

188970.2

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**Uma técnica para a
modelagem de Estruturas de
Decisão**

por

Sergio Ery Cazella

Dissertação submetida à avaliação, como
requisito parcial para a obtenção do grau de
Mestre em Ciência da Computação

Prof. Dr. José Palazzo Moreira de Oliveira

Orientador



SABi



Porto Alegre, janeiro de 1997.

UFRGS
INSTITUTO DE INFORMÁTICA
BIBLIOTECA

CIP - Catalogação na Publicação

Cazella, Sergio Ery

Uma técnica para a modelagem de estruturas de decisão / por Sergio Ery Cazella. - Porto Alegre: CPGCC da UFRGS, 1997.

84 f.:il.

Dissertação (mestrado) - Universidade Federal do Rio Grande do Sul. Curso de Pós-Graduação em Ciência da Computação, Porto Alegre, BR-RS, 1997. Orientador: Oliveira, José Palazzo Moreira de.

1. Modelagem de Empresas. 2. Estruturas de Decisão. 3. Redes de Petri. 4. Modelo TF-ORM. I. Oliveira, José Palazzo Moreira de. II. Título.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitora: Profa. Wrana Panizzi

Pró-Reitor de Pós-Graduação: Prof. José Carlos Ferraz Hennemann

Diretor do Instituto de Informática: Prof. Roberto Tom Price

Coordenador do CPGCC: Prof. Flávio Rech Wagner

Bibliotecária-Chefe do Instituto de Informática: Zita Prates de Oliveira

Agradecimentos

Em primeiro lugar, gostaria de agradecer a todos os professores das disciplinas que frequentei no Curso de Pós-graduação em Ciência da Computação, os quais com seus ensinamentos, ajudaram-me no aprimoramento de meus conhecimentos. Em especial, agradeço ao Prof. Dr. José Palazzo M. de Oliveira pela valiosa orientação que me conduziu para a realização deste trabalho.

Em segundo, agradeço, especialmente, à minha esposa Márcia que me acompanha nesta caminhada, sempre compreendendo e me apoiando em todos os momentos.

Administração - EBU
 Informática: Organização
 Administração: Empresas
 Sistemas: Informação gerencial
 Tomada: Decisões
 Redes: Petri
 Modelagem: Empresas
 CNPq 1.03.04.00-2

UFRGS INSTITUTO DE INFORMÁTICA BIBLIOTECA		
N.º CHAMADA 658:681.32(043) C386T	N.º REG: 32937	
		13.05.97
ORIGEM: ①	LATA: 16 04 97	PREÇO: R\$ 30,00
FUNDO: II	FORN.: IF	

Sumário

Lista de Figuras	5
Lista de Tabelas	6
Resumo	7
Abstract	8
1 Introdução	9
2 Conceitos	11
2.1 Estruturas de Decisão	11
2.2 As variáveis temporais	14
2.3 Agentes e Recursos	16
3 Ferramentas para modelagem	18
3.1 Redes de Petri.....	18
3.2 O Modelo TF-ORM	22
4 Proposta de uma Técnica de Análise e Modelagem de Estruturas de Decisão	29
4.1 Definição dos Passos utilizados na Técnica	30
4.2 Modelagem dos Agentes e Recursos.....	43
5 O Estudo de Caso	47
5.1 O Centro de Decisão "Comprar Equipamento"	51
6 Aplicação da Técnica ao Estudo de Caso.....	54
6.1 Especificação do nível 1 com Redes de Petri - Centros de Decisão.....	54
6.2 Especificação do nível 2 com Redes de Petri - Tarefas.....	56
6.3 Especificação do nível 3 com Redes de Petri - Ações	58
6.4 Especificação do nível 4 com Redes de Petri - Operações	62
6.5 Especificação com o modelo TF-ORM.....	64
7 Conclusões	66
Anexo 1 Modelo TF-ORM do Centro de Decisão "Comprar Equipamentos"	68
Bibliografia	81

Lista de Figuras

FIGURA 2.1 - A Função de Decisão e o Processo	11
FIGURA 2.2 - O modelo macro GRAI	13
FIGURA 2.3 - O modelo das atividades nos Centros de Decisão	14
FIGURA 2.4 - As decisões e a disponibilidade do recurso.....	16
FIGURA 2.5 - A Janela de Planejamento de um Centro de Decisão.	16
FIGURA 3.1 - Exemplo da aplicação das Redes de Petri	20
FIGURA 4.1 - Hierarquia de procedimentos.....	29
FIGURA 4.2 - Representação do Centro de Decisão.....	31
FIGURA 4.3 - Representação dos Agentes e Recursos envolvidos	32
FIGURA 4.4 - Representação da atualização de um recurso	32
FIGURA 4.5 - Exemplo nível 1	34
FIGURA 4.6 - Representação de uma tarefa	35
FIGURA 4.7 - Exemplo nível 2	36
FIGURA 4.8 - Exemplo nível 3	39
FIGURA 4.9 - Rede de Petri da ação R1	43
FIGURA 6.1 - Modelo Nível 1 - Estrutura de Decisão	56
FIGURA 6.2 - Modelo Nível 2 - Tarefas do centro de decisão "Comprar equipamentos".....	57
FIGURA 6.3 - Modelo Nível 3 - Ações da tarefa "Especificação"	60
FIGURA 6.4 - Modelo Nível 3 - Ações da tarefa "Licitação"	62

Lista de Tabelas

TABELA 4.1 - Exemplos de Centros de Decisão	33
TABELA 4.2 - Exemplos de Tarefas.....	36
TABELA 4.3 - Exemplos de Ações.....	37
TABELA 4.4 - Exemplos de Operações.	40
TABELA 5.1 - Estudo de Caso.....	50
TABELA 6.1 - Nível 1: Centros da Estrutura de Decisão.	54
TABELA 6.2 - Nível 2: Tarefas de "Comprar equipamentos".	57
TABELA 6.3 - Relação das ações por tarefa.....	58
TABELA 6.4 - Nível 3: Ações da Tarefa "Especificação".	59
TABELA 6.5 - Nível 3: Ações da Tarefa "Licitação".	60
TABELA 6.6 - Nível 4: Operações associadas à ações da tarefa "Especificação"	63

Resumo

Considerando as complexidades envolvidas no processo de tomada de decisão nas organizações de hoje, é fácil ver que este processo representa um ponto importante para a obtenção de vantagens na competição pelo mercado. Portanto, observa-se que a função de gerenciamento é fundamental para a empresa, seja qual for o seu setor de atividade, tamanho, complexidade e tecnologia dos produtos ou serviços por ela oferecidos. Por este motivo, é importante que seja dada uma atenção particular ao aperfeiçoamento desta função e, para que isto seja possível, é necessário que, primeiro, seja feita uma análise detalhada do Sistema de Gerenciamento existente na empresa através da utilização de uma ferramenta que permita criar um modelo que represente as informações necessárias. De acordo com esta necessidade, este trabalho tem como objetivo propor uma técnica que auxilia na criação do modelo da Estrutura de Decisão associada ao Sistema de Gerenciamento da empresa. A técnica descrita está baseada no método de refinamentos sucessivos, e visa permitir que, a partir do modelo macro da estrutura de decisão, sejam criados modelos específicos, com um nível de formalização adequado para a implementação de sistemas de informação automatizados, que possam apoiar o processo da tomada de decisão. A preocupação principal foi propor uma técnica que forneça a facilidade de uma ferramenta gráfica, a qual permita um rápido aprendizado e proporcione ao analista um canal de comunicação com o usuário, assim como também permita a criação de um modelo com um alto grau de detalhamento. Para que isto seja possível a proposta alia o uso das Redes de Petri e do modelo TF-ORM. Ao final é apresentado um estudo de caso focando a estrutura de decisão em um ambiente de manufatura, o qual é utilizado para mostrar a aplicação prática da técnica.

Palavras-chaves: Modelagem de Empresas, Estruturas de Decisão, Redes de Petri, TF-ORM.

Title: A technique for the Decision Structure Modelling

Abstract

Considering the complexities involved in the process of decision-taking in organizations nowadays, it is easy to realize that this process represents an important point in the acquisition of advantages concerning the competition for the market. Therefore, it is observed that the function of the management is fundamental to the enterprise, whatever its section of activity, size, complexity, technology of its products or services are. That is why it is important to give particular attention to the improvement of that function, and for that to be possible, first it is necessary to do a detailed analysis of the Management System that exists in the enterprise through the utilization of a tool which permits to create a model to represent the needed information. According to this necessity, this paper proposes a technique which helps in the creation of a model of the Decision Structure associated with the Enterprise Management System. This technique has been based on the method of successive refinement and aims at allowing to be created, from the macro model of the decision structure, specific models and with a formalization level adequate to the implement of automatic information's systems that can support the process of decision-taking. The goal was to propose a technique that provides the facility of a graphic tool and permits a fast learning, providing the analyst with a way to communicate with the user, as well as it should permit the creation of a model with a high level of details. For this to be possible, the proposal associates the use of Petri Nets and TF-ORM model together. At the end a case study focusing on the decision structure in a manufacturing environment, with is used to show the practical application of the technique is presented.

Key words: Enterprise Modeling, Decision Structures, Petri Nets, TF-ORM model.

1 Introdução

A tomada da decisão certa no momento certo pode significar a diferença entre o sucesso e o fracasso de uma empresa em muitas situações. Quando são consideradas as complexidades envolvidas no processo da tomada de decisão nas organizações de hoje, é fácil ver que este processo representa um ponto crítico para a obtenção de vantagens na competição pelo mercado [BAS 94].

Por este motivo, o Gerenciamento é uma função fundamental para as empresas, seja qual for o seu setor de atividade, tamanho, complexidade e tecnologia dos produtos ou serviços oferecidos. Portanto é importante que seja dada uma atenção particular ao aperfeiçoamento desta função. Na maioria dos casos, este aperfeiçoamento necessita que, primeiro, seja feita uma análise detalhada do Sistema de Gerenciamento existente na empresa para que seja possível entender o seu funcionamento e se verifiquem as possíveis falhas existentes, sendo que, para isto, torna-se necessário que seja estabelecida uma metodologia adequada.

A qualidade de um Sistema de Gerenciamento está estreitamente relacionada com a qualidade de seu processo decisório e uma das principais características de um bom processo decisório é a integração entre os seus vários níveis. Por este motivo é importante obter o modelo da Estrutura de Decisão da empresa e melhorar a integração de tal maneira que o desempenho combinado de duas ou mais funções deste processo aumente [OLI 94]. É importante que através deste modelo seja possível identificar os agentes responsáveis pelas tomadas de decisão e os recursos envolvidos, assim como as variáveis de tempo associadas ao processo de decisão, tais como o tempo de validade da decisão e o tempo entre a tomada da decisão e o início dos seus efeitos.

Em especial, pode-se citar o caso dos Sistemas de Manufatura, abordado em [CAR 92], os quais têm sido objeto de várias pesquisas no sentido de melhorar a integração entre os diversos sistemas que dele fazem parte, centrado na preocupação da coordenação das atividades e a conformidade destes sistemas existentes na organização com a realização dos seus objetivos. Uma estrutura conceitual que mostra uma visão abrangente desta integração é o Modelo Conceitual GRAI [DOU 84]. Este modelo divide o sistema de manufatura em quatro subsistemas, são eles: físico, de operação, de decisão e de informação. O modelo mostra três pontos de vista a partir dos quais o grau de integração do sistema de manufatura pode ser examinado. Em primeiro lugar, o subsistema de operação pode ser investigado para determinar o seu nível de integração com o subsistema físico e com o suporte oferecido pelo sistema de informação. Segundo, o subsistema de decisão, ou podemos chamar, de gerenciamento, deve ser investigado para avaliar se o nível de cada decisão está de acordo com o responsável por ela e se as altas decisões tomadas no nível estratégico são refletidas para os níveis mais baixos. E por

último, deve ser examinada a integração do subsistema de informação com os outros subsistemas.

O objetivo deste trabalho é propor uma técnica que auxilie na criação do modelo da Estrutura de Decisão associada ao Sistema de Gerenciamento da empresa. Este modelo deverá auxiliar na identificação dos Centros de Decisão que compõem esta estrutura, assim como os Agentes e Recursos envolvidos na execução de cada centro. O modelo também deverá permitir a identificação das relações de dependência entre os diversos centros de decisão. Além disto, ajudará a estimar os tempos de duração de cada centro de decisão, o que é importante para que se observe com quanto tempo de antecedência é necessário que a decisão seja tomada, de forma que os seus resultados estejam implementados na data desejada.

Com a obtenção do modelo, este servirá como base para a implementação de um Sistema de Informação associado à Estrutura de Decisão da empresa, o qual poderá ser utilizado como suporte para um Sistema de Apoio à Decisão (SAD) ou um "Executive Information System" (EIS).

Visando o objetivo descrito acima, o resultado obtido com o trabalho foi a descrição de uma técnica que permite, através de refinamentos sucessivos do modelo macro da estrutura de decisão, criar modelos detalhados dos centros de decisão com um nível de formalização necessário para que seja possível a implementação, a partir destes, de um sistema de informação que sirva para apoiar as tomadas de decisão.

O trabalho está estruturado da seguinte forma: a seção 2 reúne todos os conceitos de Estrutura de Decisão e Centro de Decisão utilizados, mostra o modelo conceitual GRAI e as variáveis de tempo associadas a um processo decisório, os conceitos de Agente e Recurso e como estes se relacionam com os Centros de Decisão. Na seção 3 são abordadas noções básicas das duas ferramentas de modelagem, Redes de Petri e TF-ORM, utilizadas na técnica proposta na seção 4. Na seção 5 é apresentado o estudo de caso utilizado para exemplificar a aplicação da técnica. Na seção 6 é mostrada a aplicação da técnica no estudo de caso descrito.

2 Conceitos

Neste capítulo serão comentados os conceitos utilizados no trabalho, os quais estão baseados na Ciência da Decisão e no Subsistema de Decisão do modelo macro do método GRAI.

2.1 Estruturas de Decisão

O conceito básico da chamada Ciência da Decisão é a **Função de Decisão**, definida em [WOR 84]. Uma Função de Decisão expressa um conjunto de decisões similares caracterizadas pelos seguintes elementos:

- visam controlar um conjunto de **variáveis alvo** para as quais existem um conjunto de **regras**;
- podem manipular somente as chamadas **variáveis controláveis**;
- as suas **variáveis alvo** são influenciadas pelas **variáveis controláveis**;
- existe um conjunto de **variáveis de ambiente** que influenciam as **variáveis alvo**, mas que podem simplesmente ser observadas pela função de decisão, isto é, não podem ser manipuladas;
- ao relacionamento entre suas **variáveis de ambiente**, **variáveis controláveis** e **variáveis alvo**, é dado o nome de **processo** a ser controlado. A natureza de um processo é definida pelo conjunto das **variáveis alvo** e **variáveis controláveis**; e
- utilizam um **modelo** do processo para tomar decisões. As partes destes modelos que possuem semelhanças podem ser incluídas em Sistemas de Suporte à Decisão (SSD).

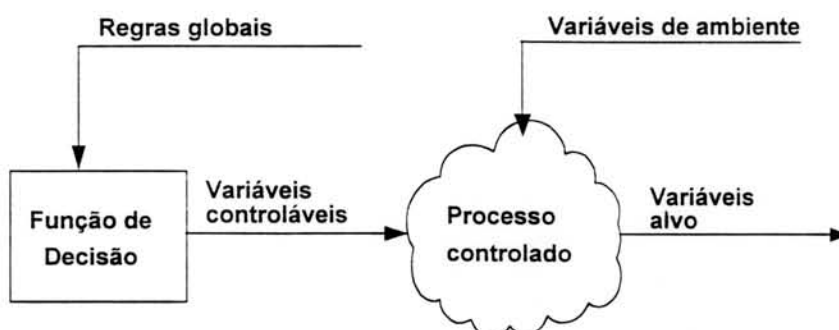


FIGURA 2.1 - A Função de Decisão e o Processo

Ao conjunto de funções de decisão distintas, interconectadas e, que dizem respeito a um determinado aspecto do sistema (como qualidade dos produtos, manutenção dos equipamentos, inventário e outros) é dado o nome de **Estrutura de Decisão** [TIM 91].

O modelo GRAI, mostrado em [BUC 84], [DOU 84], [DOU 87], [DOU 87a] e [DOU 92], foi criado dentro do contexto do método GRAI, o qual foi desenvolvido pelo laboratório GRAI da Universidade de Bordeaux I da França especialmente para o estudo de sistemas de decisão para produção. O método GRAI é um dos poucos métodos encontrados na bibliografia que se propõe a oferecer ferramentas para dar o suporte necessário à modelagem de sistemas de decisão no caso específico de um ambiente de manufatura.

O modelo conceitual GRAI é a referência através da qual vários elementos do mundo real podem ser identificados e fornece as regras básicas para o projeto de um Sistema de Manufatura. Conforme é mostrado em [DOU 84], o modelo é genérico o bastante para se adaptar a qualquer tipo de Sistema de Manufatura e preciso o suficiente para permitir a perfeita identificação dos componentes de um sistema. O modelo é abordado neste trabalho por ser adequado para servir como referencial para o entendimento do conceito de estrutura de decisão associada a um sistema de gerenciamento genérico.

O Modelo GRAI é composto por duas partes: o modelo macro, que descreve a **estrutura do Sistema de Gerenciamento da Produção**, e o modelo micro, o qual **detalha as atividades nos Centros de Decisão** (ou também podemos chamar de **Função de Decisão**). O modelo macro é usado para expressar a percepção e as idéias de um sistema de manufatura, o qual é decomposto em vários subsistemas. Particularmente, o subsistema de decisão é dividido em vários níveis e centros de decisão. O modelo micro é usado para representar os elementos internos de um centro de decisão.

Segundo o modelo macro, o Sistema de Manufatura é dividido em três subsistemas:

- **O subsistema Físico:** é composto por máquinas, fluxos de materiais, trabalhadores, técnicas; seu papel é transformar matéria-prima em produto final.
- **O subsistema de Decisão:** seu propósito é controlar o subsistema físico com a finalidade de alcançar os objetivos econômicos e sociais. Está dividido em vários níveis de tomada de decisão de acordo com vários critérios e, cada nível composto de um ou mais **Centros de Decisão**. Abaixo deste, foi colocado um **subsistema de operação** que liga a parte física e de decisão (este nível inclui o controle das máquinas e todos os procedimentos controlados: segurança, qualidade, manutenção)

- **O subsistema de Informação:** o qual provê suporte aos outros subsistemas.

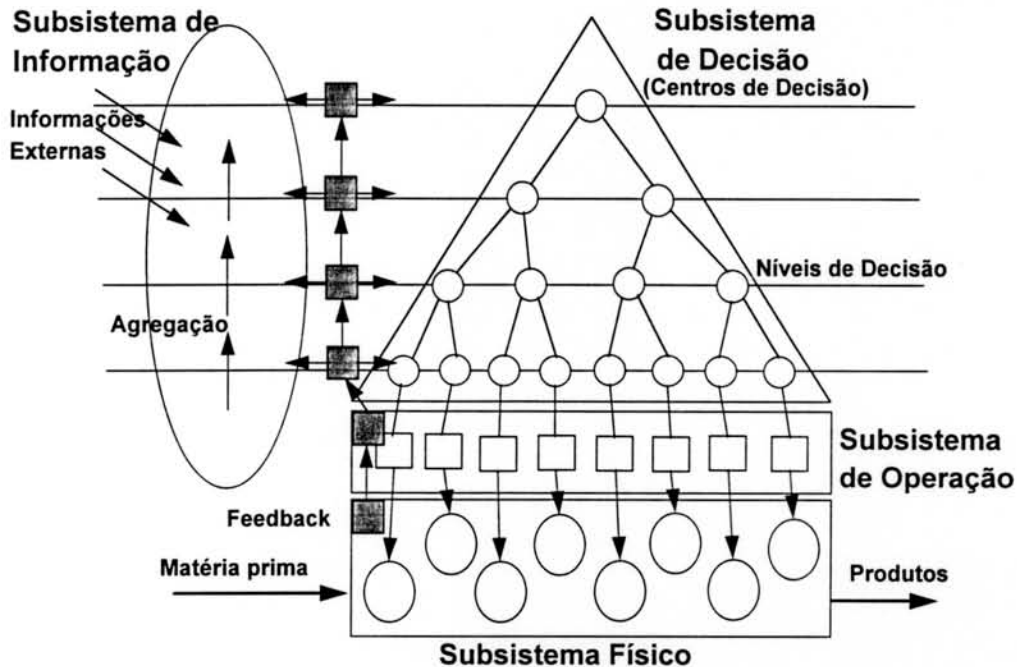


FIGURA 2.2 - O modelo macro GRAI

A decomposição em níveis do subsistema de Decisão depende, por um lado, das funções do centro de decisão e, do outro lado, das variáveis de tempo **horizonte (H)** da decisão e **período (P)** de reavaliação da decisão.

As informações e decisões possuem um detalhamento diferente em cada nível. Como consequência, a informação deve ser estruturada também de forma hierárquica para que esta esteja de acordo com a decisão em cada nível. Quanto menor fica o horizonte da decisão, isto é, quanto mais próximo do subsistema físico, mais precisa e detalhada deve ser a informação.

O modelo micro tem por objetivo conceituar as operações em cada Centro de Decisão. Para isto, devem ser definidas:

- As várias atividades do Centro de Decisão;
- seu escopo de decisão (variáveis e limitações);
- as decisões tomadas; e
- as informações usadas pelo Centro de Decisão.



FIGURA 2.3 - O modelo das atividades nos Centros de Decisão

O modelo da figura 2.3, mostra que a informação vinda de um nível hierárquico inferior é agregada de forma que esteja adaptada às necessidades do nível de decisão superior. Com base nestas informações, o tomador de decisão daquele Centro define, conforme o seu Escopo de Decisão, a distribuição de recursos, o desempenho desejado e a abrangência da responsabilidade para o nível imediatamente inferior.

Apesar do modelo GRAI ter sido criado dentro do contexto de um método que tem como objetivo auxiliar na modelagem de sistemas de gerenciamento da produção, podemos estender os seus conceitos, em especial, os propostos na estrutura do subsistema de decisão, para utilização na criação de modelos em outras áreas de negócio, sendo esta uma das propostas deste trabalho.

2.2 As variáveis temporais

Como vimos na seção anterior, as variáveis de tempo são utilizadas pelo modelo GRAI para a decomposição do subsistema de decisão em níveis hierárquicos. Os conceitos utilizados neste trabalho estão baseados nos conceitos abordados em [OLI 94], onde três propriedades temporais básicas e uma derivada podem ser associadas a uma Função de Decisão ou, como também podemos chamar, Centro de Decisão, são elas :

B: Intervalo de tempo entre a tomada da decisão e o início dos seus efeitos;

P: Período de tempo para a reavaliação da decisão tomada;

E: Intervalo de tempo entre a tomada de decisão e o fim dos seus efeitos; e

H: Horizonte da decisão ($H = E - B$).

As propriedades **B** e **E** definem a chamada **Janela de Planejamento** do Centro de Decisão, isto é, o intervalo de tempo no qual as decisões daquela classe têm efeito sobre, por exemplo, a produção no caso de Sistemas de Manufatura.

O **período de reavaliação (P)** está relacionado com a frequência com que ocorrem novos fatos que alteram o contexto em que a decisão foi tomada, por exemplo, no caso da situação da produção, fatos tais como novas ordens de compra e peças em equipamentos. Uma vez ocorrido um evento deste tipo este exigirá que a decisão tomada inicialmente seja reavaliada para que possa se adaptar ao novo contexto.

O **início da Janela de Planejamento (B)** é definido em função de uma série de fatores tais como, por exemplo: tempo para a liberação dos recursos, prazo de entrega de equipamentos ou insumos, tempo para instalação de equipamentos, tempo para testar os insumos (controle de qualidade), treinamento para utilização de novos equipamentos, fase de testes de equipamentos, etc.. Observa-se que **B** aumenta a medida em que maior quantidade de recursos são envolvidos e os fatores relevantes a sua definição são dependentes de um maior número de outros centros de decisão, ou seja, o **Fator de Dependência** aumenta.

Deve-se considerar também que cada recurso tem um ou mais **Graus de Disponibilidade** em relação ao processo para o qual ele é necessário, e cada um destes graus pode requerer uma decisão. Um equipamento, por exemplo, precisa ser (1) comprado, (2) deslocado de um departamento para outro, e (3) preparado com as ferramentas adequadas para a execução do processo. As decisões em cada um destes graus sofrem diferentes atrasos e podem envolver conflitos com outros processos que necessitem do mesmo recurso. A cada grau de decisão podemos associar uma variável **Profundidade da Decisão**, a qual será um importante indicador para avaliar a demora necessária para disponibilizar o recurso.



FIGURA 2.4 - As decisões e a disponibilidade do recurso.

A Janela de Planejamento, ou o **horizonte de planejamento (H)**, se estende de acordo com a provável duração dos efeitos físicos e econômicos do centro de decisão. É importante observar que o horizonte de planejamento deve-se estender pelo menos por tempo suficiente para que seja possível se justificar os investimentos feitos antes de B.

As propriedades período de reavaliação (P) e horizonte de planejamento (H) são de grande importância para a definição da **Estrutura de Decisão**, pois são freqüentemente usadas como parâmetros de decomposição da estrutura em uma hierarquia de **Centros de Decisões**.

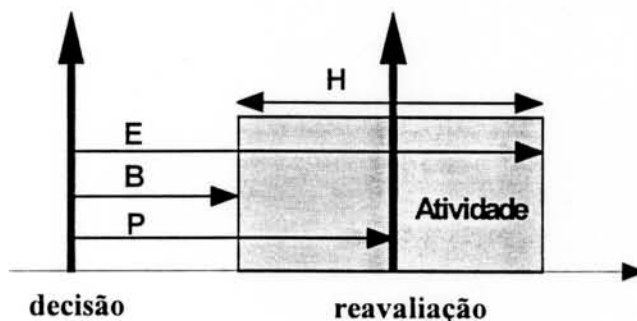


FIGURA 2.5 - A Janela de Planejamento de um Centro de Decisão.

2.3 Agentes e Recursos

Para que seja possível definir um Centro de Decisão, é importante que sejam definidos os Agentes e os Recursos envolvidos no processo; onde Agentes são as entidades responsáveis pela execução de alguma tarefa dentro do Centro de Decisão e Recursos são as entidades que devem ser disponibilizadas para serem utilizadas pelos primeiros. Dependendo da situação, Agentes e Recursos podem estar envolvidos com um Centro de Decisão de diferentes maneiras.

Os **Agentes**, os quais representam entidades ativas, ou seja, que executam algum procedimento dentro da execução do centro de decisão, podem estar envolvidos com os mesmos das seguintes formas:

- como responsáveis pelo controle da execução do centro de decisão;
- enviando algum recurso, tal como uma informação ou material necessário para a execução de algum processo interno;
- recebendo algum recurso gerado pela execução; ou
- como responsáveis por alguns processos internos.

No caso dos **Recursos**, os quais representam informações, documentos ou objetos físicos, podem estar envolvidos das seguintes formas:

- sendo utilizados pelo centro de decisão e, assim, tornam-se temporariamente indisponíveis enquanto as atividades deste são executadas (recursos alocados) ou tornam-se permanente-mente indisponíveis (recursos consumidos) em função da execução destas atividades;
- sendo gerados durante a execução do centro de decisão;
- sendo consultados quando se tratam de recursos do tipo informação; ou
- sendo suas características ou dados atualizados por alguma atividade do centro.

Os conceitos de Agentes e Recursos usados neste trabalho estão baseados nos conceitos apresentados no modelo TF-ORM [EDE 94] [EDE 94a], os quais serão abordados na próxima seção.

3 Ferramentas para modelagem

Neste capítulo serão descritas as ferramentas de modelagem utilizadas na proposta deste trabalho.

3.1 Redes de Petri

As Redes de Petri constituem uma excelente ferramenta gráfica de modelagem, as quais permitem criar modelos que representem as propriedades dinâmicas de um sistema de informação. Sua principal característica é a capacidade de modelar a concorrência [WAL 93] e a sincronização entre as diversas atividades identificadas em um sistema. Além disto, a Rede de Petri pode ser usada como uma técnica de modelagem hierárquica, isto é, por refinamento, tornando possível representar sistemas complexos através de modelos simples em vários níveis de abstração [BHA 93] [SIL 89] [SIL 90].

Basicamente, uma Rede de Petri permite representar os possíveis **estados** alcançados por um sistema e as **transições** realizadas, as quais causam as alterações de estado. Portanto, analisando o modelo do sistema representado através de uma Rede de Petri, é possível saber quais são os estados alcançáveis por ele e a dependência entre as transições, ou seja, as seqüências de transições e avaliar os possíveis conflitos.

Uma Rede de Petri é um grafo onde dois conjuntos de vértices são definidos: o conjunto dos **lugares** (ou **estados**), descrito como conjunto P, e o conjunto das **transições**, dito conjunto T. Os lugares ligam-se a transições, e estas, a outros lugares, através dos **arcos**. Na notação usual tem-se **círculos**, representando os lugares, e **retângulos**, representando as transições. Passando de um lugar para o outro através do disparo das transições da rede, existem as **marcas**, as quais indicam, conforme a posição, o estado atual do sistema. Cada vez que uma transição é disparada, conforme as regras de habilitação de disparo de transições, as marcas que se encontram nos lugares de entrada desta transição (pré-estados) são retiradas e novas marcas são adicionadas aos lugares de saída (pós-estados). Uma determinada quantidade de marcas pode ser retirada e/ou adicionada aos lugares conforme o **peso** dos arcos que ligam estes lugares à transição. Chamamos de **marcação** ao conjunto de marcas distribuídas pela rede em um determinado momento e, a qual, indica o estado atual do sistema modelado.

A teoria sobre a modelagem utilizando Redes de Petri pode ser encontrada com mais detalhes nos primeiros capítulos de [HEU 91] ou em [PET 81].

Uma grande variedade de Redes de Petri são encontradas na literatura, todas baseadas nos conceitos de lugar e transição em diferentes níveis de abstração. As principais variedades de Redes de Petri adicionam

diferentes componentes ao modelo básico, isto com o objetivo de melhorar a semântica do modelo de acordo com as diversas aplicações. Em [ANT 93] encontramos a "árvore genealógica" das Redes de Petri. Neste trabalho foram utilizados conceitos que estão espalhados pelas diversas variedades de redes de forma que a ferramenta se mostrasse adequada às necessidades de representação encontradas.

O primeiro tipo de Rede de Petri de interesse neste trabalho é o chamado Canal/Atividade como apresentado em [HEU 91], o qual permite criar um modelo informal do sistema, ou seja, um modelo sem regras exatas de funcionamento e que, portanto, não se presta para análises matemáticas ou simulações. Um modelo deste tipo é bastante útil nas fases iniciais da modelagem, quando a precisão dos modelos formais não é desejável e para facilitar a comunicação com leigos em modelagem, pois estes levariam muito tempo para aprender todos os conceitos envolvidos em um modelo formal. Em um modelo Canal/Atividade, os elementos da rede representam as **unidades funcionais**. Os **lugares** representam as **unidades funcionais passivas**, isto é, unidades funcionais do tipo estado, como, por exemplo, estados de informações, de pessoas, de máquinas, etc., e a estas unidades funcionais passivas dá-se o nome de **canais**. As **transições** representam as **unidades funcionais ativas**, ou seja, unidades funcionais do tipo transição entre estados, como, por exemplo, mudanças nos estados de pessoas, equipamentos, etc., e às unidades funcionais ativas dá-se o nome de **atividades**.

A interpretação dada aos arcos nos modelo Canal/Atividade permite criar modelos com diversos níveis de riquezas de informações. De forma genérica, um arco representa apenas uma relação de dependência entre o canal e a atividade envolvidos. O tipo de arco (entrada/saída, alterador/restaurador) não tem importância nesta interpretação, servindo, apenas, como mais um registro informal sobre o sistema modelado.

Para exemplificar a aplicação das Redes de Petri Canal/Atividade é mostrada uma pequena parte do estudo de caso descrito no capítulo 5. Neste exemplo, as unidades funcionais ativas, isto é, as atividades representam os Centros de Decisão "Definir os Recursos Necessários" e "Comprar Equipamento". No primeiro Centro de Decisão são geradas as relações dos recursos que devem ser adquiridos para alcançar a nova capacidade de produção. Para isto, o Centro de Decisão depende, conforme representam os arcos, da existência de verba disponível para a aquisição ou contratação e do indicador de variação da capacidade de produção da fábrica (VCap). Existe uma regra para execução do Centro de Decisão, " $VCap > 0$ ", o que significa que este só será executado se a variação for positiva, isto é, se for necessário aumentar a capacidade de produção. Após a sua execução são geradas duas relações: a dos "Equipamentos Necessários" e dos "Funcionários Necessários". Além disto, o Centro de Decisão atualiza o VCap, descontando a variação representada pelos equipamentos e funcionários descritos nas relações.

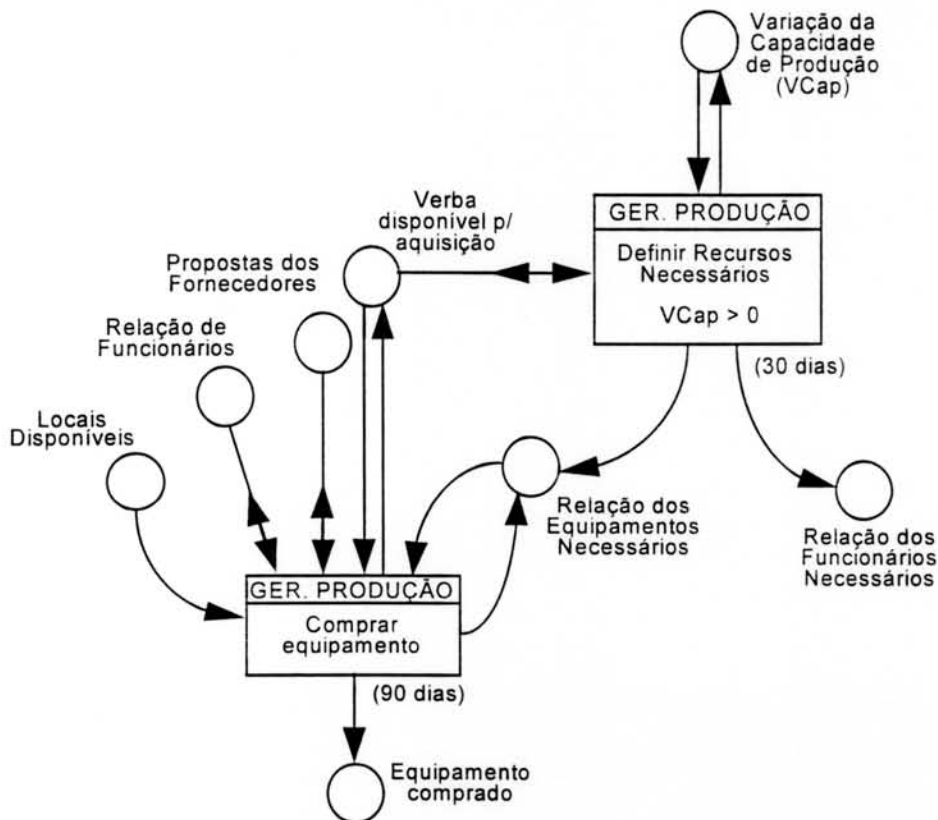


FIGURA 3.1 - Exemplo da aplicação das Redes de Petri

Uma vez gerada a Relação dos Equipamentos Necessários, o Centro de Decisão "Comprar Equipamento" pode ser executado, este Centro de Decisão envolve todo o processo de aquisição de equipamentos. Para isto é necessário existir uma relação dos funcionários disponíveis para compor a comissão de compra, as propostas de fornecedores, locais disponíveis para a instalação dos equipamentos e verba para a compra. Após a execução do Centro de Decisão, o equipamento fica disponível dentro da fábrica, a verba disponível é atualizada, assim como a relação dos equipamentos necessários. Pode-se observar neste exemplo a relação de dependência entre os dois Centros de Decisão, pois a compra de um equipamento só pode ser realizada uma vez que a relação dos equipamentos necessário estiver pronta.

Neste exemplo, os arcos de entrada representam o uso de informações existentes no canal e os de saída, a geração de informações no canal.

O outro tipo de Rede de Petri de interesse para este trabalho é chamado de Redes Compactas apresentadas em [HEU 91]. Este tipo de rede permite criar um modelo formal da realidade modelada com regras bem definidas. Uma linguagem formal chamada Linguagem de Anotação (LA) é utilizada para definir as conexões e os lugares da rede. A LA designa as entidades, as quais representam os objetos existentes na realidade modelada, através de expressões chamadas de termos. O conjunto de todas as entidades

que são consideradas no modelo forma o Universo de Discurso (UD) do modelo.

O funcionamento da rede se dá através do aparecimento e desaparecimento de entidades nos lugares da rede. Quais entidades podem aparecer em um determinado lugar será definido pelo domínio daquele lugar. Cada conexão, ou transição, ao invés de definir somente uma alteração, define um conjunto de alterações. A ocorrência de uma alteração depende e/ou afeta a presença/ausência de entidades nos lugares. Quais as alterações são definidas por uma conexão são indicadas pelos termos dos ramos, pela fórmula da conexão e pelos domínios dos lugares conectados. Os termos dos ramos designam um conjunto de entidades que é parte do conjunto de entidades definido pelo domínio do lugar conectado pelo ramo. A fórmula da conexão define a regra que deve ser satisfeita pelos elementos das entidades para executar as alterações.

As Redes Compactas são semelhantes às Redes Predicado-Transição a partir das quais foram derivadas as chamadas Redes de Petri a Objetos apresentadas em [BAK 90] na implementação de um Sistema de Regras de Produção Controlado. Neste tipo de rede as marcas representam instâncias de tipos de objetos e à cada transição são associadas condições e uma ação. Quando uma transição é disparada a ação é executada. Na literatura também é possível encontrar a definição das chamadas Redes de Petri Coloridas [ANT 93] [VER 94], as quais incorporam a possibilidade de associar estruturas de dados às marcas e são ditas coloridas porque as diferentes estruturas são representadas graficamente por marcas com cores diferentes.

Como pode ser visto nas seções anteriores deste trabalho, o tempo é um parâmetro importante na modelagem de estruturas de decisão. Por este motivo, será abordada também neste capítulo uma variedade de Redes de Petri chamada "Timed Petri Nets" (abreviada por TPN), a qual associa a variável tempo às transições. Esta variedade das Redes de Petri é amplamente utilizada na criação de modelos em diversas áreas como pode ser visto em [CAJ 88], [HAN 89] e [RAV 86].

Em uma TPN, a variável tempo não é modelada no nível da rede assim como acontece com as transições e os estados. A cada transição T é associado um tempo de duração desta, representado por $d(T)$ medida em uma unidade de tempo pré-definida pelo modelador. Desta forma, o disparo de uma transição no tempo t leva a duas ações sobre a marcação da rede: no tempo t , as marcas são removidas dos lugares de entrada (pré-estados) conforme os pesos dos arcos; no tempo $t+d(T)$, marcas são adicionadas aos lugares de saída (pós-estados) conforme os pesos dos arcos.

As diferentes variedades de Redes de Petri são empregadas na modelagem de diversos tipos de sistemas em diferentes áreas, mas pode-se destacar a aplicação desta ferramenta na área de Sistemas de Manufatura. Entre estas aplicações é possível citar: a resolução de problemas da tomada

de decisão relacionados à utilização de recursos com a finalidade de executar a produção conforme o planejamento [BIL 93], a melhoria do processo de tomada de decisão na geração da programação da produção [PLO 84], o projeto e implementação de sistemas de controle da produção [VAL 87] [SIL 89] [SIL 90] e na avaliação de desempenho [ANT 93]. As redes de Petri também são usadas como técnica básica para representar arquiteturas organizacionais como pode ser visto em [REM 88].

3.2 O Modelo TF-ORM

A outra ferramenta de modelagem utilizada é o modelo TF-ORM [EDE 94] [EDE 94a]. O modelo TF-ORM foi utilizado porque permite a modelagem dos aspectos temporais da aplicação e a implementação de um sistema de informação diretamente a partir do mapeamento da especificação obtida para um Sistema Gerenciador de Banco de Dados Orientados a Objetos (SGBDOO) ou também para um Sistema Gerenciador de Banco de Dados Relacional (SGBDR), conforme os estudos mostrados em [OLI 95] e [CAV 95].

O modelo de dados TF-ORM é um modelo orientado a objetos. O seu desenvolvimento iniciou-se com o modelo ORM (Object with Roles Model), no qual foi introduzido o conceito de papéis para representar a evolução do comportamento de um objeto. Um objeto representa uma instância de somente uma classe, mas ele pode desempenhar diferentes papéis durante o seu ciclo de vida. O modelo ORM foi posteriormente estendido dando origem ao modelo F-ORM (Functionality in Object with Roles Model) visando incorporar conceitos apropriados à especificação de funcionalidades de sistemas de informação para escritórios. Uma terceira extensão deu origem ao TF-ORM (Temporal Functionality in Objects with Roles Model), incorporando ao modelo anterior a capacidade de representação dos aspectos temporais.

O modelo de dados TF-ORM utiliza o conceito de papéis para representar a dinâmica da evolução do comportamento dos objetos de uma classe, facilitando o processo de análise da aplicação e de representação da evolução dos objetos através do tempo e fornecendo um mecanismo básico de abstração para a especificação de sistemas de informação. Além disso, o modelo incorpora a capacidade de representação de aspectos temporais do sistema.

A introdução do conceito de papéis tem por objetivo separar a representação dos aspectos estáticos de um objeto dos seus aspectos dinâmicos. Um objeto continua a ser uma instância de uma única classe, mas poderá assumir ao longo de sua existência um ou mais papéis, isolada ou simultaneamente, em diferentes tempos; deste modo é possível representar a evolução dinâmica do objeto no tempo.

No modelo TF-ORM as classes são decompostas em três tipos distintos, pré-definidos: classes **agente**, **recurso** e **processo**.

As pessoas são representadas através de classes do tipo **agente**, incluindo a possibilidade de representar uma parcela de trabalho não estruturado executado por eventuais tomadas de decisão. Os papéis representam comportamentos diferentes de pessoas, como por exemplo, funcionário, cliente e proprietário. Cada um destes papéis apresenta um conjunto de propriedades, de estados, de decisões que este agente pode tomar e de mensagens que pode enviar e receber para comunicação com outras classes. São também definidas regras de integridade para cada um dos papéis.

Uma classe do tipo **recurso** define a estrutura de um recurso (dado, documento ou objeto físico) em termos dos papéis que este recurso pode apresentar durante o seu ciclo de vida. Considerando, por exemplo, um livro, este pode representar o papel de elaboração, de distribuição ou de aquisição. Cada papel de um recurso apresenta um conjunto de propriedades, de estados, de mensagens e de regras (de transição e de integridade). Sua representação é análoga a das classes de agentes, com exceção das decisões que são exclusivas daquelas classes.

As classes do tipo **processo** permitem a integração entre as classe de agentes e recursos, permitindo a descrição do trabalho realizado. Representam a manipulação de informações através de procedimentos executados. Uma classe do tipo processo também é descrita através de papéis, os quais representam as diferentes tarefas executadas no procedimento. Em uma aplicação cada procedimento executado é descrito através de uma classe processo. Os procedimentos são divididos em tarefas que são representadas pelos papéis da classe processo. Uma tarefa é constituída de operações, as quais são descritas através das mensagens que a classe objeto naquele papel pode enviar e receber; os estados pré e pós-operações são representados através dos estados dos papéis.

O processo de modelagem de uma aplicação através do modelo TF-ORM é decomposto em duas etapas executadas paralelamente: (i) a definição das classes do tipo agente e recurso, independente das atividades nas quais estarão envolvidas; e (ii) a integração destes agentes e recursos na especificação das classes do tipo processo. Deste modo os agentes, os recursos e os processos são representados separadamente, sendo entretanto todos os seus inter-relacionamentos completamente definidos.

Assim, no modelo TF-ORM uma classe é definida por um nome único e por um conjunto de papéis. Cada papel consiste em um conjunto de propriedades que descrevem os atributos do objeto, um conjunto de estados que o objeto pode apresentar neste papel, um conjunto de mensagens que podem ser enviadas ou recebidas pelo objeto neste papel e um conjunto de regras de transição de estado e de integridade. Além destes itens, no caso das classes do tipo agente, a classe apresenta ainda um conjunto de decisões.

As propriedades são definidas em dois tipos: estáticas, as quais mantêm o mesmo valor durante a existência do objeto, e dinâmicas, que podem variar com o passar do tempo.

As mensagens podem estar associadas a uma lista de argumentos e, ao serem enviadas ou recebidas, provocam uma mudança de estado do objeto.

As regras de transição de estado descrevem o comportamento de um objeto ao desempenhar determinado papel. Mostram as seqüências de estados que o objeto pode apresentar naquele papel, definindo as mensagens as quais podem ser recebidas ou enviadas em cada estado. Uma regra de transição de estado mostra uma transição válida entre dois estados de um papel dependendo eventualmente do recebimento de uma mensagem e de uma condição a ser satisfeita. A forma geral de uma regra de transição de estados é a seguinte:

Rn: state(S1), {msg(MR1), msg(MR2), ..., msg(MRn)}
 => msg(ME1), msg(ME2), ..., msg(MEn), state(S2);
 (<Condição de Transição>)

A regra acima indica que quando o objeto desempenhando um determinado papel está no estado S1 (dito estado inicial), recebe as mensagens MR1, MR2, ..., MRn e satisfaz a Condição de Transição, ocorre a transição para o estado S2 (dito estado final) o que causa o envio das mensagens ME2, ME3, ..., MEn. Uma regra de transição não precisa apresentar todos os elementos acima. As seguintes formas alternativas de representação destas regras são as seguintes:

- (i) **Rn:** {msg(MR1), msg(MR2), ..., msg(MRn)} => msg(ME1), msg(ME2), ..., msg(MEn), state(S2) [; (<Condição de Transição>)], onde a regra é válida para todos os estados;
- (ii) **Rn:** state(S1), {msg(MR1), msg(MR2), ..., msg(MRn)} => msg(ME1), msg(ME2), ..., msg(MEn) [; (<Condição de Transição>)], onde a transição é efetuada através da troca de mensagens e o estado permanece o mesmo;
- (iii) **Rn:** state(S1) => msg(ME1), msg(ME2), ..., msg(MEn), state(S2) [; (<Condição de Transição>)], onde a transição é efetuada independentemente da chegada de alguma mensagem; e
- (iv) **Rn:** state(S1), {msg(MR1), msg(MR2), ..., msg(MRn)} => state(S2) [; (<Condição de Transição>)], onde a transição para outro estado ocorre mas nenhuma mensagem é enviada;

No caso de uma regra associada a um papel de um objeto da classe agente, a mesma pode ter a seguinte forma:

Rn: state(S1), decision(D1) => msg(ME1), msg(ME2),
..., msg(MEn), state(S2); (<Condição de Transição>)

Isto é, no lugar do recebimento de uma ou mais mensagens, a transição ocorre do estado S1 para o S2 no momento em que a decisão D1 é tomada e a condição for válida.

Para exemplificar o uso do modelo TF-ORM será utilizada uma pequena amostra do estudo de caso que será descrito na seção 5, a qual corresponde a uma parte do modelo do Centro de Decisão "Comprar Equipamentos". Nesta amostra temos o exemplo de cada tipo de classe existente no modelo TF-ORM.

A classe agente mostrada corresponde ao "Decision Maker" e que pode assumir os papéis de "Production_manager", o Gerente de Produção responsável por autorizar o início do processo de compra e do controle durante a sua execução, e "Purchase_commission", ou seja, a Comissão de Compra formada por um grupo de funcionários responsáveis pela operacionalização de todas as etapas que compõem a compra de equipamentos.

```
Agent class (
  DECISION_MAKER,
  <Base_role,
    static properties = { ... },
    dynamic properties = { ... },
    rules = {
      r1: msg(create_object) => state(active),
      r2: state(active) => msg(add_role(Production_manager)),
      r3: state(active) => msg(add_role(Purchase_commission)) }
  >,
  <Production_manager,
    static properties = { },
    dynamic properties = { },
    messages = {
      area_necessary_report(area_code: integer, necessary_description: string)
        to EXTERNAL_WORLD,
      purchase_authorization(purchase_description: string, begin: date, max_allocation:
        real, expiry_time: date)
        from EXTERNAL_WORLD,
      purchase_stoppage(reason: string)
        from EXTERNAL_WORLD,
      begin_authorization(purchase_description: string, begin: date, max_allocation: real,
        expiry_time: date, auction_limit_time: date)
        to EQUIPMENT_PURCHASE_FASES.Specification,
      end_authorization(reason: string)
        to EQUIPMENT_PURCHASE_FASES,
      info_supplier_choice(supplier_winner_code: integer)
        from EQUIPMENT_PURCHASE_FASES.Auction,
      provide_order_authorization
        to EQUIPMENT_PURCHASE_FASES.Auction,
      info_test_result_OK
        from EQUIPMENT_PURCHASE_FASES.Test },
    decision = {
```

```

    send_necessary_report(area_code: integer, necessary_description: string) },
states = {
    available, occupied, wait_authorization, provide_authorized },
rules = {
    begin:
        msg(add_role) => state(available),
    necessary_report_send:
        state(available),
        decision(send_necessary_report(Area_code, Necessary_description)) =>
        msg(area_necessary_report(Area_code, Necessary_description)),
        state(wait_authorization),
    begin_purchase_authorization:
        state(wait_authorization),
        msg(purchase_authorization(Purchase_description, Begin, Max_allocation,
        Expiry_time)) =>
        msg(begin_authorization(Purchase_description, Begin, Max_allocation, Expiry_time,
        Auction_limit_time)), state(occupied),
    end_purchase_authorization:
        msg(purchase_stoppage(Reason)) =>
        msg(end_authorization(Reason)), state(available); (Reason ≠ Null),
    authorize_provide_order:
        state(occupied),
        msg(info_supplier_choice(Supplier_winner_code)) =>
        msg(provide_order_aythorization), state(provide_authorized),
    purchase_complete:
        state(provide_authorized), msg(info_test_result_OK) =>
        state(available) }
>,
<Purchase_commission,
...
>)

```

Para exemplificar uma classe do tipo recurso, é mostrada a modelagem do objeto "Equipment", o qual descreve o recurso a ser adquirido pela execução do processo de compra.

```

Resource class (
    EQUIPMENT,
    <Base_role,
        static properties = { ... },
        dynamic properties = { ... },
        rules = {
            r1: msg(create_object) => state(active),
            r2: state(active) => msg(add_role(Equipment)) }
    >,
    <Equipment,
        messages = {
            define_specification(description: string, features: string, functions: string)
            from EQUIPMENT_PURCHASE_FASES.Specification,
            reception(product_code: integer, supplier_code: integer, delivery_time: date)
            from EQUIPMENT_PURCHASE_FASES.Reception,
            test_result_OK(setup_time: date, lifetime: date, technical_conditions: string)
            from EQUIPMENT_PURCHASE_FASES.Test },
        states = {
            created, specificated, receipted, approved },
        rules = {
            begin:
                msg(add_role) => state(created),
            specification:
                state(created), msg(define_specification(Description, Features, Functions)) =>

```

```

    state(specificated),
reception:
    state(specificated),
    msg(reception(Product_code, Supplier_code, Delivery_time)) =>
    state(receipted),
receive_test_result_OK:
    state(receipted),
    msg(test_result_OK(Setup_time, Lifetime, Technical_conditions)) =>
    state(approved) }
> )

```

Por último, o exemplo de classe de objeto do tipo processo é o "Equipment_purchase_fases", o qual reúne as etapas da compra de um equipamento, e que são representadas no modelo através dos seguintes papéis: Especificação, Licitação, Pagamento, Recebimento e Testes.

```

Process class (
    EQUIPMENT_PURCHASE_FASES,
    <Base_role,
        static properties = { ... },
        dynamic properties = { ... },
        rules = {
            r1: msg(create_object) => state(active),
            r2: state(active) => msg(add_role(Specification))
            r3: state(active) => msg(add_role(Auction))
            r4: state(active) => msg(add_role(Payment))
            r5: state(active) => msg(add_role(Reception))
            r6: state(active) => msg(add_role(Test)) }
    >,
    <Specification,
    ...
    >,
    <Auction,
    ...
    >,
    <Payment,
    ...
    >,
    <Reception,
        messages = {
            delivery(equipment_description: string, product_code: integer, supplier_code:
            integer, delivery_time: date)
                from SUPPLIER.Supplier,
            reception(product_code: integer, supplier_code: integer, delivery_time: date)
                to EQUIPMENT.Equipment,
            prepare_ready
                from PLACE.Place,
            installation(installation_time: date)
                to PLACE.Place,
            end_authorization(reason: string)
                from DECISION_MAKER.Production_manager,
            fase3_ready(supplier_winner_code: integer)
                from EQUIPMENT_PURCHASE_FASES.Payment,
            fase4_ready(supplier_winner_code: integer)
                to EQUIPMENT_PURCHASE_FASES.Test },
        states = {
            no_active, active, wait, ending },
        rules = {
            begin:

```

```

    msg(add_role) => state(no_active),
begin_reception:
    state(no_active), msg(fase3_ready(Supplier_code) =>
    state(active),
end_authorization:
    msg(end_authorization(Reason)) =>
    state(no_active),
reception:
    state(active), msg(delivery(Equipment_description, Product_code, Supplier_code,
    Delivery_time)) =>
    msg(reception(Product_code, Supplier_code, Delivery_time)),
    state(wait),
prepare_place:
    state(wait), msg(prepare_ready) =>
    msg(installation(Delivery_time)), state(ending),
reception_ready:
    state(ending) =>
    msg(fase4_ready(Supplier_code)), state(no_active) }
>,
<Test,
...
>)

```

Utilizando o modelo TF-ORM, podemos construir uma especificação passo-a-passo, com diferentes graus de refinamento [EDE 96]. Usando o método "top-down", uma classe pode ser refinada em várias classes, onde cada um dos papéis apresentados serão representados como classes distintas no próximo passo de refinamento. Esta técnica é especialmente importante quando se trata com classes do tipo processo, onde cada papel representa uma tarefa que, após o refinamento, torna-se uma nova classe do tipo processo.

4 Proposta de uma Técnica de Análise e Modelagem de Estruturas de Decisão

Este capítulo descreve a proposta de uma técnica que tem como propósito permitir a criação de um modelo detalhado da estrutura de decisão associada ao sistema de gerenciamento de uma empresa. O objetivo da técnica é definir passo-a-passo como, partindo de um primeiro modelo macro que representa a estrutura de decisão, chegar, por meio de refinamentos sucessivos, a um modelo detalhado que sirva como base para a implementação de um sistema de informação que possa dar suporte a este sistema de gerenciamento. Cada novo nível de refinamento corresponde à decomposição do nível anterior.

Para a definição dos níveis de refinamento é utilizada uma hierarquia para os procedimentos executados no Centro de Decisão. A hierarquia proposta baseia-se na hierarquia utilizada no modelo TF-ORM [EDE 94a], com o acréscimo somente do nível das ações. A figura 4.1 mostra a hierarquia de refinamento onde no primeiro nível são representados os Centro de Decisão, no segundo, estes centros são decompostos em tarefas, no terceiro, cada tarefa decomposta em ações e, no último nível, cada ação em operações. No mínimo estes quatro níveis devem ser identificados para que seja possível chegar-se ao produto final proposto pela técnica. Mas é possível que em determinadas situações seja necessário a decomposição dentro de um mesmo nível, por exemplo, as tarefas de um centro de decisão identificadas em um primeiro momento podem se mostrar muito complexas, exigindo, desta maneira, a sua decomposição em tarefas mais simples. O mesmo pode acontecer para as ações e operações. A definição da necessidade de decomposição dentro de um mesmo nível fica à critério do modelador.

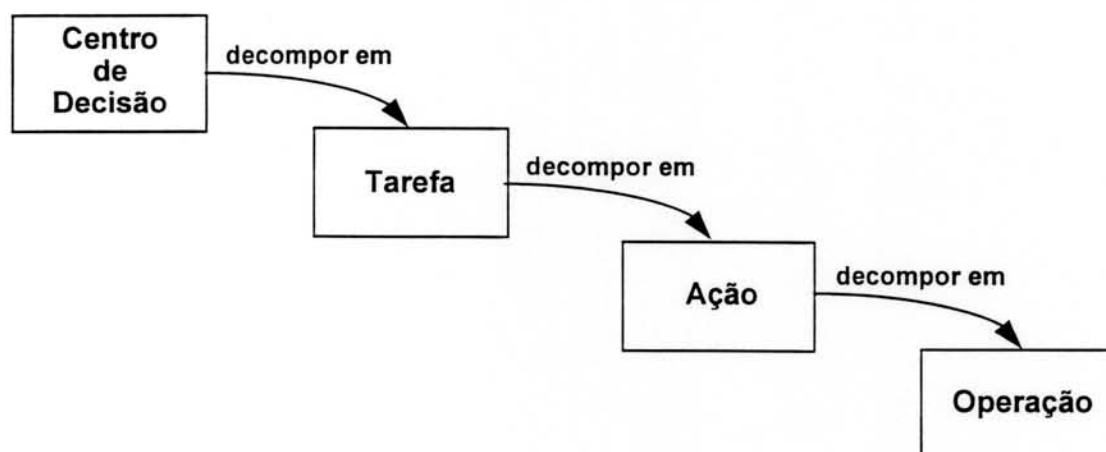


FIGURA 4.1 - Hierarquia de procedimentos

Para representar os modelos dos diversos níveis de refinamento são utilizadas as Redes de Petri com conceitos encontrados em duas modalidades destas redes, as do tipo Canal/Atividade para o primeiro nível

(Estrutura de Decisão) e as Compactas [HEU 91] para os níveis que identificam as tarefas, ações e operações. Em todos os níveis onde são utilizadas as Redes de Petri, encontra-se associada à transição a variável de tempo duração, o que caracteriza a utilização de um conceito encontrado numa variedade de Rede de Petri chamada de "Timed Petri Nets" [CAJ 88] [HAN 89]. No último nível, ou seja, no nível mais refinado, o qual servirá como base para a implementação de um sistema de informação, utiliza-se para a sua representação o modelo TF-ORM [EDE 94a].

Para representar o modelo macro no chamado nível 1 são utilizadas as Redes de Petri do tipo Canal/Atividade porque estas permitem a criação de um modelo sem formalismos matemáticos. Isto se deve ao fato de que neste nível inicial não é necessária uma representação completa, tornando, desta forma, a representação do modelo de fácil entendimento por parte dos usuários leigos, de forma a facilitar a comunicação entre estes e o modelador. Já a partir do primeiro nível de refinamento, ou seja, o nível 2, é importante o uso de um tipo de rede que permita representar os objetos envolvidos na estrutura (agentes, recursos e processos) de uma maneira mais completa, por isto foram usados os conceitos de representação gráfica encontrados nas Redes Compactas, os quais permitem representar o comportamento das entidades existentes no mundo real. Devido à sua importância na definição do modelo, foi acrescentada a variável duração da transição para indicar o tempo necessário para a realização da transição.

Já no último nível é necessário que seja utilizada uma ferramenta que permita criar um modelo que represente todas as informações importantes para a implementação de um sistema de informação associado à estrutura de decisão, por este motivo foi escolhido o modelo TF-ORM. A partir do modelo representado pelo TF-ORM é possível executar a implementação direta através de um Sistema de Banco de Dados Orientado a Objetos (SGBDOO) ou, por meio de mapeamento, a implementação em um Sistema de Banco de Dados Relacional (SGBDR).

4.1 Definição dos Passos utilizados na Técnica

Foi definida uma técnica de análise e modelagem através de uma seqüência de passos que devem ser seguidos até chegar ao produto final que é o modelo descrito pelo TF-ORM. Estes passos serão detalhados a seguir.

PASSO 1: No primeiro passo, chamado de nível 1, são descritos os **Centros de Decisão** que pertencem ao escopo do modelo. Este nível é representado através do uso das Redes de Petri do tipo Canal/Atividade, utilizando o conceito de duração da transição vindo das "Timed Petri Nets". Para cada centro de decisão devem ser identificados os seguintes elementos:

- o responsável pela sua execução;
- o tempo estimado para sua execução, isto é, a sua duração, a qual indica o tempo necessário para que a tomada de decisão produza algum efeito sobre

o sistema gerenciado, o que corresponde à variável B mostrada na seção 2.2 deste trabalho;

- os agentes envolvidos, isto é, as entidades responsáveis por algum procedimento durante a execução;
- os recursos envolvidos, ou seja, as entidades que são necessárias para a execução ou são geradas após; e
- as regras de transição que definem um critério que deve ser verdadeiro para que o centro de decisão possa ser executado.

A variável duração não precisa necessariamente ser definida neste momento pois trata-se somente de uma estimativa, portanto fica a critério do modelador se esta deve ou não ser colocada no modelo. Se a estimativa for feita, esta poderá ser corrigida a cada novo nível de refinamento. Isto cria um mecanismo de validação desta estimativa, de forma que sempre após o modelador refinar mais um nível conforme a hierarquia dos procedimentos mostrada anteriormente, o mesmo deve retornar ao nível anterior para recalcular as durações estimadas.

Após relacionar todos os Centros de Decisão que deverão fazer parte do modelo, parte-se para o desenho da Rede de Petri conforme a seguinte notação:

- Os **Centros de Decisão** identificados serão modelados como **transições**, representadas graficamente por um retângulo dividido em duas partes, onde na parte superior será preenchido o **nome do responsável** pela execução do centro, e na inferior a **identificação do centro ou função**, ou seja, o objetivo do centro, e a **regra de transição**, a qual deve ser satisfeita para que o centro possa ser executado. Abaixo da transição será indicada, se necessário, a **duração estimada** em dias e colocada entre parênteses.



FIGURA 4.2 - Representação do Centro de Decisão

- Os **Recursos** e/ou **Agentes** envolvidos serão representados como **lugares** da rede, com a exceção do agente responsável, já indicado na própria transição. A **direção dos arcos** indica, no caso dos Agentes, se o mesmo é responsável pelo envio (**arcos de entrada**), recebimento de um recurso (**arcos de saída**) ou simplesmente consultado (**arcos restauradores**). No caso dos Recursos, estes podem ser consumidos ou alocados temporariamente (**arcos de entrada**), simplesmente consultados (**arcos restauradores**) ou gerados (**arcos de saída**) pelo Centro de Decisão.

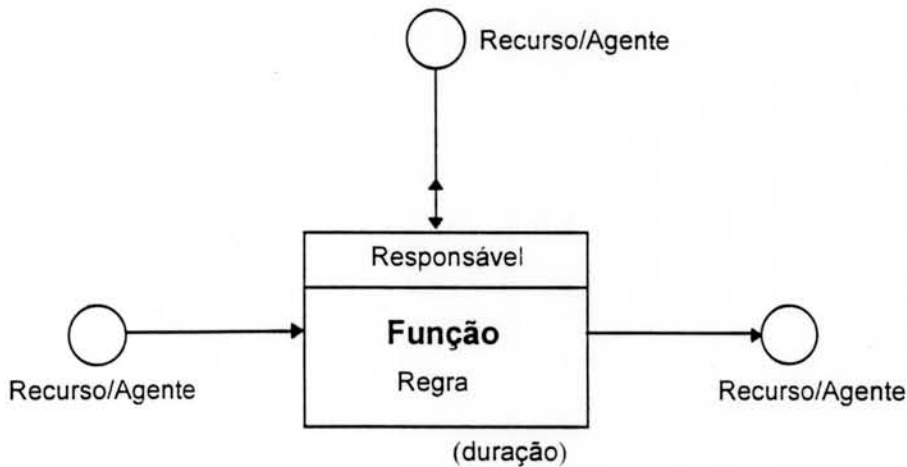


FIGURA 4.3 - Representação dos Agentes e Recursos envolvidos

- No caso de um Recurso sofrer atualização em virtude da execução da função de decisão, este será ligado através de dois arcos, um de entrada e o outro de saída.

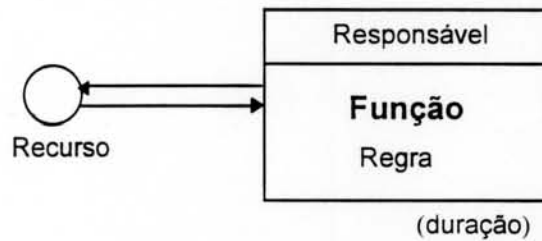


FIGURA 4.4 - Representação da atualização de um recurso

Após este primeiro passo ter sido concluído, serão mostrados todos os Centros de Decisão envolvidos em uma única rede, o que permitirá ver como estes centros se relacionam através dos recursos e/ou agentes requisitados e gerados por eles. Neste momento é possível, portanto, ter a visão de como é a chamada **Estrutura de Decisão**.

A seguir será apresentado um exemplo de modelo segundo o passo 1 onde são representados três centros de decisão. Em primeiro lugar é criada uma tabela com a relação dos centros que fazem parte do escopo. A partir desta tabela é, então, criado o modelo utilizando as Redes de Petri.

TABELA 4.1 - Exemplos de Centros de Decisão

CENTRO DE DECISÃO	RESPONSÁVEL	DURAÇÃO (B)	AGENTES	RECURSOS	REGRAS DE TRANSIÇÃO
Definir Recursos Necessários	Gerente de Produção	30 dias	<ul style="list-style-type: none"> • Gerente de Produção 	<ul style="list-style-type: none"> • Verba disponível p/ aquisição (info) • Variação da Capacidade de Produção - VCap (info) • Relação dos Equipamentos Necessários • Relação dos Funcionários Necessários 	<ul style="list-style-type: none"> • VCap > 0
Comprar Equipamentos	Gerente de Produção	90 dias	<ul style="list-style-type: none"> • Gerente de Produção • Fornecedores 	<ul style="list-style-type: none"> • Funcionários • Locais p/ instalação • Equipamentos comprados • Verba disponível p/ aquisição • Rel. Equip. necessários (info) 	<ul style="list-style-type: none"> •
Contratar Funcionários	Gerente de Pessoal	30 dias	<ul style="list-style-type: none"> • Gerente de Pessoal • Candidatos 	<ul style="list-style-type: none"> • Relação dos Funcionários necessários (info) • Funcionários contratados 	<ul style="list-style-type: none"> •
...			<ul style="list-style-type: none"> • 	<ul style="list-style-type: none"> • 	<ul style="list-style-type: none"> •

Podemos observar no exemplo que o centro "Definir Recursos Necessários" possui uma regra de transição que mostra que este só será executado caso a variação da capacidade de produção (VCap) seja positiva. O recurso "Verba disponível p/ aquisição" é consultado pelo centro de decisão "Definir Recursos Necessários", por isso o arco restaurador, e atualizado pelo centro "Comprar equipamentos", como mostram os arcos de entrada e saída.

Observe a utilização da palavra "info" ao lado de um recurso, isto indica que este se trata de uma informação ou documento e não de um recurso físico.

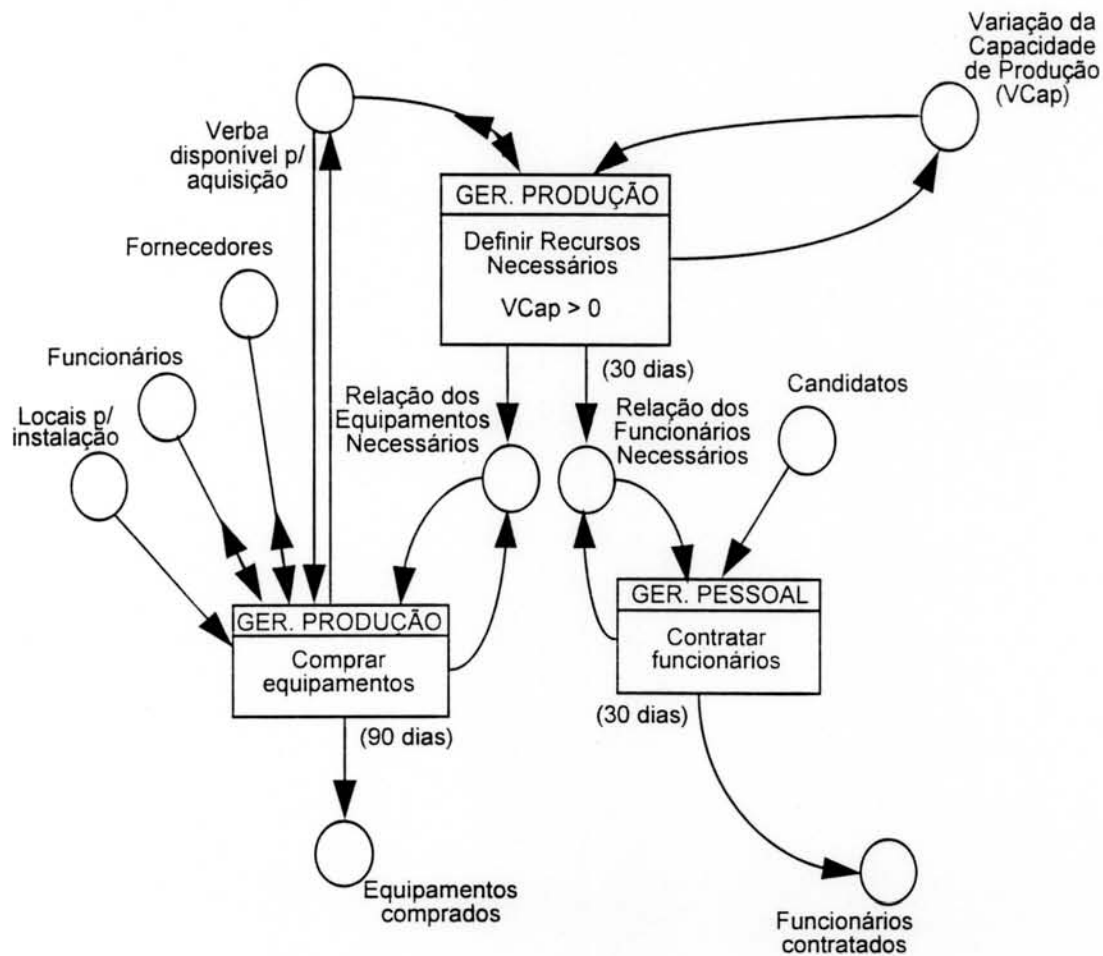


FIGURA 4.5 - Exemplo nível 1

PASSO 2: A partir deste passo inicia-se o refinamento do modelo com a criação de modelos através da utilização de conceitos extraídos das Redes de Petri Compactas. Como já foi dito, o primeiro nível de refinamento significa decompor os centros de decisão em **Tarefas** que são realizadas durante a sua execução, definindo:

- a duração estimada da tarefa;
- a regra de condição para a execução da tarefa;
- os agentes e recursos envolvidos na execução da tarefa; e
- os estados destes agentes e recursos, antes e depois da execução da tarefa.

Desta forma, serão criados vários modelos, um para cada Centro de Decisão mostrado na rede desenhada no passo 1. As transições correspondem às tarefas e dentro do retângulo são anotados o nome que identifica a tarefa e a regra que define a condição para que esta possa ser executada. Ao lado de cada arco da rede serão anotados, entre chaves, os identificadores dos objetos (no caso, agentes e recursos) que poderão ocupar os lugares os quais são ligados por estes arcos. A partir deste nível, os lugares passam a

representar os estados em que estes objetos se encontram antes e após a execução da tarefa à qual estão conectados.

Por isso, os agentes e recursos já representados no nível 1 aparecem neste nível com a mesma descrição adicionada do identificador do estado. Por exemplo, o recurso "Funcionários" mostrado na figura 4.5 (exemplo do nível 1) aparece como "Funcionários livres" na figura 4.7 (exemplo nível 2).



FIGURA 4.6 - Representação de uma tarefa

Caso sejam identificados novos agentes e/ou recursos durante a definição das tarefas, o modelo da Estrutura de Decisão criado no passo 1 deve ser revisado. Este procedimento deverá ser executado a cada novo nível de refinamento, pois este servirá como mecanismo de validação do nível anterior.

Como exemplo, a figura 4.7 mostra a representação de apenas duas das várias tarefas identificadas no centro de decisão "Comprar Equipamento", são elas: a "Especificação", onde são definidas as características necessárias dos equipamentos que serão adquiridos; e a "Licitação", onde é feita a escolha da melhor entre as propostas dos fornecedores consultados. Como podemos observar no modelo apresentado, a execução da tarefa "Especificação" tem uma duração estimada em 15 dias e depende, conforme mostram os arcos e lugares de entrada, da existência de fornecedores ($\{S\}$), de funcionários ($\{E\}$) livres, uma relação de locais ($\{P\}$) disponíveis para instalação e de equipamentos ($\{Eq\}$) relacionados na "Lista de equipamentos necessários". Os arcos e lugares de saída indicam que, após o término desta tarefa, os fornecedores passam a um estado de convidados, os funcionários escolhidos passam para o estado de alocados e, agora, fazem parte da comissão encarregada da operacionalização da compra ($\{PC\}$), a qual aguarda as propostas dos fornecedores, o local para instalação já está definido e o equipamento a ser adquirido está especificado.

Com a representação do modelo através das redes de Petri é possível identificar que, como os lugares de saída da tarefa de especificação correspondem aos lugares de entrada da tarefa de licitação, então esta segunda tarefa só poderá iniciar após a primeira ter sido concluída. Portanto, para que se inicie a tarefa de licitação, é necessário que a comissão de compra esteja aguardando as propostas, os fornecedores tenham sido convidados e já exista o local definido para a instalação do equipamento a ser adquirido. A duração da tarefa de licitação foi estimada em 45 dias e, ao final, o local para instalação estará pronto para receber o novo equipamento, a comissão de

compra estará aguardando os testes dos equipamentos após o recebimento e o fornecedor escolhido estará autorizado a fornecer o material licitado.

Antes de partir para a confecção do modelo, a seguinte tabela é construída contendo as informações necessárias:

TABELA 4.2 - Exemplos de Tarefas

TAREFA	DURAÇÃO	AGENTES	RECURSOS
Especificação	15 dias	<ul style="list-style-type: none"> Gerente de Produção (PM) Comissão de Compra (PC) Fornecedores (S) 	<ul style="list-style-type: none"> Funcionários (E) Locais p/ instalação (P) Relação dos Equipamentos necessários (Eq) (info)
Licitação	45 dias	<ul style="list-style-type: none"> Gerente de Produção (PM) Comissão de Compra (PC) Fornecedores (S) 	<ul style="list-style-type: none"> Locais p/ instalação (P)
...	...	• ...	• ...

Observe que na tabela de tarefas, aparecem as colunas com os agentes e os recursos. O gerente de produção, por ser o responsável geral pela execução do centro de decisão ao qual as tarefas pertencem, não necessariamente precisa estar representado na rede deste nível. A comissão de compra e os fornecedores são relacionados como agentes porque ambos são responsáveis por algumas das decisões tomadas.

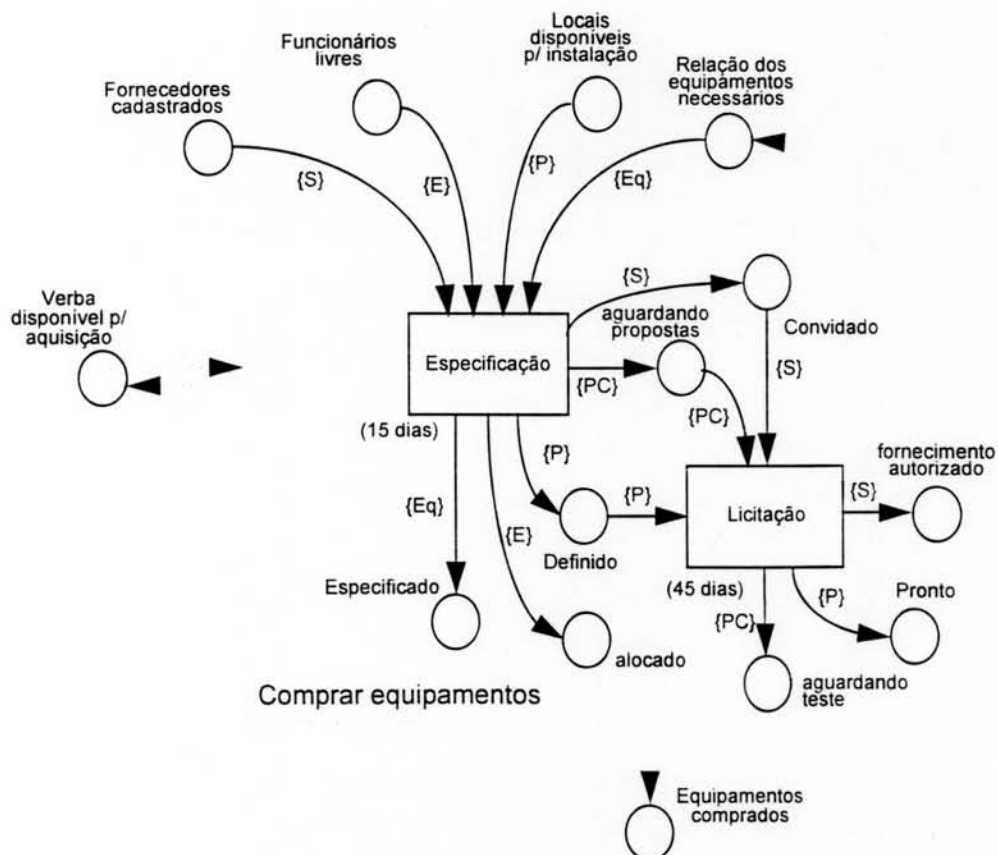


FIGURA 4.7 - Exemplo nível 2

PASSO 3: Neste passo é executado o segundo refinamento da estrutura de decisão, onde são criadas as Redes de Petri que modelam as **Ações** que compõem as tarefas identificadas no passo anterior. Nestas redes serão identificadas as mudanças no estado da tarefa provocadas pela execução das ações, mudanças de estado dos agentes e recursos envolvidos, assim como as regras que devem ser satisfeitas para que a ação possa ser executada. Para a execução desta etapa, primeiro é gerada uma tabela com a relação das ações identificadas em cada tarefa e, com base nesta tabela, são geradas as redes de Petri para cada tarefa onde são identificados os estados das entidades. Observe o exemplo abaixo:

Tabela 4.3 - Exemplos de Ações.

TAREFA	AÇÕES	RECURSOS	AGENTES	REGRAS
Especificação	• Autorizar o início do processo de compra.	• -	• Gerente de Produção	• -
	• Solicitar lista de funcionários para comissão.	• -	• AGENTE EXTERNO	• -
	• Definir Comissão de Compra.	• Funcionário	• AGENTE EXTERNO e Comissão de Compra	• -
	• Definir local para a instal. do equipamento.	• Local	• AGENTE EXTERNO	• -
	• Enviar a descrição da compra p/ a comissão.	• -	• AGENTE EXTERNO e Comissão de Compra	• -
	• Especificar equipamento.	• Eqpto.	• Comissão de Compra	• -
	• Solicitar Lista de fornecedores.	• -	• AGENTE EXTERNO	• -
	• Receber a lista dos fornecedores.	• -	• AGENTE EXTERNO	• 3 fornecedores distintos
	• Enviar especificação para os fornecedores.	• -	• Fornecedor	• -
	• Cancelar autorização de compra.	• -	• Gerente de Produção	• -

TAREFA	AÇÕES	RECURSOS	AGENTES	REGRAS
Licitação	<ul style="list-style-type: none"> • Iniciar licitação. • Cancelar autorização de compra. • Receber as propostas. • Preparar local para a instalação do equipamento. • Escolher a melhor proposta. • Solicitar negociação dos itens da proposta escolhida. • Não solicitar negociação. • Aceitar negociação. • Não aceitar negociação. • Enviar ordem de fornecimento. • Encerrar licitação. 	<ul style="list-style-type: none"> • - • - • - • Local • - • - • - • - • - • - • - 	<ul style="list-style-type: none"> • - • Gerente de Produção • Fornecedor e Comissão de Compra • - • Fornecedor e Comissão de Compra • Fornecedor e Comissão de Compra • Comissão de Compra e Gerente de Produção • Fornecedor e Gerente de Produção • Fornecedor • Fornecedor e Gerente de Produção • - 	<ul style="list-style-type: none"> • - • - • 3 fornecedores distintos • - • Antes da data limite • - • - • Novas cond. = condições solicitadas • Novas cond. ≠ condições solicitadas e razão ≠ Nulo • - • -
...	• ...	•	•	•

Como exemplo de modelo do nível 3, é mostrada a rede de Petri que representa apenas três ações que fazem parte da tarefa "Especificação", a qual aparece na figura 4.7 como exemplo do nível 2. O retângulo pontilhado indica a tarefa e nele são ligados os arcos que representam as entradas e saídas da tarefa conforme a figura anterior. A primeira ação mostrada é aquela que "Envia descrição da compra" para a comissão de compra, esta ação é executada no momento em que a tarefa "especificação", representada por {S}, está ativa e a composição da comissão de compra ({PC}) já foi definida, após a sua execução, a comissão de compra passa para o estado de ocupada, pois está criando a especificação, e a tarefa {S} fica aguardando a especificação. Feito isto, a segunda ação, onde se "Define a especificação" é executada. Para que seja possível iniciar a sua execução, esta ação depende de que uma instância do objeto equipamento tenha sido criada, após o encerramento da execução desta segunda ação, a tarefa passa para o estado especificação pronta, a comissão de compra fica aguardando as propostas dos fornecedores e o equipamento passa para um estado de especificado. A última ação do exemplo representa a "Solicitação da Lista de Fornecedores" a serem convidados para a licitação. Após a execução desta ação, a tarefa fica aguardando o recebimento da lista de fornecedores.

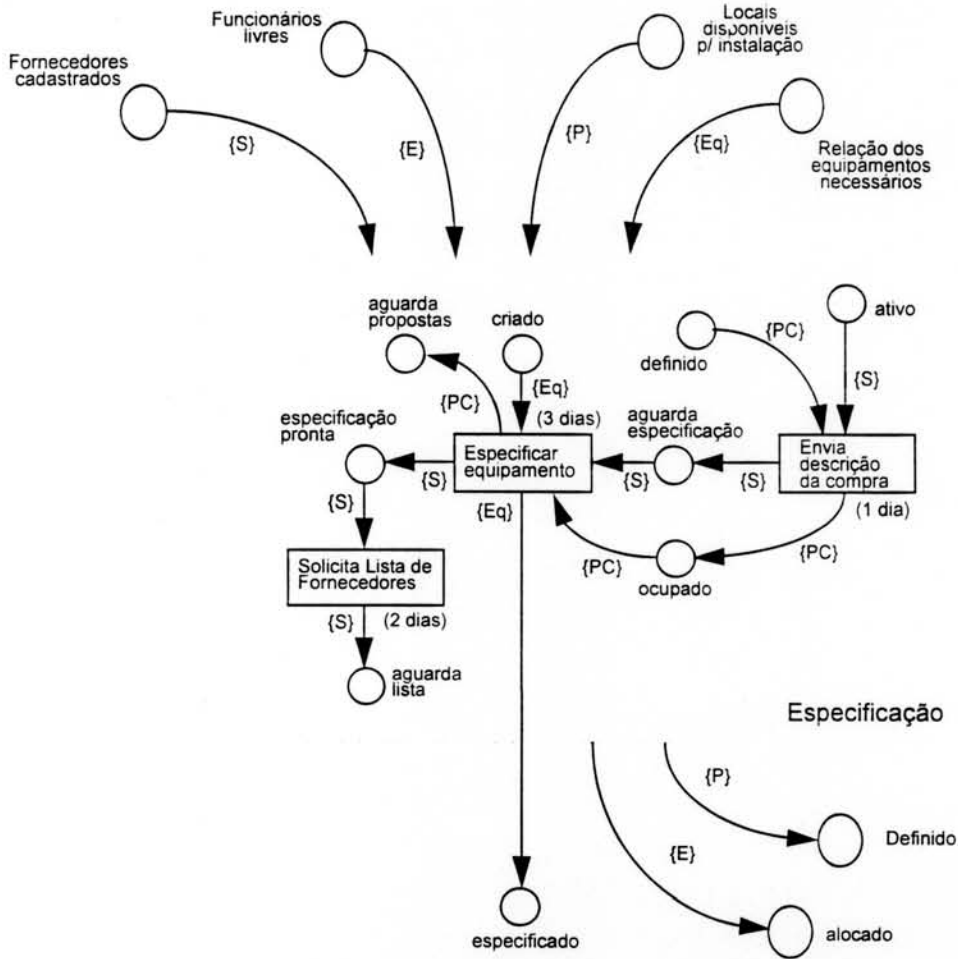


FIGURA 4.8 - Exemplo nível 3

PASSO 4: Neste passo é executado o terceiro e último nível de refinamento onde são identificadas as **operações** executadas dentro de cada ação. Caso as operações sejam simples, torna-se desnecessária a utilização das Redes de Petri para representá-las.

As operações representam a forma como os objetos do tipo agente e/ou recurso representados no modelo interagem com os objetos do tipo processo, isto significa, que podem ser interpretadas como mensagens enviadas ou recebidas, as quais servem de elo de ligação entre os agentes/recursos e os processos. Para cada operação deve-se identificar: 1) quem (agente) executa a operação, isto é, envia a mensagem, ou sobre quem (agente) ou o que (recurso) a operação atua, ou seja, recebe a mensagem; e 2) as informações associadas a operação.

Para que seja possível entender com maior clareza a coluna da tabela "QUEM EXECUTA? SOBRE QUE(M)?" observe, por exemplo, a operação "Autorização de Início", esta é executada pelo agente "Gerente de Produção", ou seja, este é o objeto que envia a mensagem, como é assinalado pela palavra "envia" ao lado do nome do agente; já no caso de "Definição da comissão", esta é executada sobre o agente "Comissão de Compra", isto é, este é o objeto que recebe a mensagem, por isso a palavra "recebe" entre

parênteses; e, como último exemplo, na operação “Ordem de alocação”, a mesma é executada sobre o recurso “Funcionário”, o qual é quem recebe a mensagem.

Continuando o exemplo utilizado, observe a tabela apresentando o refinamento de algumas das ações ligadas à tarefa “Especificação”. No caso do exemplo, já que as operações mostradas são bastante simples, foi dispensada a representação das mesmas com Redes de Petri.

TABELA 4.4 - Exemplos de Operações.

AÇÃO	OPERAÇÕES	QUEM EXECUTA? SOBRE QUE(M)?	INFORMAÇÃO ASSOCIADA
• Autorizar o início do processo de compra.	• Autorização de início	• Gerente de Produção (envia)	• Descrição da compra, data de início, verba máxima disponível, prazo para compra e prazo da licitação
• Solicitar lista de funcionários para comissão.	• Solicitação de alocação de funcionário	• Agente Externo (envia)	• Data de início e data de fim da alocação
• Definir Comissão de Compra.	• Recebe lista dos funcionários disponíveis • Ordem de alocação • Definição da comissão	• Agente Externo (envia) • Funcionários escolhidos (recebe) • Comissão de Compra definida (recebe)	• Componentes • Matrícula do funcionário, função na comissão e data de início da alocação • Componentes, data de início e data de fim da comissão
• Definir local para a instalação do equipamento.	• Recebe a lista dos locais disponíveis • Definição do local	• Agente Externo (envia) • Local escolhido (recebe)	• Localização, área e características técnicas • Localização, área e características técnicas
• Enviar a descrição da compra para a comissão.	• Recebe a lista dos equipamentos necessários • Informa objeto da compra	• Agente Externo (envia) • Comissão de Compra (recebe)	• Equipamentos necessários • Descrição do objeto para compra
• ...	•	•	•

PASSO 5: Este último passo da técnica consiste em mapear todas as informações modeladas nos passos anteriores para um modelo TF-ORM. O mapeamento é feito diretamente através das regras listadas abaixo:

1. Os Centros de Decisão, identificados no passo 1 da técnica, aparecem no modelo TF-ORM como classes de objeto do tipo Processo.
2. As Tarefas de cada centro de decisão, modeladas durante o passo 2, darão origem aos Papéis das classes de objeto do tipo Processo.

3. Os Agentes e Recursos associados às tarefas serão representados como Papéis das classes de objeto do tipo Agente ou Recurso. Conforme a natureza destes agentes e/ou recursos, é possível agrupá-los em classes de objeto.

4. As Ações, apresentadas no passo 3, serão representadas como Regras de Transição associadas aos papéis das classes de objeto Processo.

5. As Operações, identificados no passo 4, serão representadas através das Mensagens enviadas e recebidas pelas classes de objetos. É importante salientar uma restrição necessária para o uso da técnica, de que as mensagens são sempre enviadas ou recebidas por um processo, isto quer dizer, não existem mensagens trocadas diretamente entre os agentes e recursos. Desta forma, todas as operações existentes serão mapeadas dentro de um processo.

6. Os Estados dos objetos representados nas Redes de Petri do nível 3, as quais mostram as tarefas, serão mapeados para os próprios Estados associados aos papéis das classes.

7. As Regras das Transições serão apresentadas no TF-ORM como condições de transição das regras de transição de estado.

Para mostrar de forma mais clara o passo do mapeamento para TF-ORM, será utilizado o exemplo a seguir.

No exemplo mostrado no passo 1 da técnica, onde é modelada a estrutura de decisão, foi identificado um centro de decisão chamado “Comprar equipamentos”, no passo 2 foram mostradas as tarefas executadas neste centro de decisão, entre elas aparece no modelo a tarefa “Especificação”, ao refinarmos esta tarefa no passo 3, entre as diversas ações mostradas existe a ação “Especificar equipamento”, a qual envolve o agente “Comissão de compra” e o recurso “Equipamento”. As operações executadas nesta ação, mostradas na tabela do exemplo do passo 4, são: “Recebe a especificação” da Comissão de compra com os campos de informação descrição, características e funções do equipamento e; “Definição da especificação” do equipamento a ser adquirido com os mesmos campos de informação.

Estes elementos do modelo serão representados no TF-ORM da seguinte maneira:

```
Process class (
  EQUIPMENT_PURCHASE_FASES,
  <Base_role,
  static properties = {
    (max_allocation, real),
    (expiry_time, date) },
  dynamic properties = {
    (auction_limit_time, date) },
```

```

rules = {
  r1: msg(create_object) => state(active),
  r2: state(active) => msg(add_role(Specification))
  r3: state(active) => msg(add_role(Auction))
  r4: state(active) => msg(add_role(Payment))
  r5: state(active) => msg(add_role(Reception))
  r6: state(active) => msg(add_role(Test)) }
>,
<Specification,
  messages = {
  . . .
  specification(description: string, features: string, functions: string)
    from DECISION_MAKER.Purchase_commission,
  . . .
  define_specification(description: string, features: string, functions: string)
    to EQUIPMENT.Equipment,
  . . . }
  states = {
  no_active, active, wait_employee1, wait_employee2, wait_specification, specified,
  wait_supplier_list, sending1, sending2, sending3 },
  rules = {
  . . .
  define_specification:
    state(wait_specification), msg(specification(Description, Features, Functions)) =>
    msg(define_specification(Description, Features, Functions)), state(specified),
  . . . }
>,
<Auction,
. . .
>,
<Payment,
. . .
>,
<Reception,
. . .
>,
<Test,
. . .
>)

```

O Centro de decisão “Comprar equipamento” foi modelado no TF-ORM como a classe processo “Equipment_purchase_fases”, a tarefa “Especificação” é representado como um papel da classe processo chamado “Specification”, a ação “Especificar equipamento” foi mapeada como uma regra de transição chamada “Define_specification”, por último, as operações “Recebe a especificação” mapeada como a mensagem “specification” enviada pelo agente “Purchase_commission”, ou seja, a “Comissão de compra”, e a operação “Definição da especificação” como “define_specification” enviada para o recurso “Equipment”, ou seja, o “Equipamento”.

É possível observar que a técnica descrita acima mostra como, através do refinamento do modelo, podemos descrever o comportamento dos objetos tipo processo no modelo TF-ORM. Agora é necessário criar os modelos que descrevam o comportamento dos objetos agentes e recursos, estes modelos são derivados a partir dos modelos dos processos nos quais estes estão envolvidos.

4.2 Modelagem dos Agentes e Recursos

Executando os passos mostrados, o resultado será o modelo TF-ORM dos processos de decisão, mas paralelamente a isto, é necessário que sejam criados os modelos dos agentes e recursos. A seguir será mostrado um exemplo de como são derivados os modelos dos agentes e/ou recursos associados aos processos.

Considere a porção de uma Rede de Petri que representa uma ação através da transição *R1*, a qual faz parte da rede de uma tarefa *obj1* e que envolve o recurso e/ou agente representado pelo objeto *obj2* e onde as condições 1 e 2 precisam ser satisfeitas para que a ação possa ser executada. Antes da execução desta transição, o objeto do tipo processo *obj1* está no estado *S11* e o objeto do tipo recurso/agente *obj2*, no estado *S21*, após a execução, o estado de *obj1* passa a ser *S12* e o do *obj2*, *S22*.

Os agentes e/ou recursos estão ligados aos processos através das operações, as quais são executadas por estes agentes, executadas sobre estes agentes ou que atuam sobre os recursos. Como, conforme o passo 5 da técnica, todas as operações são mapeadas como mensagens das classes do tipo processo. Desta forma, todas estas mensagens correspondem também a mensagens enviadas ou recebidas pelas classes de agentes e/ou recursos relacionadas. Portanto, a medida que vão sendo definidas as operações, é possível também definir as mensagens dos agentes e recursos envolvidos. Por fim, através das Redes de Petri que identificam as ações, pode-se obter as regras de transição de agentes e recursos. No caso das regras identificadas nas classes agentes, nas quais somente aparecem mensagens enviadas, torna-se necessário um estudo para tentar identificar se existe alguma parcela de trabalho não estruturada, sendo esta representada por meio de uma decisão.

No exemplo abaixo, a operação que relaciona o processo *obj1* ao recurso e/ou agente *obj2* é mapeado no modelo TF-ORM através da mensagem *M1*.

Rede de Petri

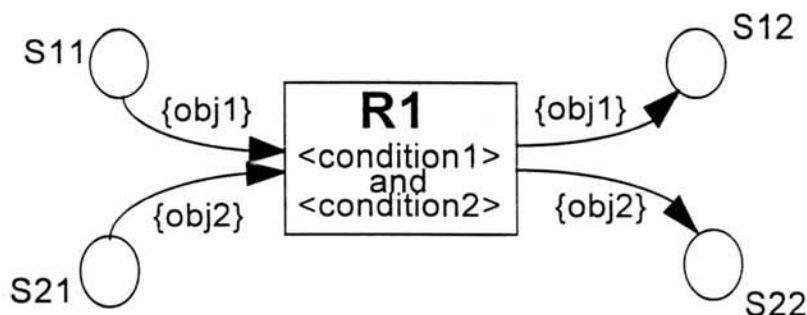


FIGURA 4.9 - Rede de Petri da ação R1

TF-ORM

```

Process class (
  PROCESS_CLASS_NAME,
  < Base_role,
    ...
  >,
  < Obj1,
    static properties = { ... },
    dynamic properties = { ... },
    messages = {
      ... ,
      M1( ... ) from RESOURCE/AGENT_CLASS_NAME.Obj2,
      ... },
    states = { ... , S11, S12, ... },
    rules = {
      ... ,
      R1:
        state(S11), msg(M1( ...)) =>
        state(S12); <condition1>,
      ... }
  >,
  <
  ...
  >)

```

```

Resource (ou Agent) class (
  RESOURCE/AGENT_CLASS_NAME,
  < Base_role,
    ...
  >,
  < Obj2,
    static properties = { ... },
    dynamic properties = { ... },
    messages = {
      ... ,
      M1( ... ) to PROCESS_CLASS_NAME.Obj1,
      ... },
    states = { ... , S21, S22, ... },
    rules = {
      ... ,
      Rn:
        state(S21) =>
        msg(M1( ...)), state(S22); <condition2>,
      ... }
  >,
  < ...
  >)

```

Para exemplificar com maior clareza a criação dos modelos dos agentes e recursos com o TF-ORM, será dada continuidade ao exemplo da tarefa “Especificação” mostrado no passo 5 da técnica. Neste exemplo é mostrada apenas a modelagem da ação “Especificar equipamento”, a qual envolve o agente “Comissão de compra” e o recurso “Equipamento”. Agora serão apresentados parte dos modelos do agente e recurso envolvidos derivados a partir do modelo da tarefa:

```

Agent class (
  DECISION_MAKER,
  <Base_role,
    static properties = {
      (code, integer),
      (cpf, integer),
      (date_birth, date) },
    dynamic properties = {
      (name, string),
      (address, string),
      (function, string),
      (salary, real) },
    rules = {
      r1: msg(create_object) => state(active),
      r2: state(active) => msg(add_role(Production_manager)),
      r3: state(active) => msg(add_role(Purchase_commission)) }
  >,
  <Production_manager,
  ...
  >,
  <Purchase_commission,
    messages = {
  ...
      specification(description: string, features: string, functions: string)
        to EQUIPMENT_PURCHASE_FASES.Specification,
  ... },
    decision = {
      define_specification(description: string, features: string, functions: string),
  ... },
    states = {
      undefined, defined, occupied, evaluated, wait_proposal, in_negotiation, wait_test},
    rules = {
  ...
      specify_equipment:
        state(occupied), decision(define_specification(Description, Features, Functions)) =>
        msg(specification(Description, Features, Functions)), state(wait_proposal),
  ... }
  > )
Resource class (
  EQUIPMENT,
  <Base_role,
    static properties = {
      (code, string),
      (description, string),
      (features, string),
      (functions, string) },
    dynamic properties = {
      (supply_product_code, integer), /* Código utilizado pelo fornecedor*/
      (model, string),
      (supply, string),
      (site, string),

```

```
(setup_time, date),
(lifetime, date),
(delivery_time, date),
(technical_conditions, string) },
rules = {
  r1: msg(create_object) => state(active),
  r2: state(active) => msg(add_role(Equipment)) }
>,
<Equipment,
  messages = {
    define_specification(description: string, features: string, functions: string)
    from EQUIPMENT_PURCHASE_FASES.Specification,
  ... },
  states = {
    created, specificated, receipted, approved },
  rules = {
  ...
  specification:
    state(created), msg(define_specification(Description, Features, Functions)) =>
    state(specificated),
  ... }
>)
```

5 O Estudo de Caso

Para exemplificar a utilização prática da técnica proposta no capítulo anterior será apresentado um estudo de caso com uma estrutura de decisão baseada em um modelo de hierarquia de planejamento e programação mostrada em [WAL 89] e geralmente encontrada em um típico ambiente de fábrica no que diz respeito aos processos decisórios ligados ao gerenciamento da produção.

Neste ambiente, a capacidade produtiva da fábrica é reavaliada conforme os dados informados sobre a demanda do mercado, ou seja, a quantidade de pedidos que chegam no setor de vendas, e o orçamento anual previsto para investimento na produção, o qual é definido pela diretoria e dentro da hierarquia representada pelo nível estratégico. Com base nestas informações, pode-se chegar à conclusão de que a capacidade de produção deve ser ampliada ou reduzida e, esta variação, após ter sido determinada quantitativamente pela gerência de produção, se dá através da aquisição e/ou venda de máquinas e equipamentos e da contratação e/ou demissão de funcionários. Após os recursos serem redistribuídos pelas áreas conforme as necessidades destas, a programação da produção é reavaliada com base na lista de pedidos e nas relações atualizadas dos recursos (equipamentos e funcionários) disponíveis.

Analisando, portanto, a estrutura de decisão associada ao gerenciamento da capacidade de produção da fábrica podemos identificar alguns centros de decisão de maior destaque.

O primeiro centro de decisão identificado corresponde a "**Aceitar novos pedidos**", no qual a decisão é tomada pelo Gerente do Setor de Vendas da empresa. Neste centro, para que seja possível tomar uma decisão, o Gerente de Vendas precisa analisar os Pedidos que chegam, consultar os Indicadores de Demanda do Mercado e a Lista dos pedidos que ainda estão aguardando. A partir destas informações, ele define se aceita ou não os novos pedidos, atualizando a previsão de demanda e, em caso positivo, a lista dos pedidos que estão aguardando, onde é informada a data de entrega prometida para cada pedido. Tomando como base as informações fornecidas pela Gerência de Vendas, o tempo estimado entre o momento da chegada de um pedido e a decisão da aceitação ou não deste é de aproximadamente 1 dia.

No nível Estratégico existe o centro de decisão "**Define o Orçamento Anual**", onde a diretoria define, entre outras, a verba disponível no ano para que sejam feitos investimentos na alteração da capacidade de produção e informa esta previsão para a Gerência de Produção. Entre a data limite para que as áreas encaminhem as suas necessidades e a aprovação do orçamento deve decorrer no máximo 15 dias.

Baseado nas informações fornecidas pelos dois Centros de Decisão identificados acima, existe o centro encarregado de "**Determinar a**

Capacidade de Produção da Fábrica", sendo que o responsável pela decisão é o Gerente de Produção. Este Centro de Decisão define, baseado nos Indicadores de Desempenho medidos na linha de produção, na previsão de demanda e na Lista de pedidos que estão aguardando, a Variação da Capacidade de Produção (VCap) necessária para que esta capacidade esteja de acordo com a previsão de demanda, caso a decisão seja de ampliar esta capacidade, isto será feito de acordo com a verba anual disponível conforme o orçamento para investimentos na produção. São necessários pelo menos 5 dias para que seja avaliada toda a informação necessária e o Gerente de Produção defina a capacidade que deverá ser alcançada, a qual é reavaliada mensalmente.

Existem outros dois Centros de Decisão, que também sob a coordenação da Gerência de Produção, são responsáveis por definir quais os ajustes a serem feitos na capacidade de produção da fábrica. Um destes centros "**Define os recursos necessários**", ou seja, quais os recursos devem ser adquiridos caso a capacidade de produção deva ser ampliada ($VCap > 0$), e o outro, "**Define os recursos a serem eliminados**", isto é, quais recursos devem ser dispensados caso a capacidade deva ser reduzida ($VCap < 0$). Nestes dois centros, a decisão é tomada com base na variação da capacidade (VCap), a qual foi definida pelo centro de decisão encarregado de determinar este valor. Além disso, o centro que define os recursos a serem adquiridos também precisa verificar a verba disponível para o investimento em recursos para a produção. Após a decisão ter sido tomada, a variação da capacidade de produção é atualizada, isto é, desconta-se a variação da capacidade a ser implementada em função da aquisição ou eliminação de recursos que foi decidida, e a partir daí são definidas as relações dos equipamentos e funcionários que necessitam ser adquiridos/contratados ou vendidos/dispensados. Estas decisões são reavaliadas mensalmente, portanto após ter sido definida a nova capacidade de produção a ser atingida, a Gerência de Produção tem 30 dias para definir as relações de equipamentos e funcionários necessários ou dispensáveis.

Após a definição da relação dos equipamentos necessários, o centro de decisão "**Comprar equipamentos**" é encarregado de executar todo o processo de compra de um, ou de um lote de equipamentos, sendo que para isso são necessários: uma lista de fornecedores para a apresentação de propostas, possuir verba disponível para a aquisição, a relação de funcionários que irão participar da comissão de compra a ser convocada, além da própria relação dos equipamentos necessários e da relação dos locais disponíveis para a instalação do novo equipamento. Este centro de decisão se encontra a cargo da Gerência de Produção e deve informar, após a compra, os novos equipamentos adquiridos, atualizar a relação dos equipamentos necessários, excluindo os equipamento que foram adquiridos, e atualizar a verba disponível conforme o saldo restante. Desde o momento em que é decidido pela compra de um equipamento, passando pelas diversas etapas que devem ser cumpridas, até a sua instalação, o prazo é, na média, de 90 dias.

Por outro lado, se existir a necessidade de "**Contratar funcionários**", a Gerência de Pessoal será encarregada de executar esta contratação, baseada na lista de candidatos e na relação dos funcionários necessários anteriormente definida. Este centro de decisão deve informar os dados dos funcionários contratados e atualizar a relação de funcionários que ainda necessitam ser contratados. O tempo mínimo necessário para que seja concluído o processo de contratação de um novo funcionário é em média de 30 dias.

Após novos equipamentos terem sido adquiridos, a Gerência da Produção, com base nas necessidades colocadas pelas áreas de produção, deve "**Definir a área dos equipamentos**", ou seja, onde estes deverão ser colocados e, depois desta definição, atualizar a relação dos equipamentos disponíveis na fábrica, todo o processo demora no máximo 5 dias.

No caso dos funcionários contratados, no primeiro dia de trabalho destes, a Gerência de Pessoal deve "**Definir a área dos funcionários**", isto é, onde estes funcionários serão lotados, considerando as necessidades apresentadas pelas próprias áreas. Após os funcionários terem sido designados, o Supervisor da Área será encarregado de outro centro de decisão, o qual tem como missão "**Definir as atividades dos funcionários**" dentro da área e atualizar a relação dos funcionários existentes complementando a sua ficha com as funções de sua responsabilidade, isto consome em média em 1 dia.

No caso de existirem recursos que devem ser dispensados, então a Gerência da Produção é encarregada de decidir sobre "**Vender equipamentos**", necessitando para tomar esta decisão, de verificar a relação dos equipamentos dispensáveis, consultar a relação dos equipamentos disponíveis e relacionar os possíveis interessados em efetuar a compra. Após a realização da venda, o que ocorre em média num prazo de 60 dias, é necessário que sejam atualizadas a relação dos equipamentos que ainda faltam ser dispensados e a relação dos equipamentos disponíveis na fábrica.

No caso da decisão ser a de "**Demitir funcionários**", esta é tomada pela Gerência de Pessoal, a qual se baseia na relação dos funcionários dispensáveis definida anteriormente e na relação de funcionários da fábrica. Este centro de decisão informa os dados dos funcionários demitidos e atualiza a relação dos funcionários, além de atualizar a relação dos funcionários que ainda devem ser dispensados. Todo o processo para a demissão de um funcionário leva em torno de 30 dias.

O último centro de decisão enfocado no estudo de caso é "**Emitir a Programação da Produção**", o qual é realizado pelo Setor de Programação e Controle da Produção (PCP) da Área. Este centro define e corrige, se necessário, a Programação da Produção e emite as Ordens de Produção para a fábrica, conforme a relação dos equipamentos disponíveis, a relação dos funcionários disponíveis e a lista de pedidos pendentes, emitida pelo Setor de

Vendas. A Programação da Produção é preparada em 2 dias e é definida para a próxima semana.

Devido à grande extensão do modelo que irá resultar deste estudo de caso, o mesmo não será desenvolvido completamente neste trabalho. Portanto foi escolhido um centro de decisão da estrutura para que este seja detalhado na próxima seção. Em função da sua complexidade foi escolhido para este detalhamento, o centro de decisão "Comprar equipamento".

TABELA 5.1 - Estudo de Caso.

Entradas	Centro de Decisão	Responsável	Duração (B)	Saídas
<ul style="list-style-type: none"> • Pedido • Indicadores de Demanda do Mercado • Lista de pedidos aguardando 	Aceitar Novos Pedidos	Gerência de Vendas	1 dia	<ul style="list-style-type: none"> • Previsão de Demanda • Lista de Pedidos aguardando atualizada
<ul style="list-style-type: none"> • Previsão de Demanda • Lista de pedidos aguardando • Verba Orçada • Indicadores de desempenho da produção 	Determinar a Capacidade de Produção	Gerência da Produção	5 dias	<ul style="list-style-type: none"> • Variação da Capacidade de Produção
<ul style="list-style-type: none"> • Objetivos e Metas da Empresa 	Definir Orçamento Anual	Diretoria	15 dias	<ul style="list-style-type: none"> • Orçamento
<ul style="list-style-type: none"> • Variação da Capacidade • Verba Disponível 	Definir Recursos Necessários	Gerência da Produção	30 dias	<ul style="list-style-type: none"> • Variação da Capacidade atualizada • Relação dos Equip. e Funcionários necessários
<ul style="list-style-type: none"> • Variação da Capacidade 	Definir Recursos a serem eliminados	Gerência da Produção	30 dias	<ul style="list-style-type: none"> • Variação da Capacidade atualizada • Relação dos Equip. e Funcionários dispensáveis
<ul style="list-style-type: none"> • Propostas de Fornecedores • Relação dos Equip. Necessários • Verba disponível • Locais disponíveis • Funcionários 	Comprar equipamentos	Gerência da Produção	90 dias	<ul style="list-style-type: none"> • Relação dos Equip. Necessários atualizada • Equip. comprados • Verba disponível atualizada
<ul style="list-style-type: none"> • Candidatos • Relação dos Funcionários Necessários 	Contratar funcionários	Gerência de Pessoal	30 dias	<ul style="list-style-type: none"> • Relação dos Func. Necessários atualizada • Func. contratados

Entradas	Centro de Decisão	Responsável	Duração (B)	Saídas
<ul style="list-style-type: none"> • Compradores • Relação dos Equipamentos Disponíveis • Relação dos Equip. Dispensáveis 	Vender equipamentos	Gerência da Produção	60 dias	<ul style="list-style-type: none"> • Relação dos Equip. Dispensáveis atualizada • Equipamentos vendidos • Relação dos Equip. disponíveis atualizada
<ul style="list-style-type: none"> • Relação de Funcionários • Relação dos Funcionários Dispensáveis 	Demitir funcionários	Gerência de Pessoal	30 dias	<ul style="list-style-type: none"> • Relação dos Func. Dispensáveis atualizada • Relação de Funcionários atualizada • Funcionários demitidos
<ul style="list-style-type: none"> • Equipamentos comprados • Necessidades das áreas 	Definir área dos equipamentos	Gerência da Produção	5 dias	<ul style="list-style-type: none"> • Relação dos Equip. Disponíveis atualizada
<ul style="list-style-type: none"> • Funcionários contratados • Necessidades das áreas 	Definir área dos funcionários	Gerência de Pessoal	1 dia	<ul style="list-style-type: none"> • Funcionários designados
<ul style="list-style-type: none"> • Funcionários designados 	Definir atividades dos funcionários	Supervisor da área	1 dia	<ul style="list-style-type: none"> • Relação de Funcionários atualizada
<ul style="list-style-type: none"> • Relação dos Equipamentos Disponíveis • Relação de Funcionários • Lista de pedidos • Programação da produção 	Emitir Programação da Produção	PCP da área	2 dias	<ul style="list-style-type: none"> • Programação da produção • Ordens de Produção

A tabela acima mostra um resumo onde estão relacionados todos os centros de decisão que fazem parte do estudo de caso, apresentando informações necessárias como: as suas entradas e saídas, seus responsáveis e a duração estimada.

5.1 O Centro de Decisão “Comprar Equipamento”

No centro de decisão a ser detalhado, é executada a compra de uma determinada máquina ou um lote de máquinas com o objetivo de aumentar a capacidade de produção da fábrica. Conforme já foi visto na seção anterior, a execução do processo associado a este centro de decisão depende da existência de verba disponível, de fornecedores para enviar propostas de venda, de funcionários disponíveis para formar a comissão de compra e de locais adequados disponíveis para a instalação dos novos equipamentos. Além de tudo isso, é preciso a “Relação de Equipamentos Necessários”, onde os equipamentos que devem ser adquiridos são definidos, esta relação é elaborada pelo centro de decisão “Definir Recursos Necessários”. Após a efetivação da compra, a relação dos equipamentos necessários é atualizada, pois o material que foi adquirido é retirado da relação, e os equipamentos adquiridos são colocados à disposição do setor de produção.

Em primeiro lugar, analisando este centro de decisão, podemos identificar os seguintes agentes, ou seja, as entidades que tomam decisões: O **Gerente de Produção**, encarregado de iniciar o processo de compra encaminhando ao seu escalão superior os relatórios de necessidades provenientes das áreas de produção e encarregado da coordenação de todo o processo; a **Comissão de Compra** composta por funcionários previamente escolhidos os quais são responsáveis pela operacionalização do processo, isto é, definir com a precisão necessária os equipamentos que devem ser adquiridos, fazer o convite para os possíveis fornecedores e decidir, baseados nas propostas recebidas, o fornecedor escolhido; e os **Fornecedores**, os quais são responsáveis, quando convidados, por definir e enviar a sua proposta de venda e, se solicitado, executar a negociação de alguns itens da proposta.

Também podemos identificar os seguintes recursos envolvidos na execução deste centro: os **Equipamentos** que são objetos da compra, os **Funcionários** alocados para formar a comissão de compra, a qual é responsável pela operacionalização do processo, os **Locais** disponíveis para instalação dos novos equipamentos e a **Verba** disponível para a aquisição.

No processo de aquisição de uma máquina podemos identificar cinco tarefas diferentes. A primeira, chamada **Especificação**, onde é criado um documento com a definição de todas as características necessárias aos equipamentos os quais serão adquiridos, este documento é enviado aos fornecedores convidados para que estes possam confeccionar as suas propostas. A **Licitação**, onde é escolhida a melhor proposta e, após, é feita a negociação de itens desta proposta de forma que as condições sejam mais adequadas conforme às pretensões do comprador. O **Pagamento**, onde a verba necessária, conforme o valor e os prazos acertados, é liberada para ser repassada ao fornecedor. O **Recebimento**, onde os equipamentos são recebidos pelo comprador e são feitas as instalações destes nos locais escolhidos e, por último, são realizados os **Testes** para a aceitação dos equipamentos, onde é emitido o parecer técnico informando o resultado dos testes realizados, sendo que no caso dos resultados serem positivos, a relação dos equipamentos necessários é atualizada e, em caso negativo, a troca do equipamento é providenciada. Baseado na experiência de compras executadas, estima-se que ao todo o processo de aquisição leva em torno de 90 dias, portanto, uma vez tomada a decisão de adquirir um novo equipamento, este só estará disponível na linha de produção aproximadamente 90 dias depois.

O processo é iniciado quando o Gerente de Produção toma a decisão de enviar ao seu escalão superior um relatório informando as necessidades das áreas em termos de equipamentos. A seguir, aguarda a autorização para iniciar o processo de aquisição. No momento em que a autorização é recebida, o objeto da compra é descrito, juntamente com a definição das datas para o início do processo, para a sua conclusão e o limite de verba disponível para a compra.

Autorizada a compra, o Gerente inicia a primeira fase do processo, dita fase de Especificação dos equipamentos. A primeira tarefa é solicitar a lista dos funcionários disponíveis para compor a comissão de compra. Criada esta comissão, o objeto da compra é especificado e um documento com a descrição das características a serem observadas para o fornecimento do equipamento é gerado e enviado para três fornecedores escolhidos previamente a partir do cadastro de fornecedores da empresa. Também neste momento, caso seja necessário devido ao porte do equipamento, será especificado o local onde o mesmo será instalado após a aquisição.

Conforme o prazo definido, os fornecedores encaminham as suas propostas com as informações sobre os equipamentos os quais fornecem e que se enquadram na especificação criada pela comissão, indicando as características destes equipamentos, vantagens, preço, condições de pagamento e os prazos de entrega.

Com o recebimento das propostas, inicia-se a segunda fase, chamada Licitação. Após terem sido recebidas as três propostas, a comissão faz a avaliação das mesmas e escolhe o fornecedor cuja a proposta seja a melhor conforme os itens avaliados. O fornecedor vencedor será consultado sobre a possibilidade de negociação das condições de pagamento e dos prazos de entrega.

Após a negociação, o Gerente de Produção é informado e este autoriza a emissão de uma "Ordem de Fornecimento". Se necessário, ainda na fase de Licitação, será solicitada a preparação do local onde será instalado o novo equipamento.

Com a emissão da ordem de fornecimento, é iniciada a fase de Pagamento, onde é feito o pedido da liberação da verba pelo Gerente de Produção e o envio do valor ao fornecedor. Efetuado o pagamento, o fornecedor deverá fazer a entrega do equipamento dentro do prazo indicado na proposta. No momento do Recebimento são observadas se as características do equipamento entregue estão de acordo com as descritas na proposta. A instalação só é realizada após o local ter sido preparado.

Feita a entrega do equipamento, inicia-se a última fase, que é a fase de Testes, onde serão verificadas as condições do equipamento entregue, verificando se tudo esta de acordo com as especificações colocadas na proposta. Se existir algum problema, o fornecedor será avisado e, este deverá recolher o equipamento com defeito e colocar um substituto em condições num determinado prazo. Caso contrário, simplesmente será enviado um comunicado ao fornecedor informando o recebimento do equipamento em perfeitas condições.

No final, o Gerente de Produção será informado de que o processo foi concluído e que o equipamento está à disposição da área de produção.

6 Aplicação da Técnica ao Estudo de Caso

Para demonstrar de forma prática o desenvolvimento de um modelo utilizando a técnica proposta, será apresentado nesta seção o exemplo da sua aplicação ao estudo de caso descrito no capítulo anterior. No primeiro nível deste exemplo é mostrada a estrutura de decisão completa associada ao estudo de caso, porém, a partir do nível 2, para que este exemplo não se torne muito extenso, é apresentado somente uma parte do modelo que corresponde ao refinamento do centro de decisão "Comprar equipamentos". Este é refinado passo a passo até chegar a sua representação através do modelo TF-ORM.

6.1 Especificação do nível 1 com Redes de Petri - Centros de Decisão

Através da análise do estudo de caso descrito, é possível criar a tabela abaixo com os seguintes centros de decisão. Como esta tabela servirá de base para o desenvolvimento do modelo, são apresentadas as propriedades do centro de decisão necessárias para a construção do mesmo conforme foi definido no passo 1 da técnica.

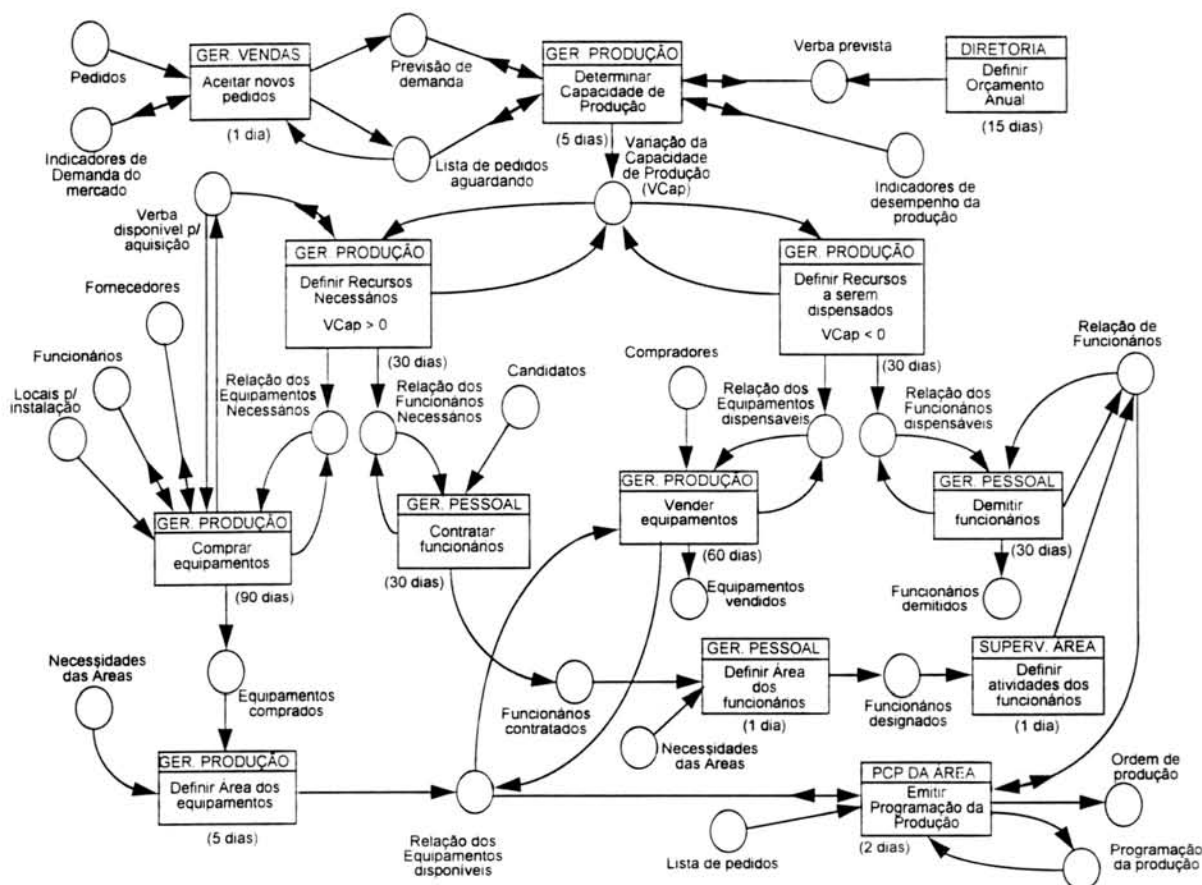
TABELA 6.1 - Nível 1: Centros da Estrutura de Decisão.

CENTRO DE DECISÃO	RESPONSÁVEL	DURAÇÃO (B)	AGENTES	RECURSOS	REGRAS DE TRANSIÇÃO
Aceitar novos pedidos	Gerente de Vendas	1 dia	<ul style="list-style-type: none"> Gerente de Vendas 	<ul style="list-style-type: none"> Pedidos Indicadores de Demanda do mercado (info) Lista dos pedidos aguardando (info) Previsão de demanda (info) 	<ul style="list-style-type: none">
Determinar a capacidade de produção	Gerente de Produção	5 dias	<ul style="list-style-type: none"> Gerente de Produção 	<ul style="list-style-type: none"> Previsão de demanda (info) Lista de pedidos aguardando (info) Verba prevista no orçamento anual (info) Indicadores de desempenho da produção (info) 	<ul style="list-style-type: none">
Definir o orçamento anual	Diretoria	15 dias	<ul style="list-style-type: none"> Diretoria 	<ul style="list-style-type: none"> Previsão verba para investimentos na produção (info) 	<ul style="list-style-type: none">
Definir recursos necessários	Gerente de Produção	30 dias	<ul style="list-style-type: none"> Gerente de Produção 	<ul style="list-style-type: none"> Verba disponível p/ aquisição (info) Variação da Capacidade de Produção (info) Relação dos equipamentos necessários (info) Relação dos funcionários necessários (info) 	<ul style="list-style-type: none"> $V_{Cap} > 0$

CENTRO DE DECISÃO	RESPONSÁVEL	DURAÇÃO (B)	AGENTES	RECURSOS	REGRAS DE TRANSIÇÃO
Comprar equipamentos	Gerente de Produção	90 dias	<ul style="list-style-type: none"> • Gerente de Produção • Fornecedores 	<ul style="list-style-type: none"> • Funcionários • Locais p/ instalação • Equipamentos comprados • Verba disponível p/ aquisição • Relação dos equipamentos necessários (info) 	•
Contratar funcionários	Gerente de Pessoal	30 dias	<ul style="list-style-type: none"> • Gerente de Pessoal • Candidatos 	<ul style="list-style-type: none"> • Funcionários contratados • Relação dos funcionários necessários (info) 	•
Definir recursos a serem dispensados	Gerente de Produção	30 dias	<ul style="list-style-type: none"> • Gerente de Produção 	<ul style="list-style-type: none"> • Variação da Capacidade de Produção (info) • Relação dos equipamentos dispensáveis (info) • Relação dos funcionários dispensáveis (info) 	• VCap < 0
Vender equipamentos	Gerente de Produção	60 dias	<ul style="list-style-type: none"> • Gerente de Produção • Compradores 	<ul style="list-style-type: none"> • Relação dos equipamentos dispensáveis (info) • Equipamentos 	•
Demitir funcionários	Gerente de Pessoal	30 dias	<ul style="list-style-type: none"> • Gerente de Pessoal 	<ul style="list-style-type: none"> • Relação dos funcionários (info) • Relação dos funcionários dispensáveis (info) • Funcionários demitidos 	•
Definir área dos equipamentos	Gerente de Produção	5 dias	<ul style="list-style-type: none"> • Gerente de Produção 	<ul style="list-style-type: none"> • Necessidades das áreas (info) • Equipamentos comprados • Relação dos equipamentos disponíveis (info) 	•
Definir área dos funcionários	Gerente de Pessoal	1 dia	<ul style="list-style-type: none"> • Gerente de Pessoal 	<ul style="list-style-type: none"> • Funcionários contratados • Necessidades das áreas (info) • Funcionários designados 	•
Definir atividades dos funcionários	Supervisor de Área	1 dia	<ul style="list-style-type: none"> • Supervisor de Área 	<ul style="list-style-type: none"> • Funcionários designados • Relação de funcionários (info) 	•
Emitir programação da produção	PCP da área	2 dias	<ul style="list-style-type: none"> • PCP da área 	<ul style="list-style-type: none"> • Relação dos equipamentos disponíveis (info) • Lista de pedidos (info) • Relação de funcionários (info) • Ordem de produção • Programação da Produção (info) 	•

A partir das seguintes informações necessárias: os responsáveis pelas decisões, o tempo necessário para que as mesmas tenham efeito, os agentes e os recursos envolvidos e as regras que devem ser respeitadas para que os processos decisórios possam ser iniciados, com todas estas informações reunidas na tabela e a descrição de como estes recursos e agentes estão envolvidos com a decisão, é possível gerar o modelo com a utilização das Redes de Petri. O modelo criado mostra claramente como os centros de decisão estão relacionados através dos recursos envolvidos, as dependências existentes entre eles, formando a estrutura de decisão.

FIGURA 6.1 - Modelo Nível 1 - Estrutura de Decisão



6.2 Especificação do nível 2 com Redes de Petri - Tarefas

A partir do modelo da estrutura de decisão, é executado o primeiro refinamento, o qual corresponde ao passo 2 da técnica e consiste em identificar as tarefas associadas a cada centro de decisão apresentado no modelo do nível 1. Para isso, serão criados vários modelos separados, um para cada centro de decisão identificado anteriormente.

Para que este exemplo não se torne muito extenso, serão mostradas apenas as tarefas ligadas ao centro de decisão "Comprar equipamentos". Em um primeiro momento, todas as tarefas são listadas em uma tabela juntamente com todas as informações necessárias para criação do modelo com Redes de Petri.

Como foi visto na descrição do estudo de caso, o centro de decisão "Comprar equipamentos" é composto pelas seguintes fases: especificação dos equipamentos a serem adquiridos, a licitação para escolha da melhor proposta, o pagamento do equipamento, o recebimento e fase de

testes. Cada uma destas fases é considerada uma tarefa. Na tabela abaixo encontramos reunidas as informações necessárias para a criação do modelo.

TABELA 6.2 - Nível 2: Tarefas de "Comprar equipamentos".

TAREFA	DURAÇÃO (B)	AGENTES	RECURSOS
Especificação	15 dias	<ul style="list-style-type: none"> • Ger. de Produção • Comissão de Compra • Fornecedores 	<ul style="list-style-type: none"> • Funcionários • Locais disponíveis • Rel. Equipamentos necessários (info)
Licitação	45 dias	<ul style="list-style-type: none"> • Ger. de Produção • Comissão de Compra • Fornecedores 	<ul style="list-style-type: none"> • Locais definidos
Pagamento	5 dias	<ul style="list-style-type: none"> • Ger. de Produção • Fornecedor vencedor 	<ul style="list-style-type: none"> • Verba disponível
Recebimento	5 dias	<ul style="list-style-type: none"> • Ger. de Produção • Fornecedor 	<ul style="list-style-type: none"> • Especificação do Equipamento • Local pronto
Teste	20 dias	<ul style="list-style-type: none"> • Ger. de Produção • Comissão de Compra • Fornecedor 	<ul style="list-style-type: none"> • Equipamento instalado • Rel. Equipamentos necessários (info)

A partir das informações sobre o tempo necessário para a execução da tarefa e os agentes e recursos envolvidos é gerado o modelo que mostra as relações existentes entre as tarefas executadas no centro de decisão "Comprar equipamentos".

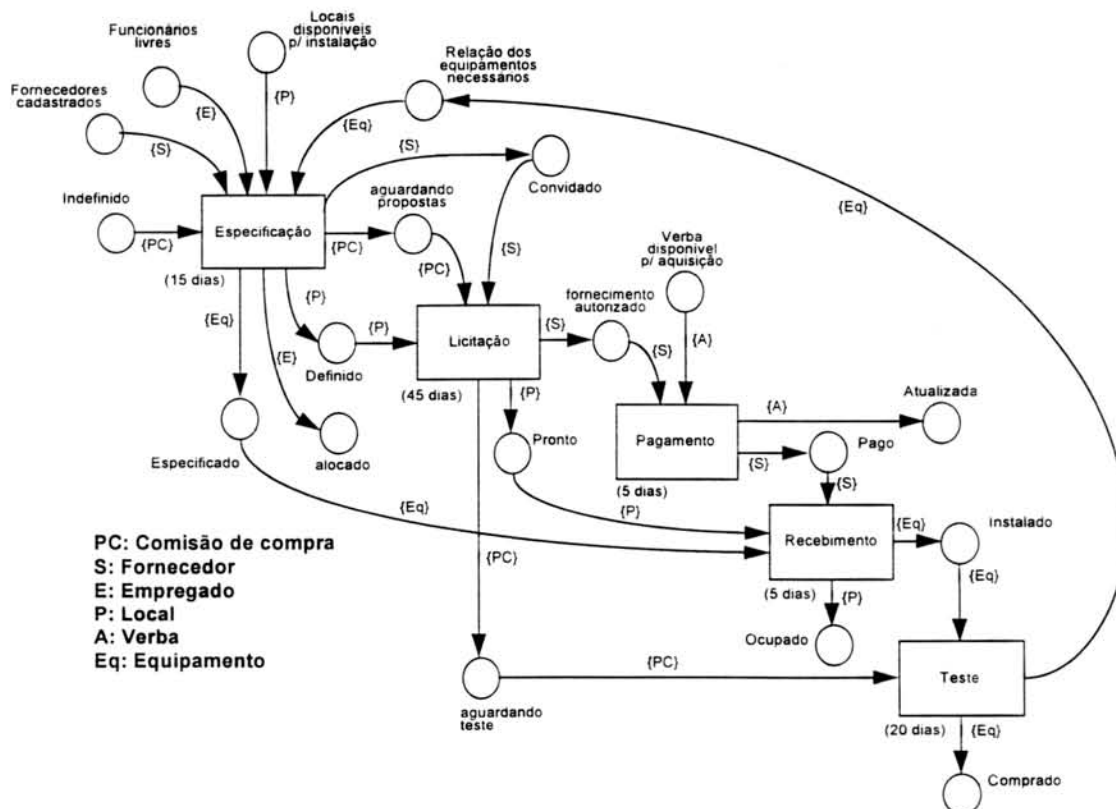


FIGURA 6.2 - Modelo Nível 2 - Tarefas do centro de decisão "Comprar equipamentos"

Desta forma, podemos identificar através do modelo a seqüência em que estas tarefas são executadas e as situações em que são executadas

em paralelo. No caso do exemplo, todas as tarefas são seqüenciais. Os tempos de duração estimados para cada tarefa são utilizados para validar o tempo de duração total estimado para o centro de decisão. O modelo da figura 6.2 descreve as tarefas do centro de decisão.

6.3 Especificação do nível 3 com Redes de Petri - Ações

Terminados os modelos que especificam as tarefas, inicia-se a construção do segundo nível de refinamento, o qual corresponde ao passo 3 da técnica e onde são identificadas as ações que compõem cada uma das tarefas descritas nos modelos do nível 2. Para isso serão criados outros modelos, cada um associado a uma tarefa representada no nível de refinamento anterior.

Por exemplo, a tarefa de especificar os equipamentos a serem adquiridos é composta das seguintes ações: a autorização do início do processo, a solicitação de uma lista de funcionários para compor a comissão encarregada de operacionalizar a compra, a definição desta comissão a partir da lista recebida, a definição de um local onde deverão ser instalados os equipamentos adquiridos, o envio da descrição do objeto da compra para a comissão, a especificação dos equipamentos feita pela comissão, a solicitação e o recebimento da lista de fornecedores capacitados, o envio da especificação para os fornecedores escolhidos e, caso aconteça, o cancelamento da autorização de compra. A tabela abaixo apresenta, de forma resumida, as ações associadas a cada uma das tarefas identificadas no centro de decisão "Comprar equipamentos".

TABELA 6.3 - Relação das ações por tarefa.

TAREFA	AÇÕES
Especificação	<ul style="list-style-type: none"> • Autorizar o início do processo de compra. • Solicitar lista de funcionários para comissão. • Definir Comissão de Compra. • Definir local para a instalação do equipamento. • Enviar a descrição da compra para a comissão. • Especificar equipamento. • Solicitar lista de fornecedores. • Receber a lista dos fornecedores. • Enviar especificação para os fornecedores. • Cancelar autorização de compra.

TAREFA	AÇÕES
Licitação	<ul style="list-style-type: none"> • Iniciar licitação. • Cancelar autorização de compra. • Receber as propostas. • Preparar local para a instalação do equipamento. • Escolher a melhor proposta. • Solicitar a negociação dos itens da proposta escolhida. • Não solicitar negociação. • Aceitar negociação. • Não aceitar negociação. • Enviar ordem de fornecimento. • Encerrar a licitação.
Pagamento	<ul style="list-style-type: none"> • Iniciar pagamento. • Cancelar autorização de compra. • Solicitar liberação da verba. • Receber verba liberada. • Pagar fornecedor escolhido.
Recebimento	<ul style="list-style-type: none"> • Iniciar recebimento. • Receber equipamento. • Instalar equipamento.
Teste	<ul style="list-style-type: none"> • Iniciar testes. • Informar resultados dos testes ok. • Informar problemas encontrados. • Solicitar troca do equipamento.

Nas duas próximas tabelas estão relacionadas as informações necessárias sobre as ações associadas às duas primeiras tarefas do centro de decisão, “Especificação” e “Licitação”, isto porque, para exemplificar a aplicação da técnica neste passo sem estender muito o trabalho, serão apresentados somente os modelos destas duas tarefas. As informações contidas nas tabelas a seguir são: o tempo necessário para realizar a ação, os recursos e os agentes envolvidos na sua execução e as regras que devem ser respeitadas para que a ação possa ser executada.

TABELA 6.4 - Nível 3: Ações da Tarefa “Especificação”.

AÇÃO	DURAÇÃO (B)	RECURSOS	AGENTES	REGRAS
Autorizar o início do processo de compra	1 dia	-	Gerente de Produção	-
Solicitar lista de funcionários para comissão	1 dia	-	AGENTE EXTERNO	-
Definir Comissão de Compra	2 dias	Funcionário	AGENTE EXTERNO e Comissão de Compra	-
Definir local para a instalação do equipamento	1 dia	Local	AGENTE EXTERNO	-
Enviar a descrição da compra para a comissão	1 dia	-	AGENTE EXTERNO e Comissão de Compra	-
Especificar equipamento	3 dias	Equipamento	Comissão de Compra	-
Solicitar Lista de fornecedores	2 dias	-	AGENTE EXTERNO	-

AÇÃO	DURAÇÃO (B)	RECURSOS	AGENTES	REGRAS
Receber a lista dos fornecedores	2 dias	-	AGENTE EXTERNO	3 fornecedores distintos
Enviar especificação para os fornecedores	2 dias	-	Fornecedor	-
Cancelar autorização de compra	zero	-	Gerente de Produção	-

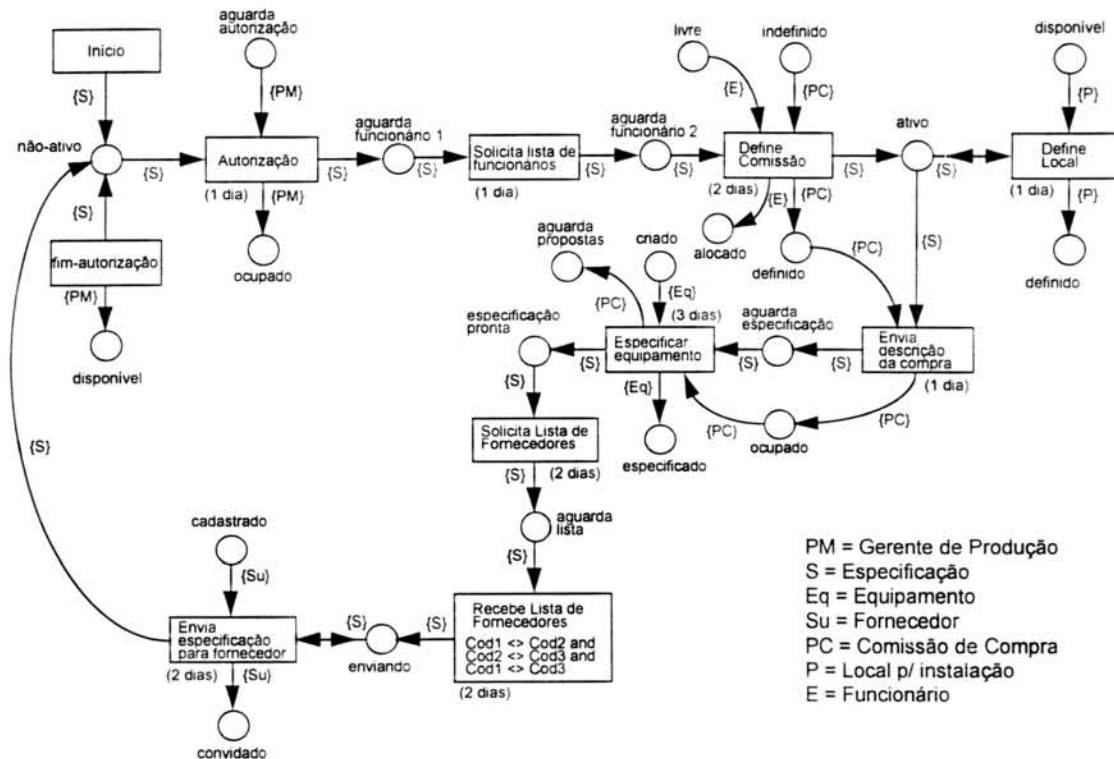


FIGURA 6.3 - Modelo Nivel 3 - Ações da tarefa "Especificação"

A partir das informações reunidas na tabela podemos construir o modelo com Redes de Petri. Observe que o "AGENTE EXTERNO" refere um objeto que não faz parte do escopo do modelo e, embora seja indicado na tabela, não é representado no modelo.

TABELA 6.5 - Nivel 3: Ações da Tarefa "Licitação".

AÇÃO	DURAÇÃO (B)	RECURSOS	AGENTES	REGRAS
Iniciar licitação	1 dia	-	-	-
Receber as propostas	10 dias	-	Fornecedor e Comissão de Compra	3 fornecedores distintos
Preparar local p/ a instalação do equipamento	6 dias	Local	-	-
Escolher a melhor proposta	10 dias	-	Fornecedor e Comissão de Compra	Antes da data limite

AÇÃO	DURAÇÃO (B)	RECURSOS	AGENTES	REGRAS
Solicitar negociação dos itens da proposta escolhida	10 dias	-	Fornecedor e Comissão de Compra	-
Não solicitar negociação	3 dias	-	Comissão de Compra e Gerente de Produção	-
Aceitar negociação	3 dias	-	Fornecedor e Gerente de Produção	Novas Condições = Condições Solicitadas
Não aceitar negociação	2 dias	-	Fornecedor	Novas Condições <> Condições Solicitadas e Razão <> Nulo
Enviar ordem de fornecimento	4 dias	-	Fornecedor e Gerente de Produção	-
Encerra Licitação	1 dia	-	-	-
Cancelar autorização de compra	zero	-	Gerente de Produção	-

A partir das informações colocadas na tabela é criado o modelo com Redes de Petri. Após a criação do modelo é possível observar como estão relacionadas as ações de uma mesma tarefa e, e além disso, definir quais os estados que podem ser alcançados pela tarefa e pelos agentes e recursos envolvidos durante a execução da tarefa. Para exemplificar, observe o modelo da figura 6.4 a seguir, ele representa a tarefa "Licitação" e mostra que esta tarefa, representada pelo objeto "A", pode assumir os seguintes estados: não ativo, ativo, aguardando resultados, em negociação, aguardando resposta, aguardando Ordem de Fornecimento, aguardando preparação e finalizando.

Ainda com relação ao modelo de "Licitação", podemos observar os estados que os agentes e recursos podem assumir, como exemplo podemos citar os estados assumidos pelo agente "Comissão de Compra" (PC), são eles: aguardando propostas, avaliando, em negociação e aguardando teste.

Com relação à duração da tarefa "Licitação", é possível ver que, considerando que nem todas as ações representadas serão executadas, a duração equivale à soma das durações de todas as ações que compõem o roteiro completo, isto é, com a execução de uma negociação com o primeiro fornecedor escolhido.

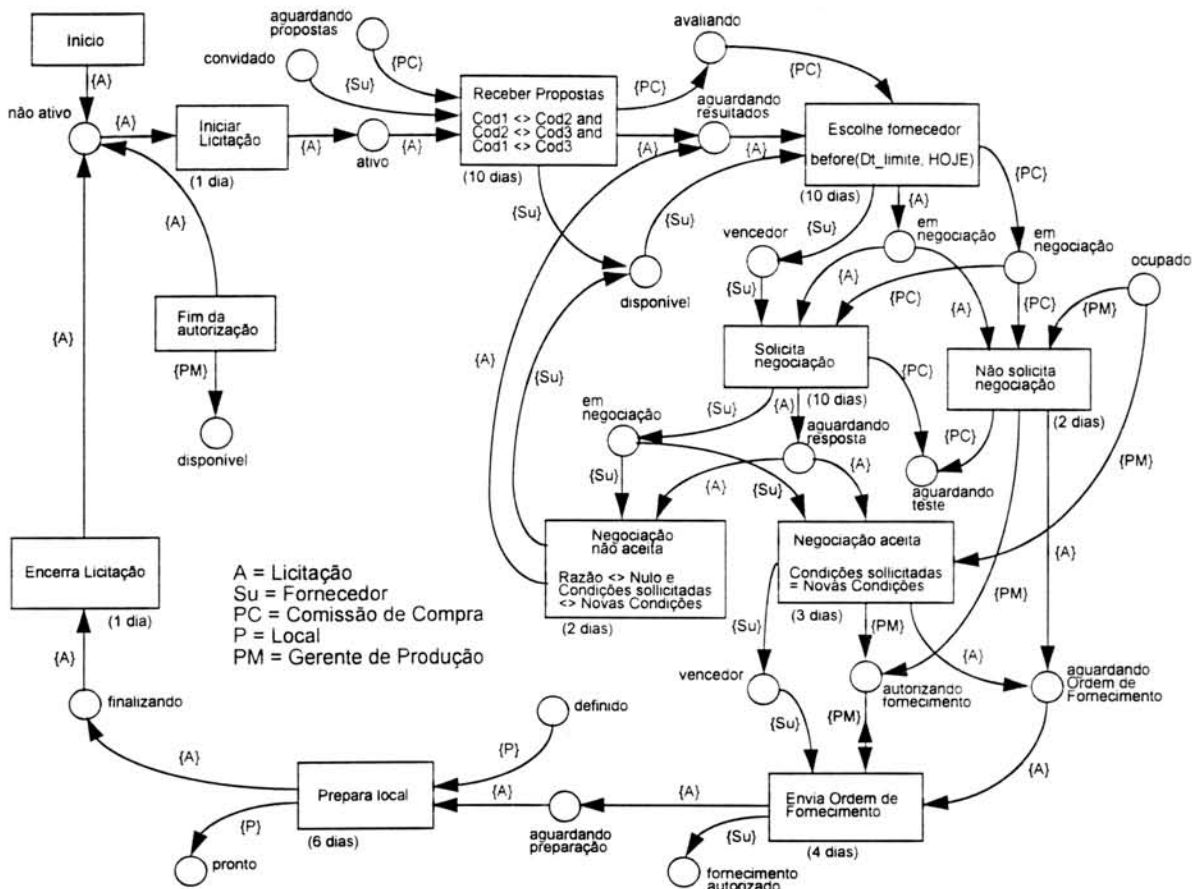


FIGURA 6.4 - Modelo Nível 3 - Ações da tarefa "Licitação"

6.4 Especificação do nível 4 com Redes de Petri - Operações

Depois de criados os modelos da estrutura de decisão, dos centros de decisão e das tarefas, agora é executado o último passo da técnica antes de criar o modelo TF-ORM. Neste passo são identificadas as operações que compõem as ações.

Como já foi dito na descrição da técnica proposta no capítulo 4, as operações correspondem às mensagens que servem como meio de comunicação entre os processos e os agentes e recursos. Portanto nesta etapa, a qual corresponde ao passo 4 da técnica, devem ser identificadas as seguintes propriedades para cada operação: Os agentes e/ou recursos ligados à operação, a forma como o agente ou recurso esta ligado, ou seja, através do envio ou recebimento de uma mensagem e a informação associada.

Os agentes enviam mensagens quando são responsáveis pela execução das operações e, desta forma, desencadeiam a execução das ações. Quando os agentes recebem mensagens, significa que estes recebem informações que resultam da execução das ações. Já os recursos recebem mensagens quando a execução de uma ação solicita alguma informação sobre ele ou provoca uma alteração do status do recurso. Quando uma ação solicita uma informação, o recurso deve retornar uma mensagem para a ação com a

informação solicitada. Como exemplo, observe a tabela abaixo onde estão relacionadas as informações necessárias sobre as operações identificadas na tarefa "Especificação".

TABELA 6.6 - Nível 4: Operações associadas à ações da tarefa "Especificação".

AÇÃO	OPERAÇÕES	QUEM EXECUTA? SOBRE QUE(M)?	INFORMAÇÃO ASSOCIADA
• Autorizar o início do processo de compra	• Autorização de início	• Gerente de Produção (envia)	• Descrição da compra, data de início, verba máxima disponível, prazo para compra e prazo da licitação
• Solicitar lista de funcionários para comissão	• Solicitação de alocação de funcionário	• Agente Externo (envia)	• Data de início e data de fim da alocação
• Definir Comissão de Compra	• Recebe lista dos funcionários disponíveis • Ordem de alocação • Definição da comissão	• Agente Externo (envia) • Funcionários escolhidos (recebe) • Comissão de Compra definida (recebe)	• Componentes • Matrícula do funcionário, função na comissão e data de início da alocação • Componentes, data de início e data de fim da comissão
• Definir local para a instalação do equipamento	• Recebe a lista dos locais disponíveis • Definição do local	• Agente Externo (envia) • Local escolhido (recebe)	• Localização, área e características técnicas • Localização, área e características técnicas
• Enviar a descrição da compra para a comissão	• Recebe a lista dos equipamentos necessários • Informa objeto da compra	• Agente Externo (envia) • Comissão de Compra (recebe)	• Equipamentos necessários • Descrição do objeto para compra
• Especificar equipamento	• Recebe a especificação • Definição da especificação	• Comissão de Compra (envia) • Equipamento especificado (recebe)	• Descrição, características e funções do equipamento • Descrição, características e funções do equipamento
• Solicitar Lista de fornecedores	• Envia requisição da lista de fornecedores	• Agente Externo (recebe)	• -
• Receber lista de fornecedores	• Recebe a lista dos fornecedores	• Agente Externo (envia)	• Três códigos de fornecedores diferentes
• Enviar especificação para os fornecedores	• Envia especificação do equipamento	• Fornecedor (recebe)	• Código do fornecedor, descrição, características e funções do equipamento
• Cancelar autorização de compra	• Fim da autorização	• Gerente de Produção (envia)	• Razão do cancelamento

Observando a tabela acima podemos ver que, por exemplo, a ação "Autorizar o início do processo de compra" é composta apenas pela operação "Autorização de início", a qual é executada pelo Gerente de

Produção e através desta são passadas as informações sobre a descrição do objeto da compra, a data de início, a verba máxima disponível, o prazo para compra e prazo da licitação.

Já no caso da ação “Definir Comissão de compra” podemos identificar três operações: “Recebe lista dos funcionários disponíveis”, “Ordem de alocação” e a “Definição da comissão”, onde a primeira é executada por um “Agente Externo” que envia uma relação dos possíveis componentes desta comissão, a segunda envia para os funcionários escolhidos as informações sobre a matrícula do funcionário, a sua função na comissão e data a partir da qual ele estará à disposição da comissão, a última operação, de “Definição da comissão”, define uma instância do objeto “Comissão de compra” e envia os dados dos componentes que a formarão e as data de início e fim da comissão.

No caso deste exemplo, as operações identificadas dispensam a modelagem com Redes de Petri por serem bastante simples.

6.5 Especificação com o modelo TF-ORM

O último passo da técnica é a conversão dos modelos em Rede de Petri para um modelo utilizando TF-ORM. Reunindo todos os elementos identificados nos modelos criados nos passos anteriores é gerado um modelo único em TF-ORM através do mapeamento destes elementos.

O modelo abaixo mostra, como exemplo, a especificação apenas do centro de decisão “Comprar Equipamentos”. Analisando os modelos obtidos nos passos anteriores pode-se identificar as seguintes classes de objetos: **Agentes** - Tomador de Decisão e Fornecedor; **Recursos** - Funcionários, Equipamentos, Locais e Verba; e o **Processo** de Compra de um equipamento.

Realizando o mapeamento dos modelos obtidos nos passos anteriores para o modelo TF-ORM, no exemplo do centro de decisão “Comprar equipamento”, conforme a definição do passo 5, podemos citar algumas correspondências:

1. O centro de decisão “**Comprar equipamento**” dá origem à classe processo “**Equipment_purchase_fases**”;
2. as tarefas “**Especificação**”, “**Licitação**”, “**Pagamento**”, “**Recebimento**” e “**Teste**” dão origem, respectivamente, aos papéis “**Specification**”, “**Auction**”, “**Payment**”, “**Reception**” e “**Test**” da classe processo “**Equipment_purchase_fases**”;
3. as ações associadas à tarefa “**Especificação**”, como por exemplo: “**Autorizar o início do processo de compra**”, “**Cancelar a autorização de compra**”, “**Solicitar lista de funcionários para a comissão**” e “**Definir Comissão de compra**” são mapeadas, respectivamente, como as regras de

transição "Authorization", "End_authorization", "Request_employee_list" e "Define_purchase_commission" do papel "Specification";

4. as operações associadas às ações da tarefa "Especificação", como por exemplo: "Autorização de início", "Fim da autorização", "Solicitação de alocação de funcionário", "Recebe lista dos funcionários disponíveis", "Ordem de alocação" e "Definição da comissão" dão origem, respectivamente, às mensagens "Begin_authorization", "End_authorization", "Request_employee", "Employee_list", "Allocate_order" e "Define_commission" do papel "Specification"; e

5. os agentes "Gerente de produção" e "Comissão de compra" dão origem aos papéis "Production_manager" e "Purchase_commission" da classe do tipo agente "Decision_maker", o agente "Fornecedor" dá origem ao papel "Supplier" da classe "Supplier", os recursos "Funcionário", "Equipamento", "Local" e "Verba" são mapeados, respectivamente, para papéis das classes "Person", "Equipment", "Place" e "Cash".

A seguir está a relação que descreve as classes de objetos identificados no modelo juntamente com os seus papéis:

Process class = (EQUIPMENT_PURCHASE_FASES, Specification, Auction, Payment, Reception, Test)

Agent class = (DECISION_MAKER, Production_manager, Purchase_commission)

Agent class = (SUPPLIER, Supplier)

Resource class = (PERSON, Employee)

Resource class = (EQUIPMENT, Equipment)

Resource class = (PLACE, Place)

Resource class = (CASH, Allocation)

No anexo 1 deste trabalho tem-se o modelo TF-ORM completo que descreve o centro de decisão "Comprar equipamentos".

7 Conclusões

O bom gerenciamento de uma empresa é uma função fundamental nos dias de hoje. Uma decisão tomada no momento certo pode significar o sucesso ou o fracasso de um empreendimento em muitas situações e, portanto, este processo é considerado como um ponto crítico para obtenção de vantagens na competição pelo mercado. Por este motivo é dada muita atenção ao aperfeiçoamento da função de gerenciamento no sentido de melhorar a qualidade dos processos decisórios envolvidos através da sua melhor integração.

Mas para que isto seja possível é importante que o analista de negócios disponha de uma ferramenta que permita criar um modelo da Estrutura de Decisão da empresa, identificando as pessoas e os recursos envolvidos, assim como as variáveis de tempo associadas a cada uma das decisões desta estrutura.

A técnica proposta neste trabalho vem ao encontro da solução para esta necessidade, pois através de uma seqüência de etapas de refinamento, a técnica permite que, a partir de um estudo inicial realizado junto ao usuário, com a utilização de ferramentas gráficas de modelagem de fácil entendimento, se chegue a um modelo que permita representar as informações necessárias com um grau de detalhamento adequado de forma que, a partir deste seja possível implementar diretamente um sistema de informações que dê suporte à função de gerenciamento.

Para isto são utilizadas as vantagens da representação gráfica das Redes de Petri, o que vem a facilitar o diálogo com o usuário, e da completude do modelo TF-ORM, o qual permite implementar o sistema de informação diretamente a partir da especificação gerada. Uma vez criado o primeiro modelo com TF-ORM, é possível ainda que a especificação seja refinada, conforme mostra [EDE 96], de forma sempre a garantir o nível de detalhamento desejado. Outra vantagem do uso do modelo TF-ORM é a possibilidade deste servir como integrador de ferramentas, o que permite gerar modelos ligados a um mesmo escopo utilizando mais de uma ferramenta de modelagem de forma que as propriedades do sistema modeladas em uma venha a complementar as propriedades obtidas na outra e, após, tudo possa ser reunido através da representação em TF-ORM como é mostrado em [EDE 96]. Com isso é possível que seja utilizada outra ferramenta, como por exemplo, modelo E-R, para identificar as estruturas de dados, as quais não são representadas com as Redes de Petri.

É importante observar que existe uma restrição nos modelos TF-ORM gerados com a utilização desta técnica no que diz respeito à troca de mensagens, pois estas devem ser trocadas entre os recursos e agentes sempre através dos processos, ou seja, a técnica não permite que existam trocas de mensagens diretamente entre os agentes e recursos. Isto acontece porque o enfoque principal da técnica é a modelagem dos processos e, a partir

destes, é que são derivados os modelos dos agentes e recursos justamente através das mensagens enviadas e recebidas pelo processo. Portanto, toda mensagem enviada por um papel de processo deve ser recebida por um papel de um recurso ou agente e vice-versa.

Desta forma, a técnica se mostra bastante útil para auxiliar os analistas na análise e projeto de aplicações para suporte às decisões gerenciais.

Através da utilização do estudo de caso genérico para um ambiente de manufatura, como o apresentado no capítulo 5, foi possível verificar a dimensão do trabalho necessário para atingir os objetivos da técnica. Para que se possa ter uma idéia, a estrutura de decisão modelada para este estudo identificou treze centros de decisão, dos quais foi refinado somente o centro de decisão "Comprar equipamentos", no qual foram identificadas cinco tarefas e, ligadas a estas, foram identificados três agentes e quatro recursos. Integrando estas cinco tarefas, foram identificadas, no passo seguinte, trinta e três ações, as quais, após o refinamento, deram origem, considerando somente as duas primeiras tarefas, à trinta e uma operações. No primeiro nível da modelagem foi gerado um modelo que descreve a estrutura de decisão e, a partir deste, seriam necessários, no segundo nível, treze modelos indentificando as tarefas associadas a cada centro de decisão representado. Analisando-se somente o centro de decisão "Comprar equipamentos", seriam necessários cinco modelos representando as ações de cada uma das tarefas. Somente analisando um dos treze centros de decisão da estrutura, o resultado foi o extenso modelo TF-ORM mostrado no anexo 1 deste trabalho.

Portanto, a técnica mostrou resultar na criação de vários modelos com Redes de Petri, e a partir destes, vários modelos bastante extensos em TF-ORM mas com o grau de detalhamento desejado. Por este motivo, torna-se necessário um estudo com o objetivo de desenvolver uma ferramenta que permita a automatização de algumas (ou, se possível, todas) etapas da técnica, principalmente no que diz respeito à derivação dos modelos dos agentes e recursos em TF-ORM junto aos modelos dos processos.

Para continuação deste trabalho seria interessante em primeiro lugar, um estudo mais aprofundado sobre os conceitos de decisão e, após, estudos mais detalhados sobre a modelagem das variáveis de tempo e o envolvimento dos agentes e recursos com os centros de decisão.

Anexo 1 Modelo TF-ORM do Centro de Decisão “Comprar Equipamentos”

Agent Classes

```

Agent class (
  DECISION_MAKER,
  <Base_role,
    static properties = {
      (code, integer),
      (cpf, integer),
      (date_birth, date) },
    dynamic properties = {
      (name, string),
      (address, string),
      (function, string),
      (salary, real) },
    rules = {
      r1: msg(create_object) => state(active),
      r2: state(active) => msg(add_role(Production_manager)),
      r3: state(active) => msg(add_role(Purchase_commission)) }
  >,
  <Production_manager,
    static properties = { },
    dynamic properties = { },
    messages = {
      area_necessary_report(area_code: integer, necessary_description: string)
        to EXTERNAL_WORLD,
      purchase_authorization(purchase_description: string, begin: date, max_allocation:
        real, expiry_time: date)
        from EXTERNAL_WORLD,
      purchase_stoppage(reason: string)
        from EXTERNAL_WORLD,
      begin_authorization(purchase_description: string, begin: date, max_allocation: real,
        expiry_time: date, auction_limit_time: date)
        to EQUIPMENT_PURCHASE_FASES.Specification,
      end_authorization(reason: string)
        to EQUIPMENT_PURCHASE_FASES,
      info_supplier_choice(supplier_winner_code: integer)
        from EQUIPMENT_PURCHASE_FASES.Auction,
      provide_order_authorization
        to EQUIPMENT_PURCHASE_FASES.Auction,
      info_test_result_OK
        from EQUIPMENT_PURCHASE_FASES.Test },
    decision = {
      send_necessary_report(area_code: integer, necessary_description: string) },
    states = {
      available, occupied, wait_authorization, provide_authorized },
    rules = {
      begin:
        msg(add_role) => state(available),
      necessary_report_send:
        state(available),
        decision(send_necessary_report(Area_code, Necessary_description)) =>
        msg(area_necessary_report(Area_code, Necessary_description)),
        state(wait_authorization),
      begin_purchase_authorization:
        state(wait_authorization),
        msg(purchase_authorization(Purchase_description, Begin, Max_allocation,

```

```

    Expiry_time)) =>
    msg(begin_authorization(Purchase_description, Begin, Max_allocation, Expiry_time,
    Auction_limit_time)), state(occupied),
end_purchase_authorization:
    msg(purchase_stoppage(Reason)) =>
    msg(end_authorization(Reason)), state(available); (Reason ≠ Null),
authorize_provide_order:
    state(occupied),
    msg(info_supplier_choice(Supplier_winner_code)) =>
    msg(provide_order_aythorization), state(provide_authorized),
purchase_complete:
    state(provide_authorized), msg(info_test_result_OK) =>
    state(available) }
>,
<Purchase_commission,
dynamic properties = {
    components, set of PERSON.Employee),
    begin_composition_time, date),
    end_composition_time, date) },
messages = {
    define_commission(components: set of PERSON.Employee, begin_composition_time:
    date, end_composition_time: date)
    from EQUIPMENT_PURCHASE_FASES.Specification,
    purchase_info(purchase_description: string)
    from EQUIPMENT_PURCHASE_FASES.Specification,
    specification(description: string, features: string, functions: string)
    to EQUIPMENT_PURCHASE_FASES.Specification,
    proposal_grid(codes: set of integer, descriptions: set of string, product_codes:
    set of integer, advantages: set of string, prices: set of real, pay_conditions: set of
    string, due_times: set of date)
    from EQUIPMENT_PURCHASE_FASES.Auction,
    supplier_winner(supplier_code: integer)
    to EQUIPMENT_PURCHASE_FASES.Auction,
    request_negotiation_authorization
    to EQUIPMENT_PURCHASE_FASES.Auction,
    request_negotiation_not_authorization
    to EQUIPMENT_PURCHASE_FASES.Auction,
    begin_test
    from EQUIPMENT_PURCHASE_FASES.Test,
    test_result(result:string)
    to EQUIPMENT_PURCHASE_FASES.Test },
decision = {
    define_specification(description: string, features: string, functions: string),
    choice_supplier(supplier_code: integer),
    request_negotiation(pay_conditions: strings, due_time: date),
    not_request_negotiation(pay_conditions: strings, due_time: date) },
states = {
    undefined, defined, occupied, evaluated, wait_proposal, in_negotiation, wait_test},
rules = {
    begin:
    msg(add_role) => state(undefined ),
    receive_definition_order:
    state(undefined), msg(define_commission(Components, Begin_composition_time,
    End_composition_time)) =>
    state(defined),
    receive_purchase_description:
    state(defined), msg(purchase_info(Purchase_description)) =>
    state(occupied),
    specify_equipment:
    state(occupied), decision(define_specification(Description, Features, Functions)) =>
    msg(specification(Description, Features, Functions)), state(wait_proposal),

```

```

receive_proposal_grid:
  state(wait_proposal), msg(proposal_grid(Codes, Descriptions, Product_codes,
  Advantages, Prices, Pay_conditions, Due_times)) =>
  state(evaluated),
decide_supplier_winner:
  state(evaluated), decision(choice_supplier(Supplier_code)) =>
  msg(supplier_winner(Supplier_code)), state(in_negotiation),
decide_negotiation:
  state(in_negotiation), decision(request_negotiation(Pay_conditions, Due_time)) =>
  msg(request_negotiation_authorization), state(wait_test),
decide_not_negotiation:
  state(in_negotiation), decision(not_request_negotiation(Pay_conditions, Due_time)) =>
  msg(request_negotiation_not_authorization), state(wait_test),
send_test_result:
  state(wait_test), msg(begin_test) =>
  msg(test_result(Result)), state(undefined) }
> )

```

```

Agent class (
  SUPPLIER,
  <Base_role,
  static properties = {
    (supplier_code, integer),
    (cgc, integer) },
  dynamic properties = {
    (name, string),
    (address, string),
    (product_line, set of string),
    (contact, set of string),
    (phone, set of integer) },
  rules = {
    r1: msg(create_object) => state(active),
    r2: state(active) => msg(add_role(Supplier)) }
>,
  <Supplier,
  dynamic properties = {
    (equipment_description, string),
    (product_code, integer),
    (advantages, string),
    (price, real),
    (pay_conditions, string),
    (due_time, date) },
  messages = {
    equipment_specification(description: string, features: string, functions: string)
      from EQUIPMENT_PURCHASE_FASES.Specification,
    proposal(number: integer, supplier_code: integer, equipment_description: string,
    product_code: integer, advantages: string, price: real, pay_conditions: string,
    due_time: date)
      to EQUIPMENT_PURCHASE_FASES.Auction,
    receive_proposal_OK(number: integer, supplier_code: integer)
      from EQUIPMENT_PURCHASE_FASES.Auction,
    info_supplier_winner(supplier_code: integer, supplier_winner_code: integer)
      from EQUIPMENT_PURCHASE_FASES.Auction,
    provide_order(supplier_code: integer, purchase_description: string)
      from EQUIPMENT_PURCHASE_FASES.Auction,
    request_negotiation(new_pay_conditions: string, new_due_time: date)
      from EQUIPMENT_PURCHASE_FASES.Auction,
    negotiation_reply(final_pay_conditions: string, final_due_time: date)
      to EQUIPMENT_PURCHASE_FASES.Auction,
    negotiation_reply_OK
      from EQUIPMENT_PURCHASE_FASES.Auction,
    negotiation_reply_NOK(reason: string)

```

```

    from EQUIPMENT_PURCHASE_FASES.Auction,
    payment(supplier_winner_code: integer, amount: real)
    from EQUIPMENT_PURCHASE_FASES.Payment,
    delivery(equipment_description: string, product_code: integer, supplier_code:
    integer, delivery_time: date)
    to EQUIPMENT_PURCHASE_FASES.Reception,
    info_problems(problems_description: string)
    from EQUIPMENT_PURCHASE_FASES.Test,
    info_tests_OK(technical_opinion: string)
    from EQUIPMENT_PURCHASE_FASES.Test,
    equipment_change(change_time: date)
    to EQUIPMENT_PURCHASE_FASES.Test },
decision = {
    proposal_define(equipment_description: string, product_code: integer,
    advantages: string, price: real, pay_conditions: string, due_time: date),
    make_negotiation(final_pay_conditions: string, final_due_time: date),
    no_negotiation(reason: string) },
states = {
    available, invited, wait1, wait2, wait3, winner, provide_authorized, in_negotiation },
rules = {
    begin:
        msg(add_role) => state(available),
    receive_specification:
        state(available), msg(equipment_specification(Description, Features, Functions)) =>
        state(invited),
    send_proposal_values:
        state(invited),
        decision(proposal_define(Equipment_description, Product_code, Advantages,
        Price, Pay_conditions, Due_time)) =>
        msg(proposal(Number, Supplier_code, Equipment_description, Product_code,
        Advantages, Price, Pay_conditions, Due_time)), state(wait1);
        (before(value(Due_time),
        value(EQUIPMENT_PURCHASE_FASES.Specification.Expiry_time))),
    receive_proposal_OK:
        state(wait1), msg(receive_proposal_OK(Number, Supplier_code)) =>
        state(available),
    receive_auction_result_winner:
        state(available), msg(info_supplier_winner(Supplier_code, Supplier_winner_code)) =>
        state(winner); (value(Supplier_code) = Supplier_winner_code),
    receive_request_negotiation:
        state(winner), msg(request_negotiation(New_pay_conditions, New_due_time)) =>
        state(in_negotiation),
    send_negotiation_reply:
        state(in_negotiation),
        decision(make_negotiation(Final_pay_conditions, Final_due_time)) =>
        msg(negotiation_reply(Final_pay_conditions, Final_due_time)), state(wait2);
        (before(Final_due_time, value(Due_time))),
    reply_NOK:
        state(wait2), msg(negotiation_reply_NOK(Reason)) =>
        state(available); (Reason ≠ Null),
    reply_OK:
        state(wait2), msg(negotiation_reply_OK) =>
        state(winner),
    receive_provide_order:
        state(winner), msg(provide_order(Supplier_code, Purchase_description)) =>
        state(provide_authorized),
    no_negotiation:
        state(in_negotiation), decision(no_negotiation(Reason)) =>
        state(winner); (Reason ≠ Null),
    reception:
        state(provide_authorized), msg(payment(Supplier_code, Amount)) =>

```

```

    msg(delivery(Equipment_description, Product_code, Supplier_code, Due_time)),
    state(wait3),
tests_problems:
    state(wait3), msg(info_problems(Problems_description)) =>
    msg(equipment_change(Change_time)),
tests_OK:
    state(wait3), msg(info_tests_OK(Technical_opinion)) =>
    state(available)
> )

```

Resource Classes

```

Resource class (
  PERSON,
  <Base_role,
    static properties = {
      (registration, integer),
      (cpf, integer),
      (date_birth, date) },
    dynamic properties = {
      (name, string),
      (address, string),
      (department, string),
      (function, string),
      (salary, real) },
    rules = {
      r1: msg(create_object) => state(active),
      r2: state(active) => msg(add_role(Employee)) }
  >,
  <Employee,
    dynamic properties = {
      (purchase_function, string),
      (allocate_time_beginning, date) },
    messages = {
      allocate_order(registration: integer, purchase_function: string,
      allocate_time_beginning: date)
      from EQUIPMENT_PURCHASE_FASES.Specification,
      allocate_end_order(registration: integer)
      from EQUIPMENT_PURCHASE_FASES.Test,
    states = {
      free, allocate },
    rules = {
      begin:
        msg(add_role) => state(free),
      receive_order_allocation:
        state(free),
        msg(allocate_order(Registration.Purchase_function, Allocate_time_beginning)) =>
        state(allocate),
      receive_allocate_end_order:
        state(allocate), msg(allocate_end_order(Registration)) =>
        state(free) }
  > )

```

```

Resource class (
  EQUIPMENT,
  <Base_role,
    static properties = {
      (code, string),
      (description, string),
      (features, string),

```



```

    (functions, string) },
dynamic properties = {
    (supply_product_code, integer), /* Código utilizado pelo fornecedor */
    (model, string),
    (supply, string),
    (site, string),
    (setup_time, date),
    (lifetime, date),
    (delivery_time, date),
    (technical_conditions, string) },
rules = {
    r1: msg(create_object) => state(active),
    r2: state(active) => msg(add_role(Equipment)) }
>,
<Equipment,
    messages = {
        define_specification(description: string, features: string, functions: string)
            from EQUIPMENT_PURCHASE_FASES.Specification,
        reception(product_code: integer, supplier_code: integer, delivery_time: date)
            from EQUIPMENT_PURCHASE_FASES.Reception,
        test_result_OK(setup_time: date, lifetime: date, technical_conditions: string)
            from EQUIPMENT_PURCHASE_FASES.Test },
    states = {
        created, specificated, receipted, approved },
    rules = {
        begin:
            msg(add_role) => state(created),
        specification:
            state(created), msg(define_specification(Description, Features, Functions)) =>
            state(specificated),
        reception:
            state(specificated),
            msg(reception(Product_code, Supplier_code, Delivery_time)) =>
            state(receipted),
        receive_test_result_OK:
            state(receipted),
            msg(test_result_OK(Setup_time, Lifetime, Technical_conditions)) =>
            state(approved) }
> )

Resource class (
    PLACE,
    <Base_role,
        static properties = {
            (locality, string),
            (area, real) },
        dynamic properties = {
            (technical_features, set of string) },
        rules = {
            r1: msg(create_object) => state(active),
            r2: state(active) => msg(add_role(Place)) }
    >,
    <Place,
        messages = {
            define_place(locality: string, area: real, technical_features: set of string)
                from EQUIPMENT_PURCHASE_FASES.Specification,
            requery_prepare(technical_preparations: set of strings)
                from EQUIPMENT_PURCHASE_FASES.Auction,
            prepare_ready
                to EQUIPMENT_PURCHASE_FASES.Reception,
            installation(inst_time: date)
                from EQUIPMENT_PURCHASE_FASES.Reception },

```

```

states = {
  available, defined, ready, occupied },
rules = {
  begin:
    msg(add_role) => state(available),
  definition:
    state(available),
    msg(define_place(Locality, Area, Technical_features)) =>
    state(defined),
  preparation:
    state(defined), msg(requery_prepare(Technical_preparations)) =>
    msg(prepare_ready), state(ready),
  installation:
    state(ready), msg(installation(Inst_time)) =>
    state(occupied) }
> )

```

```

Resource class {
  CASH,
  <Base_role,
  static properties = {
    (account, integer) },
  dynamic properties = {
    (balance, real) },
  rules = {
    r1: msg(create_object) => state(active),
    r2: state(active) => msg(add_role(Allocation)) }
>,
  <Allocation,
  messages = {
    require_allocation_free(amount: real)
    from EQUIPMENT_PURCHASE_FASES.Payment,
    allocation_free_locked(free_time: date, amount: real)
    to EQUIPMENT_PURCHASE_FASES.Payment,
    free_order(amount: real)
    from EQUIPMENT_PURCHASE_FASES.Payment,
    allocation_free(amount: real)
    to EQUIPMENT_PURCHASE_FASES.Payment },
  states = {
    available, locked, free },
  rules = {
    begin:
      msg(add_role) => state(available),
    receive_request:
      state(available), msg(require_allocation_free(Amount)) =>
      msg(allocation_free_locked(Free_time, Amount), state(locked);
      (Amount <= value(balance)),
    receive_free_order:
      state(locked), msg(free_order(Amount)) =>
      msg(allocation_free(Amount)), state(free) }
> )

```

Process Classes

```

Process class (
  EQUIPMENT_PURCHASE_FASES,
  <Base_role,
  static properties = {
    (max_allocation, real),
    (expiry_time, date) },

```

```

dynamic properties = {
  (auction_limit_time, date) },
rules = {
  r1: msg(create_object) => state(active),
  r2: state(active) => msg(add_role(Specification))
  r3: state(active) => msg(add_role(Auction))
  r4: state(active) => msg(add_role(Payment))
  r5: state(active) => msg(add_role(Reception))
  r6: state(active) => msg(add_role(Test)) }
>,
<Specification,
messages = {
  begin_authorization(purchase_description: string, begin: date, max_allocation: real,
  expiry_time: date, auction_limit_time: date)
    from DECISION_MAKER.Production_manager,
  end_authorization(reason: string)
    from DECISION_MAKER.Production_manager,
  request_employee(begin_time: date, end_time: date)
    to EXTERNAL_WORLD,
  employee_list(components: set of PERSON.Employee)
    from EXTERNAL_WORLD,
  allocate_order(registration: integer, purchase_function: string,
  allocate_time_beginning: date)
    to PERSON.Employee,
  define_commission(components: set of PERSON.Employee, begin_composition_time:
  date, end_composition_time: date)
    to DECISION_MAKER.Purchase_commission,
  necessary_resources_list(equipments: set of equipment)
    from EXTERNAL_WORLD,
  purchase_info(purchase_description: string)
    to DECISION_MAKER.Purchase_commission,
  specification(description: string, features: string, functions: string)
    from DECISION_MAKER.Purchase_commission,
  request_supplier_list
    to EXTERNAL_WORLD,
  supplier_list(code1: integer, code2: integer, code3: integer)
    from EXTERNAL_WORLD,
  equipment_specification(code_supplier: integer, description: string, features: string,
  functions: string)
    to SUPPLIER.Supplier,
  define_specification(description: string, features: string, functions: string)
    to EQUIPMENT.Equipment,
  place_list(locality: string, area: real, technical_features: set of string)
    from EXTERNAL_WORLD,
  define_place(locality: string, area: real, technical_features: set of string)
    to PLACE.Place,
  fase1_ready(auction_limit_time: date, code1: integer, code2: integer, code3: integer,
  purchase_description: string)
    to EQUIPMENT_PURCHASE_FASES.Auction },
states = {
  no_active, active, wait_employee1, wait_employee2, wait_specification, specified,
  wait_supplier_list, sending1, sending2, sending3 },
rules = {
  begin:
    msg(add_role) => state(no_active),
  authorization:
    state(no_active),
    msg(begin_authorization(Purchase_description, Begin, Max_allocation, Expiry_time,
    Auction_limit_time)) =>
    state(wait_employee1),
  end_authorization:
    msg(end_authorization(Reason)) =>

```

```

    state(no_active),
request_employee_list:
    state(wait_employee1) =>
    msg(request_employee(Begin_time, End_time)), state(wait_employee2),
define_purchase_commission:
    state(wait_employee2), msg(employee_list(Components)) =>
    msg(allocate_order(Registration, Purchase_function, Begin_composition_time)),
    msg(define_commission(Components, Begin_composition_time,
    End_composition_time)), state(active),
define_place:
    state(active), msg(place_list(Locality, Area, Technical_features)) =>
    msg(define_place(Locality, Area, Technical_features)),
send_purchase_description:
    state(active), msg(necessary_resources_list(Equipments)) =>
    msg(purchase_info(Purchase_description)), state(wait_specification),
define_specification:
    state(wait_specification), msg(specification(Description, Features, Functions)) =>
    msg(define_specification(Description, Features, Functions)), state(specified),
request_supplier_list:
    state(specified) =>
    msg(request_supplier_list), state(wait_supplier_list),
receive_supplier_list:
    state(wait_supplier_list), msg(supplier_list(Code1, Code2, Code3)) =>
state(sending1);
    (Code1 ≠ Code2 And Code2 ≠ Code3 And Code1 ≠ Code3),
send_specification_to_supplier1:
    state(sending1) =>
    msg(equipment_specification(Code1, Description, Features, Functions)),
    state(sending2),
send_specification_to_supplier2:
    state(sending2) =>
    msg(equipment_specification(Code2, Description, Features, Functions)),
    state(sending3),
send_specification_to_supplier3:
    state(sending3) =>
    msg(equipment_specification(Code3, Description, Features, Functions)),
    msg(fase1_ready(Auction_limit_time, Code1, Code2, Code3, Purchase_description)),
    state(no_active) }
>,
<Auction,
messages = {
    end_authorization(reason: string)
    from DECISION_MAKER.Production_manager,
    info_supplier_choice(supplier_winner_code: integer)
    to DECISION_MAKER.Production_manager,
    proposal(number: integer, supplier_code: integer, equipment_description: string,
    product_code: integer, advantages: string, price: real, pay_conditions: string,
    due_time: date)
    from SUPPLIER.Supplier,
    receive_proposal_OK(number: integer, supplier_code: integer)
    to SUPPLIER.Supplier,
    proposal_grid(codes: set of integer, descriptions: set of string, product_codes:
    set of integer, advantages: set of string, prices: set of real, pay_conditions: set of
    string, due_times: set of date)
    to DECISION_MAKER.Purchase_commission,
    supplier_winner(supplier_code: integer)
    from DECISION_MAKER.Purchase_commission,
    info_supplier_winner(supplier_code: integer, supplier_winner_code: integer)
    to SUPPLIER.Supplier,
    request_negotiation_authorization
    from DECISION_MAKER.Purchase_commission,

```

```

request_negotiation_not_authorization
  from DECISION_MAKER.Purchase_commission,
request_negotiation(new_pay_conditions: string, new_due_time: date)
  to SUPPLIER.Supplier,
negotiation_reply(final_pay_conditions: string, final_due_time: date)
  from SUPPLIER.Supplier,
provide_order_authorization
  from DECISION_MAKER.Production_manager,
provide_order(supplier_code: integer, purchase_description: string)
  to SUPPLIER.Supplier,
negotiation_reply_OK
  to SUPPLIER.Supplier,
negotiation_reply_NOK(reason: string)
  to SUPPLIER.Supplier,
reqquery_prepare(technical_preparations: set of strings)
  to PLACE.Place,
fase1_ready(auction_limit_time: date, code1: integer, code2: integer, code3: integer,
purchase_description: string)
  from EQUIPMENT_PURCHASE_FASES.Specification,
fase2_ready(supplier_winner_code: integer)
  to EQUIPMENT_PURCHASE_FASES.Payment },
states = {
  no_active, active, wait_result, in_negotiation, wait_reply, wait_provide_authorized,
  wait_prepare, ending },
rules = {
  begin:
    msg(add_role) => state(no_active),
  begin_auction:
    state(no_active), msg(fase1_ready(Auction_limit_time, Code1, Code2, Code3,
Purchase_description)) =>
    state(active),
  end_authorization:
    msg(end_authorization(Reason)) =>
    state(no_active),
  receive_proposal_values:
    state(active),
    {msg(proposal(Number, Code1, Equipment_description, Product_code,
Advantages, Price, Pay_conditions, Due_time)),
msg(proposal(Number, Code2, Equipment_description, Product_code,
Advantages, Price, Pay_conditions, Due_time)),
msg(proposal(Number, Code3, Equipment_description, Product_code,
Advantages, Price, Pay_conditions, Due_time))} =>
    msg(receive_proposal_OK(Number, Code1)),
    msg(receive_proposal_OK(Number, Code2)),
    msg(receive_proposal_OK(Number, Code3)),
    msg(proposal_grid(Codes, Descriptions, Product_codes, Advantages, Prices,
Pay_conditions, Due_times)), states(wait_result);
    (Code1 ≠ Code2 And Code2 ≠ Code3 And Code1 ≠ Code3),
  choice_supplier:
    state(wait_result), msg(supplier_winner(Supplier_winner_code)) =>
    msg(info_supplier_winner(Code1, Supplier_winner_code)),
    msg(info_supplier_winner(Code2, Supplier_winner_code)),
    msg(info_supplier_winner(Code3, Supplier_winner_code)),
    state(in_negotiation); (before(value(Auction_limit_time), Now),
  not_negotiation:
    state(in_negotiation), msg(request_negotiation_not_authorization) =>
    msg(info_supplier_choice(Supplier_winner_code)), state(wait_provide_authorized),
  negotiation:
    state(in_negotiation), msg(request_negotiation_authorization) =>
    msg(request_negotiation(New_pay_conditions, New_due_time)), state(wait_reply),
  negotiation_OK:

```

```

state(wait_reply). msg(negotiation_reply(Final_pay_conditions, Final_due_time)) =>
  msg(info_supplier_choice(Supplier_winner_code)). msg(negotiation_reply_OK),
state(wait_provide_authorized): (Final_pay_conditions = New_pay_conditions and
Final_due_time = New_due_time),
negotiation_NOK:
  state(wait_reply),
  msg(negotiation_reply(Final_pay_conditions, Final_due_time)) =>
  msg(negotiation_reply_NOK(Reason)), states(wait_result);
  (Reason ≠ Null and Final_pay_conditions ≠ New_pay_conditions and Final_due_time
≠ New_due_time),
send_provide_order:
  state(wait_provide_authorized), msg(provide_order_authorization) =>
  msg(provide_order(Supplier_winner_code, Purchase_description)),
state(wait_prepare),
prepare_place:
  state(wait_prepare) =>
  msg(requery_prepare(Technical_preparations)), state(ending),
auction_ready:
  state(ending) =>
  msg(fase2_ready(Supplier_winner_code)), state(no_active) }
>,
<Payment,
messages = {
  end_authorization(reason: string)
  from DECISION_MAKER.Production_manager,
  require_allocation_free(amount: real)
  to CASH.Cash,
  allocation_free_locked(free_time: date, amount: real)
  from CASH.Cash,
  free_order(amount: real)
  to CASH.Cash,
  allocation_free(amount: real)
  from CASH.Cash,
  payment(supplier_code: integer, amount: real)
  to SUPPLIER.Supplier,
  fase2_ready(supplier_winner_code: integer)
  from EQUIPMENT_PURCHASE_FASES.Auction,
  fase3_ready(supplier_winner_code: integer)
  to EQUIPMENT_PURCHASE_FASES.Reception },
states = {
  no_active, active, wait, free_authorized, ending },
rules = {
  begin:
    msg(add_role) => state(no_active),
  begin_payment:
    state(no_active), msg(fase2_ready(Supplier_winner_code)) =>
    state(active),
  end_authorization:
    msg(end_authorization(Reason)) =>
    state(no_active),
  request_allocation_free:
    state(active) =>
    msg(require_allocation_free(Amount)), state(wait),
  receive_allocation_free:
    state(wait), msg(allocation_free_locked(Free_time, Amount)) =>
    msg(free_order(Amount)), state(free_authorized),
  payment:
    state(free_authorized), msg(allocation_free(Amount)) =>
    msg(payment(Supplier_winner_code, Amount)), state(ending),
  payment_ready:
    state(ending) =>

```



```

    msg(fase3_ready(Supplier_winner_code), state(no_active) }
>,
<Reception,
  messages = {
    delivery(equipment_description: string, product_code: integer, supplier_code:
    integer, delivery_time: date)
      from SUPPLIER.Supplier,
    reception(product_code: integer, supplier_code: integer, delivery_time: date)
      to EQUIPMENT.Equipment,
    prepare_ready
      from PLACE.Place,
    installation(installation_time: date)
      to PLACE.Place,
    fase3_ready(supplier_winner_code: integer)
      from EQUIPMENT_PURCHASE_FASES.Payment,
    fase4_ready(supplier_winner_code: integer)
      to EQUIPMENT_PURCHASE_FASES.Test },
  states = {
    no_active, active, wait, ending },
  rules = {
    begin:
      msg(add_role) => state(no_active),
    begin_reception:
      state(no_active), msg(fase3_ready(Supplier_code) =>
      state(active),
    reception:
      state(active), msg(delivery(Equipment_description, Product_code, Supplier_code,
      Delivery_time)) =>
      msg(reception(Product_code, Supplier_code, Delivery_time)),
      state(wait),
    prepare_place:
      state(wait), msg(prepare_ready) =>
      msg(installation(Delivery_time)), state(ending),
    reception_ready:
      state(ending) =>
      msg(fase4_ready(Supplier_code)), state(no_active) }
>,
<Test,
  messages = {
    begin_test
      to DECISION_MAKER.Purchase_commission,
    test_result(result:string)
      from DECISION_MAKER.Purchase_commission,
    info_test_result_OK
      to DECISION_MAKER.Production_manager,
    info_problems(supplier_code: integer, problems_description: string)
      to SUPPLIER.Supplier,
    info_tests_OK(supplier_code: integer, technical_opinion: string)
      to SUPPLIER.Supplier,
    equipment_change(supplier_code: integer, change_time: date)
      from SUPPLIER.Supplier,
    test_result_OK(setup_time: date, lifetime: date, technical_conditions: string)
      to EQUIPMENT.Equipment,
    update_necessary_resources_list(equipments: set of equipment)
      to EXTERNAL_WORLD,
    allocate_end_order(registration: integer)
      to PERSON.Employee,
    end_purchase
      to EXTERNAL_WORLD,
    fase4_ready(Supplier_winner_code: integer)
      from EQUIPMENT_PURCHASE_FASES.Reception },
  states = {

```

```
no_active, active, wait, ending },
rules = {
begin:
  msg(add_role) => state(no_active),
begin_tests:
  state(no_active), msg(fase4_ready(Supplier_code)) =>
  msg(begin_test), state(active),
results_tests_OK:
  state(active), msg(test_result(Result)) =>
  msg(info_tests_OK(Supplier_code, Technical_opinion)),
  msg(info_test_result_OK),
  msg(test_result_OK(Setup_time, Lifetime, Technical_conditions)),
  msg(update_necessary_resources_list(Equipments)),
  msg(allocate_end_order(Registration)),
  msg(end_purchase),
  state(ending); (Result = 'OK'),
problems_in_tests:
  state(active), msg(test_result(Result)) =>
  msg(info_problems(Supplier_code, Problems_description)),
  state(wait); (Result ≠ 'OK'),
change:
  state(wait), msg(equipment_change(Supplier_code, Change_time)) =>
  state(active) }
>)
```

Bibliografia

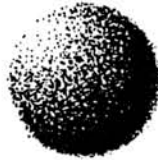
- [ANT 93] ANTONIOTTI, Marco. Conceptual and Pragmatic Tools for Design and Control of Manufacturing Systems: Petri Nets and Ramadge-Wonham Discrete Event Systems. Robotics Lab, Courant Institute of Mathematical Science, New York University, 1993. (disponível por FTP em <http://galt.cs.nyu.edu/students/marcoxa/publications.html>)
- [BAK 90] BAKO, Babou; VALETTE, Robert; CARDOSO, Janette. Implementação de um Sistema de Regras de Produção Controlado para a Aplicação na Supervisão de Sistemas Flexíveis de Manufatura. In: CONGRESSO NACIONAL DE AUTOMAÇÃO INDUSTRIAL, 4., 1990, São Paulo. **Anais...** São Paulo: CONAI, 1990. p. 164-173.
- [BAS 94] BASAGLIA, G.; GUIDA, M.; TREANOR, L. C. Supporting Industrial Enterprise Decision Making: A New Generation of Systems in Pirelli. In: IFIP WG5.7 WORKING CONFERENCE ON EVALUATION OF PRODUCTION MANAGEMENT METHODS, 1994, Gramado, Brasil. **Proceedings...** Amsterdam: North-Holland, 1994. p. 171-178.
- [BHA 93] BHARALI, R. C.; TABUCANON, Mario T. Petri Net-Based Cell Controller for a Flexible Manufacturing System. In: INTERNATIONAL CONFERENCE ON INDUSTRIAL ENGINEERING AND PRODUCTION MANAGEMENT (IEPM'93), 1993, Mons, Belgium. **Proceedings...** [S.l.: s.n.], 1993. p. 971-980.
- [BIL 93] BILLAUT, J. C.; ROUBELLAT, F. **Significant States and Decision Making for Real Time Workshop Scheduling**. Los Alamitos, CA: IEEE Computer Society, 1993. p. 493-500. Computers in Design, Manufacturing and Production, 1993, Paris-Evry, France.
- [BUC 84] BUCHEL, A.; BREUIL, D.; DOUMEINGTS, G. Comparison of Design Methodologies Characteristics and Deficiencies. In: IFIP WG5.7 WORKING CONFERENCE ON STRATEGIES FOR DESIGN AND ECONOMIC ANALYSIS OF COMPUTER SUPPORTED PRODUCTION MANAGEMENT SYSTEMS, 1983, Vienna. **Proceedings...** Amsterdam: North-Holland, 1984. p. 17-31.
- [CAJ 88] CARLIER, J.; CHRETIENNE, P. Timed Petri Net Schedules. In: Rozenberg, G. (Ed.). **Advances Petri Nets**. Berlin: Springer-Verlag, 1988. p. 62-84. (Lecture Notes in Computer Science, v. 340).

- [CAR 92] CARRIE, A. S.; MACINTOSH, R. UK Research in Manufacturing Systems Integration. In: Pels, H. J.; Wortmann, J. C.(Eds.). **Integration in Production Management Systems**. Amsterdam: North-Holland, 1992. p. 323-336.
- [CAV 95] CAVALCANTI, João M. B. et al. Uma Abordagem para a Implementação de um Modelo Temporal Orientado a Objetos usando SGBDs Relacionais. In: SIMPÓSIO BRASILEIRO DE BANCOS DE DADOS (SBBB'95), 10., 1995, Recife. **Anais...** Recife: UFPE/DI, 1995. p. 393-408.
- [DOU 84] DOUMEINGTS, Guy. Methodology to design Computer Integrated Manufacturing and Control of Manufacturing Unit. In: Rembold, U.; Dillmann, R. (Eds.). **Methods and Tools for Computer Integrated Manufacturing**. Berlin: Springer-Verlag, 1984. p. 194-265. (Lecture Notes in Computer Science, 168).
- [DOU 87] DOUMEINGTS, Guy et al. Design Methodology for Advanced Manufacturing Systems. **Computers in Industry**, Amsterdam, v. 9, n. 4, p. 271-296, Dec. 1987.
- [DOU 87a] DOUMEINGTS, Guy; POUMEYROL, Eric. Computer Aided Design for Advanced Manufacturing Systems. In: IFIP TC5/WG5.7 WORKING CONFERENCE ON NEW TECHNOLOGIES FOR PRODUCTION MANAGEMENT SYSTEMS, 1986, Tokyo, Japan. **Proceedings...** Amsterdam: North-Holland, 1987. p. 131-159.
- [DOU 92] DOUMEINGTS, Guy; CHEN, David; e MARCOTTE, François. Concepts, Models and Methods for the Design of Production Management Systems. **Computers in Industry**, Amsterdam, v. 19, n. 1, p. 89-111, Apr. 1992.
- [EDE 94] EDELWEISS, Nina. **Sistemas de Informação de Escritórios: Um Modelo para Especificações Temporais**. Porto Alegre: CPGCC da UFRGS, 1994. 187p. Tese de Doutorado.
- [EDE 94a] EDELWEISS, Nina; OLIVEIRA, José Palazzo M. de. **Modelagem de Aspectos Temporais de Sistemas de Informação**. Recife: UFPE-DI, 1994. 163p. Trabalho apresentado na Escola de Computação, 9., 1994, Recife, PE.
- [EDE 96] EDELWEISS, Nina; OLIVEIRA, José Palazzo M. de. Integrating DFD and ER Models using a Temporal Object-Oriented Model with Roles. In: ENGINEERING SYSTEM DESIGN AND ANALYSIS CONFERENCE (ESDA), 1996, Montpellier, France. **Proceedings...** New York: The American Society of Mechanical Engineers, 1996. v. 2, p. 173-181.

- [HAN 89] HANEN, Claire. Optimizing Microprograms for Recurrent Loops on Pipelined Architectures Using Timed Petri Nets. In: Rozenberg, G. (Ed.). **Advances in Petri Nets**. Berlin: Springer-Verlag, 1989. (Lecture Notes in Computer Science, v. 424).
- [HEU 91] HEUSER, Carlos A. **Modelagem Conceitual de Sistemas: Redes de Petri**. Campinas: R. Vieira, 1991. 150p. Trabalho apresentado na Escola Brasileiro-Argentina de Informática, 5., 1991, Nova Friburgo.
- [OLI 94] OLIVEIRA, José Palazzo M. de; WALTER, Cláudio. Time Aspects in the Manufacturing Decision Environment. In: ISPE/IFAC INTERNATIONAL CONFERENCE ON CAD/CAM, ROBOTICS AND FACTORIES OF THE FUTURE: CARS & FOF'94, 10., 1994, Ottawa, Canadá. **Proceedings ...** [S.l.: s.n.], 1994. p. 360-365.
- [OLI 95] OLIVEIRA, José Palazzo M. de et al. Implementation of an Object-Oriented Temporal Model. In: INTERNATIONAL CONFERENCE ON DATABASE AND EXPERT SYSTEMS APPLICATIONS (DEXA), 6., 1995, London, UK. **Proceedings...** [S.l.: s.n.], 1995. p. 35-44.
- [PET 81] PETERSON, J. L. **Petri Net Theory and the Modeling of Systems**. Englewood Cliffs: Prentice-Hall, 1981.
- [PLO 84] PLOCINSKI, Jerzy J. Optimization of Multi-variant Projects with Petri Nets. In: Hübner, H.; Paterson, I. (Eds.). **Production Management Systems: Strategies and Tools for Design**. Amsterdam: North-Holland, 1984. p. 39-49.
- [RAV 86] RAVICHANDRAN, Ramarathnam; CHAKRAVARTY, Amiya K. Decision Support in Flexible Manufacturing Systems Using Timed Petri Nets. **Journal of Manufacturing Systems**, v. 5, n. 2, p. 89-101, 1986.
- [REM 88] REMY, Pascal A.; LEVIS, Alexander H. On the Generation of Organizational Architectures Using Petri Nets. In: Rozenberg, G. (Ed.). **Advances in Petri Nets**. Berlin: Springer-Verlag, 1988. p. 371-385. (Lecture Notes in Computer Science, v. 340).
- [SIL 89] SILVA, Manuel; VALETTE, Robert. Petri Nets and Flexible Manufacturing. In: Rozenberg, G. (Ed.). **Advances in Petri Nets**. Berlin: Springer-Verlag, 1989. p. 374-417. (Lecture Notes in Computer Science, v. 424).

- [SIL 90] SILVA, Manuel; VALLETE, Robert. A Rede de Petri: uma Ferramenta para a Automação Fabril. In: **Tutoriais e Surveys**. São Paulo: CONAI, 1990. p. 181-200. Trabalho apresentado no Congresso Nacional de Automação Industrial (CONAI), 4., 1990, São Paulo.
- [TIM 91] TIMMERMANS, P. J. M. et al. Organisational and Decisional Aspects of Computer and Human Integrated Manufacturing. In: IFIP WG5.7 CONFERENCE ON ADVANCES IN PRODUCTION MANAGEMENT SYSTEMS, 1991, Espoo, Finland. **Proceedings...** Amsterdam: North-Holland, 1991. p. 129-142.
- [VAL 87] VALETTE, Robert. Nets in Production Systems. In: ADVANCES IN PETRI NETS: APPLICATIONS AND RELATIONSHIPS TO OTHER MODELS OF CONCURRENCY, 1986, Bad Honnef, West Germany. **Proceedings...** Berlin:Springer-Verlag, 1987. v. 2, p. 191-217. (Lecture Notes in Computer Science, 255).
- [VER 94] VERNADAT, F. Manufacturing System Modelling, Specification and Analysis. In: IFIP WG 5.7 WORKING CONFERENCE ON EVALUATION OF PRODUCTION MANAGEMENT METHODS, 1994, Gramado. **Proceedings...** Amsterdam: North-Holland, 1994. p. 75-83.
- [WAL 89] WALTER, Cláudio. A informática no planejamento da produção fabril: necessidades e métodos. **Revista de Informática Teórica e Aplicada**, Porto Alegre, v. 1, n. 1, p. 7-20, Out. 1989.
- [WAL 93] WALTER, Cláudio. Modelagem e Análise de Sistemas de Manufatura. Notas de Aula da Disciplina CMPP47 - Informática Industrial do CPGCC/UFRGS, 1993.
- [WOR 84] WORTMANN, J. C. Budget Control and Decision Making in Production Management Systems. In: Hübner, H. (Ed.). **Production Management Systems: Strategies and Tools for Design**, Amsterdam: North-Holland, 1984. p. 91-105.

Informática



UFRGS

CURSO DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Uma Técnica para a Modelagem de Estruturas de Decisão.

por

Sergio Ery Cazella

Dissertação apresentada aos Senhores:

Prof. Dr. Henrique Melo Rodrigues de Freitas (PPGA/UFRGS)

Profa. Dra. Lia Goldstein Golendziner

Profa. Dra. Nina Edelweiss

Vista e permitida a impressão.

Porto Alegre, 04 / 02 / 97.

Prof. Dr. José Palazzo Moreira de Oliveira,
Orientador.

Prof. Flávio Rech Wagner
Coordenador do Curso de Pós-Graduação
em Ciência da Computação
Instituto de Informática - UFRGS