa ou cliente que realizou a consulta.

Figura 19 - Diagrama de atividades da função “Consultar empresas cadastradas”

A função “Consultar andamento da demanda” é detalhada pelo diagrama de atividades da Figura 20.

Figura 20 - Diagrama de atividades da função “Consultar andamento da demanda”

Na Figura 20, a função consulta na própria base de dados da empresa ou do *framework* sobre as atualizações do processo. Com esses dados e o *workflow*, calcula-se o quanto já foi concluído do processo. Faz-se então uma estimativa de tempo para o término no processo baseado no tempo já consumido e no tempo previsto e verifica se até o momento os prazos estabelecidos foram cumpridos. Por fim, os dados são retornados para a empresa ou cliente que realizou a consulta.

A função “Avaliar atendimento da demanda” é detalhada pelo diagrama de atividades da Figura 21.

Figura 21 - Diagrama de atividades da função “Avaliar atendimento da demanda”

Na Figura 21, a função consulta o *workflow* e as empresas que participaram dele. Após essa etapa, a empresa ou o *framework* aguarda o recebimento das avaliações do cliente. A função então verifica se a avaliação feita foi específica ou geral, sendo que a geral faz com que todas as empresas participantes recebam a mesma avaliação, enquanto a específica faz com que cada empresa seja avaliada por sua atividade. Após a verificação, os dados da avaliação são registrados nas bases de dados de acordo com o tipo de avaliação.

4.4 EMPRESA-FRAMEWORK

O ator empresa é uma empresa real (na forma de uma pessoa jurídica) interessada em participar de uma colaboração para atender uma oportunidade de negócio.

O *framework* oferece as seguintes funções para as empresas:

- Consultar empresas cadastradas;
- Cadastrar empresa;
- Alterar cadastro de empresa;

- Desativar cadastro de empresa;
- Consultar confiança da empresa;
- Consultar histórico de processos concluídos;
- Cadastrar processo concluído;
- Avaliar processo concluído;
- Atualizar *status* do processo em andamento;
- Alterar avaliação de processo concluído;
- Cadastrar projeto;
- Alterar projeto;
- Consultar projeto.

Assim como para o cliente, a função “Consultar empresas cadastradas” realiza uma consulta em bases de dados distribuídos nas empresas e no sistema próprio, as empresas que estão cadastradas e que oferecem suas capacidades para uma oportunidade de negócio.

A função “Cadastrar empresa” é implementada para incluir dados de uma empresa na base de dados do *framework*. Como o *framework* é de apoio, a consulta das empresas cadastradas não fica restrita à própria base de dados do *framework*, mas está também em cada empresa, na forma de uma rede de bancos de dados.

As funções “Alterar cadastro de empresa” e “Desativar cadastro de empresa” seguem a mesma ideia da função “Cadastrar empresa”, apenas alterando o tipo de consulta realizada (alteração de dados ao invés de inclusão de dados).

A função “Consultar confiança da empresa” é uma busca nas bases de dados das empresas cadastradas por uma estimativa do valor de confiança que a empresa possui. Essa estimativa pode ser feita na forma de uma média das confianças para todas as empresas cadastradas ou uma tabela de empresas cadastradas por valor da confiança. A primeira opção é um cálculo estatístico tradicional. Já a segunda opção é usada para que a empresa que está consultando gere o próprio cálculo da confiança, como explicado na seção 4.2.

A função “Consultar histórico de processos concluídos” tem dois objetivos principais para a empresa. O primeiro é saber como um processo de uma certa demanda foi realizado por motivos de projeto, sendo nesse caso os registros dos projetos desses processos. O segundo motivo é para recuperar históricos da execução do processo, como os custos (para estimativa de custos), os prazos para realização das etapas (para estimativas de prazos) e o comportamento

de uma empresa na execução do processo (para ter informações específicas de uma ou mais capacidades oferecida por tal empresa). Essa consulta é feita na base de dados do próprio *framework* e nas bases de dados específicas das empresas cadastradas no *framework*.

A função “Cadastrar processo concluído” serve para incluir dados de um processo que tenha sido concluído, com as informações necessárias para o registro histórico de processos.

A função “Avaliar processo concluído” oferece uma conjunto de dados para que as empresas avaliem um processo do qual participaram, colocando as informações específicas da execução do processo. Esses dados são usados para cacular a confiança das empresas.

A função “Alterar avaliação de processo concluído” tem a mesma ideia da função anterior, mas especificamente para alteração de avaliações que já haviam sido cadastradas.

A função “Atualizar *status* do processo em andamento” é um mecanismo para a empresa reportar a porcentagem do processo concluído e outras informações que permitam ao cliente acompanhar sua solicitação.

A função “Cadastrar projeto” é uma função iniciada por uma empresa para realização de um projeto colaborativo com outra empresa. Nessa função, uma empresa oferece estrutura (de *software* e banco de dados) para que as empresas que desejem estipular uma parceria possam cooperar e colaborar em um projeto de demanda específica. Essa função é importante para o atendimento da atribuição de desenvolvimento de produtos personalizados.

A função “Alterar projeto” é uma função que também mantém a relação de empresas para a colaboração no projeto de produtos personalizados. A diferença dessa função para a função “Cadastrar projeto” é que essa última se aplica a mudanças no projeto atualmente registrado. A etapa de alteração de um projeto já registrado é recorrente durante a fase de projeto, uma vez que essa fase interfere em todas as demais etapas, inclusive na execução da produção.

A função “Consultar projeto” é uma função que permite às empresas estudarem o estado do projeto (não identificado, concluído ou em andamento).

O diagrama de caso de uso para a empresa é apresentado na Figura 22.

A função “Consultar empresas cadastradas” segue o mesmo diagrama de atividades e a mesma descrição da Figura 19.

A função “Cadastrar empresa” é detalhada pelo diagrama de atividades da Figura 23.

Na Figura 23, a função inicia recebendo os dados da empresa que deseja se cadastrar na base do *framework* ou de outra empresa parceira. A função então registra os dados na base de dados adequada.

As funções “Alterar cadastro de empresa” e “Desativar cadastro de empresa” são detalhadas pelos diagramas de atividades das Figuras 24 e 25, respectivamente.

Na Figura 24, a função inicia consultando os dados registrados da empresa. Após isso, a função recebe os dados da empresa que serão alterados na base de dados do *framework* ou

Figura 22 - Diagrama de casos de uso para o empresa-framework

Figura 23 - Diagrama de atividades da função “Cadastrar empresa”

da outra empresa parceira. A função então registra as alterações os dados na base de dados adequada.

Figura 24 - Diagrama de atividades da função “Alterar cadastro de empresa”

Figura 25 - Diagrama de atividades da função “Desativar cadastro de empresa”

De forma semelhante, na Figura 24, a função inicia consultando os dados registrados da empresa. A função, então, registra uma alteração nos dados da empresa que é considerar como desativados.

A função “Consultar confiança da empresa” é detalhada pelo diagrama de atividades da Figura 26.

Na Figura 26, a função inicia selecionando alguns dados de sucessos e fracassos referentes aos indicadores usados na base de cálculo, pelo histórico de processos realizados em parceria com a empresa avaliada. Depois, a função recebe (ou importa do banco de dados, caso já estejam determinadas) os valores de pesos para os indicadores, de acordo com os cálculos na seção 4.2. Então a função consulta a confiança de suas empresas cadastradas em relação a empresa avaliada. A função então calcula o valor da confiança, de acordo com os dados anteriores obtidos e retorna o valor na forma de relatório para o solicitante.

A função “Consultar histórico de processos concluídos” é detalhada pelo diagrama de ativi-

Figura 26 - Diagrama de atividades da função “Consultar confiança da empresa”

dades da Figura 27.

Figura 27 - Diagrama de atividades da função “Consultar histórico de processos concluídos”

Na Figura 27, a função consulta na própria base de dados, os projetos que estão registrados. Com os projetos, a função importa os dados referentes ao andamento e a conclusão dos projetos. A função então verifica se é preciso buscar mais informações de projetos com as empresas cadastradas em sua base de dados. Caso seja preciso, a empresa consulta as empresas cadastradas em sua base de dados por histórico de processos concluídos. Por fim, a função retorna o relatório com as informações obtidas.

A função “Cadastrar processo concluído” é detalhada pelo diagrama de atividades da Figura 28.

Na Figura 28, a função importa a estrutura do *workflow* proposta para o processo. A função então recebe o valor das informações do processo (andamento, custo, preço, prazos, técnicas, problemas e outras que julgar necessárias) realizado. Por fim a função grava as informações no banco de dados adequado.

A função “Avaliar processo concluído” é detalhada pelo diagrama de atividades da Figura 29.

Figura 28 - Diagrama de atividades da função “Cadastrar processo concluído”

Figura 29 - Diagrama de atividades da função “Avaliar processo concluído”

Na Figura 29, a função importa os dados do processo concluído (satisfatória ou insatisfatória). A função recebe as avaliações específicas da empresa avaliadora. A função registra a avaliação no banco de dados adequado.

A função “Atualizar *status* do processo em andamento” é detalhada pelo diagrama de atividades da Figura 30.

Figura 30 - Diagrama de atividades da função “Atualizar status do processo em andamento”

Na Figura 30, a função importa os dados do *workflow* do processo nas bases de dados. A função recebe a atualização de *status* de uma empresa. A função grava a atualização no banco de dados adequado.

A função “Alterar avaliação de processo concluído” é detalhada pelo diagrama de atividades da Figura 31.

Figura 31 - Diagrama de atividades da função “Alterar avaliação de processo concluído”

Na Figura 31, a função importa os dados do *workflow* do processo nas bases de dados. A função então consulta a avaliação anteriormente registrada nas bases de dados. A função recebe a alteração da avaliação do processo. A função grava a alteração no banco de dados adequado.

A função “Cadastrar projeto” é detalhada pelo diagrama de atividades da Figura 32.

Figura 32 - Diagrama de atividades da função “Cadastrar projeto”

Na Figura 32, a função recebe os dados de um novo projeto proposto. A função então cria o banco de dados necessário para armazenar o desenvolvimento do projeto e registra as

informações recebidas nesse banco de dados criado.

A função “Alterar projeto” é detalhada pelo diagrama de atividades da Figura 33.

Figura 33 - Diagrama de atividades da função “Alterar projeto”

Na Figura 33, a função importa os dados da base de dados do projeto. Depois, a função recebe os pedidos de alteração dos dados gravados do projeto. Por fim, a função registra as alterações solicitadas no banco de dados do projeto.

A função “Consultar projeto” é detalhada pelo diagrama de atividades da Figura 34.

Figura 34 - Diagrama de atividades da função “Consultar projeto”

Na Figura 34, a função importa os dados da base de dados do projeto. A função, então, retorna os dados importados para a empresa solicitante.

4.5 CLIENTE-EMPRESA

As empresas oferecem serviços específicos para os clientes. Os serviços oferecidos pelas empresas são os mesmos oferecidos pelo *framework*, com o acréscimo das funções:

- Solicitar P/S;
- Alterar solicitação de P/S;
- Remover solicitação de P/S;
- Consultar proposta;
- Confirmar proposta acordada;
- Negociar proposta;
- Negar proposta.

A função “Solicitar P/S” é utilizada pelo cliente para gerar uma demanda na forma de oportunidade de negócio. Nesse caso a empresa apresenta as informações básicas do produto ou serviço da demanda do cliente para que possa ser avaliada quanto a sua capacidade de atender e as condições da mesma.

As funções “Alterar solicitação de P/S” e “Remover solicitação de P/S” tem comportamento semelhante a função “Solicitar P/S” mas com alterações de dados ao invés de inclusão de dados.

A função “Consultar proposta” é utilizada pelo cliente para verificar a proposta que foi feita pela empresa para sua oportunidade de negócio. Essa proposta surge após a primeira parte da formação da empresa virtual, ou seja, após as empresas interessadas na oportunidade de negócio se manifestarem com as informações das etapas que podem realizar, preço, prazo e qualidade.

A função “Confirmar proposta acordada” é a aceitação, pelo cliente, da proposta que está fixada pela empresa. Essa aceitação faz com que as empresas parceiras formalizem o contrato de colaboração/cooperação para atendimento da oportunidade de negócio.

A função “Negociar proposta” é feita para que o cliente ofereça uma contraproposta à proposta feita pela empresa. De posse da contraproposta, a empresa deve transmitir para suas empresas parceiras as possíveis alterações da proposta. Após essa etapa, é gerada uma nova proposta pela empresa, que pode ser a própria contraproposta, uma outra proposta ou a rejeição.

A função “Negar proposta” é utilizada quando as tentativas de tratativas entre cliente e empresa não apresentam condições que o cliente considere satisfatórias para ensejar o acordo.

O diagrama de caso de uso para a relação cliente-empresa é apresentado na Figura 35.

Figura 35 - Diagrama de casos de uso para o cliente-empresa

Da Figura 35, as funções “Consultar empresas cadastradas”, “Consultar andamento da demanda” e “Avaliar atendimento da demanda” possuem o mesmo detalhamento em diagramas de atividades que para a relação cliente-*framework* (nas Figuras 19, 20 e 21, respectivamente).

A função “Consultar proposta” é detalhada pelo diagrama de atividades da Figura 36.

Figura 36 - Diagrama de atividades da função “Consultar proposta”

Na Figura 36, a função importa os dados da última proposta elaborada pela empresa. A

função, então, retorna essa proposta para o solicitante e oferece 3 possibilidades: Aceitar, Rejeitar ou Fazer nova proposta. Caso o solicitante opte por aceitar a proposta, a função chama a outra função “Confirmar proposta acordada”. Caso o solicitante opte por rejeitar a proposta, a função chama a outra função “Negar proposta acordada”. E caso o solicitante opte por fazer nova proposta, a função chama a outra função “Negociar proposta acordada”.

As funções “Negociar proposta acordada”, “Confirmar proposta acordada” e “Negar proposta acordada” são detalhadas pelos diagramas de atividades das Figuras 37, 38 e 39, respectivamente.

Figura 37 - Diagrama de atividades da função “Negociar proposta acordada”

Figura 38 - Diagrama de atividades da função “Confirmar proposta acordada”

Na Figura 37, a função recebe os dados da proposta de negociação. A função redistribui,

Figura 39 - Diagrama de atividades da função “Negar proposta acordada”

entre as empresas que realizaram a proposta anterior, novas solicitações de propostas com base na negociação feita.

Na Figura 38, a função responde às empresas cadastradas que fizeram ofertas na proposta aceita a confirmação da proposta. A função elabora o contrato para produção e envia para o solicitante.

Na Figura 39, a função responde às empresas cadastradas que elaboraram a proposta rejeitada, que essa foi negada. A função remove a proposta da lista de propostas que ainda podem ser respondidas.

A função “Solicitar P/S” é detalhada pelo diagrama de atividades da Figura 40.

Figura 40 - Diagrama de atividades da função “Solicitar P/S”

Na Figura 40, a função recebe a solicitação do produto ou do serviço desejado. A função verifica se há necessidade de busca de parceiras entre as empresas cadastradas para atender a demanda. Caso haja necessidade, ou seja, as atividade ainda não puderem ser cumpridas, a

função procura entre as empresas cadastradas as capacidades que ainda faltam para atender. A empresa então faz a solicitação para as empresas encontradas. Em seguida, ou caso não houvesse necessidade de outras empresas, a função elabora a proposta para o produto ou serviço solicitado.

A função “Alterar solicitação de P/S” é detalhada pelo diagrama de atividades da Figura 41.

Figura 41 - Diagrama de atividades da função “Alterar solicitação de P/S”

Na Figura 41, a função importa os dados da solicitação do produto ou do serviço que se deseja alterar. Depois, a função recebe as alterações que o solicitante deseja. Por fim, a função processa as alterações substituindo o que estava anteriormente solicitado por novos dados e refaz o processo de “Solicitar P/S”.

A função “Remover solicitação de P/S” é detalhada pelo diagrama de atividades da Figura 42.

Figura 42 - Diagrama de atividades da função “Remover solicitação de P/S”

Na Figura 42, a função interrompe a busca de empresas cadastradas para parceria da demanda. E a função remove as solicitações nas empresa que já foram solicitadas para tal demanda.

4.6 EMPRESA-EMPRESA

As funções aqui representadas servem para empresas que procuram em outras empresas opções de informações e negócios.

As funções (ou serviços) oferecidos são os mesmos das oferecidas pelo *framework* para a empresa, com acréscimo das funções:

- Solicitar P/S;
- Alterar solicitação de P/S;
- Remover solicitação de P/S;
- Consultar proposta;
- Confirmar proposta acordada;
- Negociar proposta acordada;
- Negar proposta acordada;
- Consultar andamento da atividade.

A exceção da função “Consultar andamento da atividade” todas as demais novas funções possuem a mesma descrição das homônimas presentes nos diagramas para cliente-empresa (Figura 35)

A função “Consultar andamento da atividade” é um levantamento que uma empresa pode fazer com uma empresa parceira sobre como está a execução das atividades acordadas, segundo as condições de contrato.

O diagrama de caso de uso para a empresa é apresentado na Figura 43.

A função “Consultar andamento da atividade” é detalhada pelo diagrama de atividades da Figura 44.

Na Figura 44, a função importa da base de dados o processo que se refere a atividade consultada. A função aguarda, do departamento responsável, a informação do *status* do processo. A função retorna os dados ao solicitante.

Figura 43 - Diagrama de casos de uso para o empresa-empresa

Figura 44 - Diagrama de atividades da função “Consultar andamento da atividade”

A função “Solicitar P/S”, descrita na Figura 40, oferece a possibilidade de estabelecer uma proposta para a solicitação sem necessariamente passar pela busca de parcerias entre as empresas cadastradas. Com isso, e considerando que a função “Confirmar proposta acordada” pode ser concluída mesmo não tendo outras empresas parceiras, entende-se que a arquitetura

proposta permite a contratação individual entre empresas.

4.7 ANÁLISE DAS HIPÓTESES NA ARQUITETURA

Considerando a universalização da informação como a padronização de sintaxe e semântica e a acessibilidade/facilidade do *workflow* para o ambiente colaborativo, a arquitetura proposta conecta o *workflow* com o STEP (que padroniza a sintaxe) e com a rede de Petri produtiva (RdPP). Como será mostrado posteriormente, a RdPP possui uma estrutura de informação baseada em PNML que oferece uma semântica padronizada. Ainda sobre a RdPP, sua acessibilidade e facilidade é também alvo de testes nesse trabalho.

Sobre a hipótese de bancos de dados distribuídos, a utilização de SODA e de SOA são duas soluções para a distribuição de dados na arquitetura proposta.

Sobre a hipótese da sub-contratação de empresas, como a função “Solicitar P/S” pode ser disponibilizada tanto por empresas quanto pelo *framework* de apoio, as etapas de solicitação até contratação não necessariamente precisam passar por um elemento central, sendo possível às empresas negociarem entre si. Além disso, essas funções possuem fluxos alternativos para que não seja necessária a busca de outras parceiras, tal como para contratação de serviço (na função “Confirmar proposta acordada”).

Sobre a hipótese de confiança entre as empresas, uma das funções (na forma de serviços) descritas pela arquitetura envolve o envio de avaliações das atividades realizadas nos processos de negócios. Essa função, acompanhada do cálculo da confiança baseado nas avaliações pode ser utilizada pelas empresas para escolherem suas parceiras com base nos históricos de produção.

Sobre a hipótese de projeto colaborativo, a arquitetura tem abertura para essa necessidade, uma vez que o projeto é parte dos serviços oferecidos por cada empresa.

4.8 SÍNTESE DO CAPÍTULO

Sobre as hipóteses levantadas no capítulo 3, foi feito um estudo analítico da arquitetura proposta, sobre a aderência a elas. Com base nesse estudo, pode se dizer que quase todas as hipóteses são aderentes à arquitetura. Uma que ainda necessita de proposta mais clara é a hipótese de projeto colaborativo.

As implementações próprias feitas por empresas (incluídas em seus próprios SIs) poderiam ser uma solução para o projeto colaborativo, mas para empresas que não possuem os SIs de alta complexidade, esse pode se tornar um fator limitante para participarem em todas as etapas da colaboração.

O estudo mais detalhado do *workflow* é apresentado nos próximos capítulos para que esse também seja testado quanto sua aderência às hipóteses do trabalho.

5 REDE DE PETRI PRODUTIVA

A descrição de *workflow* envolve a troca de informações entre empresas dispostas a colaborar. Essa troca de informações é prevista nesse trabalho de forma padronizada, tanto pela linguagem que ela utiliza, chamada de sintaxe, que na arquitetura proposta, adota os padrões do STEP, quanto de semântica. Além disso, uma parte importante da informação para os SPs é o fluxo das atividades descrito na forma de informação. Por necessidade das empresas, esse fluxo das atividades deve ter uma representação simples, que as permitam compreender o processo, como elas se enquadram nesse processo e propor alternativas a ele, sem que para isso necessitem ser especialistas na ferramenta utilizada.

Para isso, uma ferramenta não pode apenas dar suporte a um desses padrões e necessidades, mas a todos eles. Nesse capítulo, é proposta uma ferramenta de *workflow* que se baseia na RdP (que possui uma interpretação relativamente simples sobre fluxo) e oferece suporte para os padrões de sintaxe e semântica. Essa ferramenta se chama rede de Petri produtiva (RdPP).

Na seção 5.1, é apresentada a divisão em camadas da informação da RdPP. Na seção 5.2, é apresentada a descrição do controle em nível de equipamentos e controladores da informação em RdPP. Na seção 5.3, é apresentada a descrição em nível de supervisão da informação em RdPP. Na seção 5.4, é apresentada a descrição em nível de colaboração da informação em RdPP. Na seção 5.5, algumas características de processos produtivos são apresentadas e comparadas com as representações das técnicas existentes. Na seção 5.6, o modelo de informação com a semântica da RdPP, baseada em XML, é apresentado.

5.1 ABORDAGEM CONCEITUAL

A abordagem de informação adotada pela RdPP considera a existência de três níveis de abstração que podem ser utilizadas de forma diferentes em estratégias diferentes no contexto de produção. Não há uma correlação direta entre as camadas das informações (com diferentes níveis de abstração) na RdPP com os níveis de controle da ANSI-ISA 95 (AMERICAN NATIONAL STANDARD, 2010), pois um mesmo nível de controle da última pode utilizar mais de uma camada de informação da primeira.

Assim, as camadas de informação da RdPP são separadas de acordo com necessidades específicas para execução de processos produtivos em ambientes colaborativos e as diferenças entre as informações de camadas diferentes são pelo nível de abstração que um controle necessita. Na camada inferior, a informação descreve a comunicação entre os dispositivos de controle e de atuação e detecção dos equipamentos, como, por exemplo, o envio de sinais para o movimento de um eixo e a recepção de sinais vindo de um sensor que indica a posição

desse eixo. Numa camada imediatamente acima, a informação envolve um conjunto atômico e estruturado de ações (tarefas) que o equipamento deve executar e, combinadas com as ações de outros equipamentos, garantem a execução coordenada dos processos em uma instalação. Na camada superior, há informações que garantem a colaboração entre as unidades produtivas (empresas/instalações), e esta informação trata da interação entre os estágios do processo produtivo. Esta abordagem conceitual pode ser vista na Figura 45. A separação entre camadas permite explorar soluções tecnológicas diferentes que podem implementar funções específicas para cada camada. Entretanto, esta abordagem requer interfaces bem definidas entre as diferentes camadas para permitir a comunicação entre elas.

Figura 45 - Representação das camadas de controle de um processo produtivo de manufatura

5.2 CAMADA DE EQUIPAMENTOS E CONTROLADORES

Na abordagem conceitual da Figura 45, na camada de “Equipamentos e controladores”, o nível de abstração das informações é o utilizado por programas de controle de máquinas que atuam diretamente em dispositivos de monitoração e atuação e coleta sinais de dispositivos de comando ou detecção. Nesta camada, algumas linguagens de programação são derivadas de representações em nível de sinais elétricos, como a linguagem *ladder*, ou são derivados de técnicas de modelagem de processos produtivos, como a SIPN (IVANOVA; FREY; BATCH-KOVA, 2008), MFG (TAKAHASHI; HASEGAWA; MIYAGI, 1999), ou o Grafset (DAVID, 1995). Esse nível de abstração é mais utilizado pelo controle discreto do nível 2 da ANSI-ISA 95 (AMERICAN NATIONAL STANDARD, 2010).

A informação de envio e recepção de sinais por um controlador de um equipamento pode ser chamada de comando atômico (CA). Um CA é interpretado em 3 estados: um estado “enviando sinal”; um estado “sinal enviado”; e um estado “ocioso”. No primeiro estado, um envio de sinal ocorre do controlador para um atuador do equipamento. No segundo estado, o envio de sinal é interrompido. No terceiro estado, não ocorre nada. A mudança entre o primeiro e o segundo estados ocorre pela recepção de um sinal pelo controlador vindo de um sensor do mesmo equipamento. Quando um CA é iniciado (estado “enviando sinal”), é dito

que este CA está sendo executado. Quando recebe o sinal do sensor e muda de estado (estado “sinal enviado”) é dito que o CA teve sua execução verificada.

Um CA, sua execução, sua verificação da execução e a detecção do sinal podem ser modelados por uma SIPN composta de 2 lugares, 1 transição, 2 arcos orientados, 1 arco de sinal saindo para o atuador e 1 arco de sinal de um sensor entrando na transição. Neste contexto, se a marcação do lugar que é pré-conjunto da transição não é nulo, a interpretação é de que o CA está sendo executado, ou seja, o estado é o “enviando sinal. A Grafo 9 mostra o modelo de um CA em SIPN e em MFG. No SIPN, o lugar p_{AiS} habilita o envio de sinal para um atuador de um equipamento e a transição espera a vinda de um sinal vindo de um sensor deste mesmo equipamento para estar habilitada para o disparo. Quando a transição t_{Ai} é disparada, o lugar p_{AiS} perde sua marcação e o atuador do equipamento para de trabalhar, indicando que o CA foi finalizado.

Grafo 9 - Modelo de comando atômico

Assim, para os CAs, há um modelo SIPN $ca = (P_A, T_A, F_A, M_A, IS_A, OS_A, \varphi, \omega)$, em que: P_A para um CA é $P_{Ai} = (p_{AiS}, p_{AiV})$; T_A para um CA é $T_{Ai} = (t_{Ai})$; F_A para um CA é $F_{Ai} \subseteq ((p_{AiS} \times t_{Ai}), (t_{Ai} \times p_{AiV}))$; M_A para um CA pode variar entre $M_{Ai} = (0, 0)$ (estado “ocioso”), $M_{Ai} = (1, 0)$ (estado “enviando sinal”), $M_{Ai} = (0, 1)$ (estado “sinal enviado”); IS_A para um CA é $IS_{Ai} = (is_{Ai})$; OS_A para um CA é $OS_{Ai} = (os_{Ai})$; φ para um CA é $\varphi(t_{Ai}) = is_{Ai}$; e ω para um CA é $\omega(p_{AiS}) = os_{Ai}$.

O modelo do CA atende as condições de Wf-net (da seção 2.3.3), pois $\bullet p_{AiS} = \emptyset$ e $p_{AiV}^\bullet = \emptyset$ e caso incluído t_{aux} tal que $\bullet t_{aux} = \{p_{AiV}\}$ e $t_{aux}^\bullet = \{p_{AiS}\}$ no novo modelo $\nexists t \in \overline{T_A} | \bullet t = \emptyset$, $\nexists t \in \overline{T_A} | t^\bullet = \emptyset$, $\nexists p \in \overline{P_A} | \bullet p = \emptyset$ e $\nexists p \in \overline{P_A} | p^\bullet = \emptyset$, ou seja, um CA pode ser usado como um modelo formal de um processo produtivo.

5.3 CAMADA DE CONTROLE SUPERVISÓRIO

Na abordagem conceitual da Figura 45, na camada de “Controle supervisório”, o nível de abstração da informação é o utilizado por programas de controle que envolvem sequências de

atividades de um conjunto de equipamentos. Nesta camada, as linguagens de programação definidas pela ISO 6983 (CALABRESE; CELENTANO, 2007) são as mais usadas nos processos produtivos na área de manufatura, mas há projetos de padronização dessas linguagens, como o STEP-nc. Algumas outras áreas de processos produtivos envolvem transporte, armazenamento, montagem, (reparo/assistência) e outras. Esse nível de abstração é utilizado pelos gerenciamento de operações (nível 3 da ANSI-ISA 95 (AMERICAN NATIONAL STANDARD, 2010)), para a supervisão e o despacho da produção, e planejamento de negócios (nível 4 da ANSI-ISA 95 (AMERICAN NATIONAL STANDARD, 2010)), para determinação das capacidades da empresa.

Em processos produtivos, algumas sequências de CAs de um equipamento são similares a outras por enviarem os mesmos tipos de sinais na mesma ordem para equipamentos de mesmo tipo, ou os resultados da execução destes CAs em um equipamento com diferentes controladores são os mesmos. Neste caso, essas sequências de CAs, que podem ser vistas como etapas do processo produtivo, podem ser padronizadas para equipamentos com diferentes controladores, mas que tenham a mesma função.

Uma forma para padronização de sequências de CAs é interpretá-las como instruções, que são elementos que descrevem estas sequências.

Formalmente, uma instrução i é definida como $i = (CA_I, P_I, T_I, F_I, M_I)$, em que $CA_I = \{ca_{I1}, \dots, ca_{In}\}$ é um conjunto de CAs em sequência; P_I é um conjunto de lugares $P_I = (p_{IS}, p_{IV})$; T_I é um conjunto de transições, em que $ca_{Ii} \in CA_I$ e $ca_{I(i+1)} \in CA_I$, com $i = 1, 2, \dots, n-1$, são dois CAs consecutivos (ou que são executados sequencialmente) de uma instrução i_j , assim, defini-se que $\exists t_{Ii} \in T_I / t_{Ii} \in p_{AiV}, t_{Ii} \in p_{A(i+1)S}, \exists t_{I0} \in T_I / t_{I0} \in p_{IS}, t_{I0} \in p_{A1S}$ e $\exists t_{In} \in T_I / t_{In} \in p_{AnV}, t_{In} \in p_{IV}$; F_I é o conjunto de arcos $F_I \subseteq \bigcup_{i=1}^{n-1} ((p_{AiV} \times t_{Ii}), (t_{Ii} \times p_{A(i+1)S})) \cup ((p_{IS} \times t_{I0}), (t_{I0} \times p_{A1S}), (p_{AnV} \times t_{In}), (t_{In} \times p_{IV}))$, com $p_{AkV}, p_{AkS} \in P_{Ak} | P_{Ak} \in ca_{Ik}, t_{Ik} \in T_I, p_{IS}, p_{IV} \in P_I$; M_I é o vetor de marcação dos lugares, em que $M_I(p) = \mathbb{N}$, quando $M_I = (1, 0)$ significa que a instrução, composta de CAs sequenciais CA_I , pode ser executada, e quando $M_I = (0, 1)$ significa que a mesma instrução já foi executada e todos CAs foram verificados. A Grafo 10 mostra a representação de uma instrução aqui descrita.

Para simplificar a modelagem e facilitar o entendimento de uma instrução em um processo produtivo, um conjunto de CAs sequenciais e algumas transições e arcos de uma instrução podem ser interpretados como uma linguagem textual de alto nível. Assim, o modo simplificado de apresentar as instruções segue o modelo da Grafo 11.

Esta simplificação gráfica da interpretação esconde informações que podem ser vistas como execução automática na compreensão do processo requisitado. A informação a respeito de quais sinais de sensores dos equipamentos devem ser recebidos e para quais atuadores os sinais

Grafo 10 - Modelo de instrução que pode ser executada

Grafo 11 - Modelo simplificado de instrução que pode ser executada

devem ser enviados ficam escondidas na representação textual. O sistema supervisorio deve garantir a ativação na sequência correta dos sinais de atuadores e sensores do equipamento, como especificado na instrução. Por essa razão, a RdPP adota representações textuais de linguagens padronizadas para as instruções.

Esta simplificação também é uma forma de oferecer suporte às linguagens textuais padronizadas, principalmente no que se diz sobre a sintaxe.

O modelo de instrução atende as condições de Wf-net (da seção 2.3.3), pois $\bullet p_{IS} = \emptyset$ e $p_{IV}^{\bullet} = \emptyset$ e caso incluído t_{aux} tal que $\bullet t_{aux} = \{p_{IV}\}$ e $t_{aux}^{\bullet} = \{p_{IS}\}$ no novo modelo $\nexists t \in \overline{T_I} | \bullet t = \emptyset$, $\nexists t \in \overline{T_I} | t^{\bullet} = \emptyset$, $\nexists p \in \overline{P_I} | \bullet p = \emptyset$ e $\nexists p \in \overline{P_I} | p^{\bullet} = \emptyset$, ou seja, uma instrução pode ser usada como um modelo formal de um processo produtivo.

5.4 CAMADA DE CONTROLE COLABORATIVO

Na abordagem conceitual da Figura 45, na camada de “Controle colaborativo”, o nível de abstração da informação contida foca, principalmente, em programas de controle de processos produtivos que usam múltiplas e diferentes instalações e/ou equipamentos. Nesta camada, ainda não há técnicas padronizadas, mas há algumas derivações da RdP que podem ser aplicadas neste contexto. Entretanto, essas técnicas, como a RdP colorida e a Wf-net, são relativamente complicadas para serem exploradas sem um elevado conhecimento prévio das suas propriedades, o que dificulta o entendimento do processo produtivo descrito. A RdPP propõe uma solução para essa dificuldade estabelecendo regras e procedimentos para sistematizar a construção de modelos. Esse nível de abstração é mais utilizado pelo planejamento de negócios (nível 2 da ANSI-ISA 95 (AMERICAN NATIONAL STANDARD, 2010)), para

inclusão de novas oportunidades de negócios.

Além da ligação com os processos nas camadas inferiores das instruções, a RdPP oferece maneiras de identificar os padrões de *workflow* quando o processo produtivo tem mais de uma instrução para ser executado. Esta representação considera a hipótese de diferentes equipamentos com diferentes funções que podem ser usadas em etapas diferentes de um mesmo processo produtivo.

Nesse contexto, a RdPP possui caixas de instruções, que incluem uma ou mais instruções que podem ser executadas em uma etapa do processo produtivo global. Uma caixa de instruções inclui um conjunto de informações comuns às instruções contidas, como as propriedades dos materiais antes da execução do processos produtivos e os tipos de equipamentos necessários para a execução destas instruções.

Formalmente, uma caixa de instruções B é definida como $b = (I_B, P_B, T_B, F_B, M_B)$, em que I_B é o conjunto de instruções contidas na caixa de instruções; P_B é o conjunto de lugares $P_B = (p_{Bh}, p_{Ba})$; T_B é o conjunto de transições $T_B = (t_{BS}, t_{BE})$; F_B é o conjunto de arcos, para cada $p_{ISi}, p_{IVi}, t_{IOi}, t_{Ini} \in i_{Bi}$ com $i_{Bi} \in I_B$, $F_B \subseteq \bigcup_i ((t_{BS} \times p_{ISi}), (p_{IVi} \times t_{BE})) \cup ((p_{Bh} \times t_{IOi}), (t_{Ini} \times p_{Bh})) \cup ((p_{Ba} \times t_{BS}), (t_{BE} \times p_{BA}))$; e M_B é o vetor de marcações dos lugares, em que quando $M_B = (\mathbb{N}, 0)$ significa que as instruções da caixa de instruções estão sendo executadas, e quando $M_B = (0, 1)$ significa que a caixa de instruções não está sendo executada. A Grafo 12 mostra a representação da caixa de instruções, no estado de não sendo executada.

Grafo 12 - Modelo de caixa de instruções

Para simplificar graficamente o modelo, a caixa de instruções é indicada como um elemento retangular com os textos e suas funções dentro, como no Grafo 13. Nesta representação, há um espaço para incluir a informação de tipo de processo (ou o tipo de equipamento usado), que

pode juntar as instruções em classes (conceitualmente esse campo é uma base de referência ou biblioteca com as informações referente às instruções), e as informações do objeto (que podem incluir propriedades do material, dimensões, expressões de guarda e outros tipos de informações que podem ser indicadas de forma booleana), que provê dimensões ou outras propriedades do objeto/serviço usados como entrada na execução das instruções. Estas duas informações são opcionais no processo descrito em uma caixa de instruções, mas são importantes para se obter os resultados esperados.

Grafo 13 - Modelo simplificado da caixa de instruções

Na representação da Grafo 13, as transições t_{BS} e t_{BE} são apresentadas de modo aglutinado como uma única transição. Neste caso, o retângulo e seu conteúdo são a caixa de instruções $b_i \in B_P$ e a transição $t_j \in T_P$ estão associadas uma a outra, em que $B_P(t_j) = b_i$, ou $b_i \times t_j \neq \emptyset$. Esta representação reduz a complexidade da descrição detalhada dos processos, deixando ao controle colaborativo a função de lidar com esta complexidade, usando as técnicas de interpretação apropriadas.

A estrutura da RdPP é definida como $PPN = (B_P, P_P, T_P, F_P, M_P)$, em que B_P é o conjunto de caixas de instruções; P_P é o conjunto de lugares; T_P é o conjunto de transições; F_P é o conjunto de arcos orientados (com $F_P \subseteq (P_P \times T_P) \cup (T_P \times P_P)$); e M_P é o vetor de marcações dos lugares em P_P , dividido em dois grupos $M_P = M_{PF} + M_{PR}$, em que M_{PF} é o vetor de marcações livres (que são marcações que podem ser usadas para habilitar uma transição para execução) e M_{PR} é o vetor de marcações reservadas (que são marcações que não podem ser usadas para habilitar uma transição para execução). Para cada elemento do conjunto de B_P , há um elemento do conjunto de T_P associado a ele e para cada elemento do conjunto de T_P que tenha um elemento do conjunto de B_P associado a ele, este é único, ou seja, todas as caixas de instruções estão associadas a sua própria e única transição, mas nem todas as transições têm uma caixa de instruções associada a ela. Considerando $b \in B_P$

e $t \in T_P$, se $b \times t \neq \emptyset$ isso significa que a caixa de instruções b está associada a transição t , e se $b \times t = \emptyset$ isso significa que a caixa de instruções b e a transição t não estão associadas uma a outra.

Uma RdPP atende as condições de Wf-net se $\exists p_S \in P_P$ em que $\bullet p_S = \emptyset$ e se $\exists p_E \in P_P$ em que $p_E^\bullet = \emptyset$ e quando se inclui t_{aux} tal que $\bullet t_{aux} = \{p_E\}$ e $t_{aux}^\bullet = \{p_S\}$ no novo grafo $\nexists t \in \overline{T_P} | \bullet t = \emptyset$, $\nexists t \in \overline{T_P} | t^\bullet = \emptyset$, $\nexists p \in \overline{P_P} | \bullet p = \emptyset$ e $\nexists p \in \overline{P_P} | p^\bullet = \emptyset$.

A dinâmica da RdPP é similar a dinâmica da RdP, mas antes de “disparar” uma transição, há uma fase de “executando” as instruções da caixa de instruções associada a ela. Para uma transição $t_{PER} \in T_P$, a transição é entendida como “habilitada para execução” quando, com $\bullet t_{PER} = \{\forall p \in P_P | (p, t_{PER}) \in F_P\}$, $\forall p \in \bullet t_{PER} : M_{PF}(p) > 0$ e $\exists b \in B_P | b \times t_{PER} \neq \emptyset$. Isso significa que a caixa de instruções pode executar as instruções, caso ainda não as tenha executado. Quando uma transição está “habilitada para execução”, ela pode entrar na fase de “executando” que significa reservar marcas nos lugares de $\bullet t_{PER}$, matematicamente representando a mudança entre os estados i e $i + 1$ por $\forall p \in \bullet t_{PER} : M_{PF}^{i+1}(p) = M_{PF}^i(p) - 1$, $M_{PR}^{i+1}(p) = M_{PR}^i(p) + 1$. O processo de reservar marcas é entendido como o lugar ainda tendo esta marca, porém ela não pode ser usada no cálculo de “habilitada para execução” ou “habilitada para o disparo” de outra ou da mesma transição novamente. Para uma transição $t \in T_P$ iniciar a execução das instruções na caixa de instruções $b \in B_P$ associada a si ($b \times t \neq \emptyset$), o conteúdo da caixa de instruções deve ser analisado, pois é necessário que $M_B(p_{Ba}) = 1$ com $p_{Ba}, M_B \in b$. Quando a execução das instruções da caixa de instruções $b \in B_P$ é iniciada, $M_B = (1, 0)$ com $M_B \in b$.

Após iniciar a execução das instruções em uma caixa de instruções $b \in B_P$, cada instrução contida deve ser executada uma única vez, e no final de sua execução, isto é verificado. Quando todas as instruções de uma caixa de instruções estiverem verificadas, ou seja, $\forall i_B \in b : M_I = (0, 1)$ e $M_B(p_{Bh}) = 1$ para $M_B, p_{Bh} \in b$, é entendido que a caixa de instruções foi executada.

Se uma transição da RdPP ($t \in T_P$) está “habilitada para execução” e está associada a uma caixa de instruções b ($b \times t \neq \emptyset$) e esta caixa de instruções foi executada, é entendido que a transição t está “habilitada para o disparo”. Se a transição $t \in T_P$ não está associada a uma caixa de instruções ($\forall b \in B_P : b \times t = \emptyset$) e atende a condição de $\bullet t = \{\forall p \in P_P | (p, t) \in F_P\}$, $\forall p \in \bullet t : M_{PF}(p) > 0$, esta transição é considerada “habilitada para o disparo”. Quando uma transição está “habilitada para o disparo”, ela pode disparar e as marcas fluem como no disparo de uma transição da RdP, mas as marcas que fluem se distinguem entre os tipos de transição. Quando a transição disparada está associada a uma caixa de instruções, as marcas são removidas do vetor de marcações M_{PR} nos lugares que são pré-conjunto desta transição, e quando a transição não tem uma caixa de instruções associada, as marcas são removidas do vetor de marcações M_{PF} dos lugares no pré-conjunto desta. Além disso, quando uma

transição associada a uma caixa de instruções dispara, o estado da caixa de instruções muda para um estado de espera para execução (que também é aplicável para quando ela não está executando nenhuma instrução), ou seja, $M_B(p_{Ba}) = 1$.

5.5 CARACTERÍSTICAS DE PROCESSOS DE WORKFLOW

As características de processos de *workflow* apresentadas a seguir são comumente encontradas nos processos produtivos: sequência, paralelismo, conflito e sequência arbitrária. Esta seção apresenta essas características e compara as representações em RdP e Wf-net e RdPP.

A descrição das características de processos de *workflows*, é fundamental para a análise desses *workflows* e para obter outras informações necessárias às diferentes atividades de controle. Uma análise que pode ser feita com a RdPP é chamada de “representação sequencial do processo”. A representação sequencial do processo é uma sequência de instruções da RdPP que, se executadas na ordem, obtém o resultado do *workflow* proposto. Para isso, a representação sequencial do processo segue as regras da dinâmica da RdPP entre o “estado fornecedor” (estado em que apenas o lugar “fornecedor” tem marcação) e o “estado sorvedor” (estado em que apenas o lugar “sorvedor” tem marcação) no *workflow*. A quantidade de possíveis representações sequenciais do processo para um *workflow* varia de acordo com a quantidade os tipos de características de processos de *workflow* presentes, pois algumas delas afetam as possíveis combinações de instruções. Dessa forma, a representação sequencial do processo na RdPP transforma um modelo de processos com diferentes características de processos de *workflow* num modelo de processo com apenas uma característica, a sequência, que é a característica de processos de *workflow* mais presentes nas atuais linguagens de programação, como o STEP-nc (BHANDARKAR; NAGI, 2000).

5.5.1 Representação sequencial do processo

A representação sequencial do processo é uma forma de especificar a solução de controle do processo para implementação em dispositivos controladores.

A representação sequencial do processo depende da definição de dois estados: um estado inicial e um estado final (que no caso de um *workflow* a ser analisado, são os “estado fornecedor” e “estado sorvedor”, respectivamente). Definidos estes estados, faz-se o seguinte: dado um σ_k , isto é, uma sequência de disparos de transições tal que $M_i[\sigma_k > M_f]$ da RdP, com M_i o estado inicial definido anteriormente e M_f o estado final, entende-se que $\exists \tau_k$ que contém a representação sequencial do processo a partir do disparo das transições em sequência em σ_k . Assim, $\forall t \in \sigma_k$ caso exista um $b_i \in B_P$ tal que $b_i \times t \neq \emptyset$, o elemento b_i é colocado em τ_k na

mesma posição da sequência que o elemento t de σ_k . Entretanto, os b s ainda são conjuntos de etapas e não as etapas, propriamente ditas. Para expandir os b s em etapas (ou instruções), considera-se que quando um b_i de τ_k : possui um conjunto com apenas uma instrução a ser executada, a expansão de b_i se dá apenas substituindo essa pela própria instrução; ou possui um conjunto com mais de uma instruções a serem executadas, a expansão de b_i se dá substituindo essa por um bloco que combina esse conjunto de instruções em qualquer sequência (pois todas as sequências desse conjunto são corretas pela lógica da caixa de instruções, como descrito na seção 5.5.5).

5.5.2 Sequência

A sequência é a característica de processos de *workflow* em que um conjunto de instruções devem ser executadas apenas após outro conjunto de instruções já ter sido executado e verificado. Um possível caso prático dessa característica em processos de *workflow* é o caso de uma peça que para ser fabricada utiliza duas diferentes máquinas (A e B). Inicialmente, a peça deve ser cortada na máquina A e depois deve ser polida na máquina B. Para que o processo de polimento seja feito na máquina B, a peça tem que ter todo seu processo de corte concluído na máquina A. Dessa forma, há uma dependência lógica na execução dos processos executados em B quanto aos processos executados em A.

Para a RdPP, a dependência da conclusão de um processo para se iniciar outro processo é modelada da seguinte forma: sendo $b_w \in B_P$ uma caixa de instruções que representa os processos que devem ser executados após a execução dos processos em $b_y \in B_P$, seja $t_x \in T_P$ com $b_w \times t_x \neq \emptyset$ e seja $t_z \in T_P$ com $b_y \times t_z \neq \emptyset$, para que b_w possa ser executada, a dinâmica da RdPP exige que essa ainda não esteja em execução e que $\bullet t_x = \{\forall p \in P_P | (p, t_x) \in F_P\}$, $\forall p \in \bullet t_x : M_{PF}(p) > 0$, ou seja, para os lugares que são pré-conjunto de t_x é preciso haver marcas no vetor de marcações livres. Para gerar a dependência entre esses processos, uma estratégia é que a transição t_z forneça as marcas que a transição t_x necessita após a conclusão do seu processo. Dessa forma, uma alternativa é que $\bullet t_x = t_z^\bullet$, com $t_z^\bullet = \{\forall p \in P_P | (t_z, p) \in F_P\}$, ou seja, o pós-conjunto de t_z é igual ao pré-conjunto de t_x . Essa representação pode ser vista no Grafo 14.

Entre o “estado fornecedor” e o “estado sorvedor” do modelo da Grafo 14, a sequência de disparos de transições é $\sigma_1 = t_z t_x$. Com essa sequência de disparos, como $b_y \times t_z \neq \emptyset$, $b_w \times t_x \neq \emptyset$, i_y é instrução contida na caixa de instruções b_y e i_w é instrução contida na caixa de instruções b_w , tem-se que a representação sequencial deste processo produtivo é $\tau_1 = i_y, i_w$.

Da mesma forma que a RdPP, a Wf-net e a RdP tem uma representação para a característica de processos de *workflow* em sequência, como no Grafo 15.

Grafo 14 - Representação da característica de processos de *workflow* em sequência em RdPP Grafo 15 - Outras representações da característica de processos de *workflow* em sequência

Comparando as três representações da sequência em RdPP (no Grafo 14), em Wf-net e RdP (no Grafo 15), não há uma diferença significativa entre elas.

5.5.3 Paralelismo

O paralelismo é a característica de processos de *workflow* em que dois ou mais conjuntos de instruções são executados de forma independente. Um exemplo de paralelismo está na área de indústria automotiva. Motores de automóvel podem ser produzidos em uma instalação produtiva enquanto os chassis usados para o mesmo automóvel podem ser produzidos em outra instalação produtiva, concomitantemente. Como essas peças não afetam umas nas outras no seu processo de produção, pode-se dizer que estas duas linhas da produção são independentes e, portanto, os processos produtivos são paralelos.

Para a RdPP, a independência de execução de dois ou mais processos é feita da seguinte forma: sendo $b_w, b_y \in B_P$ duas caixas de instruções que representam os processos que podem ser executados em paralelo, seja $t_x \in T_P$ com $b_w \times t_x \neq \emptyset$ e seja $t_z \in T_P$ com $b_y \times t_z \neq \emptyset$, para que sejam independentes b_w tem que poder ser executada ao mesmo tempo que b_y , com isso, os pré-conjuntos da transição t_x e da transição t_z não podem ser as mesmas, mas podem possuir marcas num mesmo instante do modelo, ou seja, $\forall p \in \bullet t_x, p \notin \bullet t_z$. A forma simplificada de representar essa independência é fazer com que os pré-conjuntos de t_x e t_z recebam marcas a partir de uma mesma transição, ou seja, $\bullet\{t_x\} = \bullet\{t_z\} = t_a$, com isso, as transições t_x e t_z ficam “habilitadas para execução” ao mesmo tempo após o disparo de t_a . Essa representação pode ser vista no Grafo 16.

Entre o “estado fornecedor” e o “estado sorvedor” do modelo da Grafo 16, as sequências de disparos de transições são a $\sigma_1 = t_a t_z t_x$ $\sigma_2 = t_a t_x t_z$. Com essas sequências de disparos,

Grafo 16 - Representação da característica de processos de *workflow* em paralelo em RdPP

como $b_y \times t_z \neq \emptyset$, $b_w \times t_x \neq \emptyset$, $\nexists b \in B_P | b \times t_a \neq \emptyset$, i_y é instrução contida na caixa de instruções b_y e i_w é instrução contida na caixa de instruções b_w , tem-se que as representações sequenciais deste processo produtivo são $\tau_1 = i_y, i_w$ e $\tau_2 = i_w, i_y$.

Da mesma forma que a RdPP, a Wf-net e a RdP tem uma representação para a característica de processos de *workflow* em paralelo, como no Grafo 17.

Grafo 17 - Outras representações da característica de processos de *workflow* em paralelo

Comparando as três representações do paralelismo em RdPP (no Grafo 16), em Wf-net e RdP (no Grafo 17), há uma diferença significativa entre elas. O modelo de Wf-net introduz dois elementos novos para serem interpretados, que são: um do tipo AND-SPLIT e um do tipo AND-JOIN. O acréscimo de novos elementos que possuem características similares a elementos já presentes, diferenciando-os por suas regras de execução, faz com que a pessoa que vá interpretar os grafos necessite de conhecimento mais especializado (que vai contra uma

das características da RdP é a simplicidade para interpretar seu grafos (YOO; JEONG; CHO, 2010)). Assim como no requisito de universalização da informação, seção 3.3, as soluções com menor necessidade de pessoal especializado são as mais eficientes para atender as demandas do ambiente colaborativo.

5.5.4 Conflito

O conflito é a característica de processos de *workflow* que permite a escolha de um entre dois ou mais conjuntos de instruções para serem executados e que pode levar a diferentes sequências. Um exemplo de conflito é um processo de preparação de kits de material escolar. Considerando que um kit é composto por caderno e caneta, em que, o caderno pode ser com folhas normais ou de papel reciclado. Cada uma das opções oferece diferentes condições de custo e sustentabilidade, que podem ser utilizados pelo cliente para avaliar a melhor opção. A execução de um ou outro conjunto de instruções também é feita de forma independente, mas apenas uma pode ser escolhida, o que difere essa característica de processos de *workflow* de paralelismo.

Para a RdPP, da mesma forma que o paralelismo, o conflito depende da independência incompleta (entende-se a independência incompleta aqui como uma independência no fato que uma não precisa aguardar a outra para ser executada, entretanto a execução de um interfere na execução da outra) entre dois ou mais processos para execução: sendo $b_w, b_y \in B_P$ duas caixas de instruções que representam os processos que podem ser executados, seja $t_x \in T_P$ com $b_w \times t_x \neq \emptyset$ e seja $t_z \in T_P$ com $b_y \times t_z \neq \emptyset$, para que sejam independentes de forma incompleta o pré-conjunto da transição t_x não pode ser o pós-conjunto da transição t_z , nem vice-versa. A forma simplificada de representar essa independência incompleta é fazer com que os pré-conjuntos de t_x e t_z tenham ao menos um elemento comum $\exists p \in \bullet t_x | p \in \bullet t_z$, com isso, as transições t_x e t_z ficam “habilitadas para execução” ao mesmo tempo, entretanto se uma delas for disparada, a outra perde a condição de “habilitada para execução”. Essa representação pode ser vista no Grafo 18.

Entre o “estado fornecedor” e o “estado sorvedor” do modelo da Grafo 18, as sequências de disparos de transições são a $\sigma_1 = t_z$ $\sigma_2 = t_x$. Com essas sequências de disparos, como $b_y \times t_z \neq \emptyset$ e $b_w \times t_x \neq \emptyset$ e, i_y é instrução contida na caixa de instruções b_y e i_w é instrução contida na caixa de instruções b_w , tem-se que as representações sequenciais deste processo produtivo são $\tau_1 = i_y$ e $\tau_2 = i_w$.

Da mesma forma que a RdPP, a Wf-net e a RdP têm uma representação para a característica de processos de *workflow* em conflito, como no Grafo 19.

Comparando as três representações do conflito em RdPP (no Grafo 18), em Wf-net e RdP

Grafo 18 - Representação da característica de processos de *workflow* em conflito em RdPP Grafo 19 - Outras representações da característica de processos de *workflow* em conflito

(no Grafo 19), há uma diferença na quantidade de informações para tratar da divisão do processo em duas possibilidades devido a criação de novos elementos no Wf-net (do tipo OR-SPLIT e do tipo OR-JOIN) necessários para representar o caso, além do já apontado problema da necessidade de pessoal especializado.

5.5.5 Sequência Arbitrária

A sequência arbitrária é a característica de processos de *workflow* em que dois ou mais conjuntos de instruções devem ser executados um após o outro e com uma independência incompleta (a independência incompleta significa que apesar de uma impedir que a outra execute, esse impedimento é momentâneo) na execução de uma com a outra. Dessa forma, a sequência arbitrária faz uma combinação das características de processos produtivos em sequência, em paralelo e em conflito. Um exemplo de sequência arbitrária em processos é o armazenamento de duas matérias-primas de origens diferentes em dois silos diferentes feitos por uma mesma máquina. Ambas matérias-primas devem ser armazenadas e como o recurso necessário é o mesmo, apenas uma de cada vez será armazenada, entretanto a ordem desses armazenamentos é indiferente para o resultado esperado.

Na RdPP, a sequência arbitrária tem uma representação especial para quando os processos independentes a serem realizados são realizados pelo mesmo equipamento. Neste caso, a independência incompleta de dois ou mais processos para execução pode ser feita da seguinte forma: sendo $b_x \in B_P$ uma caixa de instruções que representa os processos que podem ser executados, seja $t_x \in T_P$ com $b_x \times t_x \neq \emptyset$, internamente a essa caixa de instruções tem-se 2 instruções i_w e i_y que devem ambas ser executadas, mas a ordem de execução de ambas é indiferente para o resultado final. Para que sejam independentes de forma incompleta tem que haver um elemento comum que permita a escolha entre as instruções de uma caixa de instruções (como é o elemento p_{Bh} no Grafo 12), com isso, apenas uma instrução pode ser executada (mas em qualquer sequência) quando se tem marca em p_{Bh} . Essa representação pode ser vista de forma genérica e expandida no Grafo 12 ou de forma simplificada no Grafo 20.

Grafo 20 - Representação da característica de processos de *workflow* em sequência arbitrária em RdPP

Entre o “estado fornecedor” e o “estado sorvedor” do modelo da Grafo 20, a sequência de disparos de transições é $\sigma_1 = t_x$. Com essa sequência de disparos, como i_y e i_w estão contidas na caixa de instruções associada a transição t_x , tem-se que as representações sequenciais deste processo produtivo são $\tau_{1.1} = i_y, i_w$ e $\tau_{1.2} = i_w, i_y$.

Da mesma forma que a RdPP, a Wf-net e a RdP têm uma representação para a característica de processos de *workflow* em sequência arbitrária, como no Grafo 21.

Comparando as três representações da sequência arbitrária em RdPP (no Grafo 20), em Wf-net e RdP (no Grafo 21), nota-se que a representação em RdP não pode ser simplificada sem perda de informação sobre a execução das instruções, como na RdPP, e também que a representação da Wf-net é a mesma da representação da característica de processos de *workflow* em paralelo, pois para o formalismo e dinâmica da Wf-net ambas podem ser representadas da mesma forma, via de regra essa trata sempre um único equipamento no modelo. Para diferenciar essas características de processos de *workflow*, condição necessária para o ambiente colaborativo, seria necessário acrescentar outros elementos a Wf-net e rever a sua dinâmica.

Grafo 21 - Outras representações da característica de processos de *workflow* em sequência arbitrária

5.6 ABORDAGEM DE INFORMAÇÃO

Considerando que a PNML já é amplamente utilizada como formato de troca de informação da RdP e de suas extensões, a PNML também deve ser usada para a RdPP. Neste caso, alguns elementos e estruturas não são necessárias para sua descrição. A estrutura de PNML adotada está representada na Figura 46.

Por outro lado, para a RdPP, há a necessidade de criar um novo objeto para armazenar as informações relevantes às caixas de instruções ("InstructionBox"). Esse objeto possui propriedades gráficas e deve apresentar as informações de identificação ("id") e o objeto de transição com que se relaciona ("rel"). Além disso cada elemento de "InstructionBox" pode conter múltiplas instruções, compatíveis com a descrição das caixas de instruções da RdPP, cada uma representada por um elemento "Instruction". O elemento "InstructionBox", também pode ter dois outros elementos, um para entrada de informações de tipo de processo, em "ProcessInfo", e outro para entrada de informações de entrada de material ou de serviço, em "EntryInfo". Cada um dos elementos "Instruction", "EntryInfo" e "ProcessInfo" contém um elemento do tipo "Label" para sua representação textual.

Figura 46 - Estrutura das informações no arquivo PNML adotada para a RdPP

Os elementos para múltiplos modelos, múltiplas páginas e objetos de referências não são obrigatórios no modelo básico da RdPP e, por isso, não estão aqui apresentados (“RefPlace”, “RefTransition”, “Page”, ou múltiplos “PetriNet”).

Por utilizar uma estrutura baseada no XML com especificação pública, o PNML oferece um padrão de semântica que é adequada à implementação da RdPP.

5.7 SÍNTESE DO CAPÍTULO

A arquitetura proposta no capítulo 4, que considera a RdPP para descrever o *workflow*, oferece também os mesmos suportes que a RdPP oferece, ou seja, aceita padrões de sintaxe e semântica.

O requisito de universalização da informação, além dessas hipóteses de padronização de sintaxe e semântica, também são atendidas pelo *workflow* com linguagem relativamente simples e acessível. A acessibilidade e simplicidade da linguagem utilizada pela RdPP é também discutida nos testes realizados nesse trabalho.

A RdPP, além disso, oferece a possibilidade de descer a informação ao nível de controle de

equipamento ou de atuadores para um mesmo *workflow*. Isso a torna uma ferramenta que tanto pode ser usada no nível gerencial quanto num nível de operação. Em alguns trabalhos recentes, a colaboração tem sido estudada não entre empresas, mas entre equipamentos. Neste caso, a informação deve ter a característica de poder ser usada tanto no nível gerencial quanto num nível de operação.

As características dos processos apresentadas em RdPP atendem todos os testes realizados nesse trabalho. Outras características de processo podem ser também elaboradas com a RdPP conforme a necessidade se faça para caso de outros tipos de *workflows*.

6 RESULTADOS EXPERIMENTAIS

Os testes e consultas presentes nesse capítulo visam verificar algumas características apontadas para a ferramenta proposta.

A simplicidade proposta da RdPP não se aplica apenas para usuários com grande conhecimento, mas com vários níveis de conhecimento. Assim, dois testes foram feitos com a RdPP como *workflow*.

Um teste realizado com a RdPP foi um teste prático, em que pessoas de conhecimento médio em processos produtivos e pouco ou nenhum conhecimento na RdP responderam a uma tarefa. Os resultados desse teste estão apresentados na seção 6.1.

Por fim, uma aplicação real foi estudada com a RdPP e é apresentado o resultado desse estudo na seção 6.2.

6.1 TESTE SOBRE A HIPÓTESE DE SIMPLICIDADE DO WORKFLOW COM USUÁRIOS DE MÉDIO CONHECIMENTO

O teste realizado com a RdPP para identificar se ela oferece acessibilidade e simplicidade para interpretação dos *workflows* foi feito utilizando, nesse primeiro teste, pessoas que possuem algum conhecimento de processos, mas pouco ou nenhum conhecimento de RdP. Como o projeto é uma etapa que demanda um conhecimento um pouco maior sobre o fluxo de atividades, as atividades solicitadas às pessoas que não tem conhecimento da RdP se restringiu a interpretação do *workflow* representado em RdPP.

O levantamento de dados neste teste considerou o estabelecimento de uma classe de instruções para equipamentos de manufatura. Parte do levantamento de dados contou com apoio dos técnicos de máquinas do Laboratório de Máquinas de Operação da Escola Politécnica da Universidade de São Paulo, ou operadores de máquinas.

O teste seguiu as seguintes etapas:

- Etapa 1: definição do escopo de trabalho e pessoal de apoio;
- Etapa 2: elaboração de um conjunto de instruções para compor uma classe de teste, com auxílio dos operadores de máquinas;
- Etapa 3: projeto de uma peça e um *workflow* utilizando a RdPP, sem auxílio dos operadores de máquinas;
- Etapa 4: entrega do *workflow* para os operadores de máquinas elaborarem um esboço da peça;

- Etapa 5: geração o código G para o *workflow* da peça utilizando somente o modelo em RdPP;
- Etapa 6: comparação com o mesmo *workflow* utilizando a Wf-net.

Etapa 1: definição do escopo de trabalho e pessoal de apoio

Do escopo de possibilidades oferecido pelo Laboratório de Máquinas e pela disponibilidade de operadores de máquinas disponíveis, com tempo para auxiliar nos trabalhos, o tipo de máquina escolhida foram os tornos mecânicos. O torno mecânico, como parte das máquinas de manufatura, é um dos tipos de máquinas sobre os quais o STEP-nc especificou informações de processos, o que é importante dentro do contexto da presente pesquisa.

Com isso, os operadores de máquinas de torno e os operadores de máquinas multifuncionais foram consultados para elaboração das instruções.

Etapa 2: elaboração de um conjunto de instruções para compor uma classe de teste, com auxílio dos operadores de máquinas

Com ajuda dos operadores de máquinas, foi proposta um conjunto de instruções que são realizadas com mais frequência por esses operadores em seus trabalhos. A solicitação para os operadores de máquinas foi que procurassem características geométricas comuns nas operações de rotinas desses. Essas características comuns foram então comparadas umas as outras para serem descritas como funções.

Dentre as funções que foram criadas a partir dessa comparação, algumas tiveram que ser separadas por terem peculiaridades diferentes umas das outras, por exemplo, características geométricas iguais, mas que para serem executadas necessitam de ferramentas diferentes. As funções remanescentes foram transformadas em instruções utilizando os nomes encontrados para essas funções na literatura. Para finalizar a especificação das instruções, foi definido também quais são as dependências que as funções tinham, como as medidas da característica geométrica que se esperava obter.

As instruções propostas, para uma peça padrão como da Figura 47, foram as seguintes:

- *torneamento_externo*($\emptyset, L$): que é uma instrução para um corte plano a partir da ponta livre da peça e depende da informações de diâmetro final do corte ($\emptyset$) e profundidade do corte (L), como na Figura 48;

Figura 47 - Peça padrão para o teste em manufatura

Figura 48 - Instrução proposta para torneamento externo

- $furo(\varnothing, L)$: que é uma instrução para um corte interno plano a partir da ponta livre da peça e depende das informações de diâmetro da perfuração ($\varnothing$) e profundidade da perfuração (L), como na Figura 49;

Figura 49 - Instrução proposta para furo

- $canal(D, \varnothing, L)$: que é uma instrução para um corte plano, similar ao torneamento externo, mas a partir de uma certa distância da ponta livre da peça e depende das informações de distância da ponta livre (D), diâmetro final do corte ($\varnothing$) e profundidade do corte (L), como na Figura 50;
- $torneamento_conico_direita(D, \varnothing_i \varnothing_e, L)$: que é uma instrução para um corte inclinado em que a ponta livre está mais próxima da maior profundidade de corte e a profundidade reduz conforme se afasta da ponta livre até chegar a profundidade de corte zero em

Figura 50 - Instrução proposta para corte de canal

relação a superfície da peça. Essa instrução depende da informações de distância entre a ponta livre da peça e o início do corte (D), diâmetro menor do corte ($\varnothing_i$), diâmetro maior do corte ($\varnothing_e$) e profundidade do corte (L), como na Figura 51;

Figura 51 - Instrução proposta para torneamento cônico pela direita

- *torneamento_conico_esquerda*($D, \varnothing_i, \varnothing_e, L$): que é uma instrução para um corte inclinado em que a ponta livre está mais próxima da menor profundidade de corte, ou zero em relação a superfície da peça, e a profundidade aumenta conforme se afasta da ponta livre até chegar a profundidade de corte máxima. Essa instrução depende da informações de distância entre a ponta livre da peça e o início do corte (D), diâmetro menor do corte ($\varnothing_i$), diâmetro maior do corte ($\varnothing_e$) e profundidade do corte (L), como na Figura 52;

Figura 52 - Instrução proposta para torneamento cônico pela esquerda

Essas duas primeiras etapas foram utilizadas para definir um escopo do teste para as etapas seguintes. As instruções descritas pelos operadores são encontradas, de forma semelhante, no padrão do STEP.

Etapa 3: projeto de uma peça e um workflow utilizando a RdPP, sem auxílio dos operadores de máquinas

Nem todas as instruções que foram apontadas pelos operadores seriam usadas no exemplo considerado, como o faceamento. Por isso, essas instruções não constaram na lista anterior. A peça escolhida para o teste foi baseada em propostas de testes com máquinas de torno em artigos científicos sobre o uso do STEP e reconhecimento geométrico. Esse teste utiliza todas as instruções citadas anteriormente.

A peça proposta para teste está representada na Figura 53.

Figura 53 - Peça proposta para os técnicos de máquinas avaliarem a RdPP

Para elaborar o desenho da peça em atividades do *workflow*, primeiramente foram observados quais tipos de operações, do ponto de vista geométrico, que seriam necessárias para aplicar a uma peça cônica de comprimento inicial $275mm$ e diâmetro inicial $120mm$. Para aplicação da técnica de projeto de *workflow*, a partir da desenho de peça inicial subtrai-se algumas geometrias básicas.

Antes de iniciar a busca das instruções, define-se o traço externo da peça como uma $\Delta : L \rightarrow R$ com o traço externo de x com $x \in [0, L0]$ (0 sendo a ponta livre da peça e $L0$ sendo o fim de curso da ferramenta, totalizando o comprimento inicial da peça) aponta o maior raio em que se encontra a parte da peça que não deve ser removida na usinagem. O traço externo da peça na Figura 53, é representado na Figura 54.

Figura 54 - Traço externo da peça

Define-se singularidades, quando um ponto x do traço externo tiver diferentes limites que tendem a ele pela direita ou esquerda (que significa diferentes diâmetros de peça para um mesmo ponto, ou degrau), ou seja, para x_S em uma singularidade, $\lim_{x \rightarrow x_S^-} \Delta(x) \neq \lim_{x \rightarrow x_S^+} \Delta(x)$.

A primeira parte desse procedimento é a de procurar todos os cortes de torneamento externo. Para isso, procura-se todas as cotas de Δ tal que cota é todo $\Delta(x)/\Delta'(x) = 0$ com $x \in [x_{min}, x_{max}]$. Para todos os cortes de torneamento externo, usa-se $x_{min} = 0$, ou seja, a todos iniciam na ponta livre da peça. Inicialmente, considera-se $x_{max} = L0$ e depois reduz-se até o fim dessa parte. Os torneamentos externos devem ser realizados até que o corte alcance o ponto de $\Delta(0)$.

As cotas da peça são as da Figura 55.

Figura 55 - Cotas do traço externo para a peça com $x \in [0, L0]$

Na Figura 55, as cotas são as linhas pontilhadas exceto a linha que está em $\Delta(0)$ que também é uma cota, mas não aparece o pontilhado por fazer parte do traço externo.

Cada torneamento externo terá o diâmetro $\emptyset$ de corte igual a duas vezes o maior valor de cota, desde que esse seja menor que a cota máxima anterior (para o primeiro corte, considera-se a cota máxima anterior como a metade do diâmetro inicial da peça), para $x \in [0, x_{max}]$. Para escolher o comprimento de corte L , usa-se uma linha vertical auxiliar $Aux(r) = x$, nessa condição, L será o maior valor de $x \in [0, x_{max}]$ em que a linha vertical $Aux(r) = x$ seja perpendicular com o traço externo da peça no ponto ou que x seja uma singularidade do traço externo. Após cada corte ser definido, redefini-se x_{max} para o menor valor de x em que $\Delta(x) = \emptyset$ e repete-se a operação até x_{max} ou L serem iguais a zero. Dessa forma, o primeiro corte fica como na Figura 56.

Figura 56 - Primeiro corte de torneamento externo da peça

Pela Figura 56, o diâmetro de corte é $100.0mm$ e o comprimento de corte é $275.0mm$

para o primeiro torneamento externo. Após esse corte, $x_{max} = 95$ e repetindo as operações, encontra-se um segundo torneamento externo como na Figura 57.

Figura 57 - Segundo corte de torneamento externo da peça

Pela Figura 57, o diâmetro de corte é $60.0mm$ e o comprimento de corte é $70.0mm$ para o segundo torneamento externo. Após esse corte, $x_{max} = 0$ e não é possível mais fazer cortes de torneamento externo.

Na segunda parte dessa técnica de projeto, encontra-se todos os cortes de canais. A técnica para determinar os cortes de canal é a mesma do torneamento externo, entretanto é preciso deslocar o x_{min} para encontrar o início de um canal. As demais etapas seguem a mesma sequência que a obtida com o torneamento externo. Determina-se para o canal que $x_{min} > 0$ e que $x_{min} \geq \min(x)/\Delta(x) \neq \Delta(0)$, pois nesse caso trata-se do último torneamento externo. Assim, para encontrar a posição, procura-se o menor valor de x tal que o ponto atenda as condições anteriores e que nele esteja uma singularidade ou a linha vertical auxiliar $Aux(r) = x$ seja perpendicular ao traço externo da peça no ponto (essa última condição desde que não tenha sido feito o corte rente a posição). Escolhido o ponto x_{min} , repete-se o mesmo procedimento adotado para o torneamento externo (alterando o 0 para x_{min}), buscando os vários cortes de canal a partir do mesmo ponto. O critério de parada de busca de cortes de canal é $x_{min} = L0$. Para cada x_{min} escolhido adota-se o valor da posição inicial do corte $D = x_{min}$.

O primeiro corte de canal está representado na Figura 58.

Figura 58 - Primeiro corte de canal da peça

Pela Figura 58, a posição inicial da peça é $50.0mm$, o diâmetro de corte é $20.0mm$ e o comprimento de corte é $20.0mm$ para o primeiro canal. Após esse corte, $x_{max} = x_{min}$ e não

é possível mais fazer cortes de canal para esse x_{min} . Desloca-se então o x_{min} para a próxima posição de acordo com a Figura 59.

Figura 59 - Segundo corte de canal da peça

Pela Figura 59, a posição inicial da peça é $165.0mm$, o diâmetro de corte é $40.0mm$ e o comprimento de corte é $35.0mm$ para o segundo canal. Após esse corte, $x_{max} = x_{min}$ e não é possível mais fazer cortes de canal para esse x_{min} . Desloca-se então o x_{min} para a próxima posição e encontra-se $x_{min} = L0$, fazendo com que não haja mais cortes de canal.

A próxima etapa é procurar os cortes cônicos. Para encontrar os cortes cônicos procura-se as posições $x \in [0, L0]$, de forma crescente de x tal que $\Delta'(x^+) \neq 0$. Encontrado cada ponto, define-se o ponto inicial do corte $D = x$ e se $\Delta'(x^+) > 0$ o diâmetro menor $\varnothing_i = 2 \times \Delta(x^+)$ e a instrução é o torneamento cônico pela direita ou se $\Delta'(x^+) < 0$ o diâmetro maior $\varnothing_e = 2 \times \Delta(x^+)$ e a instrução é o torneamento cônico pela esquerda.

Para determinar os demais parâmetros, procura-se o primeiro ponto $x > D$ tal que $\Delta'(x^+) \neq \Delta'(x^-)$ ou que x seja uma singularidade. Determinado esse ponto, o comprimento de corte $L = x - D$ e se $\Delta'(x^-) > 0$ o diâmetro maior $\varnothing_e = 2 \times \Delta(x^-)$ ou se $\Delta'(x^-) < 0$ o diâmetro menor $\varnothing_i = 2 \times \Delta(x^-)$.

Pela Figura 54 do traço externo, tem-se três ocorrências de corte cônico. A primeira é um torneamento cônico pela direita com ponto inicial de corte $70.0mm$, diâmetro menor $20.0mm$, diâmetro maior $100.0mm$ e comprimento de corte $25.0mm$. A segunda é um torneamento cônico pela esquerda com ponto inicial de corte $130.0mm$, diâmetro menor $40.0mm$, diâmetro maior $100.0mm$ e comprimento de corte $35.0mm$. E terceira é, também, um torneamento cônico pela direita com ponto inicial de corte $200.0mm$, diâmetro menor $40.0mm$, diâmetro maior $100.0mm$ e comprimento de corte $50.0mm$.

Por fim, para determinar os furos, define-se o traço interno $\nabla : L \rightarrow R$ para $x \in [0, L0]$ aponta o maior raio a partir do 0 em que nunca se encontra com qualquer pedaço da peça. O traço interno da peça proposta pode ser vista na Figura 60.

Para determinar os cortes de furo, usa-se o menor raio em que se encontre uma cota do traço interno. Neste caso, o diâmetro do furo é duas vezes o valor da cota. O comprimento do furo é o maior valor de x tal que $\nabla(x) = \varnothing$ em toda a extensão da peça. Após definido um furo, sua cota deve ser removida da busca da nova menor cota do traço interno.

Figura 60 - Traço interno da peça

Pela Figura 60, há apenas um corte de furo na peça, com diâmetro de furo $20.0mm$ e profundidade de furo $25.0mm$.

A ordenação das instruções encontradas nessa modelagem não são necessariamente a mesma sequência da obtenção delas pela técnica anterior, ficando ao projetista a opção de alterar a ordenação das instruções, desde que sigam uma ordem fisicamente possível de atender a produção.

Encontradas todas as instruções, é possível ver a diferença entre as peças inicial e final na Figura 61.

Figura 61 - Peças inicial e final com indicações de usinagem

Da Figura 61, retira-se os cortes a serem realizados como representados na Figura 62.

Figura 62 - Usinagem da peça inicial separada em partes

Baseado na Figura 62, as instruções propostas são:

- A: *torneamento_cônico_direita*(200.0, 40.0, 100.0, 50.0);

- *B*: *canal*(165.0, 40.0, 35.0);
- *C*: *torneamento_conico_esquerda*(130.0, 40.0, 100.0, 35.0);
- *D*: *torneamento_externo*(100.0, 275.0);
- *E*: *torneamento_conico_direita*(70.0, 20.0, 100.0, 25.0);
- *F*: *canal*(50.0, 20.0, 20.0);
- *G*: *torneamento_externo*(60.0, 70.0);
- *H*: *furo*(20.0, 25.0);

Para realizar as instruções *A*, *B*, *C*, *E*, *F* e *G*, da Figura 62, é necessário que a redução do diâmetro da peça seja de 120.0mm para 100.0mm, que é realizado pela instrução *D*. Assim, nesse arranjo, a instrução *D* antecede, praticamente, todas demais instruções. A proposta feita para este caso, isto é, a sequência de instruções a ser seguida é representada como *D*, *G*, *H*, *F*, *E*, *C*, *B*, *A*. Para utilizar a ideia de caixa de instruções com múltiplas instruções, ou seja, com a característica de sequência arbitrária, algumas instruções foram agrupadas da seguinte forma: *D*, (*G*, *H*), *F*, (*E*, *C*, *B*, *A*).

Com a sequência anterior, o modelo proposto em RdPP do *workflow* está no Grafo 22.

Grafo 22 - Modelo do *workflow* em RdPP para a peça proposta

Etapa 4: entrega do workflow para os operadores de máquinas elaborarem um esboço da peça

Nesta etapa, o *workflow* elaborado anteriormente foi entregue aos operadores de máquinas de torno e/ou multifuncionais. Foi solicitado então que eles gerassem um esboço da peça. Além disso, foi entregue um pequeno manual que explica o funcionamento básico da RdPP, quanto às características dos processos produtivos. O esboço foi elaborado pelos operadores depois de um tempo relativamente pequeno (por volta de 1 hora) após a solicitação na forma apresentada na Figura 63.

Figura 63 - Esboço da peça gerado pelos operadores de máquina

O esboço da Figura 63 é igual ao resultado esperado como a peça proposta da Figura 53.

Nessa etapa, os operadores ainda com um conhecimento muito básico da RdP tiveram resultado correto sobre sua interpretação.

Etapa 5: geração o código G para o workflow da peça utilizando somente o modelo em RdPP

Além de comprovar a validade e consistência da RdPP, é importante que a ferramenta de modelagem de *workflow* apoie efetivamente a automação das tarefas envolvidas. Para testar o funcionamento da RdPP no processo de automatização das tarefas, avaliou-se a conversão do modelo de *workflow* para a peça proposta em uma sequência de códigos de máquina. Nesse caso, as máquinas automatizadas do Laboratório de Máquinas de Operação da Escola Politécnica da Universidade de São Paulo utilizam o código G para interpretação do controlador.

Para obter o código G, é necessário primeiro sequenciar as atividades não ordenadas do

workflow. A forma adotada para esse processo é utilizar as representações sequenciais do processo produtivo que é descrita pela RdPP.

Para o modelo do Grafo 22, há 24 possíveis representações sequenciais do processo produtivo. Assim, para o operador de máquina, a peça poderia ser produzida em uma só máquina de 24 maneiras diferentes. Duas das representações sequenciais do processo produtivo são:

$$\tau_1 = \text{torneamento_externo}(\dots), \text{torneamento_externo}(\dots), \text{furo}(\dots), \text{canal}(\dots), \\ \text{torneamento_conico_direita}(\dots), \text{torneamento_conico_esquerda}(\dots), \\ \text{canal}(\dots), \text{torneamento_conico_direita}(\dots)$$

$$\tau_2 = \text{torneamento_externo}(\dots), \text{furo}(\dots), \text{torneamento_externo}(\dots), \text{canal}(\dots), \\ \text{torneamento_conico_esquerda}(\dots), \text{canal}(\dots), \text{torneamento_conico_direita}(\dots), \\ \text{torneamento_conico_direita}(\dots)$$

Para gerar um código G a partir do modelo do Grafo 22, foram usadas as duas representações sequenciais do processo produtivo. Como o código G é um conjunto de códigos que pode ser interpretado por controladores de equipamentos automatizados, se a conversão desse modelo de *workflow* para o código G for possível, é possível também executar automaticamente a produção da peça proposta.

No quadro 4 é proposta a conversão entre as instruções propostas e o código G. O código G derivado desta conversão não considera o reposicionamento do braço para a troca de ferramentas, nem a remoção do cavaco, além do que considera que a remoção de material pode ser feita com uma passada. Os códigos propostos são conhecidos e devidamente processados pelos atuais controladores de equipamentos. Para os códigos do quadro, considera-se s como uma distância de afastamento de $2.0mm$ e o diâmetro padrão da peça $\varnothing_0$ que pode ser obtido pelas informações de objeto. Outras informações necessárias, como rotação ou velocidade, dependem do equipamento e do material e devem ser inseridos previamente no código G, numa produção real.

Utilizando as conversões do Quadro 4 com a representação sequencial do processo produtivo τ_1 , obtém-se um código G como na Listagem 1.

Da mesma forma, com as conversões do Quadro 4 e com a representação sequencial do processo produtivo τ_2 , obtém-se um código G como na Listagem 2.

Os códigos das Listagens 1 e 2 foram gerados automaticamente. Mesmo sem as simplificações adotadas, existem vários programas computacionais/aplicativos de CAM que permitem gerar os códigos G para os equipamentos.

Gerar códigos de máquina sem a necessidade de pessoas acrescenta graus de automação para a cadeia produtiva como um todo.

Quadro 4 - Gerador do código G

<i>Instrução</i>	<i>Código G</i>	<i>Observações</i>
$furo(\emptyset, L)$	N10 M06 T03 N20 G90 G00 X0 Z(s) N30 G01 Z($-L$) N40 G01 Z(s)	Alguns fabricantes oferecem a função G83, que é um ciclo de furação
$torneamento_externo(\emptyset, L)$	N10 M06 T01 N20 G90 G00 X $\emptyset$ Z s N30 G01 Z($-L$) N40 G00 X($\emptyset + s$)	Alguns fabricantes oferecem a função G71, que é o ciclo fixado, com múltiplas repetições do ciclo
$canal(D, \emptyset, L)$	N10 M06 T02 N20 G90 G00 X($\emptyset_0 + s$) Z($-D$) N30 G01 X $\emptyset$ N40 G01 Z($-D - L$) N50 G00 X($\emptyset_0 + s$)	Nenhuma
$torneamento_conico_direita(D, \emptyset_i, \emptyset_e, L)$	N10 M06 T04. N20 G90 G00 X($\emptyset_0 + s$) Z($-D - L$) N30 G01 X($\emptyset_e$) N40 G01 X($\emptyset_i$) Z($-D$) N50 G01 X($\emptyset_0$) N60 G01 X($\emptyset_0 + s$)	Alguns fabricantes oferecem a função G84, que é o ciclo de torneamento cônico
$torneamento_conico_esquerda(D, \emptyset_i, \emptyset_e, L)$	N10 M06 T01 N20 G90 G00 X($\emptyset_0 + s$) Z($-D$) N30 G01 X($\emptyset_e$) N40 G01 X($\emptyset_i$) Z($-D - L$) N50 G01 X($\emptyset_0$) N60 G01 X($\emptyset_0 + s$)	Alguns fabricantes oferecem a função G84, que é o ciclo de torneamento cônico

Etapla 6: comparação com o mesmo workflow utilizando a Wf-net

Como a Wf-net é usada na literatura como um padrão para modelar o *workflow*, é importante comparar os modelos gerados entre as duas ferramentas (RdPP e Wf-net).

Listagem 1 - Código G para a representação sequencial do processo produtivo τ_1

```

1  N10 M06 T01
2  N20 G00 X100 Z02
3  N30 G01 Z(-275)
4  N40 G00 Z02
5  N50 G01 X60
6  N60 G01 Z(-70)
7  N70 G00 X62 Z02
8  N80 M06 T03
9  N90 G00 X0 Z02
10 N100 G01 Z(-25)
11 N110 G01 Z02
12 N120 M06 T02
13 N130 G90 G00 X(62) Z(-50)
14 N140 G01 X20
15 N150 G01 Z(-68)
16 N160 G00 X(62)
17 N170 M06 T04
18 N180 G00 X(102) Z(-95)
19 N190 G01 X100
20 N200 G01 X20 Z(-70)
21 N210 G00 X102
22 N220 M06 T01
23 N230 G00 X102 Z(-130)
24 N240 G01 X100
25 N250 G01 X40 Z(-165)
26 N260 G01 X102
27 N270 M06 T02
28 N280 G00 X102 Z(-165)
29 N290 G01 X40
30 N300 G01 Z(-198)
31 N310 G00 X102
32 N320 M06 T04
33 N330 G00 X102 Z(-250)
34 N340 G01 X100
35 N350 G01 X40 Z(-200)
36 N360 G01 X102

```

O modelo do *workflow* utilizando a Wf-net está representado, de forma resumida, no Grafo 23.

No modelo em Wf-net, da Grafo 23, como a representação da característica de sequência arbitrária é igual a representação da característica de paralelismo, as informações da classe da operação e de objeto, presentes na RdPP, devem ser replicadas para cada instrução, de uma caixa de instruções no modelo RdPP, para cada tarefa na Wf-net. Assim, as mesmas informações são repetidas várias vezes no modelo. Essa redundância de informações aumenta o consumo de memória, no caso de troca de mensagens para negociação no caso de uma EV. Além disso, numa contratação de EV em que a proposta para execução de duas instruções diferentes de uma mesma caixa de instruções no modelo RdPP devem ser realizadas por uma mesma empresa, já utilizando a Wf-net, a ideia de paralelismo pode fazer com que uma instrução seja executada por uma empresa enquanto a outra instrução seja executada por outra empresa, o que é um problema quando o recurso de entrada é o mesmo. Assim, um processo que possui entre suas características uma sequência arbitrária, deveria obrigatoriamente ter

Listagem 2 - Código G para a representação sequencial do processo produtivo τ_2

```

1  N10 M06 T01
2  N20 G90 G00 X100 Z02
3  N30 G01 Z(-275)
4  N40 G00 X(102)
5  N50 M06 T03
6  N60 G00 X0 Z02
7  N70 G01 Z(-25)
8  N80 G01 Z02
9  N90 M06 T01
10 N100 G00 X60 Z02
11 N110 G01 Z(-70)
12 N120 G00 X62
13 N130 M06 T02
14 N140 G00 X62 Z(-50)
15 N150 G01 X20
16 N160 G01 Z(-68)
17 N170 G00 X62
18 N180 M06 T01
19 N190 G00 X102 Z(-130)
20 N200 G01 X100
21 N210 G01 X40 Z(-165)
22 N220 G01 X102
23 N230 M06 T02
24 N240 G00 X102 Z(-165)
25 N250 G01 X40
26 N260 G01 Z(-198)
27 N270 G00 X102
28 N280 M06 T04
29 N290 G00 X102 Z(-150)
30 N300 G01 X100
31 N310 G01 X40 Z(-200)
32 N320 G01 X102
33 N330 G00 Z(-95)
34 N340 G01 X100
35 N350 G01 X40 Z(-70)
36 N360 G01 X102

```

uma representação sequencial para essa.

Por fim, nesse teste, foram gerados os códigos PNML para o modelo da Grafo 22. A representação do código foi reduzida a um trecho entre o segundo e terceiro lugares. O código PNML pode ser visto na Listagem 3.

Usando o PNML para representar os modelos da Grafo 22 e da Grafo 23, para o modelo em RdPP o tamanho (em bytes) do arquivo foi de 4119, enquanto para o modelo em Wf-net o tamanho (em bytes) do arquivo foi 6145, ou seja, um consumo 33% menor de memória para o arquivo da RdPP.

Outra comparação importante a fazer é que apenas o trecho do PNML, representado na Listagem 3, consome 935 bytes. Já esse mesmo trecho para o PNML utilizando a Wf-net, o consumo é de 1492 bytes, que tem a mesma proporção do código completo. Entretanto, para apresentar apenas a parte textual do PNML na representação do Wf-net (considerando que as informações da caixa de instruções são anotações das transições) consome 431 bytes, enquanto o mesmo trecho do *workflow* em PNML da RdPP (ou seja, apenas uma caixa de

Grafo 23 - Modelo em Wf-net para o *workflow* utilizado no teste

instruções) consome 626 bytes.

Dos dados obtidos anteriormente, pode-se concluir que o consumo excessivo de memória para representar um modelo completo da Wf-net em relação a RdPP está em utilizar elementos (lugares, arcos e transições) a mais para a mesma representação. Já o consumo de memória excessivo para representar uma caixa de instruções da RdPP em relação a Wf-net se explica pois a RdPP associa elementos específicos para cada tipo de informação, ou seja, a informação de tipo de processo está marcada especificamente no PNML da RdPP, enquanto essa separação

Listagem 3 - Código PNML para o teste com o torno

```

1 <?xml><PetriNetFile><PetriNet id="net1" type="PPN">
2 (...)
3 <Place id="P2"><graphics><position x="292" y="780"/></graphics></Place>
4 <Place id="P3"><graphics><position x="783" y="780"/></graphics></Place>
5 (...)
6 <Transition id="T2"><graphics><position x="543" y="780"/></graphics></Transition>
7 (...)
8 <Arc id="A3" source="P2" target="T2"></Arc>
9 <Arc id="A4" source="T2" target="P3"></Arc>
10 (...)
11 <ToolInfo tool="InstructionBox"><InstructionBox id="IB2" rel="T2"><graphics><position x="615" y="610"/></graphics>
12 <ProcessInfo><Annotation><value>Torno</value><graphics><offset y="-56"/></graphics></Annotation></ProcessInfo>
13 <EntryInfo><Annotation><value>L0=275.0; D0=100.0</value><graphics><offset y="-13"/></graphics></Annotation></EntryInfo>
14 <Instruction><Annotation> <value>torneamento_externo(60.0,70.0)</value><graphics><offset y="22"/></graphics></Annotation></Instruction>
15 <Instruction><Annotation> <value>furo(20.0,25.0)</value><graphics><offset y="56"/></graphics></Annotation></Instruction>
16 </InstructionBox></ToolInfo>
17 (...)
18 </PetriNet></PetriNetFile>

```

não ocorre na Wf-net.

Essa etapa envolve modelos de informação e recursos computacionais. Para o exemplo citado, o modelo com a RdPP apresentou melhor resultado que o modelo com a Wf-net por utilizar uma quantidade menor de dados para uma mesma apresentação de informações.

6.2 TESTE DE APLICAÇÃO DA FERRAMENTA DE WORKFLOW EM UMA EMPRESA REAL

O teste realizado nesta seção é a modelagem de um *workflow* para representação de uma atividade desempenhada por funcionários de uma empresa da área de extração de petróleo.

A atividade selecionada é a inspeção de poços e equipamentos realizada por técnicos de operação em uma unidade dessa empresa. Essa modelagem faz parte de um projeto de pesquisa entre o LSA e a empresa.

Para iniciar a modelagem foi realizado um mapeamento de campo para identificar as atividades realizadas pelos técnicos de operação da unidade.

Ao fim do mapeamento, a próxima etapa foi a consulta aos supervisores da unidade (funcionários da própria empresa que são responsáveis de verificar se a inspeção está de acordo com as normas) sobre o resultado do mapeamento realizado.

Após as modificações solicitadas pelos supervisores da unidade, foram organizadas as tarefas da atividade em sequências.

As sequências obtidas na aplicação não aproveitavam as características do *workflow*. Suas principais desvantagens foram representação repetitiva e desnecessária de atividades comuns, aumento da complexidade da lógica entre os elementos.

A partir dessas sequências foi gerado um *workflow* para aumentar as possibilidades de inclusão de tarefas, remover os elementos desnecessariamente repetidos e simplificar a compreensão da atividade a ser desempenhada pelos técnicos de operação.

A primeira sequência de tarefas gerada foi de uma das atividades dos técnicos de operação da unidade. Trata-se da operação de inspeção diária de poços produtores e equipamentos do tipo Bombeio Mecânico (BM) ou do tipo Bombeio de Cavidades Progressivas (BCP). A sequência de tarefas dessa atividade pode ser vista na Figura 64.

Figura 64 - Sequência de tarefas da inspeção diária de poços com BM ou BCP

Na Figura 64, cada tarefa da sequência está representada em um retângulo que pode ser de cor laranja se for uma tarefa que finaliza a sequência ou de cor azul caso contrário. A primeira tarefa descrita é para identificação do poço, que na unidade pode ser feita tanto utilizando um leitor de RFID, quanto uma entrada numérica com o identificador cadastrado para o poço produtor que se deseja inspecionar. As tarefas que possuem a característica de sequência

neste modelo são unidas por uma linha azul. A característica de conflito nesse modelo, pode ser visto pela presença de mais de uma linha azul ligando duas ou mais atividades a uma mesma. Nesta figura, as únicas características que podem ser representadas são a sequência e o conflito. Entretanto a finalização do conflito não é representada, o que causa a repetição de tarefas em partes diferentes, tal como todas as tarefa que finalizam essa atividade que são as mesmas “Ante-poço necessita de caminhão sugador?”. Os números à esquerda da Figura não são originais do modelo da representação da sequência de tarefas da aplicação, mas foram incluídos para auxiliar a metodologia de conversão para um *workflow* utilizando o PFS e a RdPP.

Primeiramente, converte-se o modelo acima em um modelo de PFS puramente substituindo as características de sequência com a sequência do PFS e as características de conflito com o conflito do PFS. As tarefas, nessa técnica, são representadas por seus números como atividades do PFS. Esse primeiro modelo pode ser visto no Grafo 24.

Grafo 24 - Primeira representação em PFS da geração do *workflow* em RdPP a partir do modelo da Figura 64

Para que seja a representação de um *workflow*, uma dos requisitos é que exista um lugar ou distribuidor no modelo que não tenha nenhuma transição ou atividade no seu pré-conjunto (fornecedor) e um lugar ou distribuidor no modelo que não tenha nenhuma transição ou atividade em seu pós-conjunto (sorvedor). Para isso, o modelo do Grafo 24 é alterado para o modelo do Grafo 25.

Grafo 25 - Representação em PFS da geração do *workflow* em RdPP a partir do modelo da Figura 64, adaptada aos requisitos de *workflow*

Na representação do Grafo 25, as atividades [8], [11], [18], [21], [27] e [30] aparecem na cor vermelha por representarem a mesma tarefa em partes diferentes da sequência do modelo da Figura 64. A próxima etapa da técnica é fundir essas atividades como uma única atividade

[1.8]. Da mesma forma, os arcos que ligam essas atividades ao sorvedor devem ser fundidos em um só arco. Para que cada parte da sequência leve a mesma atividade [1.8], os distribuidores que fazem parte dos pré-conjuntos dessas atividades também têm que ser fundidos em um só distribuidor, tal como os arcos que levam esses distribuidores as atividades que foram fundidas para gerar a atividade [1.8]. Disso, resulta a representação do Grafo 26.

Grafo 26 - Representação em PFS da geração do *workflow* em RdPP a partir do modelo da Figura 64, fusão de atividades

Na representação do Grafo 26, alguns trechos de atividades foram marcados agora com a mesma cor. Entende-se por trecho de atividades uma sequência de atividades. Os trechos de atividades que foram marcados de cor verde são sequências de tarefas semelhantes encontradas no modelo da Figura 64, assim como os trechos de atividades marcados de cor azul. Esses trechos podem ser também fundidos unindo os distribuidores e atividades dentro do trecho. As duas novas sequências de atividades serão {[2.5], [2.6], [2.7]} para as sequências de atividades marcadas de cor verde e {[2.9], [2.10]} para as sequências de atividades marcadas de cor azul. A representação dessas fusões pode ser vista no Grafo 27.

Grafo 27 - Representação em PFS da geração do *workflow* em RdPP a partir do modelo da Figura 64, fusão de trechos

Seguindo a ideia de fusão de atividades, as atividades [2.7] e [2.10] do Grafo 27 também podem ser fundidas em uma só atividade. Apesar da semelhança, as atividades [2.6] e [2.9] não podem ser fundidas utilizando a técnica de fusão de atividades. Isso porque o distribuidor que pertence ao pré-conjunto da atividade [2.9] não tem como pós-conjunto de si apenas a atividade [2.9], mas também a atividade [2.5]. Para permitir a fusão dessas atividades, considerando que elas são a mesma tarefa da sequência do modelo da Figura 64, deve-se usar uma outra técnica. A técnica a ser usada é a divisão do arco que liga o distribuidor que pertence ao pré-conjunto da atividade [2.9] à própria atividade [2.9] nos elementos seguintes: um arco, uma atividade [3.11], um arco, um distribuidor e um arco. Como isso, é mantida a estrutura do PFS. A atividade [3.11] criada não está associada a nenhuma tarefa no modelo

da Figura 64, sua função é a de manter a característica de conflito que estava presente. A técnica de divisão do arco adotada está representada no Grafo 28.

Grafo 28 - Representação em PFS da geração do *workflow* em RdPP a partir do modelo da Figura 64, divisão do arco

Após divisão do arco realizada no Grafo 28 marcada de cor fúcsia, agora é possível fundir os trechos {[2.6], [2.7]} e {[2.9], [2.10]} em um só trecho {[4.6], [4.7]}. A mesma estratégia (divisão de arco criando a atividade [4.12]) pode ser adotada para realizar a fusão entre as atividades [4], [14] e [23] (na atividade [4.13]), uma vez que o distribuidor que pertence a atividade [14] é um distribuidor está associado a um conflito. Dessa maneira, a representação do PFS utilizando as técnicas ficará como no Grafo 29.

Grafo 29 - Representação em PFS da geração do *workflow* em RdPP a partir do modelo da Figura 64, nova aplicação da divisão de arco, fusão de atividade e fusão de trechos

Por fim, ainda no Grafo 29, é possível observar duas atividades que se referem a tarefas com o mesmo enunciado no modelo da Figura 64, que são as atividades [3] e [22]. Entretanto, apesar de terem o mesmo enunciado (Instale placa de advertência "Poço em Manutenção" ou "Poço Fora de Operação"), essas tarefas são diferentes pois envolvem condições de tarefas diferentes. Por essa razão, fundir essas tarefas pode quebrar a lógica da sequência de atividades proposta no modelo da Figura 64.

A próxima etapa da técnica é converter o modelo do PFS no modelo da RdPP. Para isso, as atividades devem ser substituídas por transições com suas respectivas caixas de instruções. Nas caixas de instruções, a classe das instruções aqui presente foi chamada de "ope-unidade".

Os números que estão nos nomes das atividades do modelo do Grafo 29 devem ser substituídos por suas referências, quando existirem, até chegar no texto da tarefa no modelo da Figura 64. Esses textos devem ser formatados de forma a adequarem ao tipo de tarefa definida na classe "ope-unidade". Nessa proposta, as instruções da classe são:

- "idReader(*variavel*, *enunciado*)" ;
- "getRadiobox(*variavel*, *enunciado*, *opcoes*)";

- “*alert(enunciado)*”;
- “*getCheckbox(variavel, enunciado, opcoes)*”.

As instruções recebem os valores: *variavel* como uma variável para armazenar a resposta da tarefa; *enunciado* como um texto para descrever a tarefa a ser verificada; *opcoes* como uma lista de textos para descrever as possíveis opções selecionáveis. O texto presente em *enunciado* pode ser extraído dos textos da Figura 64. O texto presente na lista de *opções* não aparece na mesma figura, pois teve que ser extraído diretamente da aplicação. A instrução utilizada em cada tarefa também só pode ser extraída da aplicação.

As duas atividades do Grafo 29 que foram criadas pela divisão de arco não devem conter nenhuma instrução a ser realizada em suas respectivas caixas de instruções.

No campo de informações de objetos das caixas de instruções deve ser incluídas as expressões booleanas que resolvem os conflitos existentes no modelo. Ou seja, no conflito existente entre as atividades [3] e [12] do Grafo 29, o conflito é resolvido, segundo a Figura 64, pela resposta da pergunta “UB está parada?” que no modelo da RdPP fica armazenada na variável obtida a partir da instrução equivalente. Para resolver o conflito indo na direção da atividade [3], a resposta para a pergunta deve ser “Não”. Então, nas informações de objetos da caixa de instruções referente a atividade [3], deve haver a expressão que verifique se a variável que armazenou a resposta da atividade [2] tem valor “Não”. Da mesma forma, deve se aplicar a atividade [12] com o valor “Sim”.

Quando se tratar dos conflitos associados às atividades criadas pela divisão de arcos, o valor deve ser buscado na tarefa associada à atividade que estaria no lugar da atividade criada caso a divisão de arco não tivesse sido feita, ou seja, no caso da atividade [3.11] do Grafo 29, a atividade que lá estaria se não fosse feita a divisão de arco, seria a atividade [2.9] do Grafo 27 (que é também a atividade [9] do Grafo 26). Com isso, o valor da resposta para resolver o conflito para a caixa de instruções que esta associada a atividade [3.11] é “Não”.

Por fim, os distribuidores do PFS devem ser substituídos por lugares da RdPP e os arcos do PFS substituídos pelos arcos da RdPP. Uma marca deve ser adicionada ao lugar fornecedor da RdPP. o Grafo 30 mostra a representação final do *workflow* do modelo da Figura 64 em RdPP.

No Grafo 30, as variáveis *id*, *ans1*, *ans2*, *ans3*, *ans4*, *ans5*, *ans6*, *ans7* e *ans8* são definidas como variáveis globais pelo projeto de *workflow*. Essas variáveis são usadas tanto nas expressões booleanas das informações de objetos das caixas de instruções quanto para armazenar as informações da inspeção realizada.

Da mesma forma que esta, uma sequência de tarefas foi feita também para um outro processo de inspeção diária de poços de petróleo e equipamentos por Bombeio Centrífugo

Submerso (BCS). A Figura 65 mostra o modelo de sequência de tarefas da aplicação para inspeção diária em poços produtores de petróleo com equipamento por BCS.

Utilizando as mesmas técnicas anteriormente apresentadas, o *workflow* para a atividade da Figura 65 em RdPP está representado no Grafo 31.

Por se tratarem de atividades semelhantes (o *workflow* do Grafo 30 e do Grafo 31) diferindo entre si apenas pelo equipamento em que deve ser realizada a inspeção, é possível juntar ambos *workflows* em um único que atenda ambos e considere suas diferenças para diferentes equipamentos.

Para que esses dois *workflows* possam ser unificados, é preciso criar 2 caixas de instruções que não possuem instruções, ou seja, em que nenhuma tarefa precisa ser realizada. Isso é necessário para que em inspeções de poços com equipamentos BCS não sejam solicitadas tarefas que são exclusivas de poços com equipamentos BM ou BCP. O Grafo 32 mostra a unificação dos *workflows* do Grafo 30 e do Grafo 31.

No Grafo 32, a variável *equipamento* é uma variável da classe “ope-unidade” e tem valor definido pelo tipo de equipamento (BM, BCP ou BCS) que está instalado no poço produtor de petróleo em que está sendo feita a inspeção.

6.3 ANÁLISE DOS RESULTADOS

Sobre a RdPP, os testes com as devidas limitações de contexto, confirmaram que pessoas com pouco conhecimento da ferramenta são capazes de interpretar o processo solicitado. O teste realizado com operadores de máquinas teve resultado satisfatório. Apesar de não ter um espaço amostral que comprove a hipótese de que a RdPP é simples para ser compreendida por uma pessoa com pouco conhecimento dela, os resultados do teste sugerem claramente essa propriedade.

Por fim, a RdPP foi testada em caso real em uma aplicação industrial e comprovou que apresenta modelos mais simples de ser interpretados por técnicos e engenheiros da área de extração de petróleo.

Grafo 30 - Representação final do *workflow* do modelo da Figura 64 em RdPP

Figura 65 - Sequência de tarefas da inspeção diária de poços com BCS

Grafo 31 - Representação final do *workflow* do modelo da Figura 65 em RdPP

Grafo 32 - Workflow unificado para a atividade de inspeção de poços produtores de petróleo independente do equipamento utilizado

7 CONCLUSÕES

As características do mercado e do ambiente colaborativo foram apresentadas para definição e estudo do escopo do trabalho. Essas características influenciam no grau de interesse das empresas em colaborar umas com as outras nesse ambiente.

Para que as empresas colaborem umas com as outras em um mercado amplo como o oferecido pelo ambiente colaborativo, é preciso estudar as empresas para definir as parcerias que podem levar uma oportunidade de negócios ao sucesso. Isso implica em estudar a confiança que se pode atribuir a uma empresa de que a parceria com ela seria benéfica a outra. Foi, assim, proposta uma técnica para calcular e obter a confiança baseada na avaliação dos processos por outras empresas. Esse cálculo é mais efetivo quando uma empresa deseja saber o quão confiável é a outra empresa (com que nunca tenha trabalhado em projeto) para estabelecer parceria. O cálculo reaproveita as avaliações de terceiras empresas e de clientes, sobre os processos realizados pela empresa que se deseja conhecer. Essa é uma busca de informações sobre uma organização que não se conhece, numa forma de rede de contatos entre empresas, com apoio dos clientes.

Também foi proposta uma arquitetura de informação que inclui um *framework* de apoio para a utilização por empresas que desejam colaborar, mas que não possuem seus próprios SIs. O *framework* de apoio é composto de um conjunto de funções que envolvem a procura e o registro de informações sobre processos e empresas, além de dar suporte para o cálculo da confiança de outras empresas. Esse *framework* não obriga que as informações do processo tenham que transitar por seu SI, uma vez que as mesmas funções encontradas no *framework* e podem ser encontradas nas empresas que estão incluídas no ambiente colaborativo. A arquitetura proposta inclui ferramentas de coreografia, como o WS-CDL, que se foca na iteração entre as empresas, ao invés de sequenciá-las.

A rede de Petri produtiva (RdPP) foi aqui proposta para o *workflow*. Por utilizar como base a rede de Petri, sua interpretação é simples e atende plenamente uma das hipóteses geradas pelos requisitos da colaboração no ambiente colaborativo. Os testes realizados com a RdPP apresentaram resultados satisfatórios e conduzem a ferramenta a ser aplicada para outros testes em aplicações reais, como o realizado com a empresa de extração de petróleo.

A RdPP também foi testada com pessoas com pouco conhecimento na área de RdP e apresentou resultados satisfatórios para o entendimento do processo produtivo.

Como a RdPP foi testada com instruções semelhantes as encontradas no STEP-nc, é possível testá-la com o próprio STEP-nc para confirmar a hipótese de padronização da informação. Como a RdPP possui um padrão de sintaxes e uma semântica padronizada (baseada em XML) para sua troca de informações, ela atende os requisitos de uma ferramenta eficiente

para interoperabilidade da informação no ambiente colaborativo.

Por fim, a arquitetura proposta foi submetida à comparação com as hipóteses formuladas a partir dos requisitos das empresas na colaboração e no ambiente colaborativo. O estudo realizado na arquitetura foi analítico.

Entretanto, nem todas as hipóteses puderam ser estudadas analiticamente para a arquitetura por necessitarem de implementações mais detalhadas de seu funcionamento. A inadequação da arquitetura proposta às hipóteses foi nula, enquanto se identificou nas demais propostas de arquiteturas de informação de empresas virtuais, vários problemas. No Quadro 5 é apresentada uma comparação entre as arquiteturas de informação de EVs conhecidas e a arquitetura proposto nesse trabalho (AAAC - Arquitetura para Apoio ao Ambiente Colaborativo), de acordo com os requisitos apresentados.

Quadro 5 - Comparação entre arquiteturas de informação de EV existentes e a arquitetura proposta

Requisito	NIIP	PRODNET II	E-DREAM	AAAC
Universalização da informação + workflow	ISO-STEP c/ workflow	STEP EDI s/ workflow	s/ padrão e s/ workflow	STEP e RdPP
Banco de dados distribuídos	CORBA	CORBA	CORBA	SOA+ SODA + WS-CDL
Sub-contratação	Com barreiras	Sim	Não apresenta	Sim
Confiança	Não apresenta	Pode implementar	Não apresenta	Sim

Sendo assim, a arquitetura proposta atende potencialmente as necessidades da colaboração de empresas no ambiente colaborativo, utilizando o conceito de empresa virtual.

Para trabalhos futuros se pretende implementar a arquitetura proposta em um estudo de caso baseado em sistemas de manufatura integrada a computador, que são sistemas que simulam um ambiente real de manufatura. Também está em desenvolvimento uma ferramenta de modelagem e análise da RdPP e de PFS para ser utilizada em projetos e execução de manufatura.

TRABALHOS FUTUROS

Os trabalhos ainda a serem realizados com a arquitetura é estruturar e implementar ferramentas de projeto colaborativo, uma vez que essa é a única etapa desenvolvida na camada de negócios da ANSI-ISA 95 que ainda não está incluída na arquitetura para ser realizada de

forma colaborativa entre as empresas.

A arquitetura necessita também ser aplicada no mercado global para receber novos requisitos específicos das diferentes áreas.

Referências

AALST, W. van der. Workflow verification: Finding control-flow errors using Petri-net-based techniques. In: **Lecture Notes in Computer Science: Business process management**. Berlin: Springer, 2000. v. 1806, p. 161–183. DOI: 10.1007/3-540-45594-9_11.

AALST, W. van der; HEE, K. van. **Workflow management: Models, method, and systems**. Cambridge: MIT Press, 2004.

ADAM, N. R.; ATLURI, V.; HUANG, W.-K. Modeling and analysis of workflows using Petri nets. **Journal of Intelligent Information System**, Norwell, Kluwer Academic Publishers, v. 10, n. 2, p. 131–158, 1998. DOI: 10.1023/A:1008656726700.

AGÊNCIA NACIONAL DE TELECOMUNICAÇÕES. **Relatório Anual da Anatel 2012**. Brasília: Anatel, 2012. 144 p.

AMERICAN NATIONAL STANDARD. **ANSI/ISA-95.00.01 Enterprise-Control System Integration - Part 1: Models and terminology**. Research Triangle Park: International Society of Automation, 2010.

BERAN, P.; HABEL, G.; SCHIKUTA, E. SODA a distributed data management framework for the internet of services. 2008. Apresentado a 7th. International Conference on Grid and Cooperative Computing, Shenzhen, 2008. DOI: 10.1109/GCC.2008.61.

BHANDARKAR, M. P.; NAGI, R. STEP-based feature extraction from STEP geometry for agile manufacturing. **Computers in Industry**, Amsterdam, Elsevier, v. 41, p. 3–24, 2000. DOI: 10.1016/S0166-3615(99)00040-8.

BRAUER, W.; REISIG, W. Carl Adam Petri and “Petri nets”. **Informatik-Spektrum**, Berlin, Springer, v. 29, p. 369–381, 2006.

CALABRESE, F.; CELENTANO, G. Design and realization of a STEP-NC compliant CNC embedded controller. 2007. Apresentado a 12th. International Conference on Emerging Technologies and Factory Automation, Patras, 2007. DOI: 10.1109/EFTA.2007.4416894.

CAO, Q.; HOFFMAN, J. Alignment of virtual enterprise, information technology, and performance: An empirical study. **International Journal of Production Research**, Abingdon, Taylor & Francis LTD, v. 49, n. 4, p. 1127–1149, 2011. DOI: 10.1080/00207540903555478.

CHEN, S.-L.; CHEN, Y.-Y.; HSU, C. A new approach to integrate internet-of-things and software-as-a-service model for logistic systems: A case study. **Sensors**, Basel, MDPI AG, v. 14, p. 6144–6164, 2014. DOI: 10.3390/s140406144.

CHEN, Y.; CHIN, K.-S.; WANG, X. J. Internal pricing strategies design and simulation in virtual enterprise formation. **Expert Systems with Applications**, Kidlington, Pergamon-Elsevier Science LTD, v. 38, n. 11, p. 13580–13587, Oct 2011. DOI: 10.1016/j.eswa.2011.04.106.

CHEN, Y.; LI, Z.; ZHOU, M. Most permissive liveness-enforcing Petri net supervisors for flexible manufacturing systems. **International Journal of Production Research**, Abingdon, Taylor & Francis LTD, v. 50, n. 22, p. 6357–6371, Nov 2012. DOI: 10.1080/00207543.2011.637526.

CHIOLA, G. et al. Generalized stochastic Petri nets: A definition at the net level and its implications. **Transactions on Software Engineering**, New York, IEEE, v. 19, n. 2, p. 89–107, Feb 1993. DOI: 10.1109/32.214828.

CHO, J.-H.; SWAMI, A.; CHEN, I.-R. A survey on trust management for mobile ad hoc networks. **Communications Surveys Tutorials**, New York, IEEE, v. 13, n. 4, p. 562–583, 2011. DOI: 10.1109/SURV.2011.092110.00088.

CHOI, I.; PARK, C.; LEE, C. Task net: Transactional workflow model based on colored Petri net. **European Journal of Operational Research: Production, Manufacturing and Logistics**, London, Elsevier Limited, v. 136, p. 383–402, 2002. DOI: 10.1016/S0377-2217(01)00026-1.

CHOI, Y. et al. An enterprise architecture framework for collaboration of virtual enterprise chains. **International Journal of Advanced Manufacturing Technology**, London, Springer-Verlag London Limited, v. 35, p. 1065–1078, 2008. DOI: 10.1007/s00170-006-0789-7.

CHOW, T. et al. Ontology-based information sharing in service-oriented database systems. 2009. Apresentado a 6th. International Conference on Services Computing, Bangalore, 2009. DOI: 10.1109/SCC.2009.78.

DAVID, R. Grafcet: A powerful tool for specification of logic controllers. **Transactions on Control Systems Technology**, New York, IEEE, v. 3, n. 3, p. 253–268, 1995.

DELAMER, I.; LASTRA, J. Ontology modeling of assembly processes and systems using semantic web services. p. 611–617, Aug. 2006. Apresentado a 4th. IEEE International Conference on Industrial Informatics, Singapore, 2006. DOI: 10.1109/INDIN.2006.275631.

DI, X.-F.; FAN, Y.-S. Implementation of enterprises' interoperation based on ontology. **International Journal of Automation and Computing**, Beijing, Springer, v. 7, n. 3, p. 303–309, 2010. DOI: 10.1007/s11633-010-0507-7.

ESPOSITO, E.; EVANGELISTA, P. Investigating virtual enterprise models: Literature review and empirical findings. **International Journal of Production Economics**, Amsterdam, Elsevier, v. 148, p. 145–157, 2014. DOI: 10.1016/j.ijpe.2013.10.003.

ESTÉVEZ, E.; MARCOS, M. Model-based validation of industrial control systems. **Transactions on Industrial Informatics**, New York, IEEE, v. 8, n. 2, p. 302–310, 2012. DOI: 10.1109/TII.2011.2174248.

FARAHBOD, R. **Extending and Refining an Abstract Operational Semantics of the Web Services Architecture for the Business Process Execution Language**. 2001. 199 p. Dissertação (Mestrado) — Sharif University of Technology, Tehã, 2001.

FERREIRA, F. et al. Virtual enterprise process management: An application to industrial maintenance. **IFIP Advances in Information and Communication Technology: Collaborative systems for smart networked environments**. v. 434, p. 71–79, 2014. DOI: 10.1007/978-3-662-44745-1_7.

GARITA, C.; AFSARMANESH, H.; HERTZBERGER, L. The PRODNET cooperative information management for industrial virtual enterprises. **Journal of Intelligent Manufacturing**, Amsterdam, Kluwer Academic Publishers, v. 12, n. 2, p. 151–170, 2001. DOI: 10.1023/A:1011252510739.

GRIFFITHS, N. Enhancing peer-to-peer collaboration using trust. **Expert Systems with Applications**, Kidlington, Pergamon-Elsevier Science LTD, v. 31, p. 849–858, 2006. DOI: 10.1016/j.eswa.2006.01.017.

GUO, J. Global information interoperability through open information platform. 2013. Apresentado a 4th. International Conference on Software Engineering and Service Science, Beijing, 2013. DOI: 10.1109/ICSESS.2013.6615387.

GUSTIN, G. D. **Aplicação de redes de Petri interpretadas na modelagem de sistemas de elevadores em edifícios inteligentes**. 2000. Dissertação (Mestrado) — Escola Politécnica da Universidade de São Paulo, São Paulo, 2000.

HAMADI, R.; BENATALLAH, B. A Petri net-based model for web service composition. 2003. Apresentado a 14th. Australasian Database Conference, Darlinghurst, 2003.

HASEGAWA, K. et al. On resource arc for Petri net modelling of complex resource sharing system. **Journal of Intelligent and Robotic Systems**, Amsterdam, Springer, v. 26, n. 3-4, p. 423–437, 1999. DOI: 10.1023/A:1008122101686.

HE, L. et al. Modeling and analyzing the impact of authorization on workflow executions. **Future Generation Computer Systems**, London, Elsevier Limited, v. 28, p. 1177–1193, 2012. DOI: 10.1016/j.future.2012.03.003.

HORSTMANN, P. et al. Manufacturing virtual enterprises through the use of the national industrial information infrastructure protocols (NIIIP). **Proceedings of SPIE: The international society for optical engineering**, v. 2913, p. 20–47, Jan. 1997. Apresentado a Conference of Plug and Play Software for Agile Manufacturing, Boston, 1996. DOI: 10.1117/12.263456.

HUSPENINOVÁ, K. Virtual enterprise: Decision-making process of its formation. **International Journal of Mathematics and Computers in Simulation**, Oregon, NAUN Journals, v. 7, n. 2, p. 179–186, 2013.

INVESTOPEDIA. What's the difference between outsourcing and subcontracting? Disponível em: <<http://www.investopedia.com/ask/answers/032415/whats-difference-between-outsourcing-and-subcontracting.asp>>. Acesso em: 10 jun. 2015.

IP, W. et al. Genetic algorithm solution for a risk-based partner selection problem in a virtual enterprise. **Computers and Operations Research**, Amsterdam, Elsevier, v. 30, n. 2, p. 213–231, 2003. DOI: 10.1016/S0305-0548(01)00092-2.

IVANOVA, D.; FREY, G.; BATCHKOVA, I. Intelligent component based batch control using IEC61499 and ANSI/ISA S88. v. 1, 2008. Apresentado a 4th. International Conference on Intelligent Systems, Varna, 2008. DOI: 10.1109/IS.2008.4670424.

JARIMO, T.; SALO, A. Multicriteria partner selection in virtual organizations with transportation costs and other network interdependencies. **Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews**, New York, IEEE, v. 39, n. 1, p. 124–129, 2009. DOI: 10.1109/TSMCC.2008.2007251.

JASTI, N. V. K.; KODALI, R. Lean production: Literature review and trends. **International Journal of Production Research**, Abingdon, Taylor & Francis LTD, v. 53, n. 3, p. 867–885, 2015. DOI: 10.1080/00207543.2014.937508.

JENSEN, K. Coloured Petri nets and the invariant-method. **Theoretical Computer Science**, Amsterdam, North-Holland Publishing Company, p. 317–336, 1981. DOI: 10.1016/0304-3975(81)90049-9.

JENSEN, K. **Coloured Petri nets: Basic concepts, analysis methods and practical use**. Berlin: SpringerVerlag, 1992.

JIANG, Z. et al. Research on enterprise interoperability for cluster supply chain. 2012. Apresentado a 16th. International Conference on Computer Supported Cooperative Work in Design, Wuhan, 2012. DOI: 10.1109/CSCWD.2012.6221867.

KANESHIRO, P. J. et al. Modeling of collision resolution algorithm in Lonworks networks. **Proceedings of ASME International Mechanical Engineering Congress and Exposition**, p. 743–749, 2008. DOI: 10.1115/IMECE2007-43950.

KENDALL. **Make it Collaborative Design**: Kendall College of Art and Design. Folheto da Kendall College of Art and Design (2015). Disponível em: . Acessado em: 18/01/2015.

KIEPUSZEWSKI, B.; HOFSTED, A. ter; AALST, W. van der. Fundamentals of control flow in workflows. **Acta Informatica**, New York, Springer, v. 39, n. 3, p. 143–209, 2003. DOI: 10.1007/s00236-002-0105-4.

KIM, W. et al. WSCPC: An architecture using semantic web services for collaborative product commerce. **Computers in Industry**, Amsterdam, Elsevier, v. 57, p. 787–796, 2006. DOI: 10.1016/j.compind.2006.04.007.

KLEIN, M. et al. The dynamics of collaborative design: Insights from complex systems and negotiation research. **Concurrent Engineering Research & Applications**, Thousand Oaks, SAGE Publications Ltd, v. 11, n. 3, p. 201–209, 2003.

KROPOTKIN, P. A. **A conquista do pão**. Tradução de Cesar Falcão. 2. ed. Rio de Janeiro: Editora Achiamé, 2011. 152 p., 23 cm.

LEE, J.-S.; ZHOU, M.-C.; HSU, P.-L. An application of Petri nets to supervisory control for human-computer interactive systems. **Transactions on Industrial Electronics**, New York, IEEE, v. 52, n. 5, p. 1220–1226, 2005. DOI: 10.1109/TIE.2005.855694.

LEINA, Z.; TIEJUN, P. Partner trust evaluation method of virtual enterprise. **Information Technology**, Faisalabad, ANSInet, v. 11, n. 4, p. 524–527, 2012. DOI: 10.3923/itj.2012.524.527.

LEITÃO, P. Agent-based distributed manufacturing control: A state-of-art survey. **Engineering Applications of Artificial Intelligence**, Amsterdam, Elsevier, v. 22, p. 979–991, 2009. DOI: 10.1016/j.engappai.2008.09.005.

LI, J.; FAN, Y.; ZHOU, M.-C. Performance modeling and analysis of workflow. **Transactions on Systems Man and Cybernetics Part A: Systems and Humans**, New York, IEEE, v. 34, n. 2, p. 229–242, Mar 2004. DOI: 10.1109/TSMCA.2003.819490.

LIU, D. et al. Modeling workflow processes with colored Petri nets. **Computers in Industry**, Amsterdam, Elsevier, v. 49, p. 267–281, 2002. DOI: 10.1016/S0166-3615(02)00099-4.

LIU, R.; ZHANG, C. An agent-based framework for STEP-NC controllers of CNC machine tools. **Applied Mechanics and Materials**, Durnten, Trans Tech Publications, v. 44-47, p. 889–893, 2011. DOI: 10.4028/www.scientific.net/AMM.44-47.889.

LU, Y.; ZHANG, L.; SUN, J. Using colored Petri nets to model and analyze workflow with separation of duty constraints. **International Journal of Advanced Manufacturing Technology**, London, Springer-Verlag London Limited, v. 40, p. 179–192, 2009. DOI: 10.1007/s00170-007-1316-1.

MACKENZIE, C. M. et al.(Ed.). **Reference Model for Service Oriented Architecture 1.0**. [S.l.]: OASIS, aug 2006.

MAYERL, C.; VOGEL, T.; ABECK, S. SOA-based integration of service management applications integrated IT service management. 2005. Apresentado a 3rd. International Conference on Web Services, Orlando, 2005. DOI: 10.1109/ICWS.2005.114.

MEDINA, I. et al. Proposal of a methodology for implementing a service-oriented architecture in distributed manufacturing systems. 2009. Apresentado a 3rd. Manufacturing Engineering Society International Conference, Alicante, 2009. DOI: 10.1063/1.3273683.

MELO, J. I.; MORALES, R. A. G.; MIYAGI, P. E. Supervisory system for hybrid productive systems based on Bayesian networks and OO-DPT nets. 2008. Apresentado a 13th. International Conference on Emerging Technologies and Factory Automation, Hamburg, 2008. DOI: 10.1109/ETFA.2008.4638536.

MHAKNO, N. et al. Plataforma organizacional dos comunistas libertarios. **Dielo Trouda**, Group of Russian Anarchists Abroad, 1926.

MORALES, R. A. G.; MELO, J. I.; MIYAGI, P. E. Diagnosis and treatment of faults in productive systems based on Bayesian networks and Petri net. **IEEE International Conference on Automation Science and Engineering, CASE**. p. 357–362, 2007. DOI: 10.1109/COASE.2007.4341823.

MUN, J.; SHIN, M.; JUNG, M. A goal-oriented trust model for virtual organization creation. **Journal of Intelligent Manufacturing**, Amsterdam, Springer, v. 22, p. 345–354, 2011. DOI: 10.1007/s10845-009-0293-7.

MURATA, T. Petri nets: Properties, analysis and applications. **Proceedings of IEEE**, New York, IEEE, v. 77, n. 4, p. 541–580, 1989. DOI: 10.1109/5.24143.

NASSAR, M. G. V. et al. Modeling and analysing of the material entry flow system in a pickling line process using Petri nets. **ABCM Symposium Series in Mechatronics**. p. 444–453, 2008.

PIRES, A.; PUTNIK, G.; AVILA, P. A survey analysis of the resource selection models in agile/virtual enterprises. **Journal of Applied Research and Technology**, Mexico City, Universidad Nacional Autónoma de México, v. 10, n. 3, p. 416–427, Jun 2012.

POLENSKE, K. Competition, collaboration and cooperation: An uneasy triangle in networks of firms and regions. **Regional Studies**, Abingdon, Taylor & Francis LTD, v. 38, n. 9, p. 1029–1043, 2004. DOI: 10.1080/0034340042000292629.

SANTOS, I. J. G. D.; MADEIRA, E. R. M. Applying orchestration and choreography of web services on dynamic virtual marketplaces. **International Journal of Cooperative Infor-**

mation Systems, Singapore, World Scientific Publishing, v. 15, n. 1, p. 57–85, 2006. DOI: 10.1142/S0218843006001281.

SARI, B.; SEN, T.; KILIC, S. AHP model for the selection of partner companies in virtual enterprises. **International Journal of Advanced Manufacturing Technology**, London, Springer-Verlag London Limited, v. 38, n. 3-4, p. 367–376, 2008. DOI: 10.1007/s00170-007-1097-6.

SEOK, H.; NOF, S. Y. Collaborative capacity sharing among manufacturers on the same supply network horizontal layer for sustainable and balanced returns. **International Journal of Production Research**, Abingdon, Taylor & Francis LTD, v. 52, n. 6, p. 1622–1643, 2014. DOI: 10.1080/00207543.2013.842016.

SQUIT, S. et al. Orchestrating dispersed productive systems. 2013. Apresentado a 18th. International Conference on Emerging Technologies and Factory Automation, Cagliari, 2013. DOI: 10.1109/ETFA.2013.6648132.

STEIN, J. **Marxismo**: A negação do comunismo. Tradução de Luther Blisset. Rio de Janeiro: Achiamé, 2009. 116 p., 12x18 cm.

TAKAHASHI, K.; HASEGAWA, K.; MIYAGI, P. Hierarchical programming of sequential control systems by mark flow graph. v. 2, 1999. Apresentado a 7th. International Conference on Emerging Technologies and Factory Automation, Barcelona, 1999. DOI: 10.1109/ETFA.1999.813138.

TALLURI, S.; BAKER, R. Quantitative framework for designing efficient business process alliances. **Proceedings of the International Engineering Management Conference**. p. 656–661, 1996. DOI: 10.1109/IEMC.1996.547896.

TANG, D. et al. E-DREAM: A web-based platform for virtual agile manufacturing. **Concurrent Engineering Research & Applications**, Thousand Oaks, SAGE Publications Ltd, v. 10, n. 2, p. 165–183, Jun. 2002. DOI: 10.1177/1063293X02010002698.

TAO, F. et al. GA-BHTR: an improved genetic algorithm for partner selection in virtual manufacturing. **International Journal of Production Research**, Abingdon, Taylor & Francis LTD, v. 50, n. 8, p. 2079–2100, 2012. DOI: 10.1080/00207543.2011.561883.

VILLANI, E.; MIYAGI, P. E.; VALETTE, R. **Modelling and analysis of hybrid supervisory systems**: A Petri net approach. London: Springer-Verlag London Limited, 2007.

WANG, D.; YUNG, K.; IP, W. A heuristic genetic algorithm for subcontractor selection in a global manufacturing environment. **Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews**, New York, IEEE, v. 31, n. 2, p. 189–198, 2001. DOI: 10.1109/5326.941842.

WANG, H.; HU, Q. Research and application of an integration platform for e-commerce system based on SOA. 2009. Apresentado a 3rd. International Conference on Management of e-Commerce and e-Government, Nanchang, 2009. DOI: 10.1109/ICMeCG.2009.113.

WANG, J.; SONG, Y.-T.; XIONG, Y. Service evaluation in SOA: Toward business/IT alignment. 2009. Apresentado a 10th. International Conference on Software Engineering, Artificial Intelligences, Networking and Parallel/Distributed Computing, Daegu, 2009. DOI: 10.1109/SNPD.2009.105.

WANG, X.; WONG, T. N.; WANG, G. An ontological intelligent agent platform to establish an ecological virtual enterprise. **Expert Systems with Applications**, Kidlington, Pergamon-Elsevier Science LTD, v. 39, n. 8, p. 7050–7061, Jun 2012. DOI: 10.1016/j.eswa.2012.01.042.

WANG, X.; WONG, T. N.; WANG, G. Service-oriented architecture for ontologies supporting multi-agent system negotiations in virtual enterprise. **Journal of Intelligent Manufacturing**, Dordrecht, Springer, v. 23, n. 4, p. 1331–1349, Aug 2012. DOI: 10.1007/s10845-010-0469-1.

WEBER, M.; KINDLER, E. The Petri net markup language. **Lecture Notes in Computer Science: Advances in Petri Nets**, Dordrecht, Springer, v. 2472, p. 124–144, 2003. DOI: 10.1007/978-3-540-40022-6_7.

YOO, T.; JEONG, B.; CHO, H. A Petri nets based functional validation for services composition. **Expert Systems with Applications**, Kidlington, Pergamon-Elsevier Science LTD, v. 37, n. 5, p. 3768–3776, 2010. DOI: 10.1016/j.eswa.2009.11.046.

ZENG, Z.-B.; LI, Y.; ZHU, W. Partner selection with a due date constraint in virtual enterprises. **Applied Mathematics and Computation**, Amsterdam, Elsevier, v. 175, n. 2, p. 1353–1365, 2006. DOI: 10.1016/j.amc.2005.08.022.

ZHANG, W. et al. A new manufacturing resource allocation method for supply chain optimization using extended genetic algorithm. **International Journal of Advanced Manufacturing Technology**, London, Springer-Verlag London Limited, v. 53, n. 9-12, p. 1247–1260, 2011. DOI: 10.1007/s00170-010-2900-3.

ZHU, W.; MELO, W. Refactoring J2EE application for JBI-based ESB: A case study. 2009. Apresentado a 13th. International Enterprise Distributed Object Computing Conference, Auckland, 2009. DOI: 10.1109/EDOC.2009.25.

ZURAWSKI, R.; ZHOU, M. Petri nets and industrial applications: A tutorial. **Transactions on Industrial Electronics**, New York, IEEE, v. 41, n. 6, p. 567–583, 1994. DOI: 10.1109/41.334574.

Anexo A – PPNet Tool

A *PPNet Tool* é uma ferramenta desenvolvida nesse projeto para edição de modelos de processos em RdPP, ou seja, grafos RdPP. Para desenvolvimento, inicialmente foi escolhido o ambiente de trabalho esperado, para escolha das linguagens e bibliotecas que atendem as necessidades dessa.

Inicialmente se definiu que a ferramenta deve ser de código aberto com licença não comercial e foram adotados os princípios e recomendações recentes da área de Engenharia de Software que procuram assegurar compatibilidade com os diferentes sistemas operacionais do mercado. As linguagens que utilizam máquinas virtuais simplificam o desenvolvimento de aplicações compatíveis com todos os sistemas operacionais.

Devido a grande disponibilidade de documentação e aplicações recentes desenvolvidas, a linguagem escolhida para o desenvolvimento foi o *Python 2.6*.

Com a escolha da linguagem, é necessário escolher as bibliotecas que essas possuem e que auxiliam no desenvolvimento da aplicação. Uma categoria de bibliotecas importante para o desenvolvimento de ferramentas de modelagem, são as biblioteca de interface gráfica para usuários (ou *GUI*). Como a RdPP também trabalha com PNML, que é um modelo estruturado de informação baseado em XML, a outra biblioteca importante de ser utilizada é uma biblioteca de leitura e escrita de arquivos em XML.

Devido a simplicidade de instalação em sistema operacional *Windows* e *Linux*, além da documentação melhor elaborada, a biblioteca de interface gráfica de usuário escolhida foi a *Qt4* (que no *Python* também pode ser encontrada com *PyQt4*). Para a leitura e escrita de XML, a biblioteca *Qt4* oferece também classes para desenvolver aplicações utilizando XML, mas a comunidade que mantém o *Python* recomenda outras bibliotecas, como a aqui utilizada, que é o *etree*.

Com a linguagem de programação e as bibliotecas definidas, são definidas os casos de uso associados as atores que utilizam a ferramenta. Esses casos de uso podem ser vistos na Figura A.1.

changeFile: é um caso de uso para alternância entre múltiplos arquivos de modelos diferentes abertos nas abas da ferramenta. Para alternar entre as abas, o usuário deve selecionar (clicar em), entre as opções de abas, aquela que se refere ao arquivo que deseja visualizar e/ou editar. Ao escolher uma nova aba, o estado da rede deve ir para um estado inicial (ou *home state*).

closeFile: é um caso de uso para encerrar um arquivo de modelo aberto. Para encerrar o arquivo de modelo, o usuário deve selecionar o botão no formato de X presente dentro

Figura A.1 : Casos de uso da *PPNet Tool*

do espaço da aba referente ao arquivo desejado. Ao encerrar o arquivo, a função verifica se esse está salvo e pergunta se o usuário deseja salvá-lo. Ao encerrar o arquivo o caso de uso **changeFile** é então acionado para alterar a visualização para a última página acionada.

copyItems: é um caso de uso para armazenar uma cópia de elementos selecionados na tela de edição para posterior clonagem no mesmo arquivo de modelo ou em outro. Para efetuar a cópia, o usuário deve selecionar o botão “Copy” ou um atalho de teclado para cópia (“Ctrl” + “C”). Após iniciada a cópia, a função monta uma lista dos elementos que são copiáveis, de acordo com as regras de modelagem da RdPP.

createArc: é um caso de uso para criação do elemento arco na tela do modelo. Para ser criado, inicialmente, deve-se selecionar o botão de criação de arcos. Depois, deve-se selecionar um lugar ou uma transição já existentes na tela do modelo para ser a origem do arco. Por fim, seleciona-se outro lugar ou transição já existentes na tela do modelo para ser o destino do arco. A função então verifica se o arco pode ser criado com essa origem e esse destino e então cria o arco.

createFile: é um caso de uso para criação de novos arquivos de modelos. Para criar um

novo arquivo, o usuário deve selecionar o botão de criação de arquivos ou utilizar uma tecla de atalho (“Ctrl” + “N”). Ao criar o novo arquivo o caso de uso **changeFile** é então acionado para alterar a visualização para a nova página.

createInstructionBox: é um caso de uso para criação do elemento caixa de instruções na tela do modelo. Para ser criado, inicialmente, deve-se selecionar o botão de criação de caixas de instruções. Depois, deve-se selecionar uma transição já existente na tela e que ainda não tenha uma caixa de instruções associada a ela. A função verifica se as condições para a criação da caixa de instruções foram atendidas e cria a caixa de instruções.

createPlace: é um caso de uso para criação do elemento lugar na tela do modelo. Para ser criado, deve-se selecionar o botão de criação de lugares. Depois, deve-se selecionar em um espaço livre da tela, ou seja, um espaço da tela de edição em que não tenha nenhum elemento. A função então cria o elemento lugar no ponto da tela selecionado.

createTransition: é um caso de uso para criação do elemento transição na tela do modelo. Para ser criado, deve-se selecionar o botão de criação de transições. Depois, deve-se selecionar em um espaço livre da tela, ou seja, um espaço da tela de edição em que não tenha nenhum elemento. A função então cria o elemento transição no ponto da tela selecionado.

deleteItems: é um caso de uso para remover elementos selecionados na tela de edição. Para remover os elementos o usuário deve apertar a tecla de atalho (“Delete”). A função cria uma lista com os elementos selecionados e ainda adiciona outros elementos que não podem existir no modelo sem a existência dos que já estão na lista. Depois a função os remove do modelo.

editProperties: é um caso de uso para editar as informações (ou propriedades) dos elementos e/ou do arquivo do modelo. Para editar as informações, o usuário deve selecionar o botão que abre o campo para edição de informações e selecionar um item da lista de itens editáveis (cada elemento ou a página). A função então verifica se os dados editados atendem as condições especificadas para cada campo e altera as informações do elemento e/ou página.

editText: é um caso de uso para editar o conteúdo dos textos que aparecem no arquivo do modelo. Para editar o texto, o usuário deve clicar duas vezes com o botão esquerdo do mouse sobre o texto que deseja editar. A função abre uma tela para edição de texto e aguarda o usuário entrar com o conteúdo desejado. Após a entrada, a função altera o texto na tela de edição.

exportFile: é um caso de uso para salvar um arquivo de modelo em um arquivo de formato de gráficos escaláveis (SVG). Para salvar o arquivo, o usuário deve selecionar o botão para exportar no formato SVG. Depois o usuário deve selecionar um arquivo de destino. A função então grava cada elemento no arquivo SVG de saída.

moveItems: é um caso de uso para movimentar os elementos selecionados na tela edição

entre dois pontos. Para movimentar os elementos, o usuário deve clicar com o botão esquerdo do mouse em um dos elementos selecionados e manter esse botão selecionado. Depois deve movimentar o mouse com o botão esquerdo selecionado até a posição desejada. Outra alternativa utilizar as teclas de atalhos (setas) para movimentar os elementos. A função verifica se nenhum dos elementos cruza as fronteiras (bordas) do arquivo do modelo e movimenta para o ponto desejado.

openFile: é um caso de uso para abrir arquivos de modelos existentes. Para abrir um arquivo existente, o usuário deve selecionar o botão de abertura de arquivos ou utilizar uma tecla de atalho (“Ctrl” + “O”). A função então abre uma janela de diálogo para selecionar um arquivo. Após selecionado o arquivo o caso de uso **changeFile** é acionado para alterar a visualização para a nova página.

pasteItems: é um caso de uso para clonagem dos elementos selecionados em lista a partir do caso de uso **copyItems**. Para efetuar a clonagem, o usuário deve ter previamente clicado em algum local da tela de edição e então selecionar o botão de colar, ou o atalho do teclado (“Ctrl” + “V”). Os itens da lista são então clonados no arquivo do modelo.

redoAction: é um caso de uso para refazer uma ação que tenha sido desfeita. Para refazer ação, o usuário deve utilizar o botão refazer ou o atalho do teclado (“Ctrl” + “W”). Essa função então refaz a última ação desfeita e desloca essa ação para o topo da lista de ações por desfazer.

resizeItems: é um caso de uso para redimensionar graficamente os elementos selecionados na tela edição. Para redimensionar os elementos, o usuário deve clicar com o botão esquerdo do mouse sobre os botões seletores de cada elemento (que ficam próximos a eles) e manter esse botão selecionado. Depois deve movimentar o mouse com o botão esquerdo selecionado até o tamanho desejado. A função verifica se nenhum dos elementos cruza as fronteiras (bordas) do arquivo do modelo e redimensiona os elementos para o tamanho desejado.

saveFile: é um caso de uso para gravar o conteúdo de um arquivo de modelo num arquivo PNML. Para gravar o conteúdo, o usuário deve selecionar o botão salvar ou o atalho de teclado (“Ctrl” + “S”). Caso o arquivo de modelo já não tenha sido gravado nenhuma vez anteriormente a função mostra uma janela para seleção de um arquivo PNML para gravar o conteúdo. Caso já tenha sido gravado anteriormente, essa janela não é aberta. Depois a função converte os elementos em dados PNML e grava no arquivo selecionado.

selectItems: é um caso de uso para selecionar um elemento na tela de edição. Para selecionar um elemento, o usuário deve clicar com o botão esquerdo do mouse sobre o elemento na tela e soltar o botão logo em seguida. A função verifica se deve desselecionar os demais elementos antes de selecionar esse. Depois a função marca o elemento como selecionado.

undoAction: é um caso de uso para desfazer uma ação que tenha sido feita ou refeita.

Para desfazer ação, o usuário deve utilizar o botão desfazer ou o atalho do teclado (“Ctrl” + “Z”). Essa função então desfaz a última ação feita ou refeita e desloca essa ação para o topo da lista de ações por refazer.

Os casos de uso são apresentados em mais detalhes por seus diagramas de atividade.

Para o caso de uso **changeFile**, o diagrama de atividade é o da Figura A.2.

Figura A.2 : Diagrama de atividade do caso de uso **changeFile**

Para o diagrama de atividades da Figura A.2, inicialmente a função pega o valor índice da página que se deseja visualizar. Para a página que se abre, redefine os valores das variáveis usadas para determinar o estado da ferramenta e redefine o próprio estado da ferramenta para valores iniciais e estado inicial. Depois a função mostra a página que se refere ao valor índice obtido.

Para o caso de uso **closeFile**, o diagrama de atividade é o da Figura A.3.

Figura A.3 : Diagrama de atividade do caso de uso **closeFile**

Para o diagrama de atividades da Figura A.3, inicialmente a função verifica o valor da variável associada a mudanças não salvas do arquivo. Caso a variável indique que não há mudanças não salvas no arquivo, a função vai direto para a atividade de encerramento da página. Caso a variável indique que há mudanças não salvas no arquivo, a função pergunta para o usuário se

deseja salvar o arquivo. Se o usuário responder que não deseja salvar o arquivo, a função vai direto para a atividade de encerramento da página, caso contrário a função chama o caso de uso **saveFile**. Se o caso de uso **saveFile** retornar um que o arquivo não foi salvo, a função se encerra sem fazer nada, se retornar que o arquivo foi salvo, a função passa para a atividade de encerramento da página, em que ela remove a página entre as páginas abertas. Depois, a função chama o caso de uso **changeFile** com o índice de outra página aberta.

Para o caso de uso **copyItems**, o diagrama de atividade é o da Figura A.4.

Figura A.4 : Diagrama de atividade do caso de uso **copyItems**

Para o diagrama de atividades da Figura A.4, inicialmente a função cria uma lista dos elementos a serem copiados com os lugares e transições selecionados dentro dessa lista. Depois a função pega os arcos selecionados. Para cada arco selecionado, a função verifica

se os nós (lugar e transição) que esse arco ligam já estão presentes na lista criada. Caso estejam presentes, a função adiciona o arco à lista. Depois a função pega as caixas de instrução selecionadas. Para cada caixa de instrução, a função verifica se a transição associada a essa está presente na lista criada. Caso esteja presente, a função adiciona a caixa de instrução à lista. A função então grava a lista e a posição atual do mouse dentro da tela de edição.

Para o caso de uso **createFile**, o diagrama de atividade é o da Figura A.5.

Figura A.5 : Diagrama de atividade do caso de uso **createFile**

Para o diagrama de atividades da Figura A.5, inicialmente a função cria uma nova página. Depois a função chama o caso de uso **changeFile** com o valor índice da página criada, para que essa seja exibida.

Para o caso de uso **createArc**, o diagrama de atividade é o da Figura A.6.

Figura A.6 : Diagrama de atividade do caso de uso **createArc**

Para o diagrama de atividades da Figura A.6, inicialmente a função muda a cor do botão de criar arcos para que ele fique em destaque para o usuário. Depois, a função aguarda o usuário selecionar um do nós na tela de edição ou o usuário selecionar algum botão da barra de ferramentas. Se o usuário selecionar um botão da barra de ferramentas a função altera a

cor do botão de criar arcos para a cor normal e termina. Se o usuário selecionar um nó, a função aguarda o usuário selecionar outro nó ou o usuário selecionar algum botão da barra de ferramentas. Se o usuário selecionar um botão da barra de ferramentas, a função repete as etapas já anteriormente descritas para o mesmo caso. Se o usuário selecionar outro nó, a função verifica se os nós cumprem os requisitos para criação de um arco. Caso cumpram os requisitos, a função cria uma ação de retorno (é a técnica de gravar duas pilhas, uma de ações para retornar e outra de ações para refazer) para a criação do arco e define a página como alterada. Depois a função altera a cor do botão de criar arcos para o normal. Por fim, a função verifica se o usuário manteve a tecla “shift” pressionada. Caso tenha mantido, a função retorna para o início de seu processamento. Caso contrário, a função finaliza.

Para o caso de uso **createInstructionBox**, o diagrama de atividade é o da Figura A.7.

Figura A.7 : Diagrama de atividade do caso de uso **createInstructionBox**

Para o diagrama de atividades da Figura A.7, inicialmente a função muda a cor do botão de criar caixas de instruções para que ele fique em destaque para o usuário. Depois, a função aguarda o usuário selecionar uma das transições na tela de edição ou o usuário selecionar algum botão da barra de ferramentas. Se o usuário selecionar um botão da barra de ferramentas a função altera a cor do botão de criar caixas de instruções para a cor normal e termina. Se o usuário selecionar uma transição, a função cria uma ação de retorno para a criação da caixa de instruções e define a página como alterada. Depois a função altera a cor do botão de criar caixas de instruções para o normal. Por fim, a função verifica se o usuário manteve a tecla “shift” pressionada. Caso tenha mantido, a função retorna para o início de seu processamento. Caso contrário, a função finaliza.

Para o caso de uso **createPlace**, o diagrama de atividade é o da Figura A.8.

Figura A.8 : Diagrama de atividade do caso de uso **createPlace**

Para o diagrama de atividades da Figura A.8, inicialmente a função muda a cor do botão de criar lugares para que ele fique em destaque para o usuário. Depois, a função aguarda o usuário selecionar um espaço na tela de edição ou o usuário selecionar algum botão da barra de ferramentas. Se o usuário selecionar um botão da barra de ferramentas a função altera a cor do botão de criar lugares para a cor normal e termina. Se o usuário selecionar um espaço vazio, a função cria uma ação de retorno para a criação do lugar e define a página como alterada. Depois a função altera a cor do botão de criar lugares para o normal. Por fim, a função verifica se o usuário manteve a tecla “shift” pressionada. Caso tenha mantido, a função retorna para o início de seu processamento. Caso contrário, a função finaliza.

Para o caso de uso **createTransition**, o diagrama de atividade é o da Figura A.9.

Figura A.9 : Diagrama de atividade do caso de uso **createTransition**

Para o diagrama de atividades da Figura A.9, inicialmente a função muda a cor do botão

de criar transições para que ele fique em destaque para o usuário. Depois, a função aguarda o usuário selecionar um espaço na tela de edição ou o usuário selecionar algum botão da barra de ferramentas. Se o usuário selecionar um botão da barra de ferramentas a função altera a cor do botão de criar lugares para a cor normal e termina. Se o usuário selecionar um espaço vazio, a função cria uma ação de retorno para a criação da transição e define a página como alterada. Depois a função altera a cor do botão de criar lugares para o normal. Por fim, a função verifica se o usuário manteve a tecla “shift” pressionada. Caso tenha mantido, a função retorna para o início de seu processamento. Caso contrário, a função finaliza.

Para o caso de uso **deleteItems**, o diagrama de atividade é o da Figura A.10.

Figura A.10 : Diagrama de atividade do caso de uso **deleteItems**

Para o diagrama de atividades da Figura A.10, inicialmente a função cria uma lista dos elementos a serem apagados com os lugares selecionados na tela de edição dentro dessa lista. Para cada um desses lugares, a função pega os arcos com origem ou com destino nesse lugar e adiciona esses arcos na lista. Depois a função pega as transições selecionadas na tela de

edição e coloca na lista. Para cada uma dessas transições, a função pega os arcos com origem ou com destino nessa transição e adiciona esses arcos na lista. Para cada uma dessas transições, a função pega a caixa de instruções associada a essa transição e adiciona a lista. Depois a função pega os arcos selecionados e coloca eles na lista. Depois a função pega as caixas de instruções selecionadas e coloca elas na lista. Por fim, a função cria uma ação de retorno para a remoção dos elementos e define a página como alterada.

Para o caso de uso **editProperties**, o diagrama de atividade é o da Figura A.11.

Figura A.11 : Diagrama de atividade do caso de uso **editProperties**

Para o diagrama de atividades da Figura A.11, inicialmente a função apresenta uma lista de todos os elementos da tela de edição para o usuário. Depois a função pega para o elemento selecionado, o conjunto de propriedades desse com seus valores. A função então aguarda o usuário alterar algum valor de um dos itens das propriedades. Se nenhum for alterado e houver algum outro evento, a função se encerra. Se algum item for alterado, a função verifica se o valor alterado cumpre os requisitos do item da propriedade. Se não cumprir, a função se encerra sem alterar os dados. Se cumprir, a função cria uma ação de alteração das propriedades e define a página como alterada.

Para o caso de uso **editText**, o diagrama de atividade é o da Figura A.12.

Figura A.12 : Diagrama de atividade do caso de uso **editText**

Para o diagrama de atividades da Figura A.12, inicialmente a função apresenta uma caixa

de diálogo de edição de texto para o usuário. A função então aguarda o usuário digitar um texto e selecionar um dos botões da caixa. Se o botão selecionado for o “Cancelar”, a função se encerra. Se selecionado for o “ok”, a função verifica se o valor alterado cumpre os requisitos do texto. Se não cumprir, a função se encerra sem alterar o texto. Se cumprir, a função cria uma ação de alteração do texto e define a página como alterada.

Para o caso de uso **exportFile**, o diagrama de atividade é o da Figura A.13.

Figura A.13 : Diagrama de atividade do caso de uso **exportFile**

Para o diagrama de atividades da Figura A.13, inicialmente a função apresenta uma caixa de diálogo de gravação de arquivo. A função então aguarda o usuário selecionar um arquivo e selecionar um dos botões da caixa. Se o botão selecionado for o “Cancelar”, a função se encerra. Se selecionado for o “ok”, a função cria um arquivo SVG no local especificado. Depois a função escreve no arquivo cada um dos elementos da tela de edição.

Para o caso de uso **moveItems**, o diagrama de atividade é o da Figura A.14.

Figura A.14 : Diagrama de atividade do caso de uso **moveItems**

Para o diagrama de atividades da Figura A.14, inicialmente a função verifica para cada elemento selecionado se este ultrapassa os valores superiores, inferiores esquerdo ou direito dos limites da página. Se não ultrapassarem, a função define o ponto como o ponto correto. Se ultrapassarem, a função define o ponto correto como o ponto da borda da página mais próximo ao ponto desejado pelo usuário. A função, então, cria uma ação de retorno para a movimentação dos elementos e define a página como alterada.

Para o caso de uso **openFile**, o diagrama de atividade é o da Figura A.15.

Figura A.15 : Diagrama de atividade do caso de uso **openFile**

Para o diagrama de atividades da Figura A.15, inicialmente a função apresenta uma caixa de dialogo para abertura de arquivo. A função, então, aguarda o usuário selecionar um arquivo e selecionar um dos botões da caixa. Se o botão selecionado for o “Cancelar”, a função se encerra. Se selecionado for o “ok”, a função verifica se o arquivo selecionado é um arquivo

PNML. Se não for um arquivo PNML, a função se encerra. Se for um arquivo PNML, a função cria uma nova página associada a esse arquivo. Depois, para cada elemento obtido no arquivo PNML, a função adiciona o elemento na tela de edição. Por fim, a função chama o caso de uso **changeFile** com o valor índice dessa página criada.

Para o caso de uso **pastelItems**, o diagrama de atividade é o da Figura A.16.

Figura A.16 : Diagrama de atividade do caso de uso **pastelItems**

Para o diagrama de atividades da Figura A.16, inicialmente a função verifica se a página atual (na qual se deseja clonar os elementos) é a mesma em que os elementos foram copiados. Se não forem, a função define o ponto de origem da clonagem preferencialmente como o ponto atual da tela de edição. Se forem, a função define o ponto de origem da clonagem como o ponto da cópia mais o quanto o ponteiro do mouse deslocou entre a cópia e a clonagem. A função, então, cria uma ação de retorno para a clonagem dos elementos e define a página como alterada.

Para o caso de uso **redoAction**, o diagrama de atividade é o da Figura A.17.

Figura A.17 : Diagrama de atividade do caso de uso **redoAction**

Para o diagrama de atividades da Figura A.17, inicialmente a função chama a função refazer da ação no topo da pilha de ações para refazer, ou seja, a última a ser refeita (todo esse gerenciamento das pilhas e da movimentação entre as pilhas é feito pela própria biblioteca do *Qt4*). A função então verifica se o ponto alcançado após a ação refeita é o mesmo ponto do último salvamento do arquivo. Caso seja, a função define a página como não alterada. Caso não seja, a função define a página como alterada.

Para o caso de uso **resizeItems**, o diagrama de atividade é o da Figura A.18.

Figura A.18 : Diagrama de atividade do caso de uso **resizeItems**

Para o diagrama de atividades da Figura A.18, inicialmente a função monta uma lista dos elementos selecionados que são do mesmo tipo do elemento em que o seletor foi clicado. Depois a função pega a informação do tipo de seletor que foi clicado (que podem ser, para a *PPNet Tool*, "lt" ou "rb", sendo a primeira as laterais esquerda e superior e a segunda as laterais direita

e inferior). Depois a função verifica para cada elemento da lista se o redimensionamento deste o faz ultrapassar os valores superiores, inferiores esquerdo ou direito dos limites da página. Se não ultrapassarem, a função define tamanho escolhido pelo usuário como o tamanho correto. Se ultrapassarem, a função define o tamanho correto como o tamanho atual mais a distância do extremo do objeto até o ponto da borda da página mais próximo ao desejado pelo usuário. A função, então, cria uma ação de retorno para o redimensionamento dos elementos e define a página como alterada.

Para o caso de uso **saveFile**, o diagrama de atividade é o da Figura A.19.

Figura A.19 : Diagrama de atividade do caso de uso **saveFile**

Para o diagrama de atividades da Figura A.19, inicialmente a função verifica se a página foi alterada em relação ao último salvamento. Caso não tenha sido alterada, a função retorna um valor como “salvo” e finaliza. Caso tenha sido alterada, a função verifica se página já possui um arquivo associado a ela. Caso possua, a função pula direto para a atividade de

salvamento de conteúdo em arquivo. Caso não possua, a função apresenta uma caixa de diálogo de gravação de arquivo. A função então aguarda o usuário selecionar um arquivo e selecionar um dos botões da caixa. Se o botão selecionado for o “Cancelar”, a função retorna um valor como “não salvo” e se encerra. Se selecionado for o “ok”, a função verifica se o arquivo já existe e se o usuário o quer sobrescrever (essa etapa é toda realizada pela biblioteca do *Qt4*). O caso negativo para o anterior leva a função a retornar um valor “não salvo” e se encerrar. O caso positivo, a função salva o conteúdo da página no arquivo e define a página como não alterada. A função então retorna o valor “salvo” e se encerra.

Para o caso de uso **selectItems**, o diagrama de atividade é o da Figura A.20.

Figura A.20 : Diagrama de atividade do caso de uso **selectItems**

Para o diagrama de atividades da Figura A.20, inicialmente a função verifica se a tecla “shift” está pressionada. Se não estiver pressionada, a função desseleciona todos os elementos selecionados da tela de edição. Se estiver pressionada, a função não faz nada. Por fim, a função seleciona o elemento definido pelo usuário.

Para o caso de uso **undoAction**, o diagrama de atividade é o da Figura A.21.

Figura A.21 : Diagrama de atividade do caso de uso **undoAction**

Para o diagrama de atividades da Figura A.21, inicialmente a função chama a função desfazer da ação no topo da pilha de ações para desfazer, ou seja, a última a ser desfeita. A função então verifica se o ponto alcançado após a ação refeita é o mesmo ponto do último salvamento do arquivo. Caso seja, a função define a página como não alterada. Caso não seja, a função define a página como alterada.

A interface gráfica da *PPNet Tool* então tem a forma apresentada na Figura A.22.

Figura A.22 : Interface gráfica da *PPNet Tool*

Na Figura A.22, é possível identificar na parte superior um conjunto de botões responsáveis por acionar alguns dos casos de uso, que são as ferramentas padrões de interfaces gráficas. A Figura A.23 mostra esse conjunto de ferramentas em detalhe.

Figura A.23 : Ferramentas padrões de interfaces gráficas

Na barra da Figura A.23, o primeiro botão da esquerda para direita é um botão que aciona o caso de uso **createFile**. O segundo botão é um botão que aciona o caso de uso **openFile**. O terceiro botão é um botão que aciona o caso de uso **saveFile**. O quarto botão é um botão que aciona o caso de uso **exportFile**. O quinto botão é um botão que aciona o caso de uso **copyItems**. O sexto botão é um botão que aciona o caso de uso **pasteItems**. O sétimo botão é o botão que aciona o caso de uso **undoAction**. O oitavo e último botão é um botão que aciona o caso de uso **redoAction**.

Outra barra de ferramentas da Figura A.22 é a barra de ferramentas da RdPP, que pode

ser vista em detalhes na Figura A.24.

Figura A.24 : Barra de ferramentas da RdPP

Na barra da Figura A.24, o primeiro botão da esquerda para a direita é um botão que aciona o caso de uso **createPlace**. O segundo botão é um botão que aciona o caso de uso **createTransition**. O terceiro botão é um botão que aciona o caso de uso **createArc**. O quarto e último botão é um botão que aciona o caso de uso **createInstructionBox**.

Uma ferramenta que não pode ser visualizada na Figura A.22 é a tabela de propriedades, mas pode ser vista em detalhes na Figura A.25.

Figura A.25 : Tabela de propriedades

A imagem mostra uma janela de propriedades com o título "Propriedades". Abaixo do título, há quatro campos de entrada: "Item:" com o valor "Place p1", "id:" com o valor "p1", "Name:" (vazio) e "Name font:" com o valor "Arial, 12pt, 50%".

Propriedades	
Item:	Place p1
id:	p1
Name:	
Name font:	Arial, 12pt, 50%

Na Figura A.25, é mostrada uma tabela com os itens das propriedades a serem alteradas. O campo em que está escrito "Propriedades" é utilizado para acionar o caso de uso

editProperties.

Por fim, a interface gráfica possui a área da tela de edição que é apresentada em detalhes na Figura A.26.

Figura A.26 : Área da tela de edição

Na parte superior da Figura A.26, estão as abas dos arquivos abertos. Cada aba se refere a um arquivo. Ao clicar no botão na forma de X da aba, é acionado o caso de uso **closeFile**. Ao clicar no restante da aba, é acionado o caso de uso **changeFile**. Para acionar o caso de uso **editText**, é preciso dar um duplo clique com o botão esquerdo do mouse sobre um elemento de texto. Para acionar o caso de uso **deleteItems**, é preciso pressionar o botão “Delete” do teclado.

Os demais casos de uso apresentados tem dependências que podem estar em conflito umas com as outras para serem acionados. Esse conflito existe porque a área de comandos em que pode ser acionados os casos de uso é a mesma (a tela de edição que fica abaixo das abas na Figura A.26), o que difere é o que está presente na parte superior dessa área que podem ser

encontrados na tela de edição (por exemplo os elementos estão presentes na tela de edição, tal como os botões seletores e outras coisas). Os eventos que dependem da tela de edição foram divididos como na Figura A.27, para garantir que um evento acione apenas um caso de uso e que esse seja o correto.

Figura A.27 : Divisão dos eventos na tela de edição

Pela Figura A.27, para acionar o caso de uso **resizetems**, o botão esquerdo do mouse deve ser pressionado sobre Elemento->Botão seletor->Não ponto. Para acionar o caso de uso **moveltens**, o botão esquerdo do mouse deve ser pressionado sobre o Elemento->Não seletor->Selecionado. Por fim, para acionar o caso de uso **selectItems**, o botão esquerdo do mouse deve ser pressionado sobre Elemento->Não seletor->Não selecionado.

Para aumentar a capacidade de produzir, graficamente, modelos, um conjunto de 21 estados em uma máquina de estados foi proposto com alguns dos casos de usos implementados dentro deles. Outros casos de uso não alteram estados na ferramenta.

Para facilitar a visualização, esses estados foram divididos em 3 diagramas de estados, com um mesmo elemento comum aos 3, o estado “Normal”. A Figura A.28 mostra os estados referentes a adicionar elementos a tela de edição.

Figura A.28 : Máquina de estados para inclusão de elementos a tela de edição

Pela Figura A.28 os casos de uso que estão implementados são os **createPlace** (nos estados S4 e S5), **createTransition** (nos estados S6 e S7), **createArc** (nos estados S8 e S9), **createInstructionBox** (nos estados S10 e S11). As condições para mudança de estado, também conhecidas como *flags* são:

- btnPlace: o botão para criação de lugares, da Figura A.24;
- btnTransition: o botão para criação de transições, da Figura A.24;
- btnArc: o botão para criação de arcos, da Figura A.24;
- btnBox: o botão para criação de caixas de instruções, da Figura A.24;
- shiftKey: o botão “shift” do teclado;
- mouseClick: o botão esquerdo do mouse;
- inScreen: a Tela, da Figura A.27;
- inPlace: um Elemento, da Figura A.27, verificado que seu tipo é lugar;
- inTransition: um Elemento, da Figura A.27, verificado que seu tipo é transição.

Ainda sobre a Figura A.28, a presença do sinal “!” antecedendo uma *flag*, indica que o valor esperado é o valor falso. O símbolo “&” indica que deve ser realizada uma operação booleana

de E entre as duas *flags* que o antecedem e o sucedem. Também o símbolo “|” indica que deve ser realizada uma operação booleana de OU entre as duas *flags* que o antecedem e o sucedem.

A Figura A.29 mostra os estados referentes a movimento de elementos na tela de edição.

Figura A.29 : Máquina de estados para movimento de elementos na tela de edição

Pela Figura A.29 os casos de uso que estão implementados são os **moveItems** (nos estados S12, S13 e S14). Os estados S15 e S16 se referem a um tipo de ação ainda não citado, mas comum em modelagens de RdPs que é deslocar um ponto de um arco. As condições para mudança de estado são:

- inSelected: um Elemento->Não seletor->Selecionado, da Figura A.27;
- inPoint: um Elemento->Botão seletor->Ponto, da Figura A.27;
- inSelector: um Elemento->Botão seletor, da Figura A.27;
- spaceMove: é uma verificação da distância já percorrida pelo ponteiro do mouse, para filtrar movimentos de pequena grandeza;
- keyMoving: os botões direcionais (setas) do teclado;
- mouseMove: uma verificação de que o mouse está se movendo.

O caso de uso **selectItems** pode ser encontrado parcialmente no estado S13, uma vez que sendo um elemento já anteriormente selecionado, o usuário pode clicar sobre ele apenas desfazendo a seleção.

A Figura A.30 mostra os outros estados.

Figura A.30 : Máquina de estados para os demais estados

Pela Figura A.30 os casos de uso que estão implementados são os **deleteItems** (nos estados S19), **editingText** (nos estados S20), **resizingItems** (nos estados S17 e S18) e **selectItems** (nos estados S1, S2 e S3). As condições para mudança de estado são:

- inItem: um Elemento->Não seletor->Não selecionado, da Figura A.27;
- inText: um Elemento, da Figura A.27, verificado que seu tipo é texto;
- delClick: o botão “Del” do teclado;
- doubleClick: o botão esquerdo do mouse pressionado duas vezes;
- backToWindow: o retorno de uma caixa de diálogo para a tela da ferramenta.

Por fim, os diagramas de classes da ferramenta são divididos em 6 bibliotecas.

A primeira biblioteca é a biblioteca principal, representadas na Figura A.31.

Figura A.31 : Diagrama de classe da biblioteca principal

A segunda biblioteca é a biblioteca dos elementos da RdPP, representadas na Figura A.32.

Figura A.32 : Diagrama de classe da biblioteca dos elementos da RdPP

A terceira biblioteca é a biblioteca de outros elementos gráficos, representadas na Figura A.33.

Figura A.33 : Diagrama de classe da biblioteca de outros elementos gráficos

A quarta biblioteca é a biblioteca das ações de retorno, representadas na Figura A.34.

Figura A.34 : Diagrama de classe da biblioteca das ações de retorno

A quinta biblioteca é a biblioteca dos estados, representadas na Figura A.35.

Figura A.35 : Diagrama de classe da biblioteca dos estados

A última biblioteca é a biblioteca da PNML, representada na Figura A.36.

Figura A.36 : Diagrama de classe da biblioteca PNML

Anexo B – Percepção do ambiente colaborativo em um nicho de mercado

Como o ambiente colaborativo é apontado na literatura como uma realidade de alguns nichos de mercados de outros países (EUA, China e UE), foi realizada uma consulta para saber a maturidade do ambiente colaborativo em um nicho específico de mercado do Brasil.

O teste realizado para estudar a percepção das empresas sobre o ambiente envolveu empresas brasileiras.

O teste não envolveu o mercado mundial e foi realizado em apenas uma área de atuação, que é insuficiente para generalizar as conclusões obtidas, mas as informações levantadas são considerados relevantes para justificar e motivar o desenvolvimento do presente trabalho. O procedimento adotado foi o de selecionar uma área com grande quantidade de empresas atuando e de pequeno ou médio portes. A área escolhida foi a de assistência técnica em eletrônica, que atua em serviços e em grandes centros, como a cidade de São Paulo, que apresenta grande quantidade de empresas de pequeno e médio portes.

Os dados das empresas selecionadas para o teste foram consultados na internet, que é importante de ser considerado, uma vez que é o meio atual mais ágil e amplo de acesso a informação e é também fundamental para a colaboração estudada (por EVs).

Para analisar a colaboração, um questionário foi enviado para as empresas dessa área para que respondessem eletronicamente. Este questionário foi composto de uma identificação e cinco perguntas que se referiam ao tema de EVs, mas mais especificamente elaboradas para a área de atuação da empresa.

Os itens que formulam o questionário consideram os requisitos dos consumidores por produtos e serviços personalizados, com menor tempo de produção, redução de custo final e melhoria de qualidade. Também considera a necessidade de uma produção mais enxuta e a existência da heterogeneidade de processos dentro do mercado.

Uma forma de avaliar a heterogeneidade dos processos é constatar a existência de ramos de atividades específicos que podem ser classificados para um conjunto de empresas. A oposição a essa idéia é que todas as empresas do mercado podem realizar todos os processos (ou atender todo tipo de demanda).

Outra verificação importante é a disponibilidade da empresa para colaborar e cooperar com outras empresas.

O ambiente colaborativo, então, aqui se caracteriza como composto por diversos SPs (ou empresas) em que:

- Cada SP é capaz de realizar um conjunto de processos que pode diferenciá-lo de outros SPs no ambiente, sendo esse conjunto chamado de capacidades;

- Os processos realizados pelos SPs podem ser classificados em classes e a variação de processos dentro de cada classe não é tão grande que precise separá-la em classes distintas;
- Os SPs comunicam-se uns com os outros e disponibilizam as informações dos processos para que outros SPs também o possam utilizar.

O questionário enviado as empresas está representado na Figura B.37.

Figura B.37 : Formulário de perguntas utilizado na pesquisa

<p>1.1 Considerando a sua empresa e as empresas que você conhece da sua área, você considera que todos os tipos de pedidos podem ser atendidos por:</p> <p>☐ Apenas uma empresa da área</p> <p>☐ Um conjunto de empresas da área</p> <p>1.2 Você acredita que existem pedidos que não podem ser atendidos por essas empresas?</p> <p>☐ Sim</p> <p>☐ Não</p> <p>1.3 Caso tenha respondido "Sim" na pergunta anterior, qual(is) o(s) motivo(s) mais comum(s) para o não atendimento?</p> <p>☐ falta de conhecimento</p> <p>☐ falta de ferramentas</p> <p>☐ falta de matéria prima</p> <p>☐ Outro:_____</p> <p>2. Quando sua empresa atende a um pedido que você desconhecia (por ser algo muito novo), você compartilha as informações desse serviço, após feito, com outras empresas da área?</p> <p>☐ Sim</p> <p>☐ Não</p> <p>3. Um cliente pede para a sua empresa um serviço para o qual a sua empresa não possui pessoal capacitado ou ferramentas adequadas, mas você sabe que existe outra empresa da mesma área que pode atendê-lo. O quê você faz?</p> <p>☐ devolve para o cliente alegando que não pode atendê-lo</p> <p>☐ procura outra empresa conhecida que possa fazê-lo</p> <p>☐ Outro:_____</p> <p>4. No mercado em geral (todos os tipos de empresas desde produtoras de automóveis até empresas de reciclagem), você considera que as atividades que a sua empresa realiza são de domínio:</p> <p>☐ apenas da sua empresa</p> <p>☐ de um certo conjunto de empresas</p> <p>☐ de todas as empresas do mercado</p>
--

A primeira pergunta feita tinha objetivo de saber a percepção das pessoas que atuam nesse ramo de atividade quanto a capacidade de atendimento dos pedidos, tanto do ponto de vista

quantitativo quanto do ponto de vista de disponibilidade de recursos e conhecimentos. A pergunta formulada foi: *“Entre a sua empresa e as empresas que você conhece da sua mesma área (reparos em Eletrônicos), você considera que toda a diversidade de pedidos (conserto de um equipamento específico) pode ser atendida por pelo menos uma empresa ou por um conjunto de empresas? Ou existem pedidos que não podem ser atendidos por essas empresas? Qual(is) motivo(s) mais comum(s) para não atenderem (falta conhecimento, falta ferramentas, falta matéria prima, etc)?”*.

A segunda pergunta visava medir o compartilhamento de informações entre empresas especialmente no que se refere a solicitações atendidas que ainda não estavam no escopo de seus conhecimento, como o caso de PDNPs¹. A pergunta formulada foi: *“Quando sua empresa atende um pedido de cliente de conserto de um certo equipamento que você desconhecia (por ser algo muito novo), você compartilha as informações desse conserto com outras empresas da área?”*.

A terceira pergunta tinha como objetivo reconhecer se há relações de confiança e recontrações por parte das empresas do ramo para os casos de pedidos que ultrapassam as capacidades de uma empresa, que seria a busca de uma empresa parceira sem a necessidade de cancelar o pedido com o cliente. A pergunta formulada foi: *“Quando um cliente pede para a sua empresa o conserto de um equipamento que sua empresa não possui pessoal capacitado para fazê-lo ou ferramentas adequadas para o mesmo, mas que você sabe que existe outra empresa da mesma área que pode atendê-lo, o que você faz (devolve para o cliente alegando que não pode atendê-lo, ou procura outra empresa conhecida que possa fazê-lo, ou outra opção)?”*.

A quarta pergunta visava entender como as pessoas que atuam no ramo de atividade percebem a participação dela no mercado geral (não restrito unicamente a esse ramo de atividade), para identificar se essas empresas entendem a existência de competências/capacidades diferentes das delas em diversos ramos. Com as respostas, seria possível reconhecer a necessidade de classificação específica ou não das atividades realizadas em cada ramo de atividades das diferentes empresas. A pergunta formulada foi: *“No mercado geral (todos os tipos de empresas, desde produtoras de automóveis, até empresas de reciclagem), você considera que as atividades que a sua empresa é capaz de realizar são apenas da sua empresa, de um certo conjunto de empresas ou de todas as empresas do mercado?”*.

Para auxiliar na tabulação dos dados, evitando multiplicidade de respostas comuns e interpretações errôneas na análise das respostas, as respostas das perguntas foram elaboradas na forma de múltipla escolhas, deixando a parte aberta para interpretações específicas das perguntas serem informadas num campo livre para preenchimento.

¹Processo Desenvolvimento de Novos Produtos

Os contatos com empresas foram num total de 116 empresas na cidade de São Paulo, feitas por telefone e e-mail. As empresas que se dispuseram a responder o formulário foram um total de 39.

Os gráficos das respostas estão presentes na Figura B.38 e na Figura B.39.

Figura B.38 : Resultados da pesquisa de colaboração entre empresas da área de assistência técnica em eletro-eletrônicos, perguntas de 1.1 a 2

Figura B.39 : Resultados da pesquisa de colaboração entre empresas da área de assistência técnica em eletro-eletrônicos, perguntas de 3 a 5.2

Para a pergunta 1.1 da Figura B.37, 79% das empresas responderam que acreditam que um conjunto de empresas da área podem atender a todos os pedidos.

Para a pergunta 1.2 da Figura B.37, 79% das empresas responderam que acreditam que existem pedidos que não podem ser atendidos por essas empresas.

Para a pergunta 1.3 da Figura B.37, que apenas responderam as empresas que acreditam que existem pedidos que não podem ser atendidos nas empresas, nenhuma alegou que o motivo seja a falta de conhecimento, 31% alegaram o motivo ser a falta de matéria-prima, 28% a falta de ferramentas e 41% alegaram outros motivos.

Para a pergunta 2 da Figura B.37, 62% das empresas disseram que compartilham as informações sobre um serviço no caso de ter atendido a um pedido que desconhecia.

Para a pergunta 3 da Figura B.37, 72% das empresas procuram outra empresa conhecida que possa atender uma solicitação de reparo para o qual ela não possua pessoal capacitado ou ferramentas adequadas, enquanto 15% das empresas devolvem a solicitação para o cliente alegando que não podem atendê-lo.

Para a pergunta 4 da Figura B.37, 53% das empresas acreditam que um certo número de empresas tem domínio sobre as atividades que elas realizam e 35% das empresas acreditam que todas as empresas do mercado tem domínio sobre as atividades que essas realizam. Para

12% das empresas consultadas, há uma exclusividade das atividades para a própria empresa.

Pelas perguntas 1.1, 1.2 e 1.3, da percepção da maioria das empresas da área de eletrônica, que responderam ao teste, entende-se que o mercado necessita de mais empresas do ramo de atividade para que todas as capacidades sejam oferecidas a demanda, ou seja, nenhuma das empresas possui domínio de todas as capacidades da área.

Pela pergunta 2, da percepção da maioria das empresas da área de eletrônica que responderam ao teste, entende-se que a informação obtida como um novo conhecimento pode ser difundida. Apesar de não terem sido questionadas pelos motivos, pode-se entender que não vêm o compartilhamento dessa informação como prejudicial para sua estratégia de negócio.

Pela pergunta 3, da percepção da maioria das empresas da área de eletrônica que responderam ao teste, entende-se que a procura de empresas concorrentes para atender uma demanda é possível, quando essa não tem capacidade de atender.

Pela proximidade dos resultados da pergunta 4, não é possível concluir o entendimento das empresas, mesmo restritas a amostra.

Anexo C – Comparação entre a rede de Petri produtiva e a rede de Petri colorida para workflow

A proposta aqui foi a de realizar um teste com a RdPP para identificar se ela oferece acessibilidade e simplicidade para interpretação dos *workflows* foi feito utilizando pessoas que possuem pouco conhecimento de processos, e pouco ou nenhum conhecimento de RdP. Para o teste, a etapa de projeto não foi solicitada para as pessoas que responderam a um questionário, restringindo-as a interpretação do *workflow* representado em RdPP ou em RdPC.

Para uma comparação para modelos de *workflow* em RdPC e RdPP, é necessário inicialmente estabelecer a metodologia para modelagem de *workflow* utilizando RdPC. A metodologia de modelagem de *workflow* utilizada nesse teste foi baseada em Lu, Zhang e Sun (2009).

Nessa metodologia, as instruções (ou atividades do *workflow*) são representadas como as funções de expressões de arcos orientados ($A_c(f)$) atribuídas a arcos que estão entre o lugar p de execução da instrução e a transição t de término da instrução.

As informações de objetos da RdPP são representadas como as funções de guarda das transições ($G_C(t)$) e as informações de tipo de processo da RdPP são os conjuntos de cores associados aos lugares ($CF_C(p)$).

Para cada conjunto de instruções com características similares às instruções contidas em uma caixa de instruções, são definidas os nós associados como duas transições e um lugar. Uma das transições fornece marcas para o lugar e outra transição retira as marcas da forma como representada no Grafo C.1.

Figura C.1 : Representação de uma instrução e os elementos estruturais do modelo RdPC dessa

No Grafo C.1, t_{iS} representa uma transição que fornece a possibilidade de iniciar a execução das instruções e t_{iE} representa uma transição que finaliza a execução das instruções. Neste caso, $G_C(t_{iS}) = INFO.OBJETO$ e $CF(p_i) = INFO.TIPOPROC.$ e $A_C((p_i, t_{iE})) = 1''INFO.INSTRUÇÃO1 ++ \dots 1''INFO.INSTRUÇÃOm.$

Para a representação dinâmica associada a execução da RdPC como um *workflow*, deve-se considerar os estados da execução das instruções, como no Grafo C.2.

Figura C.2 : Representação dos estados da execução de uma instrução no modelo de RdPC

Fonte: (CHOI; PARK; LEE, 2002)

No Grafo C.2, estão representados 3 estados bem definidos da execução de uma instrução. O primeiro estado antecede a execução da instrução e é chamado de “recebida”. O segundo estado é a própria execução da instrução e é chamada de “executando”. O terceiro estado é posterior a execução da instrução e é chamado de “completada”. A mudança entre os estados “recebida” e “executando” é feita pelo disparo da transição t_S , similar a transição t_{jS} do Grafo C.1. A mudança entre os estados “executando” e “completada” é feita pelo disparo da transição t_E , similar a transição t_{jE} do Grafo C.1.

Para o método usado, os lugares que representam os 3 estados associados a execução de uma instrução não podem coincidir com os lugares que representam os 3 estados associados a execução de outra instrução, como pode se ver no Grafo C.2. Entretanto, essa representação acrescenta mais elementos a estrutura dos que os necessários para entendimento do modelo. Uma simplificação adotada aqui é que o lugar que representa o estado “completada” associado a uma instrução pode ser o mesmo lugar que representa o estado “recebida” da instrução que esteja em sequência a primeira.

O teste foi realizado com estudantes de engenharia que possuem pouco ou nenhum conhecimento da RdP e de suas extensões. Para o teste, o *workflow* proposto é de procedimentos para uma pessoa realizar entre o momento em que acorda até sua saída de casa. O motivo de utilizar esse *workflow* é de ser algo familiar mesmo para as pessoas que desconhecem a RdP e, por ser parte das atividades do dia-a-dia desses estudantes.

O intuito do teste foi verificar o quanto esses estudantes conseguem compreender os modelos (em RdPC e RdPP), tanto do ponto de vista estrutural, quanto do ponto de vista do *workflow*.

O *workflow* proposto inicia com uma única atividade ou instrução, que é “Desliga o despertador”. A partir dessa atividade, o processo se divide em sub-processos que envolvem “Preparar e consumir refeição”, “Tomar banho e trocar de roupa” e “Preparar o material de trabalho”. Para modelar esses *workflow* em macro atividades, foi utilizada o PFS, como no Grafo C.3.

Figura C.3 : *Workflow* em macro atividades do teste com estudantes

O refinamento do sub-processo “Preparar o material de trabalho” é feito em duas atividades que são “Pega mochila” e “Verifica mochila”. O refinamento desse sub-processo pode ser visto no Grafo C.4.

Figura C.4 : Refinamento do sub-processo “Preparar o material de trabalho” em PFS

As atividades do Grafo C.4 atingem o grau máximo de refinamento para esse teste.

No caso do sub-processo “Tomar banho e trocar de roupa”, o refinamento se divide em dois sub-processos sequenciais, que são “Tomar banho” e “Colocar roupa”. O refinamento desse sub-processo pode ser visto no Grafo C.5.

Figura C.5 : Refinamento do sub-processo “Tomar banho e trocar de roupa” em PFS

O sub-processo “Colocar roupa” do Grafo C.5, pode ser refinado em 3 atividades: “Coloca roupa de baixo”, “Coloca calça” e “Coloca camiseta”. Por dependência, a primeira atividade antecede as demais para que fisicamente esse sub-processo possa ser realizado. Já as duas outras atividades são independentes e a ordem de execução entre elas não afeta o resultado final do sub-processo. Entretanto, essas duas atividades compartilham o mesmo recurso com ferramenta. Assim, a característica encontrada entre elas é a de sequência arbitrária. O Grafo C.6 mostra o refinamento do sub-processo “Colocar roupa”.

Figura C.6 : Refinamento do sub-processo “Colocar roupa” em PFS

As atividades do Grafo C.6 também atingem o grau máximo de refinamento para esse teste.

O sub-processo “Tomar banho” do Grafo C.5, pode ser refinado em 4 sub-processos: “Entrar no banho”, “Lava o cabelo”, “Lava o corpo” e “Se seca”. Esses sub-processos são sequenciais. O primeiro sub-processo se divide em dois tipos conflitantes: “Entrar no banho sem roupa” e “Entrar no banho com roupa”. Os sub-processos “Lava o cabelo” e “Lava o corpo” possuem a característica de sequência arbitrária entre eles. O Grafo C.7 mostra o refinamento do sub-processo “Tomar banho” com o refinamento do sub-processo “Entrar no banho”.

Figura C.7 : Refinamento do sub-processo “Tomar banho” com refinamento do sub-processo “Entrar no banho” em PFS

O sub-processo “Entrar no banho com roupa” é composto por uma sequência de atividades: “Tira camiseta”, “Tira calça”, “Tira roupa de baixo”, “Pega toalha”. As duas primeiras atividades estão em uma sequência arbitrária. Já o sub-processo “Entrar no banho sem roupa” só é composto pela atividade “Pega toalha”. Apesar de possuírem o mesmo nome, as atividades

“Pega toalha” que pertencem aos sub-processos “Entrar no banho com roupa” e “Entrar no banho sem roupa” se diferem por serem chamadas em caminhos diferentes. O Grafo C.8 mostra o refinamento dos sub-processos “Entrar no banho com roupa” e “Entrar no banho sem roupa”.

Figura C.8 : Refinamento dos sub-processos “Entrar no banho sem roupa” e “Entrar no banho com roupa” em PFS

Dos sub-processos já apresentados, o único que ainda pode ser refinado é o “Preparar e consumir refeição”, do Grafo C.3. O sub-processo “Preparar e consumir refeição” é um conflito de duas possibilidades e composto de 5 atividades: “Preparar café”, “Preparar suco”, “Preparar pão”, “Consumir café e pão” e “Consumir suco e pão”. O primeiro e o segundo sub-processos estão em conflito, assim como o quarto e quinto sub-processos. O terceiro sub-processo está em paralelismo com os dois primeiros. O Grafo C.9 mostra o refinamento do sub-processo “Preparar e consumir refeição”.

Do conflito entre os sub-processos “Preparar café” e “Preparar suco” do Grafo C.9, para ambos deve ser realizado em paralelo o sub-processo “Preparar pão”. Entretanto, o projeto considera que o mesmo recurso é usado para os três sub-processos. Assim, um distribuidor foi utilizado para garantir que apenas um dos sub-processos seja realizado de cada vez.

O sub-processo “Consumir café e pão”, do Grafo C.9 pode ser refinado em duas atividades: “Come pão” e “Toma café”. Essas duas atividades estão em sequência aleatória. E o sub-processo “Consumir suco e pão” também pode ser refinado em duas atividades: “Come pão” e “Toma suco”. Nesse caso, também as duas atividades estão em sequência aleatória. O Grafo C.10 mostra o refinamento dos dois sub-processos “Consumir café e pão” e “Consumir suco e pão”.

Figura C.9 : Refinamento do sub-processo “Preparar e consumir refeição” em PFS

Figura C.10 : Refinamento dos sub-processos “Consumir café e pão” e “Consumir suco e pão” em PFS

As atividades “Come pão”, “Toma café” e “Toma suco” do Grafo C.10, não foram mais refinadas.

O sub-processo “Preparar café” do Grafo C.9 é composto pelas atividades: “Esquenta água”, “Põe pó no coador” e “Passa água no coador”. Essas três atividades devem ser realizadas em sequência com ordenação já definida. O sub-processo “Preparar suco” é composto pelas atividades: “Pega 4 laranjas”, “Pega espremedor” e “Espreme laranjas”. As duas primeiras atividades estão em sequência aleatória. Já a terceira atividade está em sequência as duas primeiras. Por fim, o sub-processo “Preparar pão” é composto pelas atividades: “Pega manteiga”, “Pega pão” e “Passa manteiga no pão”. As duas primeiras atividades também estão em sequência aleatória, enquanto a terceira atividade está em sequência as anteriores. O Grafo C.11 mostra o refinamento dos sub-processos “Preparar café”, “Preparar suco” e “Preparar pão”.

Todas as atividades do modelo atingiram o grau máximo de refinamento para o teste.

Posteriormente, é preciso definir as pré-condições para a execução das atividades (que são

Figura C.11 : Refinamento dos sub-processos “Preparar café”, “Preparar suco” e “Preparar pão” em PFS

as informações de objetos). As atividades que possuem pré-condições estão apresentadas no Grafo C.12.

As pré-condições no Grafo C.12 foram representadas como portas das atividades e transições.

O *workflow*, em PFS, é a junção desses refinamentos e pré-condições. Devido ao consumo grande consumo de espaço, a representação do *workflow* completa em PFS, não foi feita aqui.

Após obter o *workflow* em PFS, pode-se gerar os modelos em RdPP ou em RdPC.

Para obter o *workflow* em RdPP, adotou-se o seguinte procedimento:

- Para a sequência arbitrária: as portas (caso haja) das transições que paralelizam as atividades se tornam as informações de objetos e as atividades se tornam as instruções. Um exemplo dessa conversão pode ser visto no Grafo C.13;
- Para as demais características: as portas (caso haja) da atividade se tornam as informações de objeto e a atividade se torna a instrução. Um exemplo dessa conversão pode ser visto no Grafo C.14.

Figura C.12 : Pré-condições das atividades em PFS

Figura C.13 : Conversão da característica sequência arbitrária do modelo PFS para o modelo RdPP

Figura C.14 : Conversão das demais características do modelo PFS para o modelo RdPP

Da mesma forma, para obter o *workflow* em RdPC, adotou-se o seguinte procedimento:

- Para a sequência arbitrária: as portas (caso haja) das transições que paralelizam as atividades se tornam a função de guarda da transição que antecede o lugar referente a execução da atividade e as atividades se tornam a expressão do arco direcionado que liga o lugar referente a execução e a transição que termina a execução da atividade. Um exemplo dessa conversão pode ser visto no Grafo C.15;

Figura C.15 : Conversão da característica sequência arbitrária do modelo PFS para o modelo RdPC

- Para as demais características: as portas (caso haja) da atividade se tornam a função de guarda da transição que antecede o lugar referente a execução da atividade e a atividade se torna a expressão do arco direcionado que liga o lugar referente a execução e a transição que termina a execução da atividade. Um exemplo dessa conversão pode ser visto no Grafo C.16.

Figura C.16 : Conversão das demais características do modelo PFS para o modelo RdPC

Considerando para ambos a informação de classe como “Arrumar-se”, o Grafo C.17 representa o *workflow* desse teste em RdPP e o Grafo representa o *workflow* desse teste em RdPC.

Figura C.17 : *Workflow* do teste de comparação em RdPP

Figura C.18 : *Workflow* do teste de comparação em RdPC

Dos Grafos C.17 e C.18 foi gerado um questionário para comparar as estruturas das representações. As perguntas deste questionário foram:

1. No formulário de RdPP a pergunta foi “De acordo com a figura acima, o que a pessoa deve fazer na transição T23?” (com dica: “Transição é todo elemento retangular com preenchimento de cor preta”); no formulário de RdPC a pergunta foi “De acordo com a figura acima, o que a pessoa deve fazer na transição T23 (entre "s" e "e")?” (com dica: “Transição é todo elemento retangular”); e alternativas múltiplas selecionáveis:
 - (a) “Arrumar-se”;
 - (b) “Sem roupa”;
 - (c) “Coloca roupa de baixo”;
 - (d) “Verifica mochila”;
 - (e) “Não sei”.
2. No formulário de RdPP a pergunta foi “De acordo com a figura acima, o que a pessoa deve fazer na transição T15?” (com a mesma dica); no formulário de RdPC a pergunta foi “De acordo com a figura acima, o que a pessoa deve fazer na transição T15 (entre "s" e "e")?” (com a mesma dica); e alternativas múltiplas selecionáveis:
 - (a) “Arrumar-se”;
 - (b) “Com suco”;
 - (c) “Com pão”;
 - (d) “Come pão”;
 - (e) “Toma suco”;
 - (f) “Não sei”.
3. No formulário de RdPP a pergunta foi “Qual é a condição que deve ser verificada antes que a pessoa possa fazer o que é pedido na transição T18?” (com a mesma dica); no formulário de RdPC a pergunta foi “Qual é a condição que deve ser verificada antes que a pessoa possa fazer o que é pedido na transição T18 (entre "s" e "e")?” (com a mesma dica); e alternativas múltiplas selecionáveis:
 - (a) “Arrumar-se”;
 - (b) “Sem camiseta”;
 - (c) “Sem calça”;
 - (d) “Tira roupa de baixo”;
 - (e) “Não sei”.

4. No formulário de RdPP a pergunta foi “Qual é a relação existente entre o que é pedido na transição T11 e T21?” (com a mesma dica); no formulário de RdPC a pergunta foi “Qual é a relação existente entre o que é pedido na transição T11 e T21 (entre "s" e "e")?” (com a mesma dica); e alternativas exclusivas selecionáveis:
- (a) “Em sequência/um após o outro”;
 - (b) “Em paralelo/independentes”;
 - (c) “Em conflito/ou um ou outro”;
 - (d) “Não sei”.
5. No formulário de RdPP a pergunta foi “Qual é a relação existente entre o que é pedido na transição T4 e T5?” (com a mesma dica); no formulário de RdPC a pergunta foi “Qual é a relação existente entre o que é pedido na transição T4 e T5 (entre "s" e "e")?” (com a mesma dica); e alternativas exclusivas selecionáveis
- (a) “Em sequência/um após o outro”;
 - (b) “Em paralelo/independentes”;
 - (c) “Em conflito/ou um ou outro”;
 - (d) “Não sei”.
6. No formulário de RdPP a pergunta foi “Qual é a relação existente entre o que é pedido na transição T17 e T20?” (com a mesma dica); no formulário de RdPC a pergunta foi “Qual é a relação existente entre o que é pedido na transição T17 e T20 (entre "s" e "e")?” (com a mesma dica); e alternativas exclusivas selecionáveis
- (a) “Em sequência/um após o outro”;
 - (b) “Em paralelo/independentes”;
 - (c) “Em conflito/ou um ou outro”;
 - (d) “Não sei”.
7. No formulário de RdPP a pergunta foi “Qual é a relação existente entre o que é pedido na transição T9 e T12?” (com a mesma dica); no formulário de RdPC a pergunta foi “Qual é a relação existente entre o que é pedido na transição T9 e T12 (entre "s" e "e")?” (com a mesma dica); e alternativas exclusivas selecionáveis
- (a) “Em sequência/um após o outro”;

- (b) “Em paralelo/independentes”;
- (c) “Em conflito/ou um ou outro”;
- (d) “Não sei”.

8. No formulário de RdPP a pergunta foi “Qual(is) da(s) sequência(s) abaixo está(ão) correta(s), de acordo com a figura?” (com a mesma dica); no formulário de RdPC a pergunta foi “Qual(is) da(s) sequência(s) abaixo está(ão) correta(s), de acordo com a figura?” (com a mesma dica); e alternativas múltiplas selecionáveis

- (a) “Desliga o despertador”, “Esquenta água”, “Pega pão”, “Pega Manteiga”, “Passa manteiga no pão”, “Põe pó no coador”, ...”;
- (b) “Pega 4 laranjas”, “Tira camiseta”, “Pega espremedor”, “Tira calça”, ...”;
- (c) “Lava o cabelo”, “Lava o corpo”, “Pega toalha”, “Coloca roupa de baixo”, ...”;
- (d) “Coloca calça”, “Coloca camiseta”, “Pega mochila”, “Verifica mochila”, ...”;
- (e) “Não sei”.

As respostas corretas para o questionário de ambas são: 1c; 2d; 2e; 3b; 3c; 4b; 5a; 6c; 7b; 8b; 8d.

O teste foi realizado com estudantes de graduação em engenharia que não tivessem conhecimento prévio da RdP e suas extensões. Desses estudantes participantes, 58 responderam o formulário referente a RdPP e 24 responderam o formulário referente a RdPC. Foi solicitado as pessoas que não respondessem mais do que um formulário.

Sobre a primeira questão:

- das pessoas que responderam ao formulário da RdPP, 7% colocaram a alternativa correta, mas colocaram também outras alternativas como corretas e 3,5% não colocaram a alternativa correta. O índice de acerto foi 89,5%;
- das pessoas que responderam ao formulário da RdPC, 12,5% colocaram a alternativa correta, mas colocaram também outras alternativas como corretas e 16,7% não colocaram a alternativa correta. O índice de acerto foi 70,8%.

Sobre a segunda questão:

- das pessoas que responderam ao formulário da RdPP, 5% colocaram as alternativas corretas, mas colocaram também outras alternativas como corretas, 1,7% colocaram apenas uma das alternativas corretas e 5% não colocaram nenhuma alternativa correta. O índice de acerto foi 88,3%;

- das pessoas que responderam ao formulário da RdPC, 12,5% colocaram as alternativas corretas, mas colocaram também outras alternativas como corretas, 4,2% colocaram apenas uma das alternativas corretas e 12,5% não colocaram nenhuma alternativa correta. O índice de acerto foi 70,8%;

Pelas duas primeiras questões, o índice de entendimento do que é uma instrução ficou mais evidente na RdPP do que na RdPC.

Sobre a terceira questão:

- das pessoas que responderam ao formulário da RdPP, 1,7% colocaram as alternativas corretas, mas colocaram também outras alternativas como corretas, 3,5% colocaram apenas uma das alternativas corretas e nenhuma deixou de marcar alguma alternativa correta. O índice de acerto foi 94,8%;
- das pessoas que responderam ao formulário da RdPC, 4,2% colocaram apenas uma das alternativas corretas e 4,2% não colocaram nenhuma alternativa correta. O índice de acerto foi 91,6%;

Pela terceira questão, não há uma grande diferença no entendimento das pré-condições entre os modelos da RdPP e da RdPC.

Sobre a quarta questão:

- das pessoas que responderam ao formulário da RdPP, 34,5% não colocaram a alternativa correta. O índice de acerto foi 65,5%;
- das pessoas que responderam ao formulário da RdPC, 16,7% não colocaram a alternativa correta. O índice de acerto foi 83,3%.

Sobre a quinta questão:

- das pessoas que responderam ao formulário da RdPP, 12% não colocaram a alternativa correta. O índice de acerto foi 88%;
- das pessoas que responderam ao formulário da RdPC, 8,3% não colocaram a alternativa correta. O índice de acerto foi 91,7%.

Sobre a sexta questão:

- das pessoas que responderam ao formulário da RdPP, 13,8% não colocaram a alternativa correta. O índice de acerto foi 86,2%;

- das pessoas que responderam ao formulário da RdPC, 25% não colocaram a alternativa correta. O índice de acerto foi 75%.

Sobre a sétima questão:

- das pessoas que responderam ao formulário da RdPP, 17,2% não colocaram a alternativa correta. O índice de acerto foi 82,8%;
- das pessoas que responderam ao formulário da RdPC, 41,7% não colocaram a alternativa correta. O índice de acerto foi 58,3%.

Pelas questões quatro a sete, com a diversidade de resultados, torna-se difícil constatar se os erros envolviam o desconhecimento da RdP e extensões ou desconhecimento de como são as características de um processo.

Sobre a oitava questão:

- das pessoas que responderam ao formulário da RdPP, nenhuma pessoa acertou por completo. 3,5% colocaram as alternativas corretas, mas colocaram também outras alternativas como corretas, 44,8% colocaram apenas uma das alternativas corretas e 52,7% não colocaram nenhuma alternativa correta;
- das pessoas que responderam ao formulário da RdPC, 4,2% acertaram por completo. 4,2% colocaram as alternativas corretas, mas colocaram também outras alternativas como corretas, 33,3% colocaram apenas uma das alternativas corretas e 58,3% não colocaram nenhuma alternativa correta.

Sobre os resultados da oitava questão, a alternativa “b” praticamente não foi selecionada pelas pessoas que responderam em ambos formulários. Isso porque, matematicamente, ambos modelos não explicitam que as atividades de sequência aleatória não necessariamente devem ser executadas juntas numa execução global, ou seja, outra execução paralela ainda poderia ocorrer entre uma e outra, uma vez que são independentes. Além disso, a alternativa “a” foi a mais presente nas respostas e pode ser explicada pelo fato que não conhecendo as regras de disparos de transição, a pessoa não sabe que ao iniciar o processo que engloba as atividades “Esquentar água” e demais, este deve ser concluído para liberar o recurso que habilitará o processo que engloba as atividades “Pegar manteiga” e demais.

O teste com a RdPP foi feito para compará-la com outra ferramenta já aplicada de *workflow*. A RdPC que é muito encontrada na literatura de *workflow* e por esse motivo foi escolhida para efeito de comparação. Pelos resultados do teste, a RdPP apresentou melhores respostas por pessoas de pouco conhecimento em RdP. Esse teste também não tem um espaço amostral suficiente para aceitar uma hipótese de que a RdPP é mais simples de ser interpretada que a

RdPC, mas os resultados do teste sugerem também essa propriedade. De fato, considerando todo o formalismo matemático que é necessário para analisar o comportamento dinâmico de um processo modelado em RdPC e que em geral só é divulgado em nível de disciplinas de pós-graduação ou cursos avançado de especialização. A RdPP indica ser uma técnica muito mais simples de ser assimilada e utilizada por operadores de máquinas, desenhistas e gerentes de processos.