Universidade Estadual de Campinas
Faculdade de Engenharia Elétrica e de Computação

Um modelo baseado em conhecimento para simular rebanhos de bovinos de corte

Autor: Miguel Antonio Bueno da Costa
Orientador: Prof. Dr. Jurandir Fernando Ribeiro Fernandes
Co-orientador: Prof. Dr. Pedro Franklin Barbosa

Tese de Doutorado apresentada à Faculdade de Engenharia Elétrica e de Computação como parte dos requisitos para obtenção do título de Doutor em Engenharia Elétrica. Área de concentração: **Automação**.

Banca Examinadora

Jurandir Fernando Ribeiro Fernandes, Dr…..DENSIS/FEEC/UNICAMP
Carlos Guilherme Silveira Pedreira, Dr. ....DZ/ESALQ/USP
Antonio Batocchio, Dr. .... DEF/FEM/UNICAMP
Akebo Yamakami, Dr. …… ……... DT/FEEC/UNICAMP
Raul Vinhas Ribeiro, Dr..... .... .DENSIS/FEEC/UNICAMP

Campinas, SP
Dezembro/2004

FICHA CATALOGRÁFICA PREPARADA PELA BIBLIOTECA DA ÁREA DE ENGENHARIA – BAE – UNICAMP

Resumo

Esse trabalho apresenta uma abordagem sobre técnicas e ferramentas computacionais, utilizadas no desenvolvimento de modelos de simulação e, também, uma metodologia de suporte ao desenvolvimento de projetos. Um modelo de simulação de sistemas de produção de bovinos de corte foi desenvolvido utilizando conceitos de simulação de sistemas e de sistemas baseados em conhecimento com o suporte da metodologia SimuCAD: simulação e computação gráfica aplicadas ao estudo de sistemas produtivos, adaptada para atuar em sistemas de produção agropecuária. Este modelo representa um sistema de produção extensiva de bovinos de corte, genérico o suficiente para ser aplicado em qualquer rebanho. O modelo de simulação emula as políticas de manejo do pecuarista. O sistema computacional desenvolvido é amigável o suficiente para ser utilizado por pessoas leigas em informática. Os resultados extraídos desse simulador se aproximam muito dos dados associados ao sistema real de produção. O modelo foi testado e validado utilizando dados reais de uma fazenda produtora de gado de corte.

Palavras-chave: Simulação, Sistema Baseado em Conhecimento, Metodologia SimuCAD, Bovinos de Corte.

Abstract

This work presents a developing approach of simulation models, including the necessary computing tools and techniques, and, also, a supporting methodology for project design and implementation. A beef cattle simulation model was developed by using the SimuCAD methodology (graphic computing and simulation applied to the study of productive systems). This methodology, based on system simulation concepts and knowledge-based systems, was adapted to perform in the cattle breeding business. The developed model represents an extensive beef cattle production system that is generic enough to be applied in any kind of cattle breeding. It simulates the managing politics of the cattle breeders and has the advantage of being friendly enough to be used by anyone, even by non-simulating-skilled persons. The model was tested and validated by using the actual data of a beef cattle production farm.

Keywords: Simulation, Knowledge-based System, SimuCAD Methodology, Beef Cattle.

A todos aqueles que, com muita paciência e sapiência, souberam identificar o momento correto de me incentivar e o momento adequado de me cobrar quanto ao resultado deste trabalho.

Agradecimentos

Ao meu orientador Prof. Dr. Jurandir Fernando Ribeiro Fernandes e ao meu co-orientador Prof. Dr. Pedro Franklin Barbosa, sou grato pela orientação, pelo apoio e pela amizade.

Ao Isaías Torres, meu parceiro e amigo, agradeço pela ajuda no modelo computacional, pelas críticas ao trabalho e pelas ricas discussões teóricas sobre simulação.

Às pessoas que amo.

Aos meus pais que, do alto, me protegem e me orientam.

Sumário

Lista de Figuras **xi**

1 Introdução **1**

1.1 Objetivos 1

1.2 Justificativas 1

1.3 Estrutura do Trabalho 2

2 Simulação de Sistemas **5**

2.1 Pequeno Histórico dos Softwares de Simulação de Sistemas 7

2.2 Vantagens e Desvantagens da Simulação 8

2.3 Definições para Simulação de Sistemas 11

2.4 O Processo de Simulação 13

2.5 Terminologia da Simulação 14

2.6 Uma Metodologia para desenvolver Simulações 17

2.7 Comentários finais sobre simulação de sistemas 27

3 Sistemas baseados em conhecimento **29**

3.1 Inteligência Artificial e Sistemas Especialistas 29

3.2 Representação do Conhecimento 36

3.2.1 Regras de Produção 36

3.2.2 Redes Semânticas 40

3.2.3 *Frames* 41

3.3 Simulação Tradicional e Simulação Baseada em Conhecimento 43

3.3.1 Simulação & Sistemas Especialistas: Atuação Conjunta 44

3.3.2 Desvantagens da Simulação Tradicional 47

3.3.3 A Simulação Inteligente 50

3.3.4 Algumas conclusões sobre o tema simulação inteligente 52

4 Metodologia SimuCAD **56**

4.1 Métodos e técnicas utilizadas pela metodologia SimuCAD 58

4.2 A metodologia SimuCAD aplicada em sistemas agropecuários 62

5 O Modelo *SanCarlo*: Simulação de Rebanhos de Bovinos de Corte **66**
5.1 Relações sobre animais e pastagens: entidades envolvidas no modelo, suas relações lógicas e seus diagramas representativos 68
5.1.1 Programa de controle principal 73
5.1.2 Procedimento Crescimento 74
5.1.3 Procedimento Fertilidade 84
5.1.4 Procedimento Morte 87
5.2 Modelo de simulação: lógica e animação 91
5.2.1 A lógica do modelo SanCarlo 92
5.3 Módulo de decisão baseado em conhecimento 100
5.3.1 Criando uma nova regra no sistema baseado em conhecimento 101
5.3.2 Atuação conjunta entre o ARENA e o CLIPS 106
5.4 Processo de controle geral da simulação 109
6 Verificação e Validação do Modelo **111**
6.1 Comentários Gerais 116
7 Conclusões **118**
Referências Bibliográficas **120**
Apêndice 01: Lógica do Modelo de Simulação desenvolvida no ARENA **127**
Apêndice 02: Lógica do Modelo de Simulação desenvolvida nos módulos VBA – ARENA **143**

Lista de Figuras

Figura 1.1 A estrutura do trabalho 4
Figura 2.1 Passos da Simulação 18
Figura 3.1 Exemplo de casamento e execução de regras de produção 39
Figura 3.2 Exemplo de Rede Semântica 41
Figura 3.3 Exemplo de *Frame* 43
Figura 3.4 Uma taxonomia para combinar sistemas especialistas com Simuladores 45
Figura 4.1 Espaço de discussão de diferentes racionalidades 57
Figura 4.2 Macro visão da metodologia SimuCAD 58
Figura 4.3 Exemplo de um *template* de um centro de produção 60
Figura 5.1 Visão Global do Modelo *SanCarlo* 70
Figura 5.2 Módulo Simulação 71
Figura 5.3 Módulo Decisor 71
Figura 5.4 Função Emitir Relatórios 72
Figura 5.5 Função Tratar Dados da Fazenda 72
Figura 5.6 Função Tratar Dados do Mercado 73
Figura 5.7 Procedimento Crescimento 75
Figura 5.8 Procedimento Morte 90
Figura 5.9 Tela de abertura do modelo *SanCarlo* 94
Figura 5.10 Pasta Fatores I 95
Figura 5.11 Pasta Fatores II 96
Figura 5.12 Pasta Pastagem & Rebanho 97
Figura 5.13 Pasta Controle 98
Figura 5.14 Parte da Lógica do modelo *SanCarlo* 99
Figura 5.15 Parte da programação, em VBA, do procedimento Crescimento 99
Figura 5.16 Pasta referente à entrada das regras de produção 100

Figura 5.17 Tela do editor 101
Figura 5.18 Anexando uma condição baseada em um atributo dos animais 103
Figura 5.19 Complementação da condição da regra 103
Figura 5.20 A condição depois de terminada 104
Figura 5.21 Selecionando a lista de ações 105
Figura 5.22 Regra terminada 105
Figura 5.23 Módulo de Controle Geral da Simulação 109
Figura 5.24 Tela mostrando a atuação conjunta do ARENA com o CLIPS 110
Figura 6.1 Dados do rebanho experimental da Embrapa/CPPSE 114
Figura 6.2 Resultados da simulação 115

1 Introdução

Este trabalho apresenta uma abordagem sobre técnicas e ferramentas computacionais, utilizadas no desenvolvimento de modelos de simulação e, também, uma metodologia de suporte ao desenvolvimento de projetos. Um modelo de simulação de produção de bovinos de corte foi desenvolvido utilizando conceitos de simulação e de sistemas baseados em conhecimento com o suporte da metodologia SimuCAD: simulação e computação gráfica aplicadas ao estudo de sistemas produtivos, adaptada para atuar na produção agropecuária.

1.1 Objetivos

O objetivo central do trabalho foi criar um modelo de simulação para auxiliar no planejamento e controle de rebanhos de bovinos de corte. Para emular a política de manejo do pecuarista, foi estabelecida a necessidade deste modelo conter um módulo decisor, baseado em conhecimento, para fornecer resultados mais próximos da realidade do produtor.

Um objetivo complementar, associado às especificidades do ambiente de produção estudado, foi o de garantir que o modelo desenvolvido representasse uma produção extensiva de bovinos de corte, genérico o suficiente para ser aplicado em qualquer rebanho. E, para tornar viável a sua implementação, a ferramenta computacional desenvolvida precisaria ser amigável para ser utilizada por pessoas leigas em informática.

1.2 Justificativas

Uma produção extensiva de bovinos de corte, normalmente, possui algumas características que dificultam a tomada de decisão. Essas dificuldades podem ser resumidas na falta de informação sobre o rebanho e sobre as pastagens, além da fragilidade das previsões decorrentes da combinação das diversas alternativas de manejo. Um banco de dados pode solucionar o problema da falta de informações sobre os animais, mas não pode responder as questões associadas ao comportamento do rebanho no tempo. Uma análise da dinâmica desse tipo de

produção não pode ser feita utilizando ferramentas que trabalham de forma estática e determinística. A produção de bovinos de corte precisa ser estudada considerando as variáveis estocásticas, intrínsecas e exógenas, associadas a ele. Necessita de um tratamento dinâmico que forneça resultados para os diversos cenários alternativos, envolvendo combinações de política de manejo, decisões sobre suplementação alimentar e as muitas variáveis associadas ao animal, às pastagens e ao mercado. Uma forma de combinar, em um mesmo modelo, a força do tratamento numérico com o poder de análise de situações diversas, é criar um modelo de simulação. Esse modelo deve contemplar a abordagem probabilística necessária para representar a realidade e a dinâmica do sistema estudado no tempo, além de permitir interfaces amigáveis que facilitem a sua utilização.

Diante do exposto, pode-se imaginar que a complexidade associada ao desenvolvimento de um modelo desse tipo é alta. Isso leva à necessidade de se utilizar uma metodologia moderna que dê suporte ao desenvolvimento do projeto, organizando a estrutura e o inter-relacionamento dos módulos do sistema estudado, coletando, armazenando e tratando as informações e orientando no sentido de atender, de forma precisa e rápida, o objetivo traçado.

Os números associados à produção de bovinos de corte no Brasil também justificam a necessidade de tecnologias modernas para auxiliar na gestão dos negócios nesta área. Segundo ALENCAR & LIMA (2003) a América do Sul possui o segundo maior rebanho do mundo. Dentre todos os países, o maior rebanho comercial é o do Brasil, com 167,4 milhões de cabeças de gado. Devido a sua dimensão territorial o tipo de produção que predomina é o extensivo, com animais espalhados por vastas áreas de pastagens. O ANUALPEC (2004) informa que 68% do rebanho brasileiro são de bovinos de corte criados dessa forma.

1.3 Estrutura do Trabalho

- Os capítulos 2 e 3 apresentam conceitos e comentários sobre simulação e sistemas baseados em conhecimento.

- O capítulo 4 trata da metodologia *SimuCAD*: simulação e computação gráfica aplicadas ao estudo de sistemas produtivos (COSTA et al., 1996), desenvolvida e aplicada pelo grupo de pesquisa de mesmo nome, que é coordenado pelo autor deste trabalho. Essa metodologia, utilizada há dez anos em projetos nos diferentes setores da economia, foi adaptada para auxiliar no desenvolvimento do modelo de simulação de rebanhos de bovinos de corte que recebeu o nome de *"SanCarlo"*.
- O capítulo 5 apresenta, de forma detalhada, o modelo *SanCarlo*.
- O capítulo 6 mostra o processo de verificação e validação do simulador e termina com comentários gerais sobre o trabalho.
- O capítulo 7 traz as conclusões da tese.

A Figura 1.1 mostra a estruturação dos capítulos da tese, seus conteúdos, e como os mesmos estão interligados.

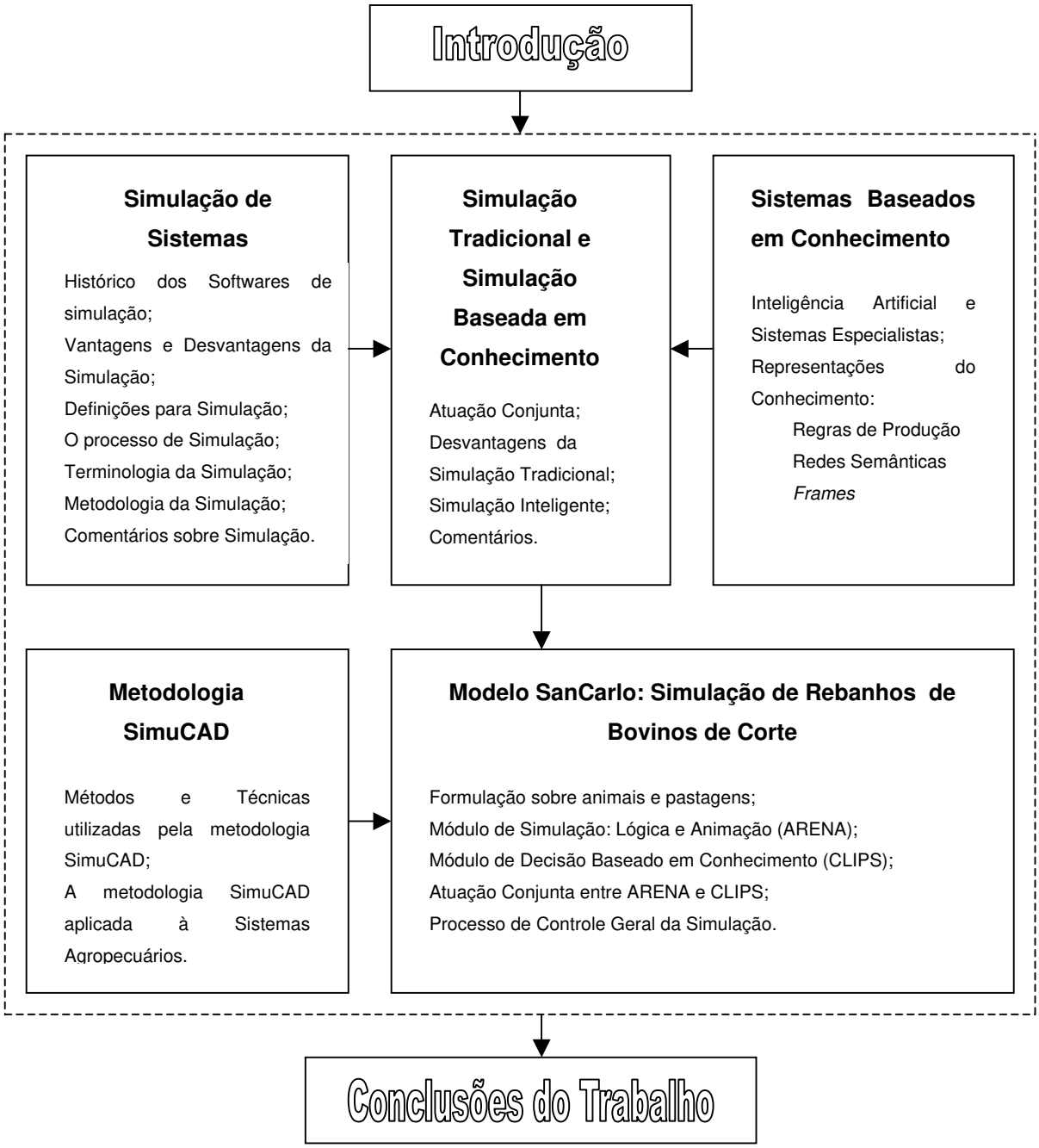


Figura 1.1: A estrutura do trabalho

2 Simulação de Sistemas

> "Simulação [ETIM lat. *Simulatio,onis*] S.f. ação ou efeito de simular... 3 imitação do funcionamento de um processo por meio do funcionamento de outro 4 teste, experiência ou ensaio em que se empregam modelos para simular o ser humano, em especial em casos de grande perigo de vida... ◆ s.analógica teste ou experiência em que os modelos empregados têm comportamento análogo ao da realidade ◆ s. digital experiência ou ensaio constituído por uma série de cálculos numéricos e decisões de escolha limitada, executados de acordo com um conjunto de normas preestabelecidas e apropriadas à utilização de computadores ... (Dicionário HOUAISS da língua portuguesa, 1ª. Edição, pág. 2575, Editora Objetiva, 2001)

Simulação é um processo de análise e síntese, útil como ferramenta de auxílio à tomada de decisão em processos produtivos complexos. Portanto, entende-se simulação como todo o processo de construção de um modelo representativo de um sistema real. O modelo gerado é chamado de simulador. A simulação é utilizada quando não é possível fazer experimentações no sistema real (devido, por exemplo, ao longo tempo necessário para realizar o experimento, ou ao alto custo do experimento ou, ainda, à dificuldade de tratar fisicamente o experimento). A simulação é utilizada, também, quando não é possível desenvolver um modelo que solucione o problema através de técnicas da Pesquisa Operacional que otimizam a solução, como a Programação Linear e a Programação Dinâmica, dentre outras. Neste sentido existe uma afirmação de FOX (1990) que diz: "Se existe um algoritmo que otimiza a solução, utilize-o".

Por outro lado, segundo GORDON (1978), a quantidade de problemas que podem ser resolvidos matematicamente é limitada. As técnicas matemáticas requerem que o problema possa ser expresso num modelo matemático, de formato particular. Ao se estudar um problema é essencial considerar os métodos analíticos primeiro e verificar se o grau de abstração, necessário para obter uma solução analítica, não é muito severo. Quando não for possível gerar soluções

para um sistema por meios analíticos, é necessário utilizar métodos de computação numérica, por exemplo a simulação.

SHANNON (1975) comenta que, ao contrário de muitas tecnologias que são classificadas de acordo com a disciplina na qual elas se originam (física ou química, por exemplo), a simulação pode ser aplicada a todas as disciplinas. Incontáveis trabalhos, relatórios, teses de mestrado e doutorado, revistas e livros mostram vários campos de aplicação para a simulação, como negócios, economia, marketing, educação, política, ciências sociais, ciência comportamental, relações internacionais, transportes, estudos urbanos, e muitos processos produtivos dos mais diferentes setores da economia.

Segundo NAYLOR (1971) o uso do verbo “simular” tem origem no trabalho de Von Newman & Ulam que, em 1940, associaram a expressão Análise de Monte Carlo a uma técnica matemática que foi utilizada para solucionar certos problemas de blindagem em reatores nucleares. O tratamento experimental desses problemas seria muito caro e uma abordagem analítica seria muito complicada.

A análise ou método de Monte Carlo consiste na amostragem experimental com números randômicos. A simulação e o método de Monte Carlo se confundem. Ambos utilizam computação numérica. Porém, segundo GORDON (1978), enquanto a simulação é aplicada em modelos dinâmicos, a análise de Monte Carlo é aplicada a modelos estáticos.

Em síntese, os modelos de simulação são valiosos e necessários, uma vez que fórmulas fechadas e equações analíticas raramente conseguem descrever, de forma adequada, sistemas reais complexos (WIDMAN & LOPARO, 1990). Uma grande vantagem da simulação é permitir estudos de sistemas reais sem modificá-los, com velocidade e baixo custo quando comparados às alterações físicas e organizacionais reais necessárias para estudar as mesmas alternativas de cenários futuros. Dessa forma, mudanças podem ser tentadas e estudadas de forma sistemática sem interferir no sistema real (BAKER, 1982).

Os simuladores podem ser de grande valia quando decisões precisam ser tomadas num ambiente não completamente conhecido, devido às diversas interações existentes no mesmo.

2.1 Pequeno Histórico dos Softwares de Simulação de Sistemas

A simulação computadorizada começou a ser usada no século XX, na década de 50, na indústria aeroespacial dos EUA. Somente na década de 60 é que começaram a surgir as primeiras aplicações em sistemas industriais. Esse processo foi lento, devido principalmente à baixa capacidade de processamento dos computadores existentes na época (HARREL & TUMAY, 1995).

As primeiras simulações foram desenvolvidas nas linguagens de programação de aplicação geral, principalmente em FORTRAN. Posteriormente evoluíram para as linguagens PASCAL e C. A primeira etapa foi longa até surgirem as linguagens específicas de simulação. Recentemente, as linguagens de simulação aumentaram sua abrangência de aplicação através da programação orientada a objetos e da simulação com base na *web*.

As linguagens de simulação são linguagens de programação com blocos de código (comandos) de alto nível, que procuram representar ações e elementos característicos dos sistemas industriais. Entre essas linguagens destacam-se GPSS, SIMSCRIPT, SIMAN, SLAM, entre outras. Os blocos de código são similares a comandos e são facilmente reutilizáveis, gerando programas menores de forma rápida.

A segunda onda de desenvolvimento das ferramentas computacionais para simulação ocorreu na década de 80, quando surgiram os simuladores, alguns destinados a aplicações industriais específicas. Segundo HARREL & TUMAY (1995), os simuladores não apresentavam capacidades de programação geral. Eram específicos e elevavam o nível de diálogo com o usuário através de elementos da manufatura parametrizáveis, o que permitiu que ocupassem o lugar das linguagens de simulação no desenvolvimento de modelos de modos produtivos industriais. Entre esses pacotes computacionais figuram ARENA®

(ARENA, 2004), AUTOMOD® (AUTOMOD, 2002), PROMODEL® (PROMODEL, 2004), entre outros. Nessa época, já se acreditava que a técnica de simulação passaria a ser uma ferramenta muito utilizada, devido ao aumento da divulgação, maior disponibilidade de *softwares*, maior capacidade e facilidade de uso da tecnologia e a redução no preço do *hardware*. Nessa linha, BANKS & CARSON (1984) argumentaram que a disponibilidade de mais sistemas de simulação amigáveis, de plataformas de *hardware* mais econômicas e poderosas e de avanços metodológicos permitiriam que a simulação se desenvolvesse mais rapidamente, tanto em número de usuários quanto em capacidade.

2.2 Vantagens e Desvantagens da Simulação

TORRES (2001) apresenta uma lista de razões, adaptada de HARREL et al. (1995) e BANKS & NORMAN (1995) que justificam a utilização da simulação na solução de problemas:

- **Simulação pode promover a solução criativa de problemas**. Através de modelos, soluções intuitivas para problemas do dia-a-dia da empresa podem ser modeladas e testadas resultando em avaliações quantitativas;
- **Simulação pode identificar causas de problemas**. Permite a experimentação com as interações entre componentes de um sistema complexo. Permite também identificar causas restritivas como gargalos e falta de materiais;
- **Simulação pode prever resultados**. A simulação serve, também, para avaliar projetos e sistemas ainda inexistentes. Além disso, alterações de processos existentes podem ser testadas sem que se tenha que alterar a rotina do sistema real, o que implicaria em desperdício de tempo e dinheiro. Exemplos dessas alterações podem incluir a escolha entre dois equipamentos diferentes, a inclusão de novos produtos em uma linha de produção, alterações de volumes de produção e muitos outros. Permite, também, explorar possibilidades para um negócio, em termos de novas

políticas, novos métodos gerenciais e diferentes procedimentos operacionais;

- **Simulação pode considerar as variações aleatórias do sistema**. É possível considerar estatisticamente as variações que atingem os sistemas reais;
- **Simulação pode promover soluções totais**. As abordagens tradicionais geralmente dividem os problemas em partes menores para facilitar a sua resolução. A simulação permite que todos os condicionantes de um problema sejam analisados simultaneamente, o que permite que sejam consideradas as relações entre eles. Além disso, essa forma de trabalho exige a participação efetiva de pessoas de todas as áreas envolvidas na modelagem e na avaliação das soluções geradas. Ocorre a construção de consensos através de opiniões objetivas. Mesmo as hipóteses menos prováveis podem ser testadas e comparadas;
- **Simulação pode ser eficiente do ponto de vista econômico**. O uso de simulação pode compensar financeiramente, pois serve para treinamento de pessoal, troca de idéias e experiências e para estruturação de soluções. A sua capacidade de avaliação de alternativas permite que se calcule rapidamente os impactos da entrada de novas estratégias e conceitos administrativos na empresa, além de permitir avaliar o desempenho de investimentos em capital sem que haja o seu comprometimento efetivo;
- **Simulação pode preparar a empresa para mudanças**. A preparação para mudanças organizacionais, ambientais e na estrutura das informações pode ocorrer pela observação dos resultados obtidos do modelo de simulação. Conforme comentado, modelos podem ser usados para treinar pessoal nas novas situações produtivas como na utilização de um novo equipamento ou em uma nova forma de organizar o trabalho;

- **Simulação pode promover agilidade**. A possibilidade de comprimir e expandir o tempo permite realizar análises sobre longos períodos de funcionamento do sistema em um curto espaço de tempo.
- **Simulação pode gerar conhecimento**. Permite entender porque determinado fenômeno ocorre na realidade. O entendimento é desenvolvido baseado na experiência sobre o funcionamento do sistema e não sobre idéias de como ele supostamente funcionaria. Além disso, o próprio processo de construção do modelo aumenta o nível de entendimento do sistema;
- **Simulação pode ser usada para verificar soluções analíticas geradas por outros métodos**;
- **Simulação pode ser mais simples que outras ferramentas**. Uma vez construído, um mesmo modelo pode ser utilizado várias vezes para realizar uma análise. Os dados utilizados em simulação, em alguns casos, podem ser obtidos mais facilmente que aqueles exigidos por outros métodos. É possível tratar o mundo real com um alto grau de complexidade, o que amplia a representatividade do modelo (SCHMIDT & TAYLOR apud BANKS & CARSON, 1984);

TORRES (2001) cita BANKS (2000) que aponta as seguintes desvantagens na adoção de simulação:

- **A construção do modelo requer treinamento especial que demanda tempo e experiência**. A falta de experiência pode ter um impacto considerável no tempo para se desenvolver o modelo conceitual;
- **A difícil interpretação dos resultados da simulação em função da sua aleatoriedade**. O processo de análise dos dados de saída da simulação necessita de conhecimentos estatísticos não triviais**;**
- **O longo tempo e o alto custo** requeridos pela análise e modelagem de sistemas complexos;

- **A utilização inapropriada da simulação** quando soluções de outra natureza resolveriam o mesmo problema com menor tempo e custo.

Como comentário geral sobre as aplicações de simulação, HARREL & TUMAY (1995) afirmam que a mesma é adequada nos casos onde:

- **o desenvolvimento de um modelo de outra natureza seja difícil ou impossível;**
- **existam duas ou mais variáveis aleatórias interdependentes;**
- **a dinâmica do sistema seja muito complexa;**
- **o objetivo principal do estudo seja observar o comportamento do sistema com o passar do tempo;**
- **o uso da animação seja importante.**

Além disso, argumentam que a simulação é indicada principalmente para analisar os aspectos operacionais do sistema, tendo pouca eficiência para avaliar aspectos humanos como níveis de habilidade, interesses e relações interpessoais, e aspectos tecnológicos como confiabilidade.

2.3 Definições para Simulação de Sistemas

Existe um grande número de definições para simulação. A seguir são apresentadas algumas delas, provenientes de livros clássicos sobre o assunto, e uma definição particular do autor desta tese. São feitas, também, algumas colocações relevantes na busca de elucidar o que vem a ser simulação de sistemas.

SHANNON (1975): "Simulação é o processo de desenvolvimento de um modelo de um sistema real, e a condução de experimentos nesse modelo, com o propósito de entender o comportamento do sistema e/ou avaliar várias estratégias (com os limites impostos por um critério ou conjunto de critérios) para a operação do sistema".

NAYLOR (1971)**:** "Simulação é uma técnica numérica para realizar experiências em um computador digital, envolvendo certos tipos de modelos lógicos que

descrevem o comportamento de um sistema econômico ou de negócios (ou um aspecto parcial de um deles) sobre extensos intervalos de tempo".

GORDON (1978)**:** "Simulação de sistemas é a técnica de solucionar problemas observando o desempenho no tempo de um modelo dinâmico do sistema".

A definição de T. Naylor especifica experimentações em computadores, enquanto que, tanto a definição dada por R. Shannon quanto a de G. Gordon são amplas o suficiente para englobarem os modelos físicos dinâmicos. Nesses modelos os resultados são derivados de medidas físicas, em vez de computações numéricas.

Na opinião do autor desta tese simulação de sistemas "*é o processo de modelagem de um sistema real e a experimentação no mesmo. Os resultados das experimentações, após análises, apresentam uma visão futura do sistema. As informações geradas auxiliam nas tomadas de decisão, necessárias no momento presente, e contribuem para uma melhor compreensão do sistema estudado*".

Além de auxiliar na tomada de decisão, é importante enfatizar a contribuição da simulação para a compreensão do sistema estudado pois, como afirma KNUTH (1969), "...freqüentemente nos enganamos, pensando saber mais do que realmente sabemos sobre uma coisa, até que tentamos simulá-la num computador". Através da simulação o usuário pode comparar seus resultados com os do sistema e validar seus próprios processos de raciocínio.

A simulação de sistemas é, portanto, uma metodologia experimental que busca descrever o funcionamento de um sistema. Esta metodologia constrói formas de quantificar o comportamento observado, prevendo o comportamento futuro. A proposta da simulação é produzir dados (e ela é uma grande geradora de números) que, quando analisados, identificarão importantes aspectos do sistema estudado, auxiliando na explicação, compreensão e melhoria do mesmo.

Segundo DOUKIDIS (1987) essa é a função primária de um modelo de simulação que busca examinar como o sistema se comporta durante um período de tempo. Para atingir este objetivo, o modelo deve providenciar facilidades, para representar o estado atual do sistema, e várias pré-condições que, se satisfeitas, irão resultar num provável estado futuro.

Outra colocação clássica foi dada por SHUBIK (1960) que centra no modelo a sua definição: "Simulação de um sistema ou de um organismo é a operação de um modelo (ou simulador) que representa esse sistema ou organismo. O modelo é passível de manipulações que seriam difíceis de levar a cabo no sistema que ele representa, quer pelo custo, quer pela impossibilidade de fazê-las. As propriedades concernentes ao comportamento de um sistema podem ser inferidas estudando-se a operação do modelo".

2.4 O Processo de Simulação

Entende-se por processo de simulação o desenvolvimento de um modelo de simulação, a experimentação do mesmo e a avaliação dos resultados.

SHANNON (1975) coloca que o desenvolvimento de um processo de simulação é mais uma extensão das artes do que das ciências. Isso talvez explique a dificuldade de se apresentar um mecanismo sistemático para que um usuário interessado possa desenvolver simulações. Apesar das dificuldades, pode-se estabelecer os elementos que participam de um modelo de simulação e alguns passos que, se seguidos, podem ajudar um modelador, mesmo novato, a ter um bom desempenho no desenvolvimento de simulações.

Todo modelo de simulação possui, de forma combinada ou isolada, os seguintes elementos:

- **Componentes**: são as partes (ou subsistemas) integrantes do sistema. Entende-se por sistema, um conjunto de objetos, que interagem entre si, para atingir um objetivo comum.
- **Parâmetros e Variáveis**: são elementos do sistema que recebem valores. Existem dois tipos de variáveis: Endógenas (Dependentes) e Exógenas (Independentes). Variáveis endógenas são aquelas produzidas dentro do sistema ou resultantes de causas internas. São também chamadas variáveis de estado (pois mostram o estado do sistema em um dado instante) ou variáveis de saída (pois são responsáveis por gerar e apresentar os resultados oriundos do sistema). Variáveis exógenas, também chamadas variáveis de entrada, são

originárias de (ou produzidas por) causas externas. Parâmetros são variáveis para as quais se estabelecem valores no início da experimentação de cada cenário.

- **Relações Funcionais**: são normalmente apresentadas na forma de equações matemáticas, que relacionam as variáveis. Essas relações podem ser de ordem determinística (onde para uma dada entrada existe uma única saída) ou estocástica (onde para uma dada entrada existe incerteza associada à saída).
- **Restrições**: são limitações, impostas pelo modelador ou pela natureza do problema, que restringem os valores das variáveis.
- **Objetivos**: é o estabelecimento das metas do sistema e como elas devem ser avaliadas. A manipulação do modelo é orientada de forma a satisfazer esses objetivos.

2.5 Terminologia da Simulação

A terminologia utilizada na simulação não é única, mas há uma certa tendência da literatura atual em aceitar a que é apresentada a seguir.

- **Modelo**: é a representação de um sistema. Existem várias formas de se classificar modelos. COSTA (2001) apresenta uma explanação sintética, segundo a qual os modelos são tipificados como icônicos (representação fiel do sistema, geralmente em escala diferente da real), analógicos (as propriedades do sistema são representadas por outras equivalentes) e simbólicos (as propriedades do sistema são representadas por símbolos). Outro tipo de classificação separa os modelos em estáticos e dinâmicos. Os modelos estáticos não consideram explicitamente a variável tempo e são, normalmente, tratados por técnicas analíticas. Os modelos dinâmicos tratam com interações que se alteram no tempo (o estudo do sistema em um certo instante interfere no estado do sistema no instante seguinte). Também existe uma caracterização importante para os modelos, que diz respeito às situações discreta e contínua. Um modelo discreto sofre alterações

repentinas no tempo. As mudanças no estado do sistema são expressas em termos do estado do sistema no início do período. As mudanças durante um intervalo de tempo são calculadas e é deduzido um novo estado do sistema para o final deste período e início do seguinte. Um modelo contínuo sofre constantes e suaves alterações no tempo. Uma situação bastante comum é tratar um sistema contínuo como sendo uma seqüência de pequenas alterações discretas no tempo. Por fim, também são diferenciados os modelos determinísticos dos estocásticos. Os modelos determinísticos não tratam variáveis aleatórias e envolvem relações funcionais exatas. Quando pelo menos uma variável do sistema recebe valores segundo uma distribuição probabilística, diz-se tratar de um modelo estocástico.

- **Entidade**: também chamada de transação, é um elemento essencial para o modelo. Cada tipo de entidade (pessoa, objeto ou informação) possui um ciclo de vida, onde estados ativos e passivos se alternam. Uma entidade pode ser classificada como temporária ou permanente. As entidades temporárias entram no sistema, percorrem o seu ciclo de vida e o abandonam. As entidades permanentes executam a sua função sem abandonar o sistema. A chegada de entidades é gerada por um procedimento externo de acordo com a necessidade imposta pelo estado do sistema. Na visão da simulação discreta convencional, as entidades temporárias são criadas primeiro. Elas participam de atividades e requisitam recursos. Uma vez terminada a atividade, a entidade é removida.
- **Atividade:** é uma seqüência de procedimentos que causa mudança no sistema. A atividade corresponde a um estado ativo, comum a uma ou mais entidades. A duração de uma atividade pode ser determinística ou estocástica. Assume-se que uma atividade é indivisível. Uma vez iniciada, ela não é mais interrompida. Segundo REDDY (1986), as atividades podem ser físicas ou cognitivas. As físicas são as formas convencionais de atividades, em torno das quais a simulação tem sido

desenvolvida. As cognitivas são formas de tomada de decisão racional e inteligente. Da mesma forma que as atividades físicas, as atividades cognitivas possuem tempo finito de duração, que pode ser aleatório ou dependente das informações e/ou decisões do sistema.

- **Atributos**: São as propriedades que caracterizam cada entidade. Os atributos possuem escopo local, ou seja, cada entidade possui um conjunto particular de valores para seus atributos.
- **Evento**: as atividades são iniciadas e terminadas por eventos. Eventos são instantes no tempo, enquanto atividades possuem durações no tempo. Evento é o ponto no tempo no qual acontece alguma mudança no sistema modelado. O processamento de um evento é realizado por uma rotina. Normalmente a ocorrência de um evento dispara a ocorrência de outros eventos. Da mesma forma que as atividades, os eventos podem ser físicos ou cognitivos. As rotinas associadas aos eventos físicos são análogas às rotinas dos eventos tradicionais da simulação discreta. Os eventos cognitivos envolvem processamento de conhecimento, análogo ao procedimento cognitivo desempenhado pelo tomador de decisão. Rotinas de eventos cognitivos irão conter conhecimentos, representados por regras de produção e heurísticas, que serão processados quando uma decisão precisar ser tomada (REDDY,1986).
- **Acumuladores**: são variáveis que permitem medir o desempenho do sistema. Possuem esse nome porque acumulam valores no tempo. Esses valores recebem tratamento e geram as estatísticas da simulação.
- **Relógio**: é uma variável que marca o tempo da simulação.
- **Lista de eventos futuros**: estrutura de dados que armazena os eventos previstos para ocorrer no futuro. Esses eventos são armazenados por ordem cronológica de ocorrência.
- **Cenário**: uma configuração estruturada de parâmetros (dados de entrada) e de entidades do sistema. Pode-se construir vários cenários

para testar várias hipóteses. Por exemplo, compra de um novo equipamento e conseqüente ampliação da capacidade produtiva, ampliação da planta industrial para comportar um novo *mix* de produtos e novas áreas de armazenagem, alteração do regime de turnos de trabalho, automatização da movimentação de material em processo, variações nos valores de demanda, variações no *lead time* e muitos outros cenários.

- **Replicação**: é a execução do modelo no computador.
- **Rodada**: é o intervalo de tempo compreendido entre o início e o fim da replicação.
- **Variáveis de estado**: é o conjunto de variáveis que identificam o estado do sistema em um determinado instante de tempo.
- **Recurso**: é uma entidade estática do modelo que atende entidades dinâmicas. Esse atendimento pode ser feito simultaneamente a várias entidades. Uma entidade pode solicitar parte da capacidade ou a capacidade total do recurso (TORRES, 2001).
- **Filas**: constituem locais de espera onde as entidades dinâmicas aguardam sua vez de seguir através do sistema. As filas podem ser chamadas de áreas de espera ou pulmões (*buffers*). Depois de dar entrada na fila a entidade é retirada seguindo algum tipo de critério, por exemplo, FIFO (*First In First Out*), LIFO (*Last In Last Out*), HVF (*Highest Value First*), LVF (*Lowest Value First*) ou um critério aleatório.

2.6 Uma Metodologia para desenvolver Simulações

Em linhas gerais a metodologia de trabalho adotada pela simulação de sistemas não difere da metodologia utilizada pela pesquisa operacional, mesmo porque a simulação é uma técnica da pesquisa operacional que, junto com a programação linear, ocupa um lugar de destaque em termos de utilização prática. Os passos que orientam o desenvolvimento de um processo de simulação e os pontos de controle desse processo são apresentados na Figura 2.1. Antes de comentar individualmente cada passo, é preciso destacar a importância de se montar, a

priori, a equipe de trabalho. Essa equipe deve agregar competências nas áreas de modelagem, de computação, de análise estatística e de gerenciamento de projetos. É importante comentar também que a tarefa de coleta de dados não é apresentada como uma etapa específica, pois ela faz parte de todos os passos do processo de simulação.

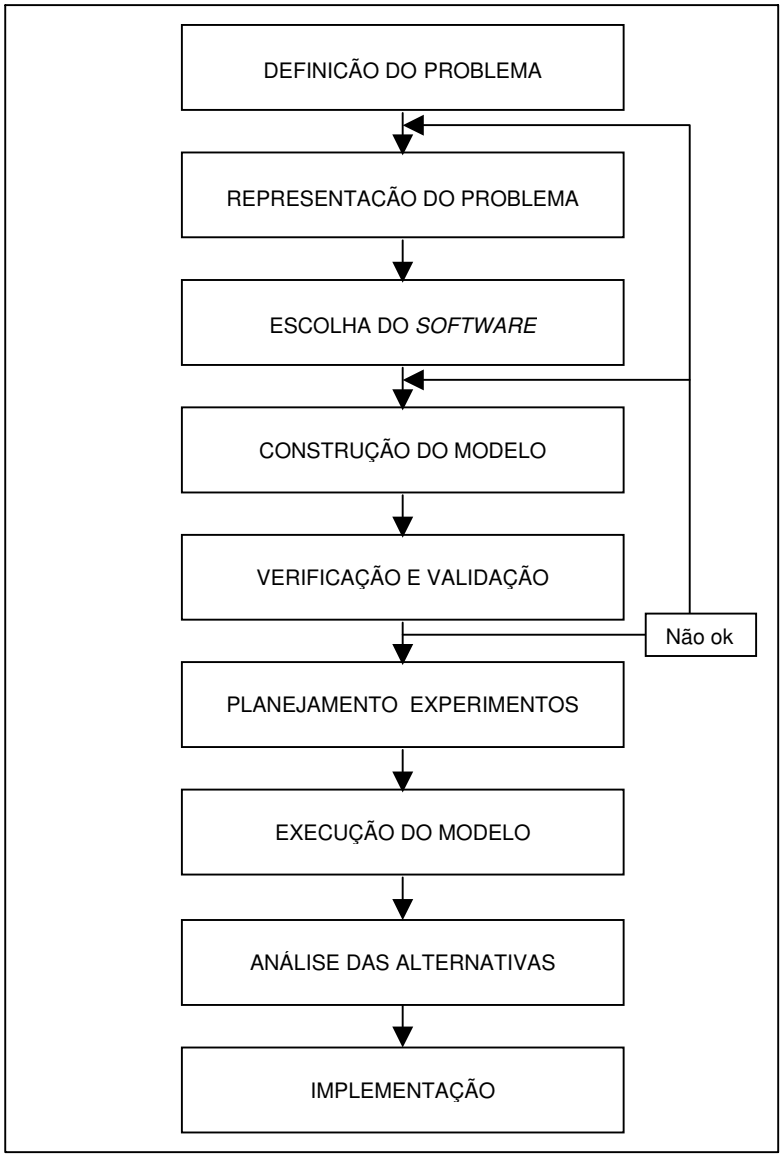


Figura 2.1 – Passos da Simulação

As atividades componentes da metodologia de simulação são:

(a) Definição do Problema e do Plano de Estudo

Não é possível solucionar um problema sem conhecê-lo a fundo. Como primeiro passo é preciso definir claramente os objetivos relacionados ao problema a ser resolvido. Não é raro acontecer do pesquisador resolver corretamente o problema errado. SHANNON (1975) fez uma afirmação que se encaixa perfeitamente nos dias de hoje. Ele diz: "...milhões de dólares são gastos anualmente trazendo respostas sofisticadas e elegantes para questões erradas".

A formulação de um problema é um processo contínuo através do estudo do seu contexto. Novas informações surgem e trazem novas restrições, alteram objetivos e fornecem novas alternativas de investigação. Essas informações devem ser utilizadas para atualizar continuamente a formulação e definição do problema. A formulação do problema é uma etapa vital para todo o processo de simulação. Nessa fase cada informação relevante deve ser analisada a fim de definir o problema no nível de objetivos, restrições e complexidade.

As situações-problema contêm, normalmente, um grande número de variáveis, parâmetros, relacionamentos funcionais, restrições e objetivos. Entretanto, se somente alguns aspectos são de interesse do estudo, muitas características do sistema real podem ser ignoradas. Trata-se, portanto, de abstrair da situação real os aspectos relevantes, formando uma versão idealizada do mundo real.

O projeto de simulação deve ser iniciado com o desenvolvimento de um modelo simples. Na medida em que as metas estabelecidas são atingidas, novos problemas são identificados. Isso levará a revisões do modelo e, conseqüentemente, melhoria da representatividade do mesmo, aproximando-o do sistema real. Trata-se, portanto, de um enriquecimento contínuo do modelo.

KELTON, SADOWSKI & SADOWSKI (1998) colocam algumas questões para facilitar a identificação dos objetivos da simulação, para descrever o sistema de estudo, estabelecendo o grau e a necessidade ou não de animação, para especificar as entradas e saídas do modelo e estabelecer os elementos de gestão

do projeto de simulação. Esta lista, mesmo não sendo exaustiva, pode oferecer uma idéia geral do nível de detalhes necessários. As questões são as seguintes:

- O que deve conter o modelo de simulação?
- Qual o nível de detalhes necessário?
- Quais os recursos primários do sistema real?
- Quais as atividades e processos associados a esse sistema?
- Existem diagramas de fluxo e de processo disponíveis?
- Eles estão atualizados?
- Esses diagramas são seguidos ou são somente prescrições?
- Sob quais condições eles não são seguidos?
- Existem restrições físicas, tecnológicas ou legais na operação do sistema?
- Essas restrições podem ser alteradas?
- Os procedimentos do sistema estão bem definidos?
- Como são tomadas as decisões?
- Existem dados disponíveis?
- Quem vai cuidar do projeto de levantamento de dados?
- Quem vai cuidar da coleta de dados?
- Qual tipo de animação é necessário?
- Quem vai verificar e validar o modelo, e como o fará?
- Quais as saídas necessárias?
- Quão geral ou específico deve ser o modelo?
- Quem vai fazer as análises de resultados?
- Quantos e quais cenários devem ser considerados?
- Quais os pontos de controle do projeto (*milestones*)?

(b) Representação do Problema

Após estabelecer a situação a ser tratada, deve-se passar para a fase de representação do seu contexto. Trata-se, no caso da simulação, de descrever o sistema real em um diagrama de fluxo lógico. O diagrama representará o problema através de associações feitas entre as entidades e as atividades.

Um diagrama bastante conhecido é o Diagrama Ciclo Atividade (*Activity Cycle Diagram - ACD*). Esses diagramas são constituídos por filas e atividades, cujos relacionamentos estabelecem o ciclo de vida de cada entidade participante do sistema. Detalhes da estrutura ACD podem ser obtidos na obra de PIDD (2001).

A utilização de um ACD é indicada para sistemas que possuem forte estrutura de filas. Todas as entidades envolvidas, a cada instante da simulação, estarão sempre em uma das situações: ativa (participando da execução de uma atividade) ou esperando numa fila. O estado de um sistema modelado é definido pela situação de cada entidade componente. Um ciclo de vida é utilizado para descrever uma seqüência de situações nas quais uma entidade deverá atuar.

Em linhas gerais a representação dos diagramas ACD é feita através de círculos e retângulos. Os retângulos representam atividades (estados ativos) que envolvem, normalmente, a co-participação de diferentes entidades. As durações dessas atividades são definidas baseadas em funções, probabilísticas ou não. Os círculos representam filas (estados passivos), compostas de entidades que aguardam alguma mudança no sistema. Essas mudanças acontecem com a ocorrência de um evento, que liberará entidades que poderão ser utilizadas na realização de uma atividade comum.

(c) Escolha da Ferramenta Computacional

Existem basicamente quatro tipos de ferramentas computacionais que podem ser utilizadas no desenvolvimento de simuladores:

- Linguagens de Aplicação Geral;
- Linguagens Específicas de Simulação;
- Geradores Automáticos de Códigos de Simulação; e
- Ambientes de Simulação.

Existe uma contraposição entre flexibilidade e facilidade de desenvolvimento, de acordo com a ferramenta escolhida. Por exemplo, uma linguagem de aplicação geral permite alta flexibilidade no desenvolvimento de um programa de simulação, porém exige grande conhecimento da linguagem e demanda um grande esforço de programação. Por outro lado um ambiente de simulação é muito amigável e

permite fácil utilização por parte do usuário, mas não oferece muita flexibilidade no desenvolvimento.

As linguagens de aplicação geral são mais poderosas ou menos poderosas de acordo com suas características individuais. Linguagens como PASCAL e C são muito utilizadas no desenvolvimento de simulações.

Linguagens específicas de simulação (ou simplesmente linguagens de simulação) proporcionam facilidades no desenvolvimento de modelos. GORDON (1978) e BULGREN (1982) citam as seguintes linguagens de simulação como sendo clássicas: GPSS, SIMSCRIPT, DINAMO, GASP, SIMPAC, CLS e SIMULA. Atualmente, uma linguagem bastante conhecida é o SIMAN (PEGDEN, SHANNON & SADOWSKI, 1995), base do *software* ARENA.

Algumas características das linguagens específicas de simulação são:

- Possuem uma estrutura padrão para criar modelos de simulação;
- Facilitam a conversão de um modelo de simulação em um programa computacional;
- Flexibilizam alterações no programa computacional;
- Facilitam as análises dos resultados da simulação.

Os geradores automáticos de códigos de simulação surgiram para facilitar a tarefa do modelador no processo de programação computacional. A partir de uma representação formal do problema estudado, esse tipo de ferramenta produz um programa fonte, que é uma versão executável do modelo especificado. São exemplos desses geradores: PASSIM (SHEARN, 1990), LANGEN (CROOKES et al., 1986), GERSIMUL (SALIBY, 1991), AUTOSIM (PAUL, 1987), CAPS (CLEMENTSON, 1982). Segundo PAUL (1987) o CAPS foi o primeiro gerador automático de códigos de simulação.

Os ambientes de simulação são sistemas que têm por objetivo auxiliar o usuário no desenvolvimento de modelos de simulação, atuando em várias etapas do processo, e não somente na geração dos códigos. As etapas nas quais um ambiente de simulação normalmente atua são: análise dos dados de entrada, análise dos resultados da simulação, facilidade na programação da animação, facilidade na programação lógica fornecendo conjuntos de comandos de alto nível

e preparação de interfaces amigáveis através de ferramentas computacionais do tipo VISUAL BASIC. São exemplos desses ambientes: KBMC (Knowledge-Based Model Construction) (MURRAY, 1988), KBS (Knowledge-Based Simulation) (REDDY et al., 1986), ARENA (PEGDEN, SHANNON & SADOWSKI, 1995; KELTON, SADOWSKI & SADOWSKI, 1998), AUTOMOD (BANKS, 2000), PROMODEL (HARREL et al., 2000).

Mais recentemente surgiram linguagens de simulação para desenvolver simulações baseadas na Internet (*web based simulations*) que usam, em geral, a linguagem JAVA. Entre os exemplos estão a SILK (THREADTEC, 2003) e a SIMJAVA (SIMJAVA, 2003). NARAYANAN (2000) comenta que, entre as vantagens dessas linguagens, estão a re-usabilidade do código e a distribuição dos modelos.

(d) Construção do Modelo

O desenvolvimento do modelo depende da ferramenta computacional escolhida. O resultado será um simulador tradicional ou um complexo sistema de simulação (por exemplo, um sistema inteligente ou baseado em conhecimento).

Nessa fase são criados os códigos de simulação, por geração automática ou não. A dificuldade, e conseqüente velocidade de desenvolvimento do simulador, está associada à ferramenta computacional utilizada, à complexidade do modelo e à experiência do modelador.

(e) Verificação e Validação do Modelo

A verificação e a validação são duas fases muito delicadas no processo de desenvolvimento de modelos de simulação.

Verificação do modelo é a etapa onde o modelador certifica se o modelo desenvolvido corresponde ao idealizado. Neste ponto é verificado se o modelo foi construído corretamente. Nesta fase busca-se fazer testes exaustivos no simulador. O modelador precisa se convencer de que o simulador não contém erros.

Na verificação de um modelo deve-se variar os valores dos parâmetros de entrada (inclusive utilizando as fronteiras do intervalo de valores) e analisar se os resultados são coerentes. Nesse momento, a simulação agiliza a etapa de testes, tornando-os viáveis em termos de tempo e dinheiro.

A validação é a etapa onde será conferido se o modelo desenvolvido representa bem o sistema real. É a resposta para a pergunta: “Foi desenvolvido o modelo correto?”. A idéia é passar confiança ao usuário, mostrando que a experimentação com o modelo irá gerar resultados que coadunam com a realidade do sistema estudado.

A validação é normalmente conseguida executando o modelo e comparando seus resultados com os oriundos do sistema real. Se os resultados da simulação se aproximarem dos valores reais, dentro do nível de confiança desejado, o simulador será validado.

Segundo ANNINO & RUSSEL (1981) a técnica de validação mais efetiva é apresentar o programa para alguém familiar ao sistema em estudo. Num esquema interativo entre o especialista no funcionamento do sistema real e o modelador, será mais fácil corrigir desvios do modelo em relação ao sistema real.

A validação do modelo é extremamente importante, pois os simuladores normalmente tendem a parecer reais e, tanto o modelador como o usuário, passam a acreditar nele.

(f) Planejamento de Experimentos

O planejamento de experimentos tem dois objetivos: (1) testar muitas vezes o modelo, a baixo custo e de forma rápida, e (2) providenciar uma estrutura para o processo de aprendizado dos investigadores, para que esses possam aprender mais a respeito do sistema.

O planejamento de experimentos pode ser dividido em duas fases: estratégica e tática. No nível estratégico é importante projetar os experimentos visando alcançar as metas estabelecidas. Portanto deve-se combinar os valores dos parâmetros

que “otimizarão” as variáveis de respostas, e explicar as relações entre as variáveis de respostas e os fatores controláveis do sistema.

Estabelecido o projeto dos experimentos, a fase tática determinará como cada um dos testes especificados nos experimentos será executado. Para isso devem ser definidas as condições iniciais, e como elas afetam o equilíbrio do sistema. O pano de fundo dessa fase é a necessidade de diminuir a variância das respostas e, ao mesmo tempo, buscar minimizar o tamanho das amostras necessárias.

Os cenários são tratados individualmente, com a preocupação de se estabelecer exatamente os pontos de início e fim da simulação, para efeito de análises estatísticas. Esses pontos correspondem à fase transiente (*warm-up*) e ao tempo total de simulação da rodada (*length of simulation*).

(g) Execução do Modelo

É a fase que consiste em executar o modelo computacional para os diversos cenários de simulação estabelecidos.

A simulação, a partir de situações iniciais diferentes, oferece linhas alternativas de ação para o tomador de decisão. E como a fase de execução do modelo é, normalmente, rápida, pode-se abusar do número de cenários a serem rodados. Isso é bom na medida que amplia a oferta de opções de cenários.

(h) Análise das Alternativas de Ação

Esta etapa consiste em fazer inferências a partir dos dados gerados na fase de execução do modelo. É uma fase crítica, demorada, difícil e importante. Uma saída de simulação é uma grande quantidade de números, que precisa ser tratada e analisada. A análise dos resultados da simulação é realizada por especialistas pouco disponíveis. Para suavizar os problemas enfrentados nessa fase surgiram ferramentas computacionais, do tipo *statfit* (*statistical fitness*), já disponíveis em boa parte das linguagens, geradores e ambientes de simulação visuais (gráficos e imagens).

Essas ferramentas fazem toda a modelagem dos dados partindo do estabelecimento da amostra representativa da população estudada. O resultado da análise dos dados de entrada (*input analyzer*) indica uma distribuição de probabilidade hipotética que é transformada em uma função geradora de números no simulador. O *statfit* tem a preocupação de verificar o grau de aderência da curva hipotética com a distribuição dos dados reais. Para isso são realizados testes de aderência, por exemplo do tipo Qui-Quadrado e KS (*Kolmogorov Smirnov*).

O *statfit* também analisa os dados de saída, mostrando as estatísticas dos diversos elementos da simulação: entidade, localização, estação, fila, recurso, variáveis e atributos. Considerando se o sistema é terminal ou não terminal, essa ferramenta utiliza técnicas de médias móveis para estabelecer o período de comportamento transiente do sistema, utiliza correlogramas para determinar o tamanho e quantidade das replicações e utiliza intervalos de confiança para analisar a consistência dos resultados da simulação. Existem muitas publicações que tratam da análise de dados da simulação. Dentre elas destacam-se: LAW & KELTON (1991) e KELTON, SADOWSKI & SADOWSKI (1998).

Tão somente a disponibilidade de ferramentas visuais não soluciona o problema de análise de resultados. É necessário possuir profissionais com condições de realizar análises concretas ao final do processo de simulação. Nos últimos anos tem ocorrido a utilização de sistemas especialistas para desempenhar o trabalho dos analistas de resultados de simulação. Esses sistemas especialistas, contendo o conhecimento de expertos na área em estudo, analisam os resultados fornecidos pelo simulador, orientando o tomador de decisão.

(i) Documentação e Implementação

Uma das grandes falhas nos projetos, em geral, é a interface que apresenta os resultados para o usuário. Isso é uma falha de desenvolvimento. O pouco tempo gasto na implementação normalmente não é suficiente para as tarefas existentes nessa fase. Fazem parte dessas tarefas: ajustes do modelo, treinamento do usuário e a validade dos resultados. Essa última tarefa, que só é possível com a

implementação, tende a ser, segundo SHANNON (1975), o problema mais difícil a ser enfrentado pelo analista.

A documentação está intimamente ligada à implementação. Além de facilitar a implementação e a possível necessidade de alterações no modelo, auxilia o modelador em futuros projetos, além de forçá-lo a questionar seus próprios procedimentos.

2.7 Comentários finais sobre simulação de sistemas

A simulação de sistemas faz análises de problemas. Ela serve como ferramenta para a análise do comportamento do sistema sob condições especificadas pelo usuário.

A simulação pode evitar problemas que ocorrem quando se faz experimentações com o sistema real. Por exemplo, na simulação:

- Não é necessário interromper as operações do sistema real;
- É fácil manter as mesmas condições de operação para cada réplica do experimento;
- É possível explorar muitos tipos de alternativas na experimentação;
- O consumo de tempo e dinheiro é extremamente menor para se trabalhar o mesmo tamanho de amostra.

SHANNON (1975) coloca que o analista deve considerar o uso de simulação sempre que:

- Uma formulação matemática completa do problema não exista, ou métodos analíticos para resolver o modelo matemático não foram ainda desenvolvidos;
- Métodos analíticos estejam disponíveis, mas os procedimentos necessários para a utilização dos mesmos são tão complexos e árduos que a simulação acaba fornecendo um método mais simples de solução;

- Soluções analíticas existam e sejam viáveis em termos de complexidade, mas estão além da habilidade matemática do pessoal disponível. O custo de desenvolvimento, teste e execução da simulação deve ser avaliado, comparativamente ao custo de obtenção de ajuda externa;
- A Simulação pode ser a única opção possível dadas as dificuldades de observação de fenômenos no ambiente real.

Uma vantagem adicional da simulação é a sua contribuição para se compreender melhor e aprender a respeito do sistema. O desenvolvimento e a utilização de um modelo de simulação permitem ao projetista enxergar e manipular o sistema. Isso leva a um crescente entendimento do mesmo, permitindo que modelos simples de simulação sofram alterações até se tornarem complexos o suficiente para representar bem a situação estudada.

3 Sistemas baseados em conhecimento

Inteligência Artificial (IA), em particular Sistemas Especialistas (SE), são tópicos importantes trabalhados na comunidade científica. As tentativas de utilização de conceitos e *softwares* de IA no desenvolvimento de simulações permitem construções fáceis e rápidas de modelos, aceleram as validações destes, facilitam as análises dos resultados da simulação, e rodam com a base de conhecimento interna ao sistema/pacote. Esse último fato permite que decisões sejam tomadas sem interromper a simulação.

A seguir é apresentada uma abordagem sobre inteligência artificial e sistemas especialistas e uma comparação entre a simulação tradicional e a simulação baseada em conhecimento, também chamada de simulação inteligente.

3.1 Inteligência Artificial e Sistemas Especialistas

A inteligência artificial, segundo SHANNON (1985), tem suas raízes nos ensaios sobre o poder dos computadores, realizados por A.M.Turing em 1950. Existem várias definições para inteligência artificial, algumas filosóficas, outras mais pragmáticas, dependendo muito do campo de atuação dos autores das mesmas, por exemplo :

- Inteligência artificial é o estudo de como fazer computadores realizarem tarefas para as quais, até o momento, o homem faz melhor (RICH, 1983).
- Inteligência artificial diz respeito ao desenvolvimento de teorias e técnicas requeridas por uma máquina computacional para eficientemente perceber, pensar e agir com inteligência em ambientes complexos (FOX, 1990).

- IA é o campo da ciência da computação que diz respeito ao raciocínio simbólico, inferência e solução de problemas (ROBERTS, 1983).
- IA é uma área da ciência da computação que objetiva desenvolver programas computacionais inteligentes. Esses programas são: solucionadores de problemas; sistemas computacionais que aprendem com a experiência; sistemas que entendem linguagens; sistemas que interpretam esquemas visuais. Enfim, programas que se comportam de uma maneira que seria considerada inteligente se observada num humano (WATERMAN, 1985).

Um ponto comum e intrínseco às definições, aqui apresentadas, é a necessidade de compreensão dos processos humanos de aprendizagem e a modelagem dos mesmos, permitindo a emulação desses processos numa máquina computacional.

Dado que a inteligência humana é muito complexa, o que a IA consegue atingir é a modelagem de aspectos particulares dela em relação a domínios específicos (DEWHURST, 1990).

Os vocábulos "inteligência artificial" e "baseado em conhecimento", de acordo com REDDY (1987), parecem implicar que sistemas baseados nesses conceitos são muito poderosos, e problemas complexos podem ser solucionados com pequeno esforço por parte do usuário. A utilização de paradigmas de IA não torna, automaticamente, um sistema forte. Mas esses paradigmas abrem novas possibilidades para construção de sistemas poderosos.

O objetivo da IA é compreender os processos humanos de aprendizagem, modelá-los e reproduzi-los num computador. Dessa forma a IA pode ser vista sob o ângulo da abordagem científica, que busca entender os mecanismos da inteligência humana, ou sob a visão da engenharia, que quer dotar o computador de capacidades inteligentes (DOUKIDIS, 1987).

Segundo SHANNON (1985), a IA atua em dois grandes campos :

- Imitação das habilidades humanas, como visão, fala, gestos, etc.; e
- Duplicação de resultados estabelecidos pelo homem através de sua habilidade e experiência.

COSTA (1991), a título de exemplificação, cita algumas áreas de pesquisa da IA:

- Solucionadores de Problemas (normalmente, através de abordagens heurísticas que, sem analisar todas as alternativas, mostra um caminho com boa chance de êxito);
- Raciocínio Lógico (dedução);
- Processamento de Linguagem Natural (tradução e compreensão de textos);
- Robótica e Visão (manipulação de objetos, seqüenciamento de tarefas, reconhecimento de padrões);
- Programação Automática (geradores automáticos de programas computacionais);
- Aprendizagem (o sistema aprendendo com sua própria experiência, melhorando seu desempenho);
- Sistemas Especialistas (sistemas que armazenam, em um programa computacional, o conhecimento de uma área específica de atuação, utilizando-o como suporte à tomada de decisão).

Portanto, inteligência artificial é um termo abrangente, que inclui diversas áreas, mas como cita MOSER (1986), uma das áreas que causou grande impacto nas ciências, principalmente administrativas, foi a de sistemas especialistas.

Da mesma forma que para IA, são muitas as definições para sistemas especialistas. Por exemplo:

- Sistema especialista é um *software* capaz de acessar e processar logicamente as informações que uma pessoa, especialista numa

área particular de conhecimento, utilizaria para tomar uma decisão (MOSER, 1986).

- Sistema especialista é um programa computacional que imita o comportamento de exploração (busca) dos especialistas humanos na solução de problemas (FOX, 1990).
- Sistemas especialistas são sistemas de processamento de informações que utilizam uma combinação de raciocínio simbólico com processamento de dados (KERCHOFFS & VANSTEENKISTE, 1986).
- São sistemas projetados para compilar a experiência de um certo número de especialistas, de um campo de atuação bem definido, numa série de regras. Essas regras serão utilizadas para gerar inferências e sugerir ao usuário um curso de ação no tratamento de um problema (SHANNON, 1985).
- É um programa computacional que utiliza conhecimentos de especialistas para alcançar altos níveis de desempenho numa área restrita de conhecimento. Esses programas representam o conhecimento simbolicamente, examinam e explicam seus processos de raciocínio, e tratam com problemas de áreas que requerem anos de treinamento especial para que um homem possa dominá-lo (WATERMAN, 1985).
- É um sistema computacional que incorpora conhecimento organizado referente a alguma área específica do conhecimento humano, suficiente para ser um consultor habilidoso e de custo acessível (FINLAY,1990).

Nas diversas definições apresentadas vários pontos são comuns e, de certa forma, caracterizam os sistemas especialistas (COSTA, 1991). Por exemplo:

- São sistemas computacionais;
- Processam informações obtidas junto a especialistas, com áreas de atuação restritas e bem definidas;
- Trabalham com regras e dados incertos;
- Utilizam raciocínio qualitativo;
- Explicam a linha de raciocínio;
- Justificam suas conclusões;
- Possuem interação amigável.

Segundo KERCKOFFS & VANSTEENKISTE (1986), um sistema especialista (SE) organiza e conserva conhecimento, permitindo ao usuário acessar um conhecimento só disponível num especialista, que nem sempre pode estar presente. O usuário pode comparar seus resultados com os do SE e validar seus próprios processos de raciocínio.

Portanto, sistemas especialistas são programas computacionais projetados para extrair o conhecimento do(s) especialista(s) numa área restrita de atuação. Esse conhecimento advém da experiência adquirida pelo(s) experto(s) no decorrer dos anos de trabalho. O conhecimento obtido é então codificado de uma forma que permite inferências e conseqüentes sugestões de cursos de ação para o usuário. As justificativas para as sugestões dadas devem ser apresentadas de forma clara e consistente. O sistema deve ser o mais amigável possível.

Um SE não precisa, necessariamente, conter em si mesmo os fatos e procedimentos heurísticos que o especialista conhece. Ele precisa reproduzir a ação do especialista no tratamento do assunto.

Para atingir seu objetivo, um SE deve conter os seguintes componentes básicos:

(i) Uma Base de Dados Global;

(ii) Uma Base de Conhecimento;

(iii) Uma Estrutura de Inferências.

A Base de Dados incorpora o conhecimento declarativo, representando a realidade do problema tratado. A Base de Conhecimento contém os fatos e heurísticas representativas do conhecimento do experto. Trata-se do conhecimento relacional. A Estrutura (ou Máquina) de Inferências estabelece como os dados e conhecimentos serão manipulados para que seja possível atingir a solução do problema.

Pode-se dizer que o desenvolvimento de um SE é apropriado (ou pode ser utilizado) quando a situação problema atende algumas pré-condições como as citadas por IGNIZIO (1990) :

- Deve existir um domínio experto. Especialistas ou registros de decisões expertas devem estar disponíveis;
- O domínio experto apresenta um grau de experiência muito acima do disponível no sistema tratado;
- O processo utilizado pelo especialista é caracterizado, principalmente, por procedimentos heurísticos;
- Os valores dos atributos são principalmente símbolos, ao invés de números;
- Existe a necessidade de uma base de conhecimento transparente;
- Existe a necessidade de tomar decisões na ausência de informações completas;
- O problema tratado explode em termos combinatórios.

WATERMAN (1985) faz a seguinte colocação quanto a optar ou não por um SE. Ele afirma que deve-se optar por um SE somente se o seu desenvolvimento for possível, justificado e apropriado.

Entende-se que o desenvolvimento do SE é possível se:

- Existirem verdadeiros especialistas;
- Eles concordarem entre si;

- Eles forem capazes de explicar os métodos que os levam às escolhas dos planos de ação; e
- As tarefas não exigirem bom senso.

Entende-se que o desenvolvimento do SE é justificado se:

- O retorno financeiro for compatível;
- Houver uma disponibilidade pequena de especialistas;
- A especialidade for requisitada em vários locais diferentes;
- O sistema for trabalhar em ambiente hostil; e
- Houver possibilidade de perda de conhecimento (troca de pessoal, por exemplo).

Entende-se que o desenvolvimento do SE é apropriado se:

- A natureza do problema exige manipulação de símbolos e/ou tratamento heurístico; e
- O escopo do problema for bem particular e restrito.

O processo de construção de um SE é muito similar ao processo de análise de sistemas (TRIPPI & TURBAN, 1990). Ele é composto de quatro tarefas básicas:

(i) Aquisição de Conhecimento, que é o trabalho de extrair o conhecimento de especialistas e de outras fontes;

(ii) Representação do Conhecimento, que corresponde à tarefa de formalizar e programar o conhecimento, além de organizá-lo em bases de conhecimentos;

(iii) Inferência do Conhecimento, que é a programação de um mecanismo que permita inferências ou raciocínios;

(iv) Criação de uma facilidade de explanação, que explique as conclusões que foram indicadas.

Para projetar um SE, portanto, deve-se extrair o conhecimento associado ao domínio do especialista e codificá-lo numa forma que o computador possa manipulá-lo. Esse procedimento é chamado de representação do conhecimento.

3.2 Representação do Conhecimento

Representação do conhecimento corresponde ao processo de formalização do conhecimento adquirido. O resultado dessa formalização é um conjunto de símbolos que, seguindo uma convenção sintática e semântica, é capaz de descrever o conhecimento que integrará o sistema especialista. A representação do conhecimento deve expressar todo o conhecimento relevante do especialista de uma forma facilmente compreensível.

Os sistemas especialistas utilizam-se de programações declarativas, não procedurais. A abordagem não procedural facilita a manutenção do sistema, permitindo que novos conhecimentos sejam incorporados sem afetar o fluxo de controle existente (HILL & ROBERTS, 1987).

3.2.1 Regras de Produção

Os especialistas tendem a expressar seus métodos de solução de problemas através de regras do tipo "Situação-Ação". Essas regras são representadas pela seguinte estrutura: **Se** <*Condição*> **Então** <*Ação*>. A esse formalismo é dado o nome de regras de produção, sistemas de produção, sistemas baseados em regras, ou simplesmente regras.

Num sistema especialista baseado em regras o conhecimento do domínio é representado por um conjunto de regras que são checadas através dos fatos e conhecimentos a respeito da situação presente. Quando a parte **Se** da regra é satisfeita pelos fatos, a ação especificada pela parte **Então** da regra é executada.

A ação de uma regra pode modificar a base de conhecimento, adicionando um novo fato ou modificando um fato já existente na base. Isso normalmente acarreta a execução de uma outra regra, repetindo o processo até que um ponto de parada seja atingido.

O casamento, das regras com os fatos, é feito pela máquina de inferência. Há duas importantes maneiras de controlar as inferências: o Encadeamento Progressivo (*Forward Chaining*) e o Encadeamento Regressivo (*Backward Chaining*).

No encadeamento progressivo, também conhecido como raciocínio orientado por dados, a parte da condição estabelece as combinações dos fatos que deverão ser comparados com os dados disponíveis. Caso ocorra o casamento dos fatos com os dados, a regra será executada, desencadeando um processo semelhante, que se repetirá até atingir a meta estabelecida no SE, ou ser abortado por um critério de parada.

No encadeamento regressivo, também chamado de raciocínio orientado por metas, a parte da ação é utilizada para fazer os casamentos. Estabelecido um objetivo final, a máquina de inferência procura uma regra cuja ação satisfaça o objetivo. Encontrada uma regra, a máquina de inferência passa a examinar o(s) atributo(s) da parte **se** da regra. Cada atributo será considerado um objetivo intermediário. A partir daí procura-se uma nova regra cuja ação corresponda a esse novo objetivo, e repete-se o processo até que não existam mais regras a serem examinadas.

COSTA & GURGEL (1984) foram didáticos ao comparar os encadeamentos progressivos e regressivos aos processos de raciocínio utilizados pelos famosos detetives novelescos Sherlock Holmes e Hercule Poirot, respectivamente. O raciocínio utilizado por Sherlock Holmes, nas suas famosas deduções, é progressivo. Ele parte dos fatos e vai deduzindo outros fatos, até alcançar a

solução do problema. Já Hercule Poirot, o famoso detetive belga criado por Agatha Christie, parte de uma lista de hipóteses e procura reunir fatos que lhe permitam concluir que uma das hipóteses é verdadeira. Portanto segue o raciocínio regressivo.

Muitas máquinas de inferências permitem que sejam atribuídas prioridades às regras, facilitando a ordem de procura das mesmas. Um fator importante é que a ordem das regras não interfere na solução final do problema. Isso permite uma alta flexibilidade ao engenheiro do conhecimento que pode adicionar, retirar uma regra ou re-arranjar o conjunto de regras, sem se preocupar com a ordenação das mesmas.

Uma outra característica que valoriza muito as regras de produção é a sua expressividade. Uma regra, por si só, já é auto-explicativa. A sua estrutura permite uma fácil compreensão do seu conteúdo.

As regras de produção constituem o formalismo mais simples e mais utilizado na representação do conhecimento. Podem ser entendidas como um conjunto de heurísticas representativas das técnicas utilizadas pelo especialista na solução de problemas. Os solucionadores de problemas, por envolverem heurísticas, não são “otimizantes” mas são bons, pois, apesar de não necessariamente pesquisarem a solução ótima, chegam a uma solução que satisfaz a maioria das restrições e condições do problema. As regras de produção são muito utilizadas em sistemas especialistas comerciais (WIDMAN & LOPARO, 1990). A Figura 3.1 apresenta alguns exemplos de regras e o ciclo interpretador das mesmas através de uma seqüência de casamento e execução de regras.

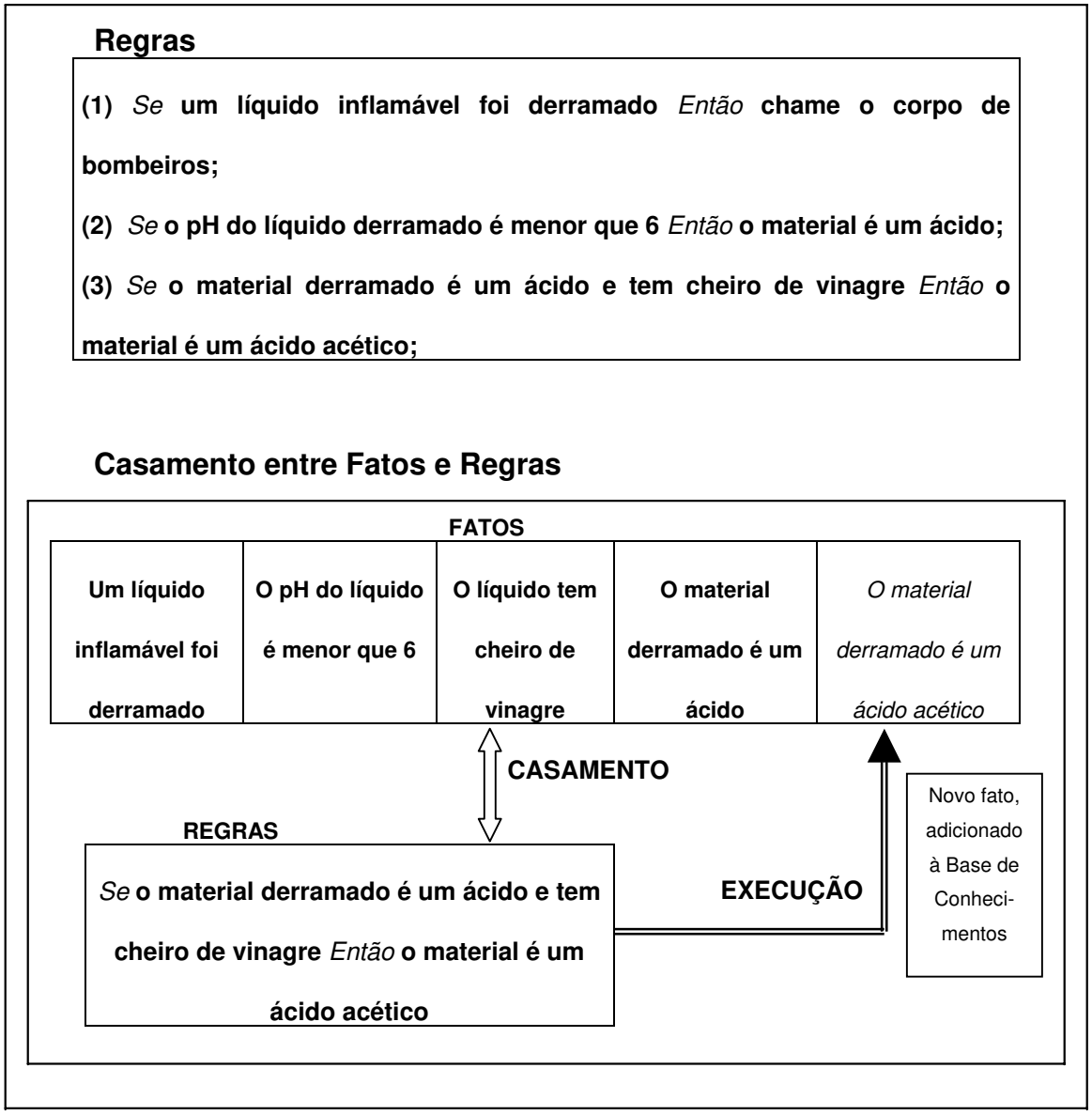


Figura 3.1 – Exemplo de casamento e execução de regras de produção

Fonte - Adaptada de WATERMAN (1985)

3.2.2 Redes Semânticas

Conforme o próprio nome indica, rede semântica é um método de representação do conhecimento que se baseia numa estrutura de rede. Uma rede semântica consiste de pontos chamados **Nós** interligados por setas chamadas **Arcos**. Os arcos descrevem as relações existentes entre os nós.

Os **Nós**, numa Rede Semântica, podem representar objetos, conceitos ou eventos. Basicamente existem dois tipos de nós: individuais e genéricos. Os nós individuais representam descrições ou afirmações referentes a uma instância individual de um objeto. Os nós genéricos referem-se a uma classe ou categoria de objetos.

Os **Arcos** são definidos de acordo com o conhecimento que será representado. Na representação de hierarquias é muito comum utilizar arcos do tipo "É UM" ("IS A") e "FAZ PARTE" ("HAS-PART"). Os arcos representam relações transitivas. Portanto, a partir de premissas, é possível inferir fatos que não estão explicitados na rede.

Utilizando-se as relações representadas pelos arcos é possível estabelecer uma propriedade da rede semântica chamada de hierarquia herdada. Essa propriedade permite que itens pertencentes a níveis menores da rede herdem características dos itens que estão nos níveis maiores. Isso facilita a representação da rede, evitando a ocorrência de repetições.

As redes semânticas são muito utilizadas para representar conhecimento em domínios que possuem uma taxonomia bem estabelecida. A Figura 3.2 mostra um exemplo de rede semântica.

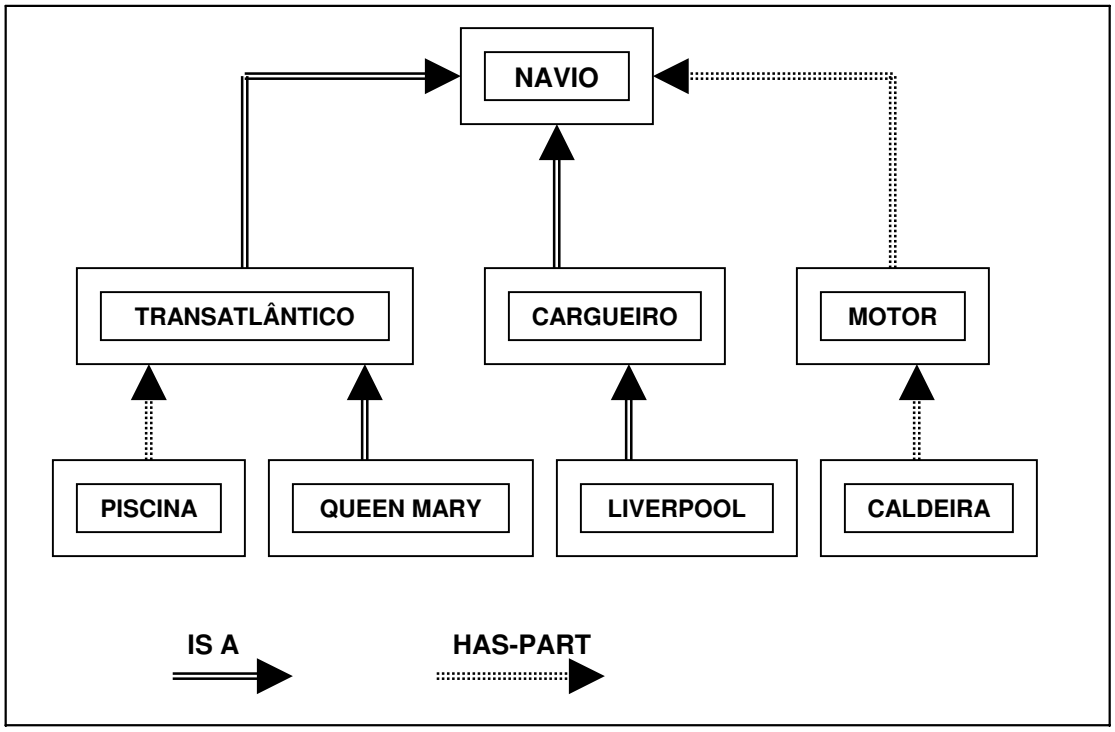


Figura 3.2 – Exemplo de Rede Semântica

Fonte - Adaptada de WATERMAN (1985)

3.2.3 *Frames*

Um *Frame* é organizado de uma forma muito parecida com uma rede semântica. É uma descrição estruturada de um objeto ou classe de objetos. Trata-se de uma rede de nós com as relações, entre eles, organizadas de forma hierárquica. Os nós superiores representam conceitos gerais, enquanto os nós inferiores são instâncias mais específicas desses conceitos. "Conceitualmente um *Frame* representa um item, que é um objeto físico ou um conceito, ou seja, uma idéia. O conteúdo do *Frame* descreve o item através de suas características, suas propriedades ou seus comportamentos" (TORSUN, 1995).

Os *Frames* são considerados como um processo evolutivo das redes semânticas. Os *Frames* podem ser vistos como nós complexos que possuem compartimentos, onde são guardadas as informações associadas aos atributos de cada nó. Esses compartimentos recebem o nome de "S*lots*". Cada nó, portanto, é caracterizado por uma coleção de atributos e seus valores.

O valor de um S*lot* é o significado (conteúdo) do mesmo. Esses valores são controlados por restrições que limitam o tipo de valor, e o intervalo no qual o mesmo pode atuar.

Cada S*lot* possui procedimentos (rotinas computacionais) que são executados quando os valores de um atributo são alterados. Os procedimentos indicam como utilizar o atributo.

Existem muitos tipos de procedimentos associados aos S*lots*. Os mais conhecidos são:

- "*IF-Added Procedure*": O procedimento é executado toda vez que novas informações são colocadas no S*lot*.
- "If-Removed Procedure": O procedimento é executado quando informações são removidas do S*lot*.
- "If-Needed Procedure": O procedimento é executado quando são requisitadas informações do S*lot* e o mesmo se encontra vazio.

Esses procedimentos monitoram a atribuição de informações para os nós, fazendo com que ações apropriadas ocorram quando acontecerem as mudanças dos valores. A Figura 3.3 apresenta um exemplo de *Frame*.

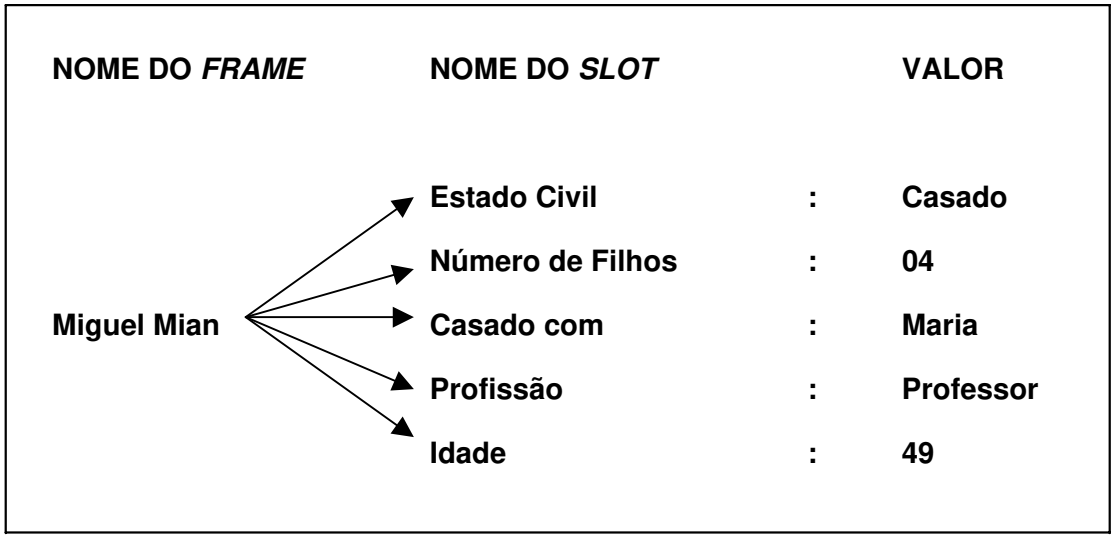


Figura 3.3 – Exemplo de *Frame*

3.3 Simulação Tradicional e Simulação Baseada em Conhecimento

Muitos pesquisadores, entre eles DOUKIDIS (1987) e O'KEEFE (1986), consideram a simulação de sistemas e a inteligência artificial, em particular os sistemas especialistas, como tecnologias que possuem uma forte similaridade metodológica.

Na verdade, tanto a simulação como a IA, tentam modelar a realidade, representando os objetos e suas interrelações. DOUKIDIS (1987) afirma que nos anos 60 e 70 a IA se valeu dos mesmos conceitos utilizados pela simulação para solucionar problemas de configuração de computadores e diagnósticos médicos .

Para O'KEEFE (1986a) o que torna os métodos de simulação e de sistemas especialistas similares é que eles são baseados em uma representação modular de um sistema, com um mecanismo que dirige essa representação. Em outro trabalho, o mesmo O'KEEFE (1986b) complementa dizendo que simulação e SE

são similares pois ambos enfrentam situações de decisão e freqüentemente empregam medidas de incerteza nas análises.

Outros pesquisadores como MURRAY & SHEPPARD (1988), MOSER (1986) e REDDY (1987), enxergam a simulação de sistemas e os sistemas especialistas como tecnologias complementares, que podem ser integradas para fornecer uma poderosa ferramenta de apoio à tomada de decisão no tratamento de problemas complexos. Essa integração é alcançada combinando-se a habilidade da simulação para prever valores de conjuntos complexos de variáveis, com a habilidade de raciocínio dos sistemas especialistas.

A união da simulação tradicional com os conceitos de inteligência artificial gera uma ferramenta de apoio a tomada de decisão chamada Simulação Baseada em Conhecimento (MOSER,1986).

Em síntese as metodologias adotadas pela simulação e pelos sistemas especialistas são realmente similares. Porém, nada impede que ambas trabalhem em conjunto utilizando as suas características potenciais. Formas de atuação entre simulação e sistemas especialistas foram muito bem abordadas por O'KEEFE (1986a).

3.3.1 Simulação & Sistemas Especialistas: Atuação Conjunta

Na Figura 3.4 são apresentadas várias formas de combinar sistemas especialistas e simulação de sistemas. Uma primeira maneira, até mesmo natural, de visualizar a simulação e o SE atuando de forma conjunta é colocando um sistema especialista dentro de um modelo de simulação, ou vice-versa. Uma situação que exemplifica esse tipo de aplicação pode ocorrer quando for conveniente manter regras, representativas de algum conhecimento específico, armazenadas em uma

base ao invés de codificá-las de forma procedural. Isso facilita qualquer alteração na base de conhecimento.

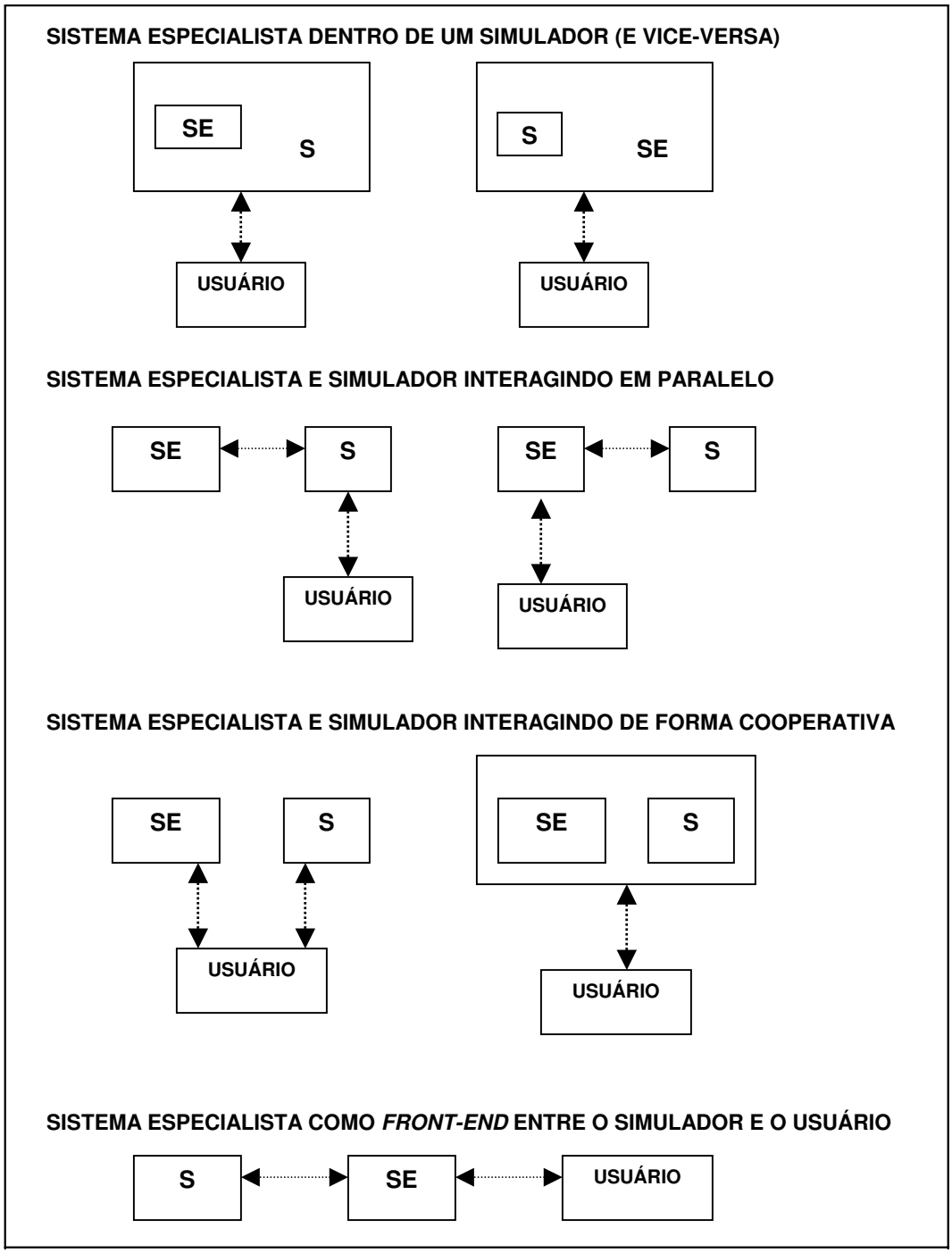


Figura 3.4 – Uma taxonomia para combinar sistemas especialistas com simuladores

Fonte - Adaptada de O'KEEFE (1986a)

A utilização de um modelo de simulação interno a um sistema especialista pode ser interessante quando: (i) o SE necessite efetuar uma simulação para obter resultados que justifique para o usuário as linhas de ação recomendadas; ou (ii) o SE utilizar variáveis temporais que precisam da simulação para serem atualizadas. Um simulador e um SE, desenvolvidos como softwares isolados, podem interagir interrogando-se um ao outro, em paralelo. Por exemplo, uma situação complexa pode exigir que um simulador interrogue um SE já existente, para auxiliá-lo nas tomadas de decisão. Por outro lado, um SE pode utilizar um simulador para ser testado e eventualmente até mesmo ser validado.

Uma outra forma de atuação conjunta é a interação cooperativa. Esse caso difere dos anteriores uma vez que o usuário de uma ferramenta pode ter acesso direto à segunda ferramenta. Nos casos citados anteriormente o usuário de uma ferramenta só acessava a segunda de forma indireta. Um claro exemplo de aplicação é quando o usuário utiliza um SE para desenvolver ou manipular um modelo de simulação. A interação cooperativa pode ser circundada por um *software*, constituindo um ambiente de simulação mais complexo. Existem várias ferramentas, de suporte à decisão, construídas segundo esse enfoque.

Pode-se utilizar um SE acoplado a um simulador. São os chamados *Front-End's*. Nesse caso um SE, localizado entre o simulador e o usuário, gera instruções que facilitam a utilização do simulador. Também, o SE pode trabalhar como um analisador dos resultados da simulação. De uma forma não muito usual o SE, nesse caso, pode ser chamado de *Back-End* (HILL & ROBERTS, 1987).

3.3.2 Desvantagens da Simulação Tradicional

A simulação comumente chamada de tradicional (aquela que não incorpora conceitos de IA) possui alguns pontos fracos que inibem a sua maior utilização, sobrecarregando os projetistas e programadores quando do desenvolvimento do processo de simulação, composto da modelagem e implantação do modelo. Algumas desvantagens dessa simulação podem ser vistas como particularidades de uma afirmação mais ampla dada por EVANS (1984), que diz: "Uma das maiores limitações da simulação tradicional é a sua inabilidade de modelar comportamentos inteligentes".

Analisando o processo de desenvolvimento e utilização de um modelo de simulação tradicional percebe-se que especialistas têm participação ativa em todas as etapas do processo. O ideal seria conservar o esforço humano, incorporando conhecimento e experiência no *software*. Com isso, muito trabalho poderia ser deslocado do homem para a máquina, deixando para o homem as etapas que realmente requerem um toque humano (SHANNON, 1988a).

Segundo REDDY (1987), nos sistemas convencionais de simulação primeiro cria-se o modelo, que inclui uma coleção de estruturas de dados interconectadas. Um conjunto de procedimentos opera nessas estruturas, representando a dinâmica do modelo. Quando o nível dessas estruturas e procedimentos é baixo, os modelos construídos são difíceis de serem entendidos e modificados.

Tradicionalmente, um modelador de simulação constrói um modelo usando uma linguagem de programação de proposta geral, ou uma linguagem específica de simulação. O modelo é então validado, executado e os resultados analisados e interpretados. Várias dessas partes do ciclo de vida da simulação podem ser automatizadas utilizando técnicas e ferramentas de IA e SE.

A fase de formulação de um modelo de simulação é um processo custoso e longo, porque normalmente os programas auxiliados por computador não se preocupam

com essa fase. As facilidades de comunicação entre usuários e projetistas são, via de regra, inadequadas, não existindo ferramentas que façam essa interação de forma mais amigável. Seria muito interessante que houvesse uma boa comunicação entre usuário e projetista, através de gráficos, planilhas, menus e diálogos de vários tipos, inclusive via linguagem natural.

Outra idéia é aproveitar esforços empreendidos no passado, conservando-se informações (de modelos já desenvolvidos, por exemplo) que podem ser aproveitadas como ponto de partida no desenvolvimento de novos modelos.

É sempre interessante reduzir o esforço computacional, tanto no momento da construção como também na manutenção dos sistemas. A simulação tradicional, por trabalhar com abordagens procedurais, amarra o sistema. As regras de decisão, uma vez codificadas, são fixas, tornando o sistema pouco flexível. Se as regras fossem construídas em separado da simulação, numa linguagem facilmente modificável, o resultado seria um sistema mais maleável. Isso facilitaria a manutenção do sistema, permitindo que novos conhecimentos fossem incorporados, sem afetar o módulo de controle existente (COSTA, 1991).

A fase de verificação e validação do modelo também é uma etapa demorada e difícil. Essa fase poderia ser automatizada, permitindo um número maior de testes e indicando ações que auxiliassem no re-projeto do modelo, se fosse o caso.

A execução de um modelo precisa ser constantemente interrompida, para que novos cenários sejam estabelecidos. É interessante que essa fase seja também automatizada, gerando cenários que permitam uma convergência que atenda às metas traçadas.

Uma limitação crítica da simulação tradicional é a falta de ferramentas que analisem os resultados das rodadas do modelo e façam as devidas inferências. Por ser uma grande geradora de números, a simulação tradicional apresenta normalmente, como saída, números sem qualquer comentário mais profundo.

Uma saída completa, bem analisada, utilizando-se de facilidades visuais na apresentação e justificação das linhas de ação a serem seguidas, permitiria um melhor aproveitamento, por parte do usuário, de todo o processo de simulação. Isso porque, na falta de um analista experiente, todo o potencial da simulação como ferramenta de auxílio à decisão cai por terra, criando um gargalo justamente na fase de utilização dos resultados.

O ideal é fazer com que a simulação não forneça apenas uma previsão, mostrando os resultados associados aos diversos cenários rodados, mas também faça uma prescrição, sugerindo o melhor caminho para atingir as metas estabelecidas.

As decisões tomadas durante a simulação normalmente são programadas de forma procedural, ou são tomadas externamente ao processo de simulação. Para isso é preciso interromper a simulação, consultar um especialista, comunicar a decisão ao simulador e dar continuidade ao processo. Portanto é muito importante, para que a simulação seja flexível e rápida, que os tomadores de decisão possam ser modelados e introduzidos no sistema de simulação. De preferência o especialista deve ser programado de forma declarativa, permitindo que um mesmo modelo de simulação possa ser utilizado em várias situações diferentes, sob a ação de diferentes especialistas.

Alguns dos problemas acima citados já foram ou estão sendo tratados pelos pesquisadores da área de simulação. A simulação de sistemas mudou muito nas décadas de 80 e 90. Essa nova simulação, com suas características e novidades, é comentada a seguir.

3.3.3 A Simulação Inteligente

A simulação inteligente, ou simulação baseada em conhecimento, tem como meta automatizar todas as atividades "inteligentes", pertencentes ao processo de simulação, que hoje são realizadas por especialistas. O objetivo desse tipo de sistema de simulação é acoplar ao *software* o máximo do conhecimento especializado necessário. Para alcançar esse objetivo os simuladores inteligentes devem possuir algumas características como:

- Organizar e conservar conhecimentos. Essa habilidade permite construir modelos a partir de esforços de estudos do passado. A conservação do conhecimento permite que o sistema, a cada estudo, torne-se mais inteligente. É preciso possuir uma base de conhecimento que possua uma biblioteca hierárquica de modelos, com diferentes níveis de abstração.
- Permitir acesso a conhecimentos só disponíveis num especialista. Esse conhecimento deve ser armazenado através de programação declarativa e não procedural. Essa estruturação facilita modificações, flexibilizando o sistema.
- Validar os processos de raciocínio do usuário, comparando-os com os resultados da simulação.
- Conseguir tratar altas complexidades, utilizando raciocínio qualitativo em vez de métodos matemáticos.
- Ter habilidade para explicar racionalmente porque uma meta não foi atingida, ou porque uma certa linha de ação está sendo recomendada.
- Deve possuir facilidades integradas para gerenciamento de dados. Deve possuir uma implementação moderna, iterativa, utilizando menus, gráficos e diálogos.

Segundo REDDY (1987), uma simulação baseada em conhecimento ideal deveria seguir a seguinte metodologia :

(i) Receber uma descrição do problema e sintetizar um modelo de simulação, consultando uma base de conhecimento apropriada.

(ii) Receber uma meta na forma de um conjunto de expectativas ou restrições, selecionar um modelo num nível apropriado de abstração, determinar as medidas de desempenho, gerar o espaço de pesquisa de cenários possíveis e, finalmente, recomendar um cenário que satisfaça a meta estabelecida.

(iii) Explicar o raciocínio através do qual certos cenários foram explorados e porque um cenário particular foi recomendado.

(iv) Aprender da experiência e divulgar seu comportamento, mostrando os eventos significativos e justificando o porquê da utilização de cada um.

(v) Mostrar o modelo resultante com um alto grau de clareza, aumentando dessa forma a confiança do usuário.

De acordo com SHANNON (1985) são várias as diferenças entre a simulação tradicional e a simulação inteligente. Talvez a maior delas esteja na forma como o modelo é construído e utilizado.

Na simulação tradicional o processo é iterativo, onde é projetado um modelo, estabelece-se um cenário (correspondentes às entradas), executa-se o modelo, e analisam-se os resultados. Decide-se por um novo cenário, executa-se novamente o modelo e analisam-se os novos resultados. E assim segue até atingir um ponto de parada. Os projetistas transportam a lógica e o comportamento do sistema real para um conjunto de comandos, e os executam de forma seqüencial (estruturada de forma *top-down*). Não ocorre uma busca objetiva da solução.

Na simulação inteligente os projetistas declaram o conhecimento a respeito do sistema real, definem os objetivos e restrições, e deixam o computador trabalhar

para encontrar a solução. Ao contrário de uma programação algoritmica, são utilizados passos não explicitados. Além disso, muitos passos do processo convencional de modelagem são executados externamente ao modelo. Na simulação inteligente há um interesse constante em minimizar o trabalho do usuário.

Uma segunda diferença diz respeito à estruturação dos módulos pertencentes ao sistema: base de dados, base de conhecimento e estrutura de controle. Nos sistemas baseados em conhecimento esses módulos são separados, podendo-se alterar qualquer um deles sem afetar os demais. O mesmo não ocorre com a simulação tradicional. Nela a programação dos módulos é única, dificultando a manutenção.

Uma diferença significativa é que o modelo de simulação tradicional não executa nada que não seja pré-especificado. Ao contrário, já existem modelos inteligentes que têm a capacidade de aprender da própria experiência. Diante do exposto são vários os pontos do processo de simulação onde a inteligência artificial, em especial os sistemas especialistas, pode atuar.

3.3.4 Algumas conclusões sobre o tema simulação inteligente

A simulação tem sido reconhecida como uma poderosa ferramenta para estudar o comportamento de sistemas complexos que não podem ou são muito difíceis de serem tratados matematicamente. Porém os usuários consideram fracos os *softwares* tradicionais de simulação. Eles necessitam de modelos mais realistas, que ofereçam maior suporte para os novatos e melhor assistência nas diversas etapas do processo de simulação.

O processo de modelagem da simulação providencia um mecanismo sistemático para o modelador entender e aumentar o desempenho de sistemas dinâmicos.

Esse processo compreende quatro grandes fases: conceituação, construção do modelo, análises da simulação, e comunicação. Na simulação tradicional algumas dessas fases são reconhecidamente pobres, por exemplo, a etapa de análise dos resultados da simulação e a tênue comunicação geralmente existente entre modelo e usuário.

IGNIZIO (1990) lembra que o tempo do analista de simulação é muito importante para ser gasto no desenvolvimento e depuração de programas. Portanto, devem ser utilizadas ferramentas que permitam que o analista concentre-se nas partes mais importantes do processo, aquelas que requerem raciocínio humano. Adotar tecnologias de IA é um passo no sentido de tornar os *softwares* de simulação mais poderosos e amigáveis.

A crescente onda de interesse em inteligência artificial e sistemas especialistas tem levado muitos pesquisadores de simulação a apontar similaridades entre IA e técnicas de simulação. Outros têm acenado com a possibilidade de atuação conjunta entre as duas tecnologias.

OREN & ZEIGLER (1979) foram uns dos primeiros pesquisadores a propor a utilização de técnicas de IA em sistemas de simulação assistidos por computador.

É um fato de extrema importância o reconhecimento de que a abordagem de análise de atividades da simulação se assemelha ao paradigma reconhecimento-ação dos sistemas expertos baseados em regras. Isso permite enxergar a simulação como uma técnica que pode evoluir muito, utilizando-se dos preceitos da IA.

A atuação conjunta entre inteligência artificial, em especial sistemas especialistas, e simulação é benéfica para ambas. Métodos de programação de IA permitem modelos de simulação mais realísticos e robustos, enquanto que algoritmos de simulação permitem que sistemas especialistas atuem sobre modelos complexos que se alteram no tempo, ou trabalham com elementos estocásticos.

São muitos os conceitos e técnicas de IA/SE que podem ser utilizadas pela simulação de sistemas, como:

- Linguagens de IA para desenvolver modelos de simulação;
- Técnicas de representação de conhecimento;
- Utilização das habilidades de aprendizagem;
- Facilidades de comunicação (linguagem Natural por exemplo);
- *Front-End's* inteligentes para facilitar a utilização de um simulador, para auxiliar na construção de um modelo ou para fazer análises das saídas da simulação;
- Sistemas conselheiros para atuarem como decisores internos;
- Conceitos como abstração, aprendizagem e introspecção.

É importante salientar que a possibilidade de casamento entre IA e simulação aumentou significativamente com a queda dos preços do *hardware* e a crescente sofisticação do *software*.

Pesquisadores como OREN (1987a,b) e MARTIN (1991) comentam que não existe nada que possa ser considerado realmente inteligente. OREN (1987a) até criou o termo "cognizante" para diferenciar do termo cognitivo, hoje fartamente utilizado. Para ele enquanto os sistemas cognitivos emulam habilidades humanas, os sistemas “cognizantes” deveriam utilizar essas habilidades para aprender com a própria execução, ampliando o seu conhecimento. Nesse caso se encaixa uma simulação chamada de introspectiva.

JIGNIZIO (1990) também é bastante radical ao afirmar que não há novidade alguma nos sistemas especialistas. Ele afirma que uma boa parte da comunidade de Pesquisa Operacional crê que sistema especialista é apenas um outro nome para a Programação Heurística. Reforça sua posição citando um trabalho de H. Simon e A. Newell, datado de 1957, como sendo o pioneiro no tema sistema especialista.

Mas a verdade é que, independentemente de opiniões no mínimo discutíveis, a inteligência artificial tem muito a oferecer à pesquisa operacional e, em particular, à simulação. Se a IA tem necessidade de simulação, pesquisadores que utilizam simulação discreta têm uma necessidade recíproca da IA (O'KEEFE,1986). Uma grande contribuição da IA na construção de sistemas de simulação é a qualidade dos ambientes computacionais, que buscam facilitar o desenvolvimento de simulações.

A combinação de conceitos de IA com metodologias da simulação tradicional produz uma poderosa ferramenta, de suporte ao projeto, conhecida como simulação baseada em conhecimento. Essa abordagem transforma a simulação, de uma ferramenta descritiva numa ferramenta prescritiva.

4 Metodologia SimuCAD

A busca da solução de um problema qualquer, independente da área de conhecimento a qual pertence, envolve a definição, estruturação e produção de um projeto. Entende-se como projeto um conjunto de atividades interligadas, que buscam um objetivo comum, dentro de um intervalo de tempo limitado (SHTUB et al., 1994). O guia PMBOK (2000), organizado pela *Project Management Institute* (PMI) enfatiza as características temporária e única de um projeto ao comentar que “um projeto é um esforço temporário levado a efeito para produzir um produto ou serviço único”. Um projeto, portanto, tem, bem definidos, um ponto de início e um ponto de término. A engenharia moderna trata com o projeto de sistemas complexos, de grande porte, onde a criação envolve preocupações com desempenho da equipe de trabalho, orçamento viável e métodos de planejamento, programação e controle de eventos.

Diante do contexto colocado, surgiu a necessidade de se utilizar uma boa metodologia de trabalho que orientasse o desenvolvimento de soluções, preferencialmente simples, para um problema de alta complexidade, no caso o modelo *SanCarlo*. O termo complexidade está associado diretamente ao conjunto de variáveis tratadas e às funções lógicas que as envolvem. A metodologia escolhida para auxiliar na modelagem do sistema de produção de bovinos de corte denomina-se SimuCAD: simulação e computação gráfica aplicadas ao estudo de sistemas produtivos. Essa metodologia, desenvolvida pelo grupo de pesquisa SimuCAD do Departamento de Engenharia de Produção da Universidade Federal de São Carlos, coordenado pelo autor desta tese, utiliza-se de referenciais conceituais oriundos de três campos distintos do conhecimento: pesquisa-ação ou pesquisa participativa, *engineering design* e simulação.

A pesquisa-ação, ou pesquisa participativa, é importante, na medida em que são estipulados objetivos de busca de conhecimento e de ação sobre a realidade, implicando em confrontações de diferentes racionalidades, as quais necessitam de um espaço de discussão e de construção de consensos no decurso do projeto. A Figura 4.1 mostra uma representação utilizada e discutida em EKLUND (1998), na qual os seminários criam esse espaço, representando a confrontação entre o

mundo da pesquisa, derivado do conhecimento teórico e de uma abstração do mundo real, e o mundo da empresa, derivado do conhecimento prático reflexivo (SCHON, 1983). Tal compreensão demanda mecanismos que permitam a integração do grupo na construção coletiva da definição do problema de projeto e do campo de soluções possíveis, partindo da realidade da situação e considerando as diferentes interpretações dos personagens envolvidos. Neste sentido, os seminários, constituem-se em instâncias de argumentação e deliberação em torno das hipóteses formuladas ao longo da condução do estudo e validadas sob critérios quantitativos e qualitativos.

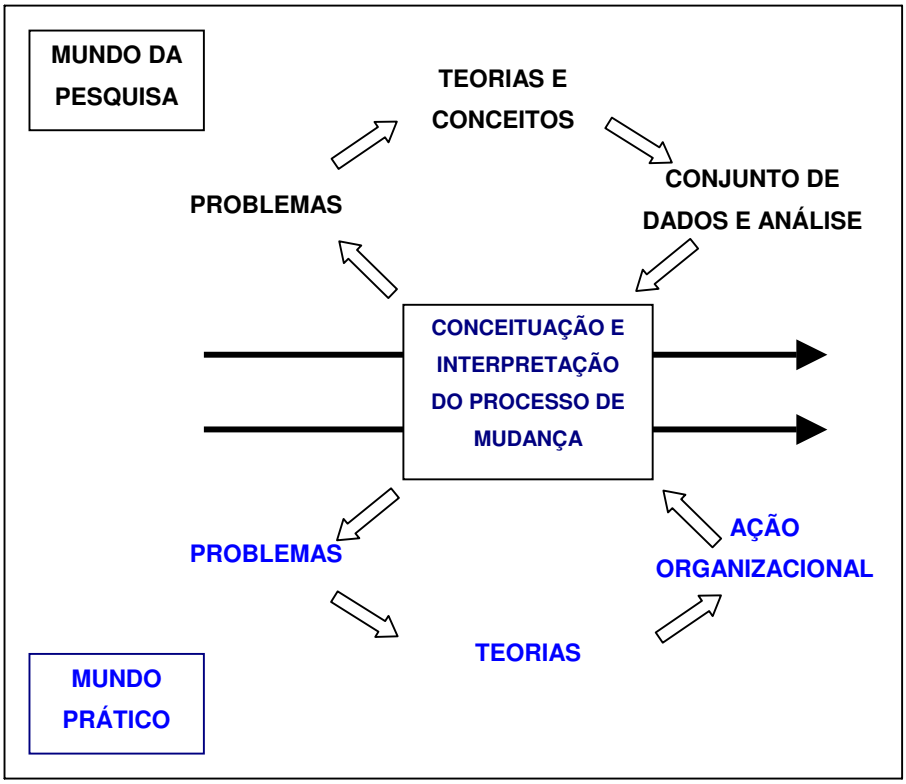


Figura 4.1 – Espaço de discussão de diferentes racionalidades

Fonte - Adaptada de EKLUND (1998)

No campo das metodologias derivadas do *engineering design*, considera-se a concepção e implantação de novos dispositivos técnicos e organizacionais de produção, envolvendo, além da descrição da realidade, a formulação de propostas

de ação. Nesta perspectiva, o modelo desenvolvido por PUGH (1991), *Business Activity Design Model*, considera o caráter desestruturado da atividade de projeto, no mesmo sentido apresentado por BUCHIARELLI (1994), destacando que todo processo de *design* envolve pessoas com diferentes "*object worlds*" construindo um consenso negociado, gerando resultados técnicos e organizacionais.

A simulação é o terceiro referencial conceitual da metodologia SimuCAD posto que as situações, a serem modeladas e analisadas, envolvem tratamentos finos dos processos de trabalho e previsão das condições futuras de funcionamento.

4.1 Métodos e técnicas utilizadas pela metodologia SimuCAD

Os módulos que compõem a metodologia SimuCAD são mostrados na Figura 4.2.

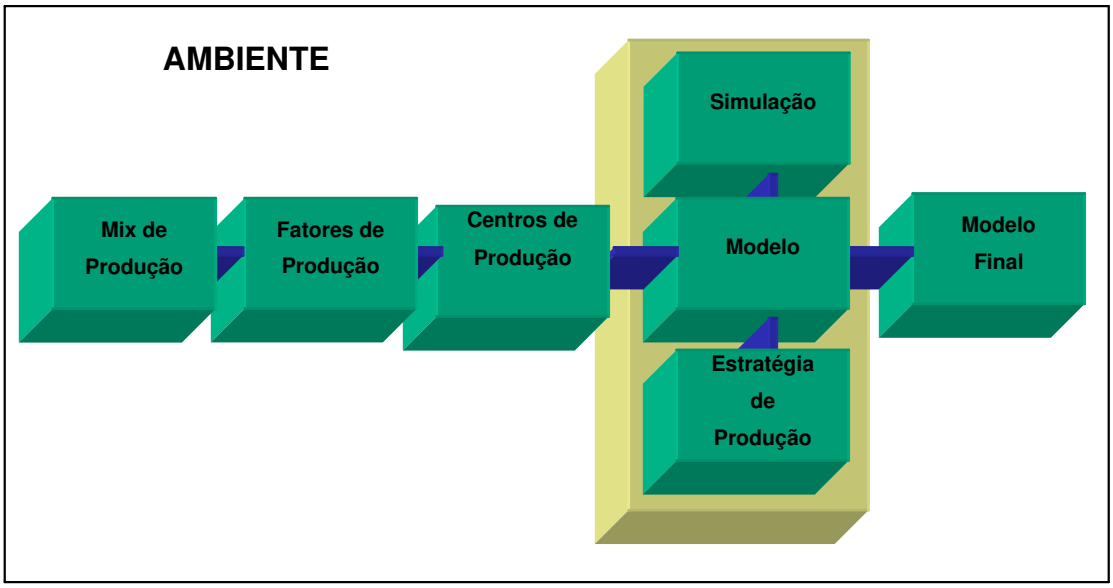


Figura 4.2 – Macro visão da metodologia SimuCAD

Fonte - Adaptada de COSTA, MENEGON & CAMAROTTO (1996)

Todo sistema em estudo está inserido em outro maior, chamado ambiente. Esse ambiente interfere no sistema em estudo através de variáveis denominadas exógenas, externas, independentes ou, ainda, não controláveis. Consideradas as

condicionantes do ambiente, o primeiro módulo do processo que orienta o projeto de sistemas produtivos busca identificar o ***mix* de produtos** e a **tecnologia de produção** adotada na unidade. Procura-se compreender quais são os produtos ou serviços a serem produzidos e os seus processos de obtenção. Trata-se de responder às questões relacionadas com o que e como produzir. Com este objetivo, deve-se neste momento, compreender precisamente todos os detalhes dos produtos, de suas partes e componentes, bem como os processos de produção associados. São documentos importantes para o levantamento destas informações: planos de processo, fluxogramas, layout das instalações e documentos gerais da produção e do negócio.

Após consolidar a primeira etapa, nos detalhes, deve-se buscar quantificar os recursos demandados no processo produtivo. Trata-se de quantificar os fatores diretos e indiretos de produção agrupados em: homens, materiais e equipamentos. Para a quantificação são fundamentais as informações relacionadas com a demanda do *mix* de produtos, estudo de tempos, rendimento dos processos, considerando as paradas técnicas e organizacionais (KOHRMANN & WIENDAHL, 1999) e dados referentes aos controles estatísticos. Dados históricos, quando disponíveis, devem ser tratados e convertidos em curvas estocásticas que alimentarão o modelo de simulação.

No módulo seguinte são representadas as demandas espaciais dos **centros de produção**, através de *templates* desenhados em duas ou três dimensões, de acordo com a necessidade. O conceito de centro de produção representa a menor unidade de recorte possível para a situação produtiva em estudo. Esta é uma etapa fundamental no processo de projeto de situações produtivas, pois dela depende a qualidade final do projeto quanto à ocupação dos espaços, as condições de trabalho e a gestão das interações entre centros de um mesmo setor. Além dos centros produtivos, devem ser consideradas todas as demais demandas espaciais, decorrentes das instalações de serviços, de utilidades e de gestão necessárias para que as atividades produtivas ocorram. Para o levantamento de dados nesta fase, deve-se recorrer às observações das atividades de trabalho, entrevistas e discussões em grupo com os elementos

atuantes no sistema. A Figura 4.3 ilustra o conceito genérico de centro de produção.

O *mix* de produtos, a tecnologia de produção, a quantificação dos fatores de produção e as demandas espaciais compõem os dados de entrada para o processo de modelagem e de simulação.

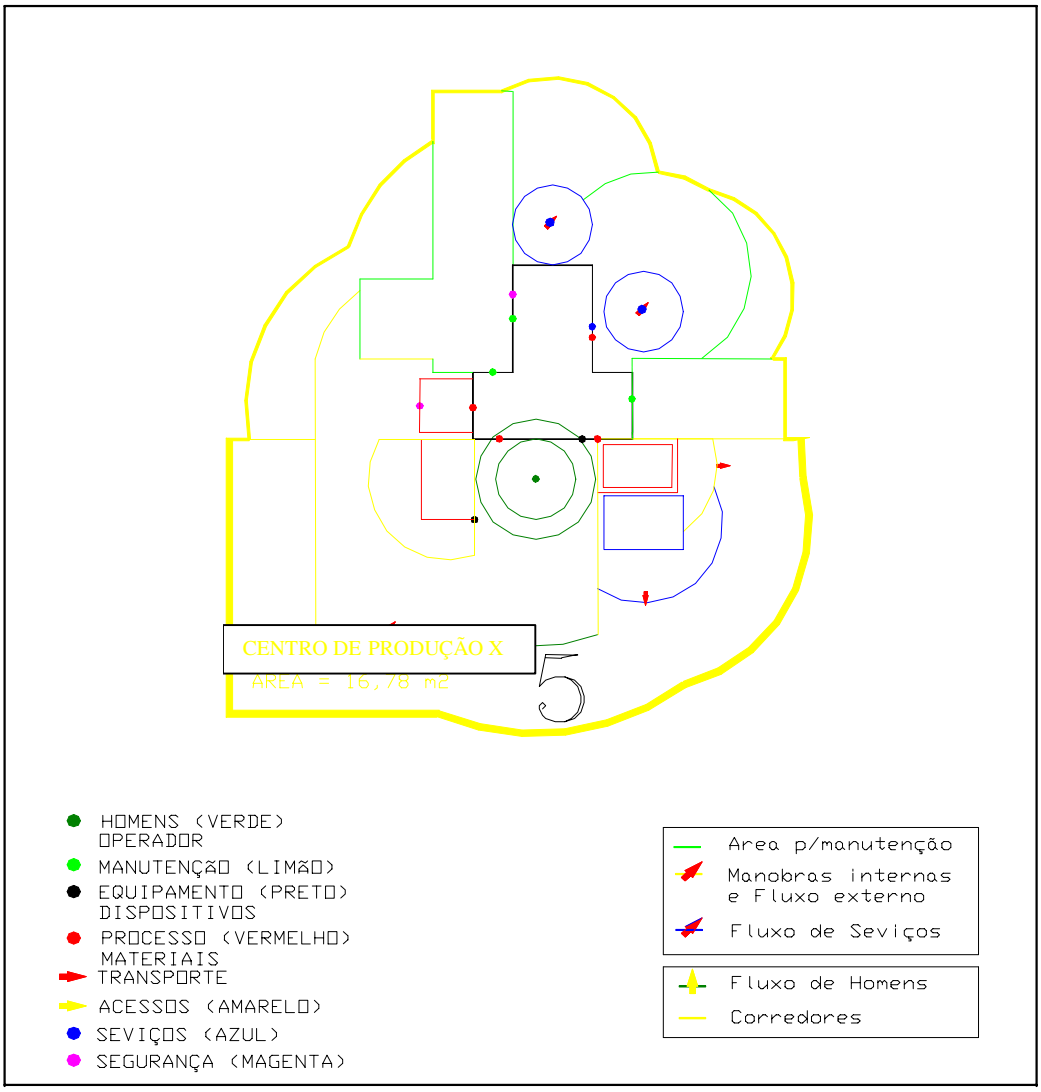


Figura 4.3 – Exemplo de um *template* de um centro de produção

Fonte - Adaptada de COSTA, MENEGON & CAMAROTTO(1996)

Na Figura 4.2, as fases de definição da **estratégia de produção**, **construção do modelo** e de **simulação,** são representadas de forma paralela pois constituem um processo cíclico de construção de modelos, verificação e validação desses modelos e geração de cenários estratégicos alternativos. Nesta fase considera-se detalhadamente as diferentes possibilidades de arranjo para os fatores de produção articulados em torno de uma dada estratégia. Neste processo agrupam-se áreas produtivas e não produtivas, são estabelecidos relacionamentos e são avaliadas soluções passíveis de implementação. As diversas alternativas geradas deverão ser comparadas frente a critérios objetivos e subjetivos. Este é um processo altamente interativo. A melhor solução irá emergir da análise do conjunto de possibilidades.

No campo instrumental, articula-se, nesta fase, os seguintes métodos de modelagem que exploram aspectos qualitativos e quantitativos (CAMAROTTO et al., 1999):

(i) **modelagem física**, envolvendo conceitos de decomposição e recomposição sistêmica de processos e operações (PHAL & BEITZ, 1995), articulados em torno do conceito de centro de produção (OLIVÉRIO, 1985);

(ii) **modelagem de fluxos** de materiais e de informações (MUTHER, 1978; SULE, 1988; e MARTIN & LEBEN, 1989);

(iii) **modelagem dinâmica**, fundamentada em pesquisa operacional e simulação (COSTA, 1996; COSTA, 1997); e

(iv) **modelagens subjetivas**, derivadas de entrevistas, questionários, seminários e observações participantes (THIOLLENT, 1997).

Os resultados devem responder as questões mais gerais impostas pelo ambiente, contra as quais as soluções são avaliadas. É importante reforçar que a simulação aparece como elemento chave do processo de avaliação, buscando evidenciar os efeitos de opções estratégicas sobre a produtividade da unidade e suas conseqüências sobre as atividades dos profissionais envolvidos. É isto, no final, que irá identificar a produtividade e a flexibilidade do sistema produtivo.

A saída deste processo é um conceito para o sistema estudado, que considera a estratégia de produção adotada e fornece a entrada para o detalhamento do modelo a ser implantado. A etapa de detalhamento do modelo envolve a especificação de todos os elementos que irão contribuir para o funcionamento da unidade. Trata-se da elaboração de um documento detalhado que irá orientar os diferentes profissionais que irão participar da implantação da nova situação.

4.2 A metodologia SimuCAD aplicada em sistemas agropecuários

Normalmente, teorias e metodologias de produção surgem para atender sistemas de manufatura, presentes no setor secundário da economia. De acordo com a necessidade, essas teorias e metodologias são adaptadas para utilização em outros tipos de sistemas, por exemplo, sistemas prestadores de serviços, sistemas agroindustriais e sistemas de produção agropecuários. É o caso da metodologia SimuCAD, criada inicialmente para orientar o desenvolvimento de projetos de instalações industriais, que foi adaptada para auxiliar no projeto de qualquer sistema de negócio.

Para ser utilizada na modelagem de um sistema de produção de bovinos de corte, essa metodologia recebeu novas interpretações para os conceitos presentes no sistema produtivo, através de analogias para definir claramente os elementos participantes desse tipo de sistema de produção e as relações lógicas e funcionais entre eles. A produção no setor primário possui particularidades que acrescentam dificuldades aos trabalhos desenvolvidos na área, como o ambiente mutável, a variabilidade da matéria prima, os longos experimentos de campo, os custos e riscos dos experimentos, a pequena quantidade de especialistas no setor, a falta de sistemas de informação que estruturem e gerenciem os dados existentes, entre outras.

Quanto às analogias, primeiro deve ser definido o ***mix*** de produtos a ser trabalhado. No caso de uma produção extensiva de bovinos de corte, os produtos de saída do sistema, os chamados produtos comerciais ou SKU (*Stock Keeping Unit*), são:

- bezerro desmamado;
- bezerra desmamada;
- novilha;
- garrote;
- fêmea descartada;
- boi magro;
- boi gordo;
- touro.

Os processos de produção dizem respeito ao ciclo de vida de cada animal no rebanho dependente das políticas de manejo e de compra e descarte do pecuarista. Por exemplo, estabelecida uma política de cria e recria, uma idade de desmama, uma taxa de suplementação alimentar, a tecnologia associada ao tipo de produção adotada e um *trade-off* que oriente os descartes, é possível descrever a seqüência de atividades do processo de produção de cada tipo de produto, os insumos necessários, as interdependências das atividades, suas durações, os recursos demandados, os custos envolvidos, históricos e previsões de demandas, prioridades e, dependendo do grau de complexidade que se deseja trabalhar, muitas outras variáveis podem ser estabelecidas.

Num segundo passo, devem ser definidos todos os **fatores** que dão apoio ao sistema de **produção**. Nesse momento entram os dados sobre pastos e pastagens, suporte administrativo, recursos de campo, movimentação e armazenagem, logística interna e externa, sistemas de informação gerencial, todos eles em um grau de detalhes que atenda os objetivos iniciais do trabalho. O resultado dessa fase é um mapeamento das condições existentes no ambiente de produção. São dados e informações que serão utilizadas no gerenciamento tático do negócio.

Os ***templates*** estão associados às necessidades espaciais demandadas pelas atividades do processo produtivo. Quando chega o momento do desenvolvimento dos *templates*, as atividades já estão definidas. Cabe agora detalhar cada uma delas, identificando suas entradas e saídas. Uma metodologia, secundária, de apoio pode ser utilizada nesse momento. Por exemplo, a metodologia de

diagramas de fluxo de dados (GANE & SARSON, 1988) que trata o sistema de uma forma "*top-down*", iniciando por um nível mais abstrato que mostra as macro-atividades, num baixo grau de detalhes, e segue num nível crescente de detalhamento até atingir um ponto onde todos os componentes do sistema não precisam mais ser divididos em componentes menores. Eles são chamados componentes elementares do sistema. Os *templates* são dependentes dos espaços definidos para abrigar os animais e das políticas de manejo, responsáveis pelo "armazenamento e movimentação" dos animais (analogia feita aos materiais circulantes em um sistema produtivo). Nesse momento fica bastante consistente o conceito de **centro de produção**, entendido como um espaço estabelecido através das especificidades do animal e da sua alimentação, onde o mesmo sofre uma transformação associada as suas características genéticas e às variáveis ambientais.

Definidos todos os *templates* e as interdependências entre eles, é construído o **modelo** representativo do sistema de produção de bovinos de corte. Este modelo é montado utilizando-se ferramentas de representação diagramática (por exemplo, diagramas ciclo atividade), ferramentas computacionais gráficas (por exemplo, AUTOCAD e 3 D Max) e *softwares* de modelagem de fluxos (FACTORY é um deles). A saída dessa fase é um modelo completo para representar o sistema em estudo.

De posse do modelo, são estabelecidas as **estratégias de produção**. Trata-se de um conjunto de alternativas de interesse que será testado considerando a dinâmica do modelo no tempo. O *software* utilizado nesse momento para auxiliar no desenvolvimento do modelo dinâmico é um software de **simulação de sistemas** (no caso, o *software* ARENA). Como existe o interesse de que o modelo de simulação represente o sistema real com toda a incerteza nele contida, é importante acoplar, no simulador, um módulo decisor que represente o tomador de decisão, no caso o pecuarista, nas suas políticas de compra e de descarte. Esse módulo, contendo todo o conhecimento do responsável pelo rebanho, é chamado sistema baseado em conhecimento e pode ser desenvolvido através de *shells* ou linguagens de sistemas especialistas. *Shell* é um programa esqueleto (*skeletal*

program) que possui no seu interior uma base de dados, uma base de conhecimento e uma máquina de inferência programadas e vazias. Dependendo do objetivo, do conhecimento tratado e da dinâmica do sistema real, são criadas representações de conhecimento (regras de produção são as mais comuns) para preencherem os espaços vazios do *shell*. Linguagens são mais flexíveis e mais difíceis de serem trabalhadas, possuem uma máquina de inferência (*engine*) mas, normalmente, não possuem bases estruturadas de dados e de conhecimentos, deixando para o analista programador a tarefa de dar entrada nas representações dos fatos e dos conhecimentos. Dentre as diversas linguagens existentes para desenvolver sistemas baseados em conhecimento uma bem interessante, criada pela NASA, é a linguagem CLIPS (CLIPS, 2003).

O modelo do sistema de produção de bovinos de corte, transportado para o simulador “inteligente” é, então, testado nas várias estratégias de produção definidas pelo usuário. Após análise dos resultados, são feitos os ajustes no modelo, gerando o modelo final, verificado através de testes e validado através de dados reais. O capítulo 5 deste trabalho apresenta os modelos desenvolvidos seguindo essa metodologia.

5 O Modelo *SanCarlo*: Simulação de Rebanhos de Bovinos de Corte

"*Transladar conhecimento humano para um modelo de Simulação é uma tentativa fascinante de condensar o mundo num sistema formal*"

(KERCKOFFS & VANSTEENKISTE, 1986)

É grande a necessidade de pesquisas e de ferramentas que auxiliem nos processos decisórios nos sistemas de produção de gado de corte. A pesquisa nesse campo no Brasil prende-se a estudos experimentais, que demandam muito tempo para chegar a conclusões que permitam alterar substancialmente os sistemas reais de produção pecuária. Além do tempo demandado, e talvez por isso mesmo, as novas soluções encontradas podem não atender ao problema originalmente proposto. Nesse caso novos experimentos devem ser realizados, com novos parâmetros de entrada, o que resultará em um novo tempo de espera para obter resultados, correndo o risco de novamente não atender às necessidades.

Os criadores brasileiros administram o maior rebanho do mundo, onde 68% dos animais são bovinos de corte, criados de forma extensiva. Esses produtores trabalham com pouca informação o que dificulta a tomada de decisão em um ambiente complexo onde, praticamente, inexistem ferramentas de apoio para auxiliá-los na análise das diversas combinações alternativas de manejo. Além dessas particularidades, identifica-se uma necessidade de se considerar a dinâmica associada às tomadas de decisão sobre compra e venda, que ocorrem durante todo o processo de produção de gado de corte. Essas decisões ampliam, e muito, o conjunto de alternativas de manejo a ser considerado para efeito de

estudo e análise. Esse fato contribui para dificultar ainda mais os estudos feitos através de experimentos no sistema real.

Neste capítulo são apresentadas as informações pertinentes ao modelo de simulação discreta de eventos futuros

que envolve as variáveis sobre os animais e as pastagens, as relações funcionais entre essas variáveis e as variáveis do ambiente, o programa computacional que emula os procedimentos de nascimento, crescimento, fertilidade e morte dos animais e o módulo decisor que reproduz as posturas do criador em relação à compra e descarte de animais.

As técnicas utilizadas para levantamento de informações associadas aos animais, ambiente e decisões de compra e venda tomadas pelos criadores foram: pesquisa bibliográfica tradicional, entrevistas, questionários e técnicas de análise de casos e tarefas.

As entrevistas seguiram os seguintes passos:

- Selecionar e localizar pessoas para serem entrevistadas, baseando-se na atuação e conhecimento de cada uma em relação ao tema tratado;
- Procurar conhecer os entrevistados anteriormente às entrevistas, recorrendo às fontes de informações disponíveis;
- Agendar as entrevistas;
- Elaborar questionários contendo perguntas e estudos de casos, que norteassem a entrevista;
- Explicar ao entrevistado porque ele foi escolhido e qual o objetivo da entrevista, na tentativa de envolvê-lo no processo de desenvolvimento do modelo;
- Conduzir a entrevista com base no questionário;
- Organizar as informações obtidas;

- Revisar as informações à procura de falhas ou dúvidas;
- Contatar novamente os entrevistados para obter informações complementares;
- Gerar as regras representativas das políticas decisórias dos entrevistados;
- Retornar aos entrevistados as regras geradas para análise e correção;
- Definir as regras que irão compor os processos decisórios do simulador.

5.1 Relações sobre animais e pastagens: entidades envolvidas no modelo, suas relações lógicas e seus diagramas representativos

As entidades tratadas no modelo de simulação *SanCarlo* correspondem às seguintes classes de animais:

- Bezerros (período entre um mês e idade de desmama);
- Bezerras (idem);
- Bezerros Desmamados (período entre idade de desmama e 12 meses);
- Bezerras Desmamadas (idem);
- Garrotes (machos com idade entre 12 meses e 24 meses);
- Novilhas (fêmeas com idade entre 12 meses e primeira cobertura);
- Boi Magro (idade entre 24 meses e 48 meses);
- Boi Gordo (idade acima de 48 meses);
- Vacas Prenhas (vacas do primeiro ao oitavo mês de prenhez);
- Vacas (fêmeas não prenhes);
- Touros.

Nas figuras numeradas de 5.1 a 5.6 são apresentados os diagramas de fluxos de dados que relacionam as entidades externas (ambiente), as funções internas do modelo, o módulo de simulação, o módulo decisor, os depósitos de dados dos animais e pastagens, o depósito de dados econômicos e as bases de fatos e regras de produção.

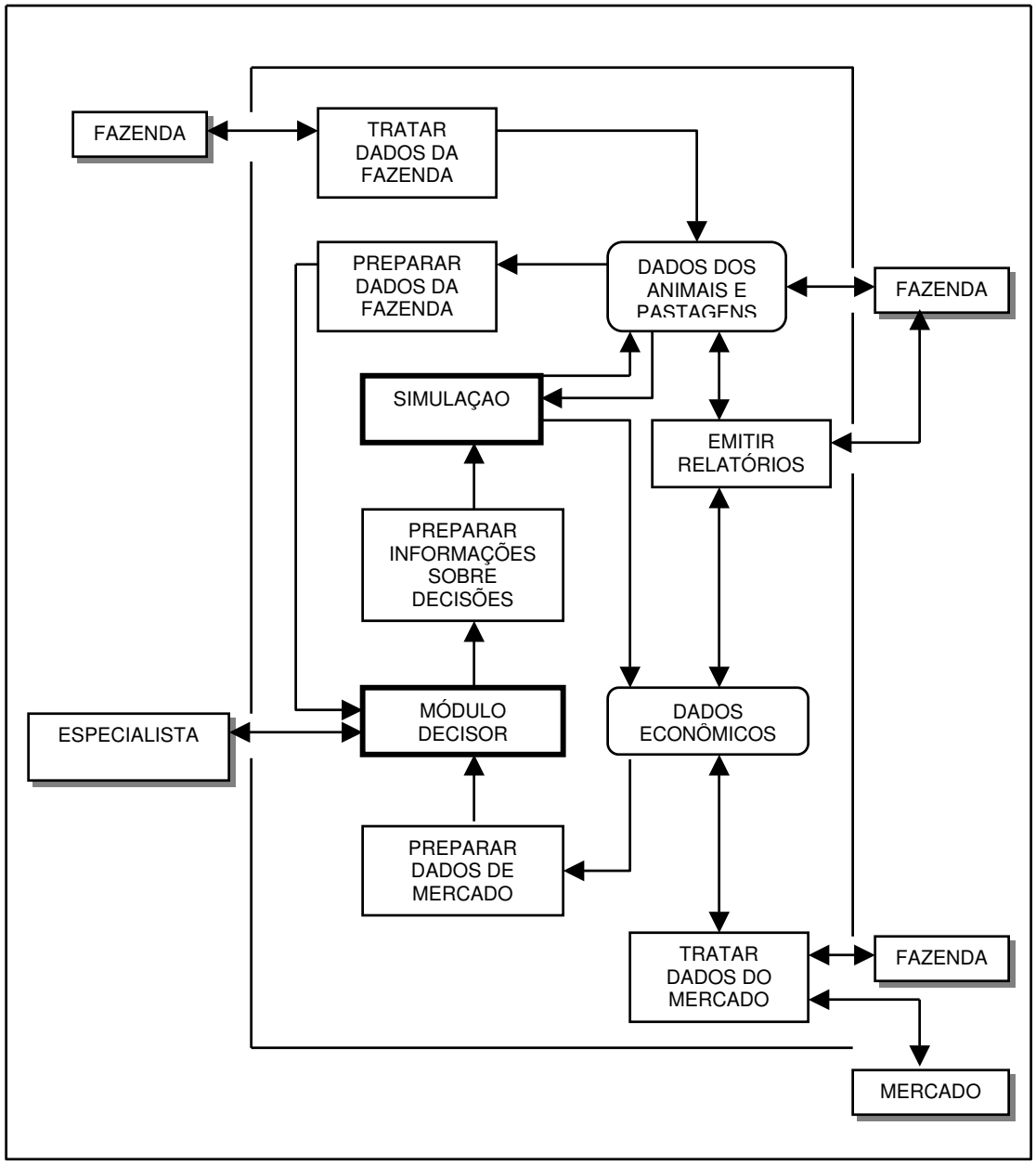


Figura 5.1 – Visão Global do Modelo *SanCarlo*

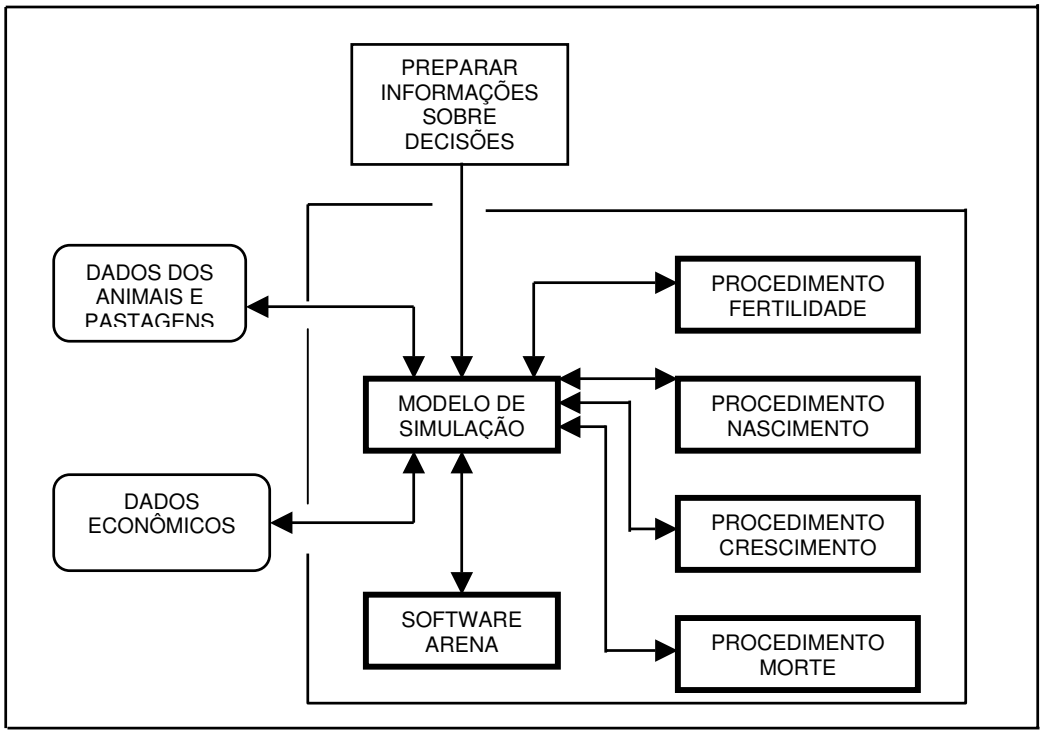


Figura 5.2 – Módulo Simulação

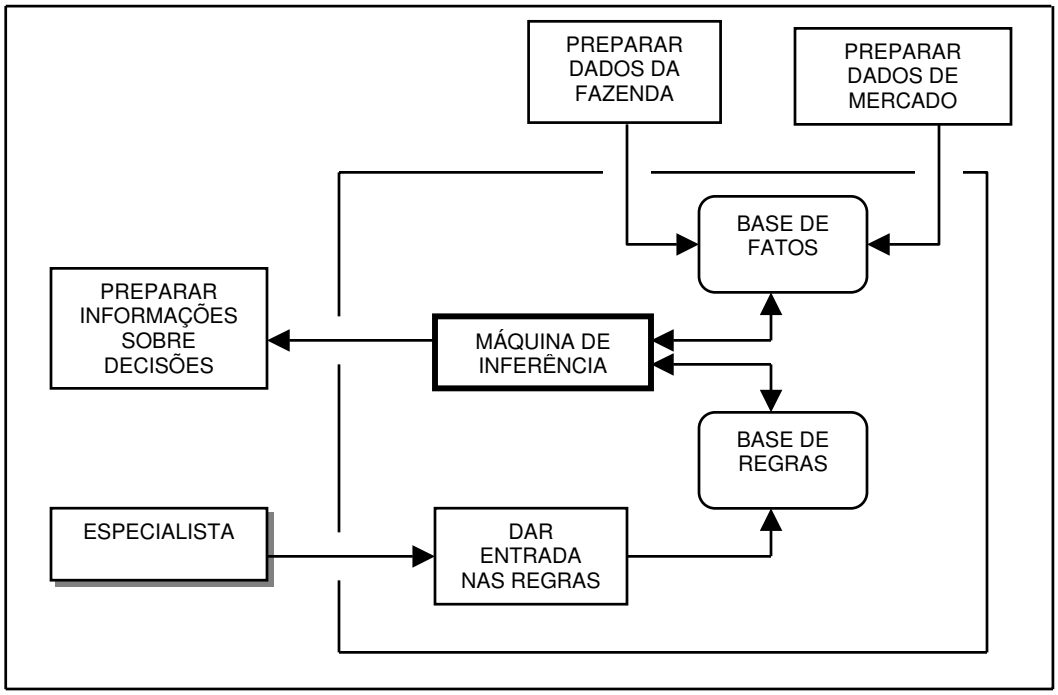


Figura 5.3 – Módulo Decisor

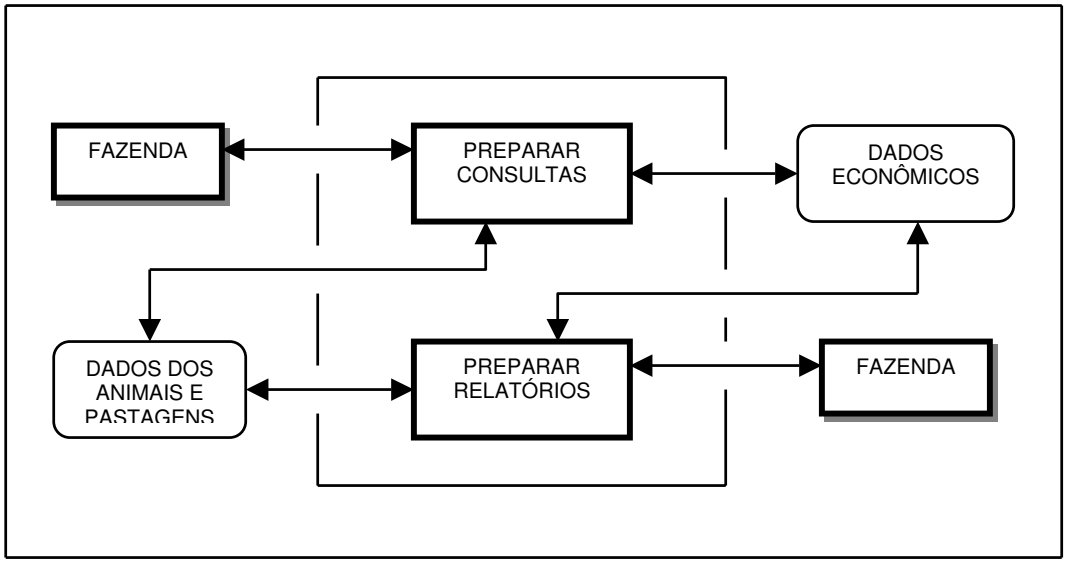


Figura 5.4 – Função Emitir Relatórios

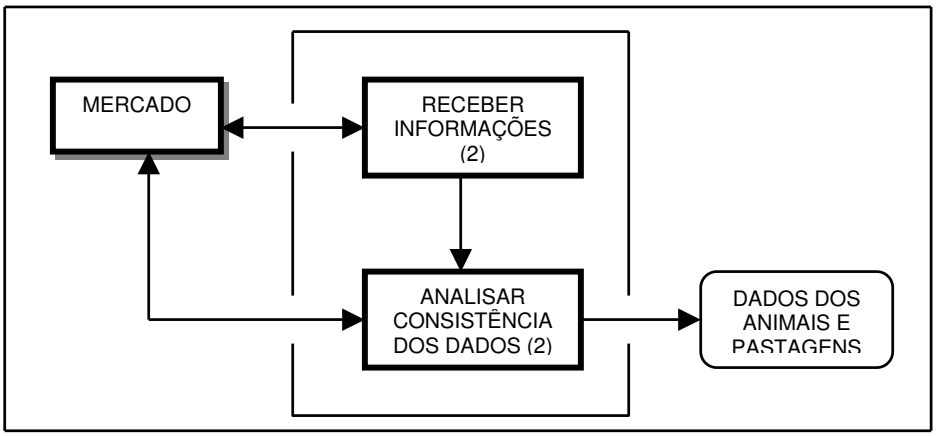


Figura 5.5 – Função Tratar Dados da Fazenda

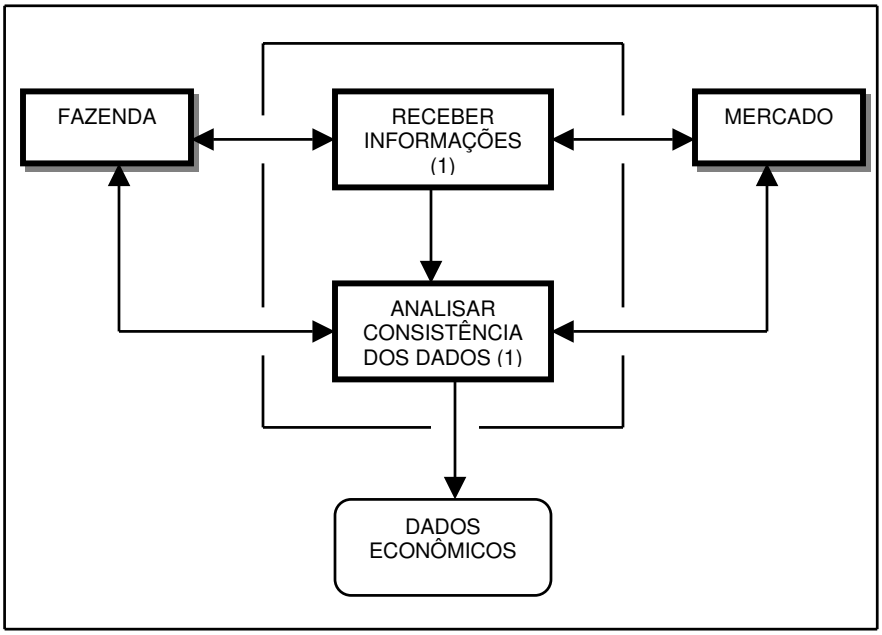


Figura 5.6 – Função Tratar Dados do Mercado

5.1.1 Programa de controle principal

Um programa recebe as informações sobre a quantidade e condição dos animais, nas diversas classes. Os animais são identificados pelo sexo, idade e mês de lactação ou gestação. O programa principal também lê os dados de entrada do modelo, para uma rodada de simulação específica, e gerencia a saída na tela ou impressa.

Não há restrições para o tamanho do rebanho, nem para as quantidades de animais em cada classe. As políticas de manejo, como idade de desmama, idade de descarte, idade para o primeiro cruzamento, períodos das estações de monta, suplementação alimentar, política de compra e venda de animais, também não são limitadas e são especificadas no módulo que gerencia as bases de fatos e de regras de produção.

As rotinas principais que compõem o modelo de simulação são: Crescimento, Fertilidade e Morte. Nesta seção são comentadas essas rotinas, são apresentados os fluxogramas representativos das mesmas, suas funções, variáveis e parâmetros. As rotinas utilizam funções genéricas que, segundo SMITH (1979), não necessitam de re-especificações para a aplicação do modelo a sistemas de produção diferentes. Apesar dessa afirmação, algumas funções receberam ajustes para se adequarem melhor à realidade brasileira. Essas alterações são comentadas quando da apresentação das funções.

O TAMU (Texas A&M University Beef Cattle Model) foi escolhido por ser, provavelmente, o mais completo modelo de simulação de bovinos de corte existente (DZAMA et al., 2001). Outra característica interessante do TAMU é que ele trabalha com os animais e, também, com as pastagens. Por ser muito conhecido e estudado, existe um farto material bibliográfico sobre esse modelo, destacam-se as seguintes referências: SANDERS & CARTWRIGHT (1979 a, b), SMITH (1979), NOTTER et al. (1979), e ROCHA (1992). As funções do modelo foram criadas a partir de outras funções primárias e seus parâmetros foram ajustados através de análises comparativas entre as saídas da simulação e os resultados obtidos na prática. Esse tipo de procedimento permitiu ajustes finos nas funções.

5.1.2 Procedimento Crescimento

Os meses considerados na simulação são todos de trinta dias e a idade dos animais, quando utilizadas nas funções, aparecem com as seguintes notações: **I** (idade em anos) e **IDADE** (idade em meses). A Figura 5.7 mostra a rotina Crescimento.

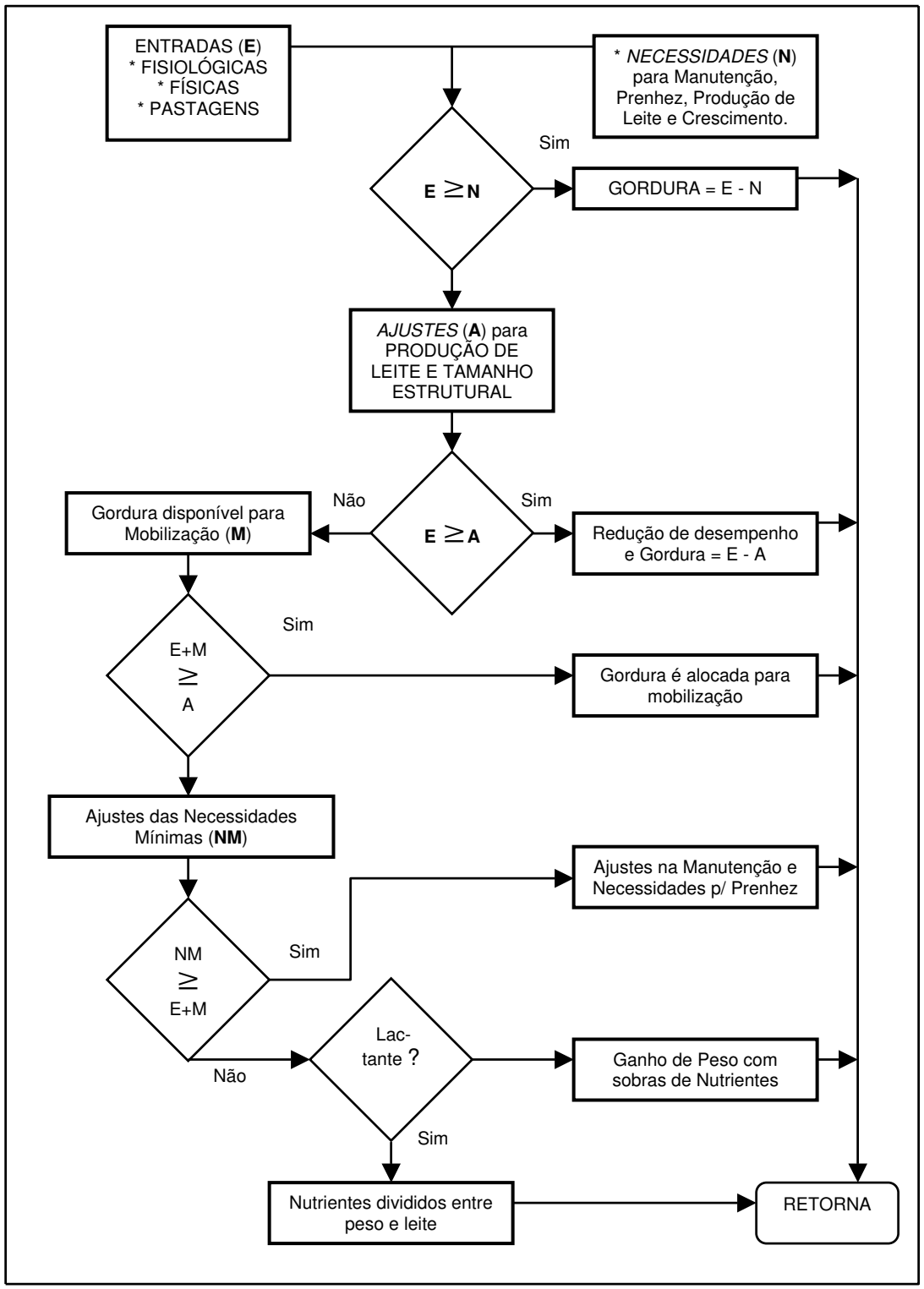


Figura 5.7 - Procedimento Crescimento

Como os animais são tratados individualmente, a cada mês simulado a rotina Crescimento é executada para cada animal, em cada classe. Os parâmetros de entrada são: Tamanho Estrutural (**WM**) e Tamanho Estrutural na Maturidade (**WMA**) na unidade quilograma, Peso Atual (**W**), Peso ao Nascimento (**BW**), Idade (**I** ou **IDADE**), Mês de Lactação (**J**) e Mês de Gestação (**K**). Os requerimentos de energia para manutenção, prenhez, produção de leite, crescimento do tecido muscular e deposição de gordura são calculados mensalmente para cada animal.

Um fator relevante nos cálculos dos valores da rotina crescimento é a condição do animal, que é calculada pela razão **W**/**WM**. No caso de dois animais com mesmo WM, será considerado em melhores condições aquele que possuir peso maior.

Os requerimentos de manutenção diária são calculados pela seguinte função:

$$\mathbf{M} = 0.0306 * \mathbf{W}^{0.75} * (\mathbf{WM}/\mathbf{W})^{0.5} * (\mathbf{WMA}/\mathbf{WM})^{0.15}$$

- $(\mathbf{WM}/\mathbf{W})^{0.5}$: é utilizado para corrigir o requerimento de manutenção diária do animal, de acordo com sua condição. Nesse caso os animais mais magros terão maior requerimento de manutenção do que os animais mais gordos;

- **W** : o peso do animal cresce ou diminui de acordo com o ganho ou perda de condição;

- **WM** : o tamanho estrutural não diminui. Uma vez atingido um valor este não regride devido à nutrição ou qualquer outra restrição;

- $(\mathbf{WMA}/\mathbf{WM})^{0.15}$: este termo corrige o requerimento de manutenção considerando o grau de maturidade do animal. Os animais mais jovens terão maiores necessidades de alimento do que os animais mais velhos.

Para as vacas prenhes e/ou lactantes são calculados requerimentos específicos que serão acrescidos aos requerimentos para manutenção. O requerimento para prenhez (**RP**) é dado pela seguinte função:

$$\mathbf{RP} = 0.0000275\ \mathbf{WMA}^{\,0.522(\mathbf{k}-1)}\ ,\ \text{onde}\ \mathbf{k}\ \text{é o mês de prenhez.}$$

O **RP** é, portanto, calculado considerando o genótipo do animal e o mês de gestação no qual se encontra.

Para um animal em período de lactação, o requerimento de manutenção é reajustado em 30%, portanto, **M**= 1.3**M**. Existe também um requerimento extra, o requerimento de lactação (**RL**), que é calculado considerando a idade, condição e genótipo do animal, além do seu estágio de lactação. O RL corresponde a 30% da Produção de Leite do animal (**PM**). O PM, de acordo com o seu genótipo, é dado por:

$$\mathbf{PM} = \mathbf{CFA} * \mathbf{CFX} * \mathbf{PMA} * e^{\,-0.08\mathbf{j}}$$

- **j** : é o mês de lactação;
- **PMA** : é o Potencial de Produção de Leite do animal. Em termos de idade considera-se que o animal com maior potencial para a produção de leite é aquele com 7 ou 8 anos. Os demais receberão ajustes de acordo com suas idades através dos coeficientes CFA e CFX;
- **CFA** = $1 + 0.01*(\mathbf{I}-7) - 0.01*(\mathbf{I}-7)^2$;
- **CFX** : Considera-se que um animal com peso inferior a 2/3 do seu tamanho estrutural não tem condição de produzir leite, portanto nessa situação CFX = 0. Por outro lado, um animal com peso igual ao seu tamanho estrutural será considerado com condição plena de produção de leite e o CFX = 1. Nos casos intermediários, tem-se: $\mathbf{CFX} = 1 + ((\mathbf{WM}-\mathbf{W})/0.33\mathbf{WM})^{3.32}$

A curva de crescimento do tamanho estrutural possui um ponto de inflexão (**WMP**) que é calculado da seguinte forma:

WMP = **WMA** (**BW**+360(0.0026)480)/(**WMA**+360(0.0026)480).

Os parâmetros aqui considerados correspondem ao tamanho do animal médio da raça Hereford, que foi utilizada para derivar essas equações. No caso de animais zebuínos basta entrar com os seus respectivos valores do tamanho estrutural na maturidade e da taxa de maturação. Por exemplo, BARBOSA, ALENCAR & SILVA (2002) apresentam, segundo seus experimentos, um tamanho estrutural na maturidade para fêmeas da raça Canchim igual a 517 kg e taxa de maturação igual a 0.056 kg/kg de peso vivo/mês, que corresponde a uma taxa de maturação diária de 0.00187.

Animais com WM abaixo do ponto de inflexão WMP terão taxas de crescimento do tamanho estrutural (**GR**) diferentes daquelas dos animais com WM maior do que WMP. Considera-se que no primeiro ano de vida o crescimento do WM é linear e ocorre segundo uma taxa diária **GR** = (**WMP**-**BW**)/360. Para animais com mais de um ano o valor de **GR** = 0.0026 * (480/**WMA**) * (**WMA**-**WM**).

Resulta das taxas de crescimento do WM um requerimento extra a ser considerado no Requerimento Total de Alimentos do animal (**REQ**), o requerimento de crescimento **RG**, dado pela seguinte relação:

RG = 0.8 **GR** + 5.56 (0.22 **GR**) * (2 **WM** + **GR** + **BW**) / (**WMA**-**BW**).

No modelo assume-se que qualquer animal cujo peso é igual ao seu tamanho estrutural, independente do seu grau de maturidade, tem 18% de proteína e 3% de gordura no tecido muscular (fato considerado no primeiro termo da função). No caso de haver diferença entre o peso e o tamanho estrutural, essa diferença é assumida como sendo gordura adicional. Essa situação é contemplada na função através do fator 5.56, que corresponde à conversão da energia metabolizável em

gordura. Portanto, o requerimento de energia líquida para ganhos está incluindo energia para a gordura e para a proteína que são depositadas.

Para finalizar, existe ainda um último requerimento alimentar associado ao ganho de condição, o valor **Q** = 5.56 (0.01) (**WM** - **W**). O valor Q é igual a zero quando o peso é maior ou igual ao tamanho estrutural.

Chega-se ao cálculo do requerimento alimentar total do animal, somando-se os valores calculados para cada um dos requerimentos particulares:

$$\mathbf{REQ} = \mathbf{M} + \mathbf{RP} + \mathbf{RL} + \mathbf{RC} + \mathbf{GC}.$$

Nesse ponto da rotina Crescimento é ativada a rotina Pastagem, que permite ao usuário fornecer seus dados particulares sobre pastagens. Nesse caso são estabelecidos três conjuntos contendo valores mensais para Digestibilidade da Matéria Seca (**DIG**), Conteúdo (Concentração) de Proteína Bruta (**CP**) e Disponibilidade (Massa) de Forragem (**AV**) disponível por animal maduro, por dia.

O conjunto AV, segundo ROCHA (1992), pode refletir diferentes circunstâncias. Em alguns casos pode ser considerado como a medida da quantidade total de forragem disponível para pastejo. Em outros pode estar relacionado com a quantidade de tempo que pode ser gasto pelo animal pastando por dia, em conjunto com a quantidade de forragem que pode ser consumida por unidade de tempo. Esse fator mostra que, para um dado conjunto de condições, existe um limite da quantidade de forragem que um animal pode consumir em um dia.

De uma forma geral o tamanho do animal, seu grau de maturidade, sua condição (normalmente medida através de gordura), seu estado fisiológico e seu potencial genético interagem com a qualidade e quantidade de forragem, simulando as condições de mudanças mensais nas taxas de crescimento do animal.

As restrições de consumo para o animal são estabelecidas pelos três limitantes: físico, fisiológico e quantidade de forragem. O menor deles será o utilizado como

referência para o consumo a ser simulado no mês. Isso é feito através do procedimento Consumo.

O limite fisiológico é estabelecido por um máximo de consumo de matéria seca de 2,25% do tamanho estrutural do animal, dividido pelo potencial de digestibilidade dessa matéria seca. No caso onde existe amamentação, o limite fisiológico considera também um parâmetro de consumo de leite (**TM**), que corresponde a 32% da produção do mesmo. Esse valor é subtraído do máximo consumo possível de matéria seca.

O limite físico corresponde à quantidade de matéria seca indigestível que o animal consome por dia. Esse limite é de 1.07% do tamanho estrutural.

O limite de massa de forragem é calculado para todos os animais, com exceção dos animais não desmamados, como sendo o máximo de forragem, medida em kg MS ha^{-1} dia^{-1}, multiplicada pelo termo $(WM/WMA)^{0.15}$. O tamanho estrutural na maturidade (WMA) representa o potencial genético do animal e corresponde ao peso de animais maduros com 25% de gordura.

Como os limites fisiológico e físico são linearmente relacionados ao tamanho estrutural, o consumo simulado é proporcional a esse tamanho, exceto no caso do limite de disponibilidade de pastagens ser menor que ambos. O menor desses três valores limitantes, em conjunto com a taxa de digestibilidade, orientará qual será o consumo total de forragem (**CN**), por dia. O consumo em excesso é utilizado como deposição de gordura adicional, aumentando o peso, mas não o tamanho estrutural. Para animais acima de oito anos de idade, o consumo é ajustado para baixo a uma taxa de 3% para cada ano que ultrapasse o limite oito.

O modelo também considera que os bezerros não desmamados consomem todo o leite de suas mães.

Com a saída da rotina Consumo é possível comparar os requerimentos alimentares necessários para o animal (REQ), com o consumo total de forragem

simulado (CN). Caso REQ < CN, o consumo ocorrido é mais do que o suficiente para atender às necessidades especificadas de alimentação. Nesse caso calcula-se as alterações que devem ocorrer no animal. A primeira alteração é a mudança no peso, dada pela equação:

$$\mathbf{DW} = \mathbf{GR} + \mathbf{FG} + 0.953*\mathbf{RP}$$

Onde o ganho de gordura $\mathbf{FG} = (\mathbf{CN} - \mathbf{REQ} + \mathbf{Q})/5.56$.

Uma segunda alteração diz respeito ao ganho no tamanho estrutural **GWM** = **GR**. Quando CN < REQ ocorrerá redução da produtividade. Com isso é calculado um novo REQ, considerando novos valores para RL (lactação) e RG (crescimento). Nessa nova situação não há cálculo de ganho de condição (Q) como ocorreu na situação anterior, pois como os requerimentos não foram atingidos, o "ganho" seria negativo e, nesse caso, considera-se Q = 0. A nova produção de leite será calculada como $PM = (0.1 + 0.9*(CN/REQ)^{0.5})*PM$ e o RL será igual a 30% desse novo PM.

Um novo ganho diário no tamanho estrutural é calculado como $\mathbf{GWM} = (0.2 + 0.8(\mathbf{CN}/\mathbf{REQ})^{0.6})* \mathbf{GR}$, e um novo requerimento para crescimento $\mathbf{RG} = 0.8\ \mathbf{GWM}$ é estabelecido.

Com esses novos valores obtém-se um requerimento total de alimentação $\mathbf{REQ} = \mathbf{M} + \mathbf{RP} + \mathbf{RL} + \mathbf{RG}$ menor que o anterior. Novamente é feita a comparação entre REQ e CN e, caso REQ < CN, estabelece-se novos valores para alteração no tamanho estrutural (**DS**), na gordura e no peso, através das seguintes funções:

$$\mathbf{DS} = (1 - 0.22\ ((2*\mathbf{WM} + \mathbf{GWM} + \mathbf{BW})/(\mathbf{WMA}-\mathbf{BW})))\mathbf{GWM}$$

$$\mathbf{FG} = (\mathbf{CN}-\mathbf{REQ})/5.56$$

$$\mathbf{DW} = \mathbf{DS} + \mathbf{FG} + 0.953\ \mathbf{RP}$$

O ganho no tamanho estrutural, na nova situação, corresponde ao crescimento no tecido muscular e é denotado por DS (é a diferença entre o ganho diário de WM e o ganho de gordura associada a esse WM, não considerando os 3% essenciais).

Esses serão os novos valores de saída do procedimento Crescimento, que mostra que o animal, apesar de não atingir plenamente o seu potencial genético para **GWM** e **PM**, receberá alguma deposição de gordura.

Um fato interessante a comentar é que quando não se atinge o CN suficiente para atender o REQ estabelecido inicialmente, ocorre do animal ter um aumento de peso menor que o aumento do tamanho estrutural, o que significa perda de condição, ou seja, o animal se torna maior, porém mais magro.

Caso CN ainda não seja suficiente para atingir o novo REQ, novas situações podem ocorrer. Uma delas é utilizar a gordura disponível para completar CN, superando REQ. Nesse caso a mesma formulação para DW é utilizada, porém com o segundo fator ((CN-REQ)/5.56) tornando-se negativo. Isso significa que não haverá deposição de gordura, e essa será utilizada para atingir REQ, diminuindo o ganho diário de peso. A gordura utilizada é calculada pela soma **FAT** = **W** - **WM** + 0.22*__WM__*(__WM__-__BW__)/(__WMA__-__BW__). O total de gordura disponível para transformação (**ANF**), por dia, será igual a 5.56*FAT/30.

Pode ocorrer ainda uma situação onde a gordura disponível não seja suficiente para, somada a CN, atingir REQ. Nesse caso, novamente o ganho no tamanho estrutural e a produção de leite serão ajustados para baixo. Independente do plano de nutrição, considera-se os requerimentos de prenhez como sendo sempre atingidos. Portanto os requerimentos mínimos (**REQM**) necessários são os de manutenção e prenhez. Numa situação onde nem mesmo esses requerimentos mínimos sejam atingidos, considera-se que não haverá ganho no tamanho estrutural (o animal não vai crescer) nem haverá produção de leite. A alteração do peso será **DW** = -**FAT**/30 + 0.953 **RP**. Se a transformação de gordura não for suficiente para contrabalançar o requerimento de prenhez, o animal perderá peso. É a situação mais crítica.

Uma última possibilidade ocorre quando a soma de CN + ANF é suficiente para atender os requerimentos mínimos de manutenção e prenhez. Caso o animal não esteja em período de lactação, as mesmas fórmulas anteriores para DS e DW são válidas, apenas que o ganho no tecido muscular será calculado considerando o consumo disponível somado à gordura convertida, menos os requerimentos mínimos necessários. Portanto **GWM**=(**ANF**+**CN**-**REQM**)/0.8.

Para vacas lactantes considera-se necessário atingir um mínimo de 25% do requerimento para produção de leite. Caso isso não aconteça, não haverá crescimento no tamanho estrutural, a produção de leite ficará restrita a (**ANF** + **CN** - **REQM**)/0.3, e a variação no peso se manterá igual à **DS** + 0.95 **R**P - **FAT**/30.

No caso de ser possível atender o mínimo exigido de 25% do requerimento para produção de leite, os nutrientes disponíveis restantes serão divididos entre a lactação e o crescimento de WM. Primeiramente considera-se o complemento de 75% de RL, o requerimento para crescimento do tecido muscular, e monta-se uma nova variável **PREQ** = 0.75**RL** + (0.8***GWM**). Chamando de soma total dos nutrientes adicionais (**ATDN**) a soma de CTF+ANF menos REQM, esse ATDN será dividido em duas partes. Uma será utilizada para produção de leite e outra para simular o crescimento do tecido muscular. As saídas serão geradas segundo as fórmulas anteriores de DW e DS, que por sua vez consideram os novos valores de GWM.

“Em linhas gerais, a rotina Crescimento simula um mês do desempenho do animal considerando a quantidade e qualidade da alimentação disponível para o mês, o potencial genético e características iniciais do animal, que inclui peso, grau de maturidade, estado de prenhez e lactação, e idade. Consumo de alimento e requerimentos energéticos para diferentes atividades fisiológicas são passos intermediários na simulação. No fim, a produção de leite para o mês e mudanças no tamanho estrutural e peso são estimadas, e as características do animal são

atualizadas para o começo do próximo mês de simulação. Dessa forma, na simulação do desempenho do animal, as características do mesmo interagem não somente com as condições de alimentação atual, mas também com as condições de alimentação dos períodos prévios de simulação" (ROCHA, 1992).

5.1.3 Procedimento Fertilidade

A rotina Fertilidade simula, mensalmente, para todas as fêmeas (novilhas e vacas) a ocorrência do cio e a probabilidade de concepção do animal. Essa simulação é função da condição, do estado de lactação, período do último parto e potencial genético de reprodução do animal.

Essa rotina é chamada após o término do procedimento Crescimento e utiliza como passagem de parâmetros por valor o tamanho estrutural, o peso, idade e mês de lactação. Utiliza como passagem por variável o mês de gestação e uma variável *booleana* que registra se o animal está ciclando ou não.

O tratamento utilizado nessa rotina é estocástico, diferentemente dos trabalhos da Universidade de Agricultura e Mecânica do Texas (TAMU). A rotina Fertilidade utiliza três medidas básicas :

- **PEST**: a partir da informação de que o animal não estava ciclando no mês anterior, estabelece se o animal vai ciclar ou não no mês atual;
- **CCYC**: na situação do animal ter ciclado no mês anterior, estabelece se o mesmo vai continuar ciclando ou não;
- **PCON**: estabelece se o animal vai conceber ou não.

Essas medidas são obtidas através do cálculo de variáveis auxiliares, comentadas a seguir. Definida cada uma dessas medidas, fica estabelecido o tamanho do intervalo de validade. Gera-se um número aleatório e verifica-se se o mesmo

encontra-se dentro desse intervalo. De acordo com o resultado a medida tratada será ativada ou não.

As quatro variáveis auxiliares utilizadas são:

- **CFW** (fator de correção para a condição do animal);
- **CFDW** (fator de correção para a mudança de peso);
- **CFM** (fator de correção para o grau de maturidade); e
- **CFT** (fator de correção de acordo com o mês de lactação).

O CFW, que corrige a simulação do cio e da concepção de acordo com a condição do animal, considera o valor do tamanho estrutural como sendo o indicador do peso mínimo necessário para que o animal tenha máxima fertilidade. Nesse caso se $W \geq WM$ o CFW=1. Considera-se que o animal terá mínima fertilidade no caso de $W \leq 0.75WM$ e, nesse caso, CFW = 0. Nos casos intermediários **CFW** = ((**W** - **WM**) - 0.75)/0.25.

O CFDW é a medida do plano de nutrição atual e é calculado a partir dos valores de mudanças do peso (DW) e mudança do tamanho estrutural corrigida (DWM). O DWM é o mínimo ganho de peso que possibilita ao animal máxima fertilidade. É calculado como DWM = GWM + 0.01(WM-W). Segundo SANDERS & CARTWRIGHT (1979 a,b), o CFDW será igual a um caso $DW \geq DWM$ e nos demais casos **CFDW** = 1-(100(**DWM**-**DW**)/**WM**).

O CFM é o fator de correção para grau de maturidade das novilhas. Considera-se que o grau de maturidade bom para uma novilha atingir o máximo de fertilidade é quando o seu tamanho estrutural atinge, no mínimo, 60% do WMA. Portanto **CFM** = ((**WM**/0.6*__WMA__)-O.67)/0.33. Caso $CFM \geq 1$ assume-se CFM = 1. No caso de $CFM \leq 0$ assume-se CFM = 0.

O CFT corrige a taxa de fertilidade de acordo com o tempo transcorrido da última parição. Essa variável busca representar o que acontece no intervalo pós-parto. Caso o animal não seja lactante ou seu mês de lactação seja superior a três,

considera-se o mesmo com total potencial, ou seja, CFT = 1. Caso o animal esteja no seu primeiro mês de lactação (j=1) o mesmo não tem condição de ficar prenhe, portanto CFT = 0. No caso de j=2 ou j=3 o animal terá, respectivamente, 40% e 99% de potencial de fertilidade e, conseqüentemente, CFT = 0.40 ou CFT = 0.99 (BARBOSA, 1991). Para $j \geq 4$ CFT = 1.

Calculados os valores das variáveis auxiliares, o procedimento Fertilidade checa o valor da variável lógica Ciclo (que indica se o animal estava ciclando no mês anterior) e caso Ciclo = 1 então calcula-se a probabilidade do animal continuar a ciclar, dada por $\mathbf{CCYC} = \mathbf{CFW}^{0.1} * \mathbf{CFDW}^{0.1}$. A seguir gera-se um número aleatório entre 0 e 1, e compara-se com CCYC. Caso CCYC seja menor ou igual ao número aleatório então o animal continuará a ciclar, caso contrário não.

No caso de Ciclo = 0, calcula-se PEST que irá dizer se o animal, que não estava ciclando no mês anterior, irá ciclar no mês atual. Parte-se do princípio que um animal que não esteja ciclando tem no máximo 95% de probabilidade de vir a ciclar. Esse valor não é fixo e pode variar de acordo com o sistema de produção. Considerando a influência de cada uma das variáveis tratadas anteriormente tem-se: $\mathbf{PEST} = 0.95 * \mathbf{CFW} * \mathbf{CFDW} * \mathbf{CFM} * \mathbf{CFT}$.

Da mesma forma que para CCYC, gera-se um número aleatório e caso PEST seja menor ou igual ao número aleatório, a variável Ciclo será igual a um.

A partir do momento que o animal tiver Ciclo = 1 calcula-se a probabilidade do mesmo vir a conceber. Aqui considera-se também que o animal possui um valor máximo de probabilidade de vir a conceber, que varia de acordo com o sistema de produção. No nosso caso esse valor foi considerado igual a 0.95. Utilizando esse valor, a função de PCON será :

$$\mathbf{PCON} = 0.95 * \mathbf{CFW}^{0.2} * \mathbf{CFDW}^{0.2} * \mathbf{CFT}^{0.5} * \mathbf{MB}(T)$$

onde MB(T) = 1 nos meses T pertencentes à estação de monta e será igual a zero nos demais.

Da mesma forma que foi tratada a probabilidade do animal vir a ciclar, gera-se um número aleatório que, comparado com PCON, estabelecerá se o animal vai conceber ou não. Caso PCON seja menor ou igual ao número aleatório gerado, o valor a ser retornado ao programa será k = 1, ou seja, uma indicação de que o animal entrará no seu primeiro mês de gestação. Caso contrário k = 0.

O tratamento dado à rotina fertilidade, neste trabalho, de certa forma foi bem diferenciada da rotina equivalente utilizada no TAMU, principalmente porque os animais são trabalhados individualmente. No TAMU contabiliza-se uma outra variável que controla a porcentagem de animais de uma classe que vai ciclar e cujo valor interferirá na taxa de concepção. Não foi preciso utilizar essa variável pois foi calculada a probabilidade individual do animal vir a ciclar e a sua probabilidade de concepção, também individual. Sem dúvida, esse tratamento é muito mais real.

5.1.4 Procedimento Morte

Essa rotina simula a probabilidade de morte de cada animal, mensalmente, em função do período do ano, idade, condição e estado fisiológico dos animais.

A produção de leite, e seu grande efeito no crescimento simulado dos bezerros, interfere significativamente na taxa de morte desses bezerros. Também influenciam na morte dos bezerros o mês do ano e a idade das mães. Mães muito jovens ou muito velhas influenciam positivamente na taxa de morte dos bezerros.

Os potenciais genéticos e os planos de nutrição afetam as condições simuladas do estado fisiológico e, portanto, afetam as taxas de mortes simuladas. Os parâmetros de chamada da rotina Morte são: idade, peso, tamanho estrutural, peso ao nascimento e estágios de prenhez e lactação.

As taxas de mortes dos animais são ajustadas para cima para animais mais magros, mais jovens e para as vacas que pariram recentemente, principalmente

se forem muito jovens ou muito velhas. Em termos ambientais as taxas mínimas de mortes são ajustadas para valores maiores nos meses que causam mais *stress* nos animais, por exemplo nos meses de seca. No modelo de simulação o conjunto de valores **CT** é o responsável pela apresentação dos valores associados à influência do mês do ano na mortalidade dos animais, com exceção dos recém nascidos que possuem um conjunto próprio, o **CB**.

Outros três conjuntos de dados de entrada do modelo são utilizados pela rotina Morte:

- **CL**: mostra o efeito da idade do bezerro na sua própria morte;
- **CBA**: mostra como a idade da vaca influencia na morte do bezerro no parto;
- **CA**: representa a influência da idade da vaca na sua mortalidade, durante a parição.

Nessa rotina, devido ao tratamento individualizado de cada animal, também foi utilizada uma abordagem estocástica para definir sobre a morte ou não dos animais no presente mês de simulação. Essa abordagem, da mesma forma que na rotina Fertilidade, diferencia muito a rotina Morte da rotina equivalente utilizada pelo TAMU.

Na entrada do procedimento Morte há um bloco de verificação da necessidade de análise de morte perinatal ou não. Se for o caso, um fator de morte **FD** é calculado como **FD** = **CB**[T] * **CBA**[Idade da Mãe]. Esse fator considera, portanto, a influência do mês do ano e a idade da mãe na morte do bezerro ao nascer. Nesse caso é gerado um número aleatório que é comparado com o fator FD. Caso o número aleatório seja menor ou igual a FD o animal recém nascido será considerado morto e encerra-se o procedimento Morte. Se não se tratar de uma situação de nascimento, a rotina tem seqüência contabilizando a influência do mês do ano na morte do animal através de **FD** = 0.001 * **CT**[T].

Essa probabilidade mínima de morte será ampliada nos passos seguintes do procedimento de acordo com a condição do animal, comparando seu peso ao seu tamanho estrutural, sua idade e, no caso das vacas, é considerado também o fato da mesma ter parido no presente mês ou no mês anterior.

Em termos de condição do animal, a razão W/WM amplia o fator de morte do mesmo segundo a função $FD = FD * e^{140(1-W/WM)**3.3}$.

Considerando a idade do animal, se for bezerro o FD = FD * CL[Idade]. Se a idade for menor que quatro meses haverá, ainda, um fator que poderá ampliar a possibilidade de morte, de acordo com o peso ao nascimento do bezerro e do seu desenvolvimento corporal. Calcula-se qual seria o ganho de tamanho estrutural diário ideal para o bezerro, dado por **EDW** = (**WMP**-**BW**)/360. De acordo com a idade do animal calcula-se qual seria o seu WM (ideal) atual através da fórmula **EWM** = **BW** + 30 * **EDW** * [Idade]. Se WM < EDW tudo bem e o fator FD se mantém, caso contrário **FD** = **FD** + $(1-(WM/EWM)^{0.64})$.

Uma última consideração ocorre quando o animal for fêmea e tiver parido no mês atual ou no anterior. Se o parto ocorreu no presente mês o fator FD recebe influência do conjunto de valores CA, e **FD** = **FD** * **CA**[Idade do animal]. Se o parto ocorreu no mês anterior, **FD** = 1,25 **FD**.

Da mesma forma como foi apresentado para a morte perinatal, um número aleatório é gerado e, ao ser comparado com FD, indica se o animal morrerá no presente mês de simulação ou sobreviverá computando mais um mês de vida à sua idade.

A Figura 5.8 apresenta um fluxograma da rotina Morte.

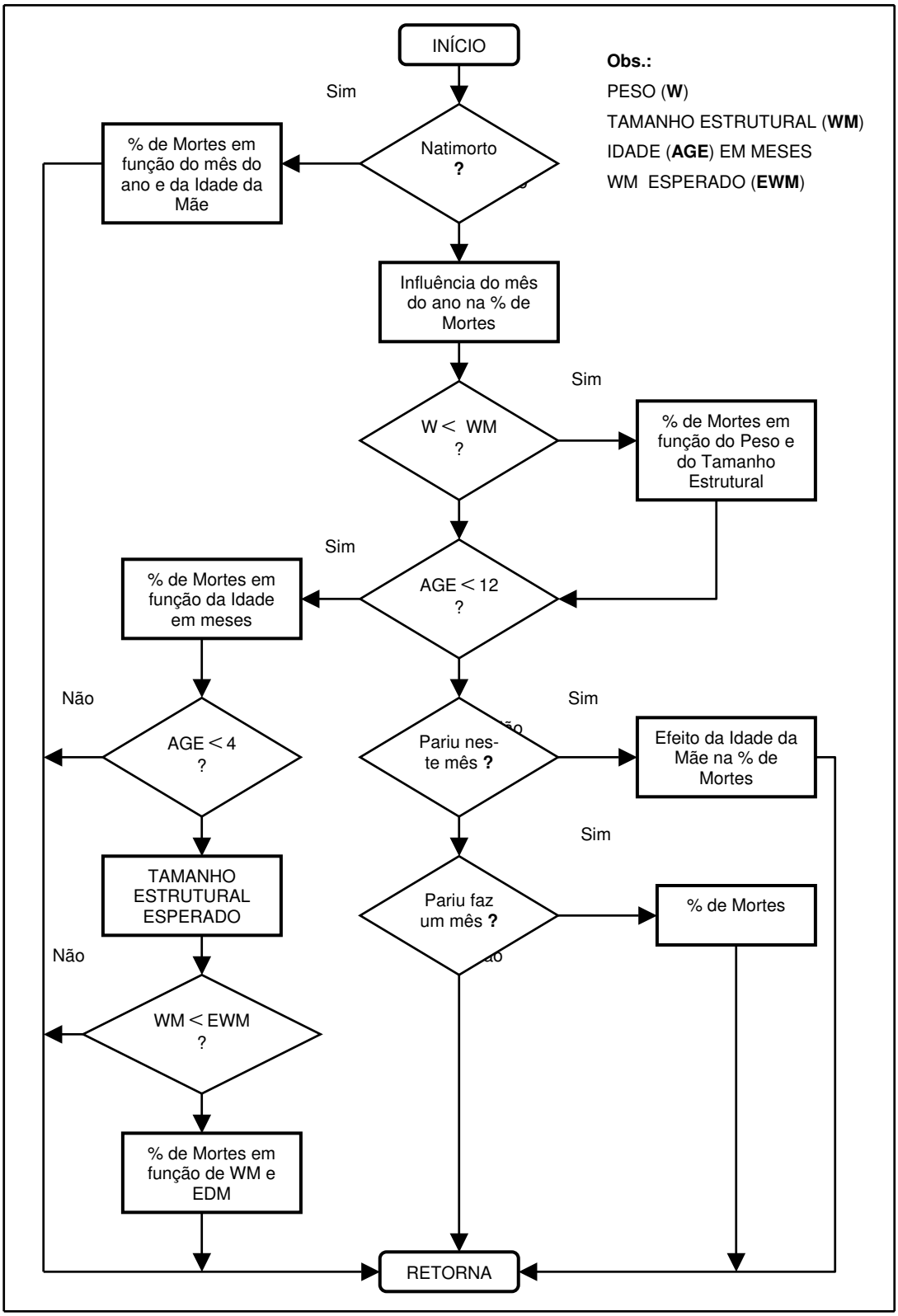


Figura 5.8 - Procedimento Morte

5.2 Modelo de simulação: lógica e animação

O funcionamento do simulador segue uma variante do Método das Três Fases (CROOKES et al., 1986; SALIBY & OLIVEIRA, 1991). A única adaptação feita foi em relação ao estabelecimento do próximo evento e, conseqüentemente, no tempo de ocorrência desse evento. Como a simulação acontece todo mês, não há necessidade de armazenar numa lista de eventos futuros os eventos que irão ocorrer, como normalmente é feito quando se utiliza o Método das Três Fases. Considera-se como o tempo de ocorrência do próximo evento o mês seguinte, atualiza-se o relógio da simulação, e desenvolve-se as fases normalmente.

O modelo de simulação foi programado no *software* ARENA. No módulo de execução é carregado o modelo de simulação e, de uma forma interativa, o usuário responde sobre o tamanho desejado da simulação e apresenta a sua política de manejo quanto à idade de desmame, idade do primeiro cruzamento e, se for utilizar estação de monta, em que período do ano ocorrerá tal estação. O usuário fornece também os dados sobre potenciais genéticos dos animais e os dados sobre as pastagens.

Os dados dos animais do rebanho são fornecidos através de um arquivo externo (**Forfait1_Input.txt**) que apresenta os animais, individualmente, através de uma identificação (uma classe e um número) que o acompanhará durante toda a simulação, por exemplo, Garrote 135, Novilha 45, Touro 3, e assim por diante. Os dados fornecidos de cada animal são: peso, tamanho estrutural, peso ao nascimento, idade, idade da mãe, identificação da mãe, e no caso das fêmeas, número de crias, estados de prenhez e lactação, e se está ciclando ou não.

O modelo cria uma fila para cada uma das classes dos animais, identificando-as como: Bezerras, Bezerros, Bezerras Desmamadas, Bezerros Desmamados, Novilhas, Garrotes, Vacas de *n* Crias (*n*:1..7), Vacas Prenhes, Vacas de Oito Meses de Gestação, Boi Magro, Boi Gordo e Touros. Cada animal é alocado a sua

fila correspondente e o conjunto dessas filas constitui o Banco de Dados (BD) do modelo.

Fornecidas as entradas, o modelo roda um procedimento central até que o tempo de simulação alcance aquele estabelecido pelo usuário. O procedimento central representa a estrutura central de controle da simulação.

Mensalmente, o simulador executa para cada animal os procedimentos Crescimento e Morte e, para as fêmeas que estão em idade de cobertura, executa também o procedimento Fertilidade. Os animais ao passarem por esses procedimentos sofrem alterações em seus dados individuais. O Banco de Dados (BD) é, portanto, atualizado. De posse dos novos dados, o modelo roda um módulo de decisão sobre compra e descarte de animais. De acordo com a política contida nas regras pertencentes a esse módulo, o BD é novamente alterado e encerra-se um mês de simulação. O módulo de decisão é apresentado em detalhes na seção seguinte. O ciclo constituído de requisitar dados do BD, simular o crescimento, morte e fertilidade dos animais, atualizar o BD, tomar decisões sobre compra e descarte de animais, e atualizar novamente o BD, é repetido mensalmente até atingir o tempo de simulação estipulado. Os resultados da rodada de simulação são apresentados em relatórios gerados automaticamente que fornecem os dados finais da simulação (sobre as classes dos animais, as pastagens consumidas, as decisões tomadas e um histórico da simulação de cada animal).

5.2.1 A lógica do modelo *SanCarlo*

O modelo *SanCarlo*, por motivos já comentados anteriormente, foi criado a partir do modelo TAMU da Texas A&M University, College Station, Texas, USA, que foi originalmente programado na linguagem de aplicação geral FORTRAN para representar um sistema de produção de bovinos de corte separando e tratando os

animais por classes. A primeira programação do modelo *SanCarlo* foi feita em PASCAL, utilizando um *shell* desenvolvido pela London School of Economics chamado ELSE (Extended Lancaster Simulation Environment; PAUL & CHEW (1987)). Nesse modelo os animais foram tratados individualmente para tentar representar de uma forma fiel o que ocorre no sistema real. Com o surgimento de novos *softwares* de simulação, o modelo foi transportado para o ambiente de simulação ARENA (PEGDEN, SHANNON & SADOWSKI (1995); KELTON, SADOWSKI & SADOWSKI (1998)). O *software* ARENA foi escolhido por ser um dos mais conhecidos e utilizados pacotes de modelagem de simulação do mundo e por conta da experiência adquirida, pelo autor desta tese, na aplicação desta ferramenta em projetos junto a empresas dos mais diferentes setores da economia brasileira. O ARENA permitiu ampliar a visualização da simulação, através da sua interface de animação que possui boas ferramentas gráficas para apresentação de dados e resultados. O ARENA também permite programações em Visual Basic (VBA), o que facilitou modular a lógica complexa, necessária para representar o sistema em estudo, e a troca de informações com outros *softwares*, no caso com sistemas computacionais baseados em conhecimento. A Figura 5.9 apresenta a tela de abertura do modelo *SanCarlo*, no software ARENA.

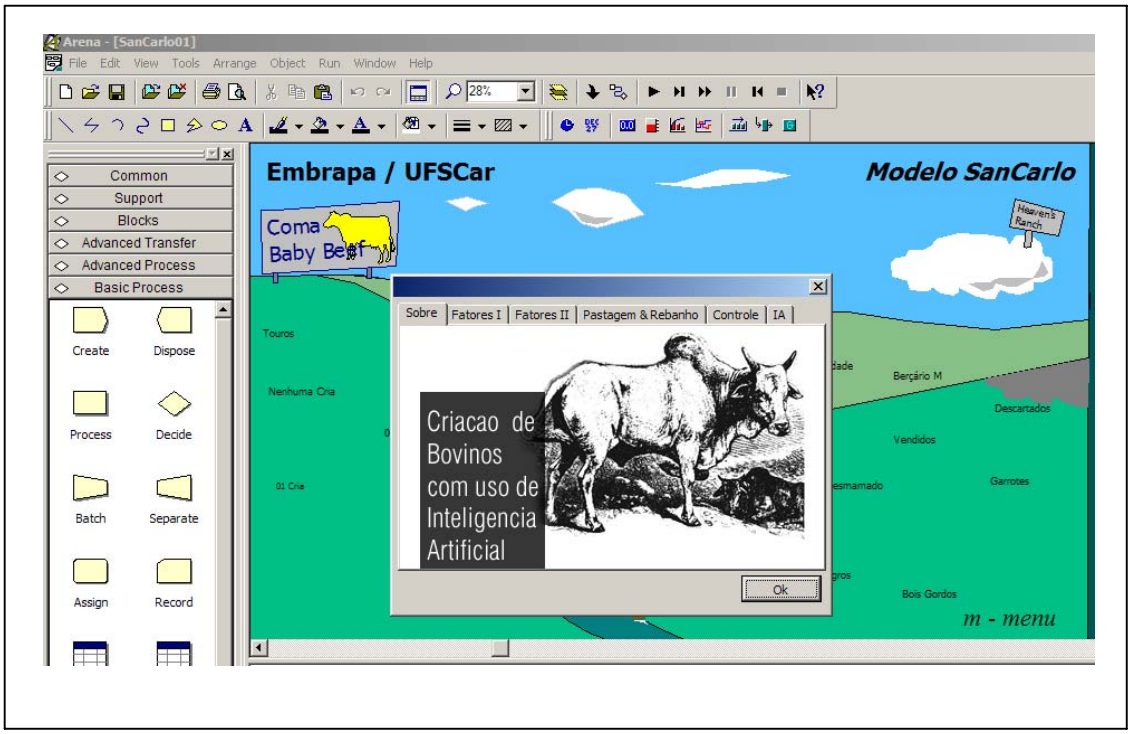


Figura 5.9 – Tela de abertura do modelo *SanCarlo*

As pastas contidas no menu principal do modelo são: Fatores I, Fatores II, Pastagem & Rebanho e IA (Inteligência Artificial).

A pasta Fatores I, apresentada na Figura 5.10, recebe os valores de entrada para os seguintes parâmetros: Idade de Desmama, Idade do Primeiro Cruzamento, Taxa de Produção de Leite, Tamanho Estrutural na Maturidade (WMA), Probabilidade Máxima da Vaca Ciclar, Probabilidade Máxima de Concepção e Fator de Correção Para a Lactação.

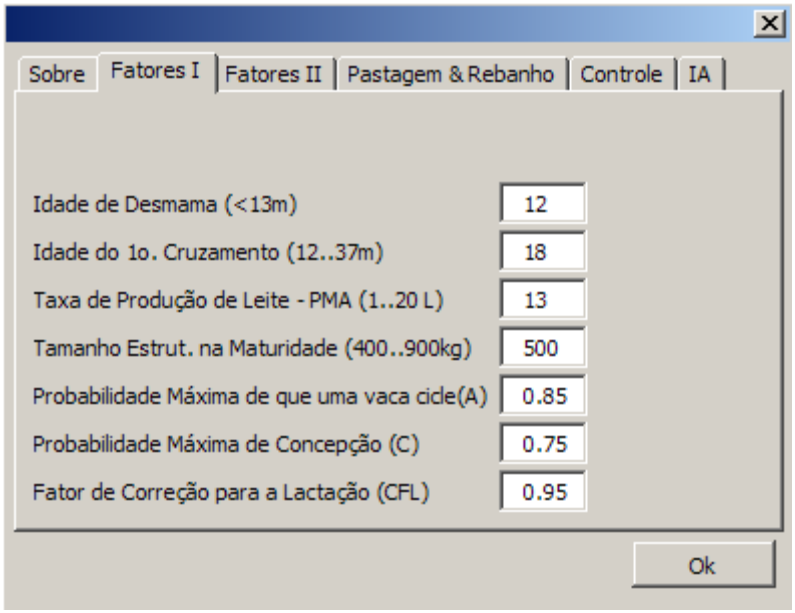


Figura 5.10 – Pasta Fatores I

A pasta Fatores II (Figura 5.11) é responsável pela entrada dos parâmetros:

- **CA**: Efeito da idade na morte do animal;
- **CB**: Efeito do mês do ano na morte perinatal;
- **CL**: Efeito da idade do bezerro na sua morte;
- **CBA**: Efeito da idade da vaca na morte do bezerro no parto;
- **CT**: Efeito do mês do ano na morte do animal;
- **MB**: Definição da estação de monta.

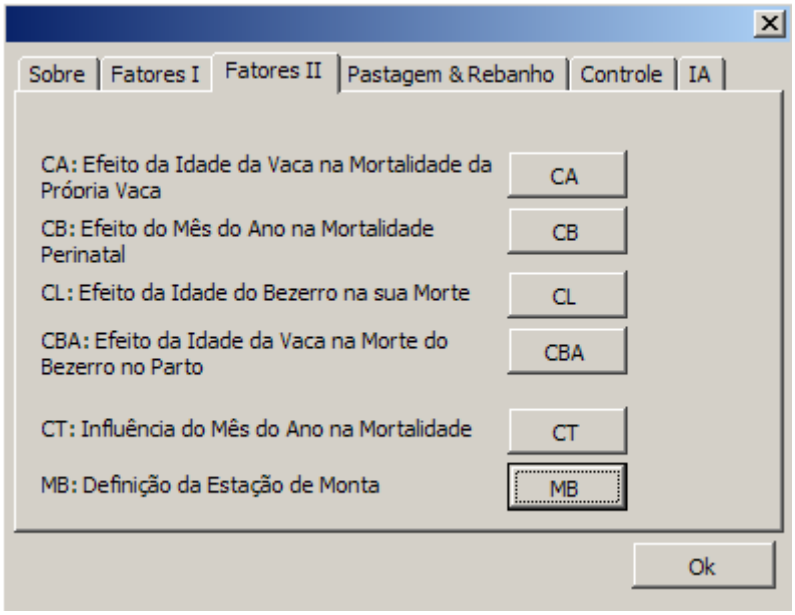


Figura 5.11 – Pasta Fatores II

Na pasta Pastagem & Rebanho (Figura 5.12) são dadas as entradas com as informações sobre os animais e sobre as pastagens. Em um arquivo texto externo (Forfait1_Input.txt) são fornecidos os dados dos animais do rebanho. Esses dados incluem: Código do Animal, Sexo (inclusive diferenciando se o animal macho for Touro), Idade, Peso ao Nascimento, Peso Atual, Tamanho Estrutural, Mês de Lactação, Mês de Gestação, Numero de Crias. Esses três últimos parâmetros são para fêmeas quando se enquadrarem em alguma das três situações (lactante, prenhe e mãe).

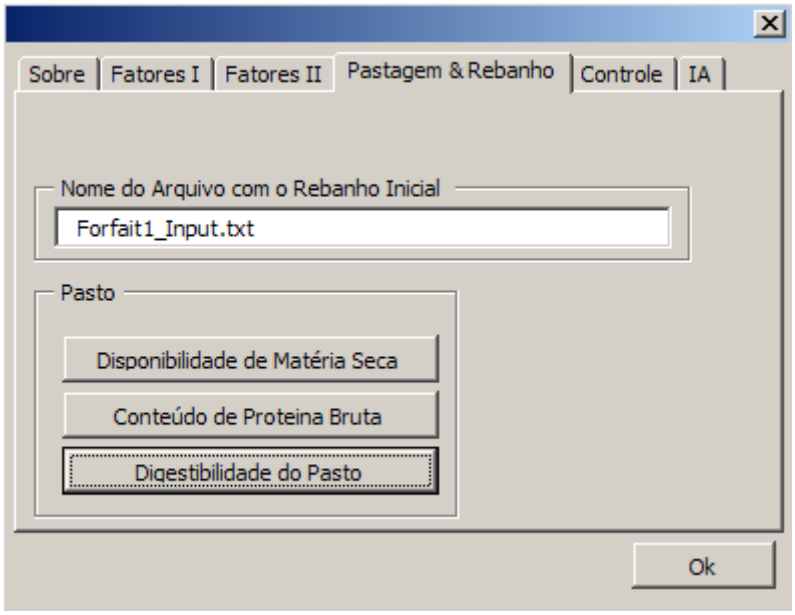


Figura 5.12 – Pasta Pastagem & Rebanho

Os dados sobre Pastagens dizem respeito à disponibilidade de massa de forragem, concentração de proteína bruta e digestibilidade da forragem.

Na pasta Controle (Figura 5.13) é estabelecido o tamanho da simulação desejada, em meses. São definidos, também, os nomes dos arquivos que conterão os dados mensais de saída da simulação referentes ao crescimento dos animais, às mortes ocorridas e às taxas de fertilidade e concepção geradas.

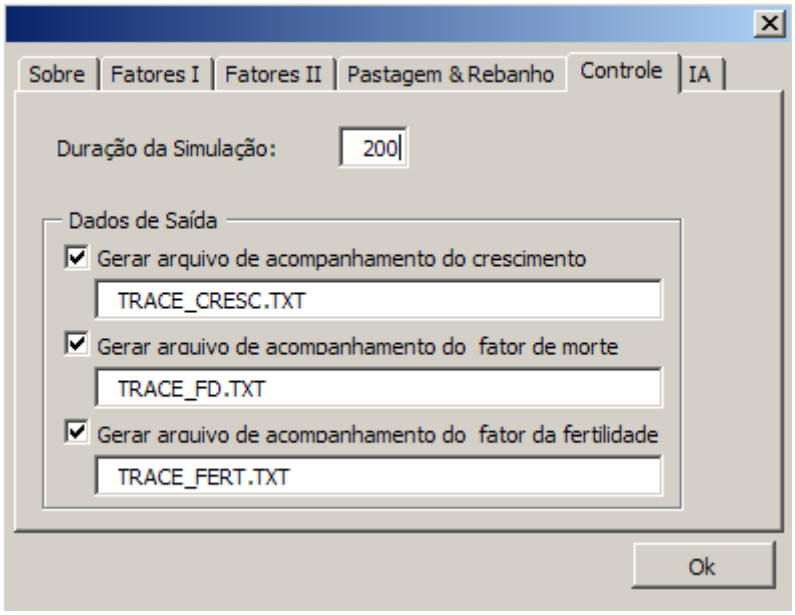


Figura 5.13 – Pasta Controle

A lógica do modelo, programada no software ARENA, devido a sua complexidade recebeu vários módulos em VBA (Visual Basic). Isso fez com que a lógica ficasse mais enxuta, deixando os detalhes para procedimentos criados no VBA interno ao próprio ARENA. Toda a lógica programada no *software* ARENA encontra-se no Apêndice 01. Os procedimentos criados no VBA estão no Apêndice 02.

Portanto, para efeito de ilustração, a programação do procedimento Crescimento (bastante complexo, como pode ser visto na seção 4.1.2) no ARENA fica com apenas um comando VBA (Figura 5.14). A programação desse comando é repassada para o módulo em VBA (parte reduzida dele é apresentada na Figura 5.15).

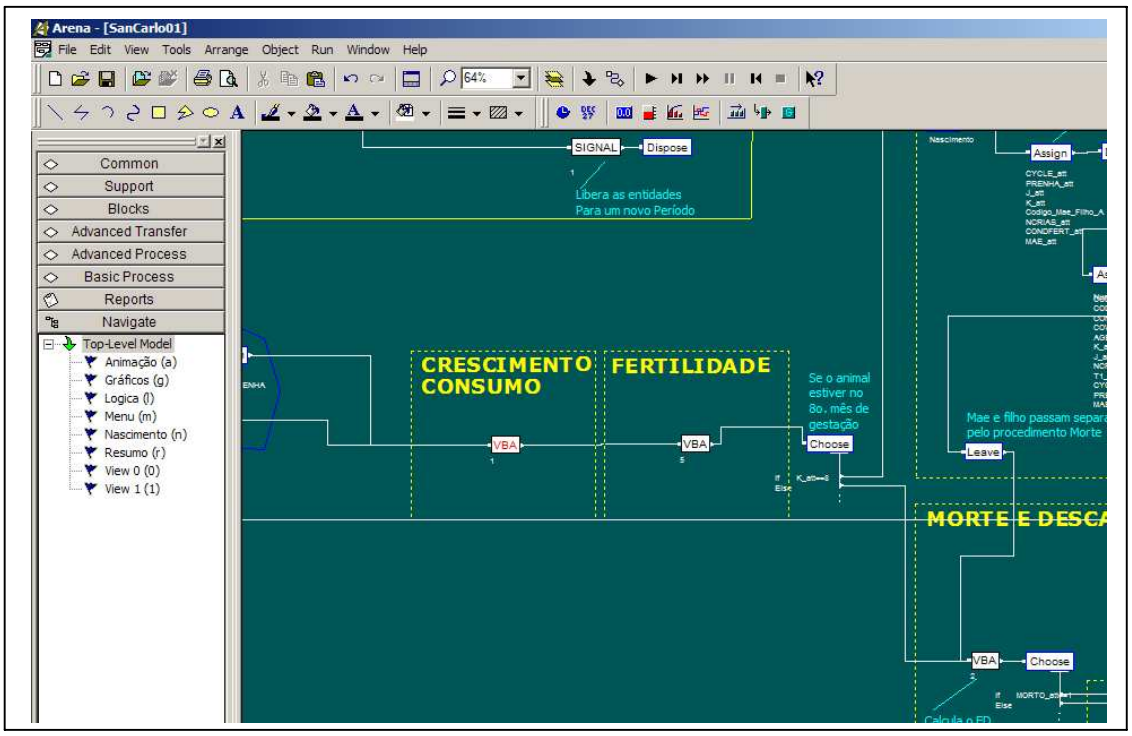


Figura 5.14 – Parte da Lógica do modelo *SanCarlo*

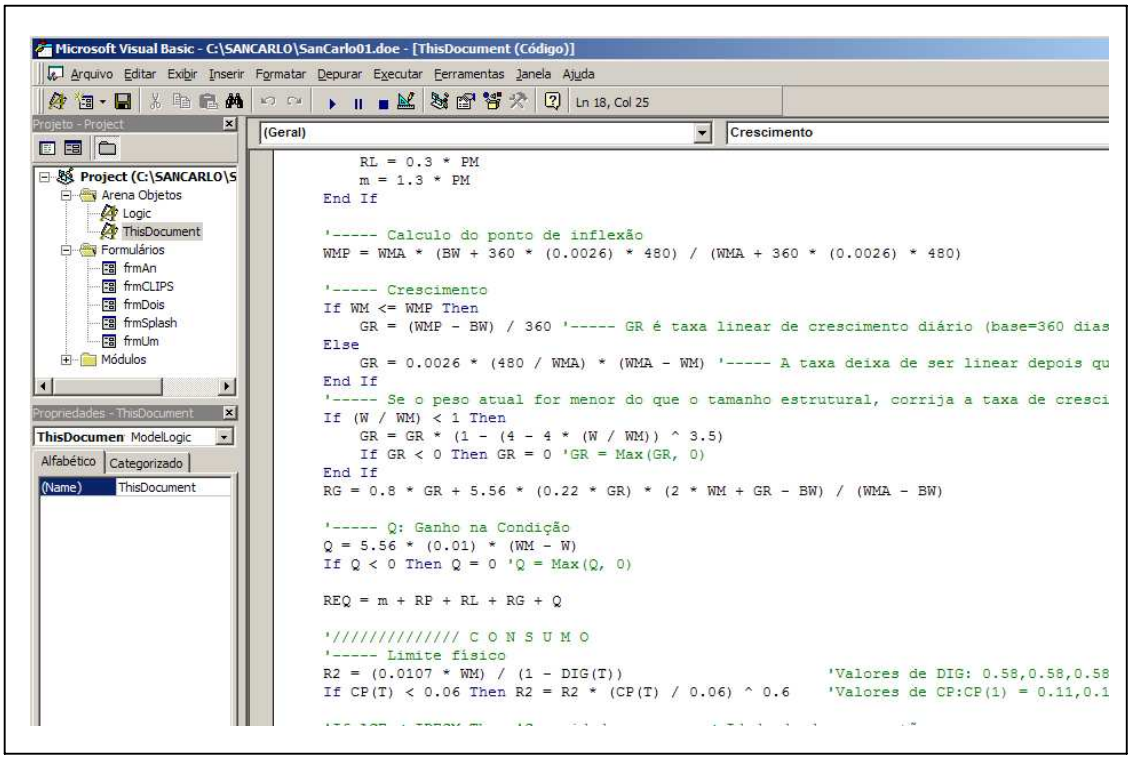


Figura 5.15 – Parte da programação, em VBA, do procedimento Crescimento

O modelo *SanCarlo* contém uma interface razoavelmente amigável para a entrada de dados e um módulo de inteligência artificial que gerencia a política de compra e descarte do pecuarista.

O ARENA, por possuir animação, permite uma visualização dos animais, no decorrer da simulação, mudando de pastagens, apartados por conta da prenhez, descartados (por venda ou morte), além de vários tipos de gráficos e tabelas que são alterados em tempo real com a simulação. Isso permite uma visualização do estado da simulação a qualquer momento.

5.3 Módulo de decisão baseado em conhecimento

A Figura 5.16 apresenta a entrada para o módulo de conhecimentos do criador, representado por regras de produção e modelado utilizando o *software* CLIPS ((CLIPS, 2002), (CLIPS, 2003), (GIARRATANO & RILEY, 1998)), desenvolvido pela NASA e disponibilizado para domínio público.

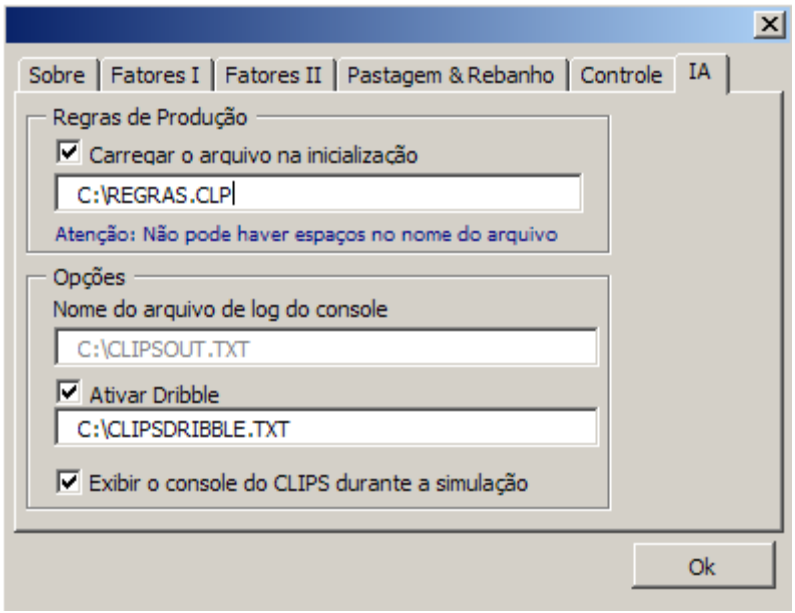


Figura 5.16 – Pasta referente à entrada das regras de produção

O CLIPS foi escolhido por ser um *software* que atendeu as necessidades estabelecidas no projeto da tese, estava disponível para uso público e permitiu uma boa integração com o ARENA.

As regras, programadas no CLIPS, são responsáveis pelas decisões sobre entrada e saída de animais do rebanho, de acordo com o desempenho dos mesmos durante a simulação levando em consideração as políticas do responsável pelo rebanho quanto à compra e descarte de animais.

5.3.1 Criando uma nova regra no sistema baseado em conhecimento

Toda regra é composta de duas partes principais: as condições e as ações. Para uma dada regra, se as condições forem verificadas as ações são executadas na máquina de inferência do sistema especialista, no caso o CLIPS. No modelo *SanCarlo* as ações acrescentam, modificam ou retiram fatos da agenda do CLIPS.

A Figura 5.17 mostra a tela do editor de regras que foi desenvolvido em português, buscando tornar amigável a interação usuário-modelo.

Figura 5.17: Tela do editor

Depois de criada a regra, pode-se acrescentar um comentário para registrar o objetivo da regra.

Para completar a regra é preciso acrescentar as condições e as ações. Uma condição pode ser baseada em atributos dos animais (por exemplo, peso ou idade) ou em alguma variável do modelo (mês do ano, se o mês é de monta ou não, área disponível ou alguma outra). No exemplo que segue será apresentada uma regra de desmama de um animal.

Para estabelecer uma condição deve-se seguir os seguintes passos:

1. Selecionar a lista adequada do elemento no qual a condição vai ser baseada. Selecionar a lista **Atributos dos Animais**
2. Clicar com o botão esquerdo do *mouse* sobre o atributo desejado, nesse caso, **Idade.**
3. Arrastar com o BOTÃO DIREITO do *mouse* o item selecionado sobre o rótulo SE da área de edição de regras, como mostrado na Figura 5.18.
4. Liberar o botão do *mouse*. Irá surgir a janela para edição da condição (Figura 5.19).

Na janela de edição da condição compara-se o elemento selecionado (no caso um atributo do animal) com um valor, outro atributo ou variável do modelo. No exemplo em questão será comparada a idade do animal (já selecionada) com o valor de uma variável definida no modelo: **Idade de desmama**.

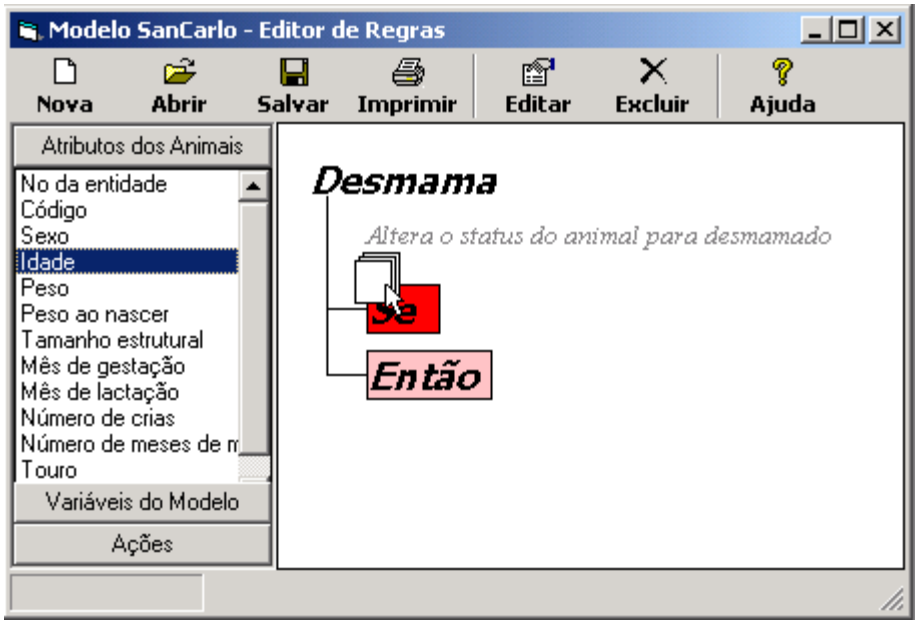


Figura 5.18 - Anexando uma condição baseada em um atributo dos animais

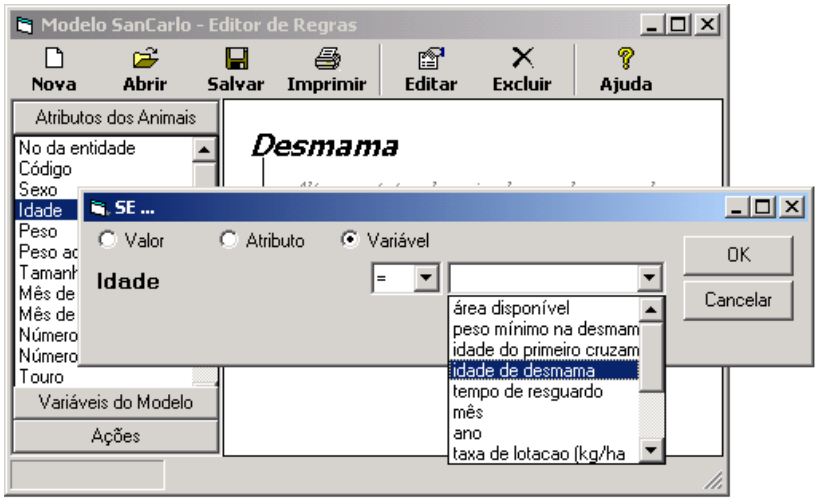


Figura 5.19 - Complementação da condição da regra

Para terminar a condição basta:

1. Selecionar o botão **Variável**
2. Selecionar o item **Idade de desmama** na lista suspensa
3. Selecionar o operador matemático
4. Clicar em **OK**. A condição ficará como mostrado na Figura 5.20.

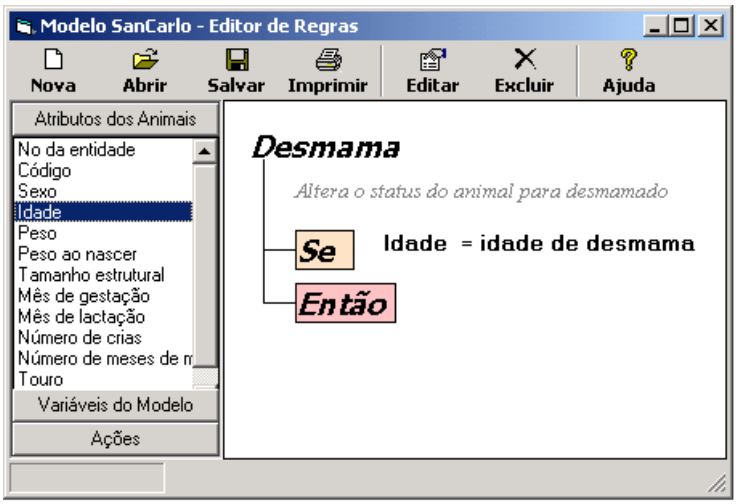


Figura 5.20 - A condição depois de terminada

As ações das regras no modelo *SanCarlo* alteram a agenda a ser submetida à máquina de inferência do CLIPS através do acréscimo, modificação ou eliminação de fatos.

Para estabelecer uma ação deve-se caminhar nos seguintes passos:

1. Clicar na lista **Ações** (Figura 5.21).
2. Selecionar com o BOTÃO ESQUERDO do *mouse* o item desejado, no caso **Alterar Status**.
3. Com o BOTÃO DIREITO do *mouse* arrastar o item sobre o rótulo ENTÃO.
4. Preencher os campos da janela de maneira semelhante à das condições
5. Clicar em OK.

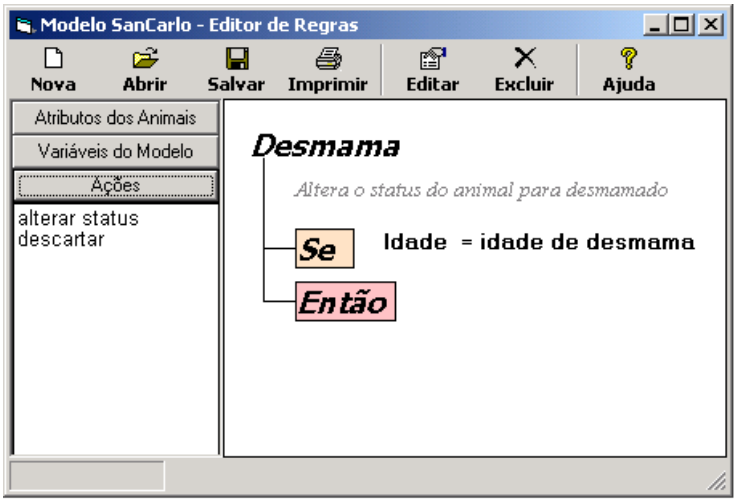


Figura 5.21 - Selecionando a lista de ações

Selecionado o *status* adequado (no caso, **desmamado**), tem-se a regra mostrada na Figura 5.22.

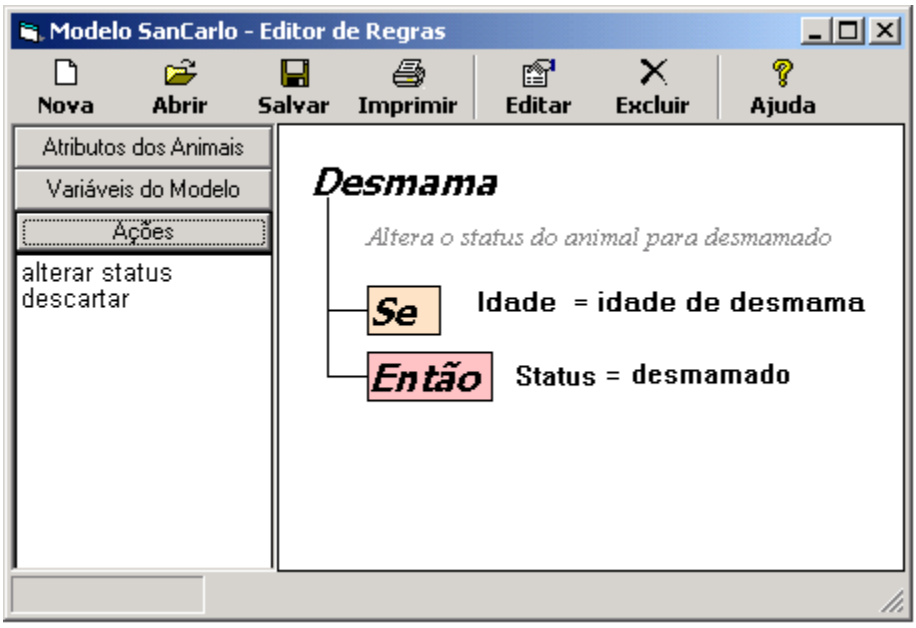


Figura 5.22 - Regra terminada

A regra em CLIPS correspondente pode ser vista clicando-se sobre o botão salvar. No exemplo seria a seguinte:

```
(defrule desmama "Altera o status do animal para desmamado"
   ?anim <- (animal (idade ?idade))
   (IDADE_DE_DESMAMA ?idesm)
  (test (= ?idade ?idesm))
=>
  (modify ?na (status desmamado))
)
```

5.3.2 Atuação conjunta entre o ARENA e o CLIPS

Em princípio, as variáveis globais do ARENA são transformadas em variáveis globais do CLIPS. Exemplos dessas variáveis:

- Peso_Mínimo_na_Desmama;
- Idade_de_Desmama;
- Idade_de Descarte_de_Touros;
- Idade_de _Descarte_de_Machos;
- Idade_de_Descarte_de_Fêmeas;
- Relação_a_Manter_entre_Touro_e_Vacas;
- Taxa_de_Lotação_do_Pasto_a_Manter.

Dessa forma, assim como no simulador, as variáveis globais do CLIPS (chamadas de **constructo** do tipo **defgloba**l) são visíveis, instantaneamente, pelas regras.
As variáveis globais do CLIPS têm que ter um prefixo "**?***" e um sufixo "*". Por exemplo, a variável **Idade_de_Desmama** será identificada no CLIPS como: **?*Idade_de_Desmama***.
No início da simulação o modelo, após criar uma instância do CLIPS, carrega as regras e demais elementos necessários da base de conhecimento. Nesse

momento são criadas as definições **defglobais** que representam as variáveis do ARENA.

Os animais são modelados como fatos (**constructo** do tipo **fact**). Para isso, utiliza-se um gabarito para os fatos que representam animais. Este gabarito de fatos é chamado de **deftemplate**. O deftemplate dos fatos que representam os animais é o seguinte:

```
(deftemplate animal
    (Slot id                    ; # da entidade do modelo ARENA
        (type INTEGER)
        (default 0)
    )
    (Slot cod                   ; Código do animal fornecido pelo pecuarista
        (type INTEGER)
        (default 0)
    )
    (Slot sexo                  ; Sexo do animal: macho ou fêmea
        (type SYMBOL)
        (allowed-symbols m f)
        (default f)
    )
    (Slot idade                 ; Idade do animal
        (type INTEGER)
        (default 0)
    )
    (Slot peso                  ; peso do animal em kg (W)
        (type NUMBER)
        (default 0)
    )

    (Slot peso_nasc             ; peso do animal em kg ao nascer(BW)
        (type NUMBER)
        (default 0)
    )
    (Slot tam                   ; tamanho estrutural do animal em kg (WM)
        (type NUMBER)
```

```
            (default 0)
        )
        (Slot gest                  ; mês de gestação: 0 = não gestante
            (type INTEGER)
            (default 0)
        )
        (Slot lact                  ; mês de lactação: 0 = não lactante
            (type INTEGER)
            (default 0)
        )
        (Slot crias                 ; número de crias da fêmea
            (type INTEGER)
            (default 0)
        )
        (Slot touro                 ; identifica se o animal é um touro
            (type SYMBOL)
            (allowed-symbols s n)
            (default n)
        )
        (Slot mae                   ; indica se o animal está com bezerro
            (type SYMBOL)
            (allowed-symbols s n)
            (default n)
        )
        (Slot status                ; define o status do animal no rebanho
            (type SYMBOL)
            (allowed-symbols ok venda descarte novo)
            (default ok)
        )
    )
```

A principal vantagem dos deftemplates é que ao se especificar um fato não é preciso especificar todos os campos (*slots*).

Toda vez que um animal termina sua etapa de crescimento é inserido, na base de conhecimentos do CLIPS, um fato que representa o animal. Depois que todos os animais são inseridos, a máquina de inferência roda e os fatos que representam os animais são alterados.

5.4 Processo de controle geral da simulação

Considerando os dados do rebanho simulado no mês atual, as regras de produção estabelecidas e os resultados da simulação, a base de dados é atualizada e uma nova simulação para o mês seguinte é iniciada. Esse processo cíclico é repetido até que o tempo de simulação, em meses, atinja o tempo estabelecido para o tamanho da simulação desejada. Esse processo está representado na Figura 5.23.

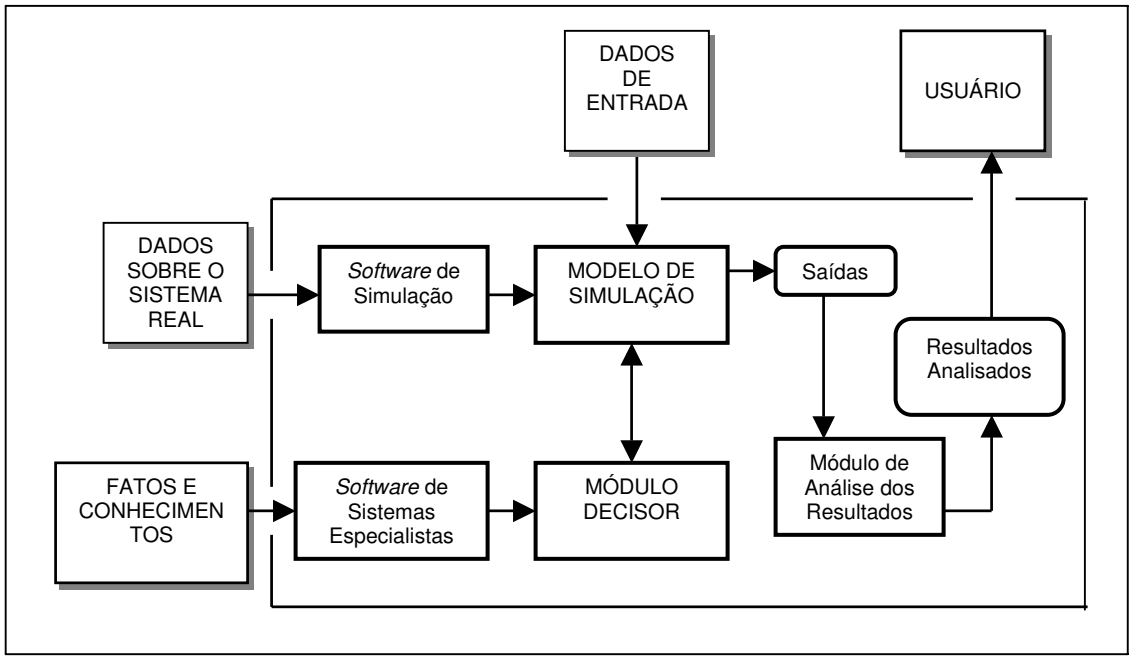


Figura 5.23 - Módulo de Controle Geral da Simulação

Da forma como é feito o controle da simulação, o usuário tem total domínio sobre os fatos gerados pelo modelo durante a rodada de simulação. Isso é feito através de uma interface, também criada em VBA, que possibilita parar a qualquer momento a replicação e acessar os fatos, permitindo ao usuário listar, acrescentar, retirar, enfim manipular os dados da simulação da forma que desejar.

A Figura 5.24 mostra a atuação conjunta do ARENA com o CLIPS. Uma caixa de diálogo permite acessar e modificar todas as informações de cada animal individualmente (COSTA & TORRES, 2004).

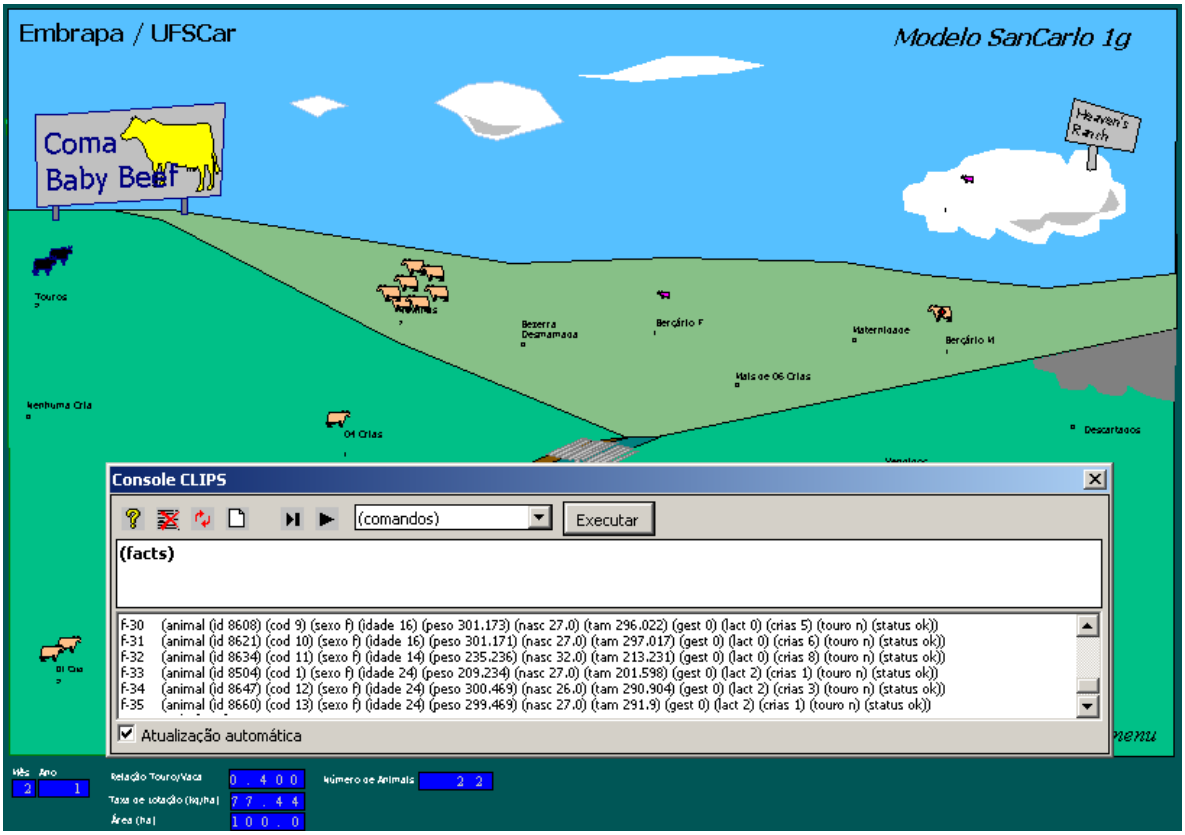


Figura 5.24 - Tela mostrando a atuação conjunta do ARENA com o CLIPS

6 Verificação e Validação do Modelo

Os processos de verificação e validação do modelo são desenvolvidos considerando os resultados das execuções realizadas no modelo de simulação. Os resultados da simulação podem ser apresentados de várias formas. HARRELL & TUMAY (1995) comentam que existem vários tipos de relatórios de simulação, por exemplo:

(i) Relatórios de uma única replicação (Mostram o desempenho da rodada de simulação utilizando média e variância);

(ii) Históricos detalhados, mostrando conjuntos de dados e "plotagens" de observações ocorridas no tempo. São as chamadas estatísticas dependentes do tempo ou estatística temporal;

(iii) Relatórios relâmpagos que são resumos estatísticos mostrando o estado do sistema em algum instante de tempo;

(iv) Relatórios de múltiplas replicações que providenciam resultados combinados de várias rodadas de simulação, fazendo tratamentos estatísticos nos resultados com estimativas de erros, dentro de uma faixa de incerteza desejada, baseando-se em intervalo de confiança (*output analyzer*);

(v) Relatórios que comparam as saídas de vários cenários analisando os conjuntos de variáveis representativas dos mesmos.

As análises dos resultados também dependem do tipo da simulação adotada. A simulação pode ser identificada como sendo **Terminal** ou **Não-Terminal**. A diferença entre as duas está na possibilidade de ser definido um comprimento para a simulação. Se um sistema tem claramente definido um instante de início e um instante de término, o sistema é considerado terminal. Caso contrário, é chamado de sistema não-terminal. No nosso caso o sistema tratado é terminal. Para a simulação terminal é importante estabelecer quantas execuções (replicações) devem acontecer para que os resultados estatísticos tenham consistência.

Por se tratar de um sistema terminal, a forma escolhida de análise de *output* foi através de múltiplas replicações, analisando os desvios, sob uma taxa de incerteza, com os resultados sendo tratados através de intervalos de confiança (próximo da opção *iv* apresentada nos tipos de relatórios). Um intervalo de confiança é um *range* (intervalo) no qual a média se encontra, com um certo nível de certeza. É calculado a partir da média e do desvio padrão, obtidos em uma quantidade de amostras (***n*** replicações).

A verificação do modelo *SanCarlo* se deu através de vários testes considerando sempre uma variável de controle por vez. É a forma tradicional de se testar um modelo. Como um conjunto de testes exaustivo seria praticamente impossível, a opção é rodar o modelo testando suas variáveis críticas, como por exemplo, o peso do animal. A forma de trabalho abordada seguiu a idéia de um sistema de controle com *feedback*. Através desse tipo de abordagem foi possível refinar fórmulas e ajustar parâmetros do modelo. Quando o modelo passou a responder bem aos cenários estabelecidos, considerou-se que o modelo estava bem construído, ou seja, foi programado de acordo com a lógica pré-estabelecida. Essa fase é chamada de verificação do modelo.

Como é possível construir corretamente um modelo errado, que não atenda a realidade, a fase de verificação precisa ser seguida de uma fase de validação. A fase de verificação considera dados hipotéticos, com a preocupação em rodar o maior número possível de replicações checando os valores das variáveis críticas. Variável crítica é aquela onde uma pequena alteração no seu valor causa uma grande interferência no sistema.

A fase de validação trata com dados reais. As comparações são entre resultados simulados e resultados obtidos no sistema real. Neste trabalho foram considerados os dados de um experimento da EMBRAPA/CPPSE de São Carlos.

Na verdade, são dois os momentos em que o modelo *SanCarlo* entrou em processo de validação. Quando o modelo foi programado pela primeira vez, na linguagem Pascal, o mesmo foi rodado inúmeras vezes de tal forma que os ajustes nas funções e parâmetros ocorreram praticamente naquele momento. De posse do modelo testado, foi feita a validação do mesmo utilizando-se dados de

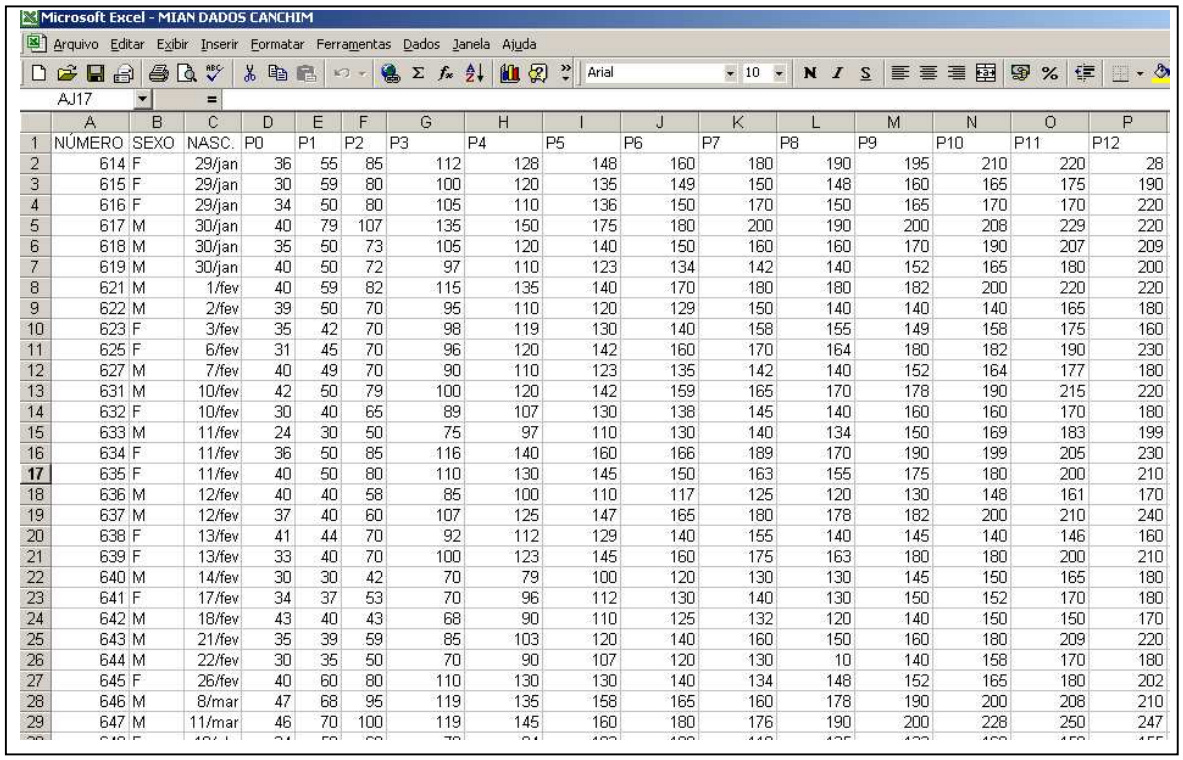
um experimento do Centro de Pesquisa de Pecuária do Sudeste, da EMBRAPA. Como o modelo sofreu modificações na sua programação, devido à troca de *softwares* de simulação utilizados, foi preciso repetir os processos de verificação e validação para a versão atual do modelo *SanCarlo*. O processo de verificação foi o mesmo adotado anteriormente, onde após um grande número de rodadas foi atestado que o modelo está correto em termos de lógica e programação.

O modelo está, no momento, sendo validado utilizando dados de uma fazenda de Nioaque, MS, que possui um banco de dados bem estruturado e atualizado. A partir dessa nova validação o modelo migrará para o formato de modelo comercial. Os tipos de dados existentes na base da fazenda em questão são os seguintes:

- Os tipos de alimentação existentes são: pasto; semi-confinado e confinado;
- A identificação do animal possui os seguintes campos: fazenda; código do animal; nome do animal; descrição (13 tipos); registro; código do pai; fazenda do pai; código da mãe; fazenda da mãe; data de nascimento; data de compra; sexo; pelagem (11 tipos); categoria; grau-sangue (30 subdivisões); raça; consangüinidade; criador; lote; regime alimentar; peso ao nascimento; peso anual;
- Categoria do animal: animal desmamado; animal mamando; engorda; animal morto; novilha em reprodução; novilha; touro múltiplo; tourinho; touro rufião; touro; vaca com bezerro ao pé; animal vendido; vaca em lactação; vaca seca; vaca solteira; prenha;
- Total de animais por lote; peso de cada animal; peso médio; desvio padrão;
- Pastos e localização;
- Causa *mortis*;
- Tipo de parto;
- Situação da cria;
- Dados de peso dos animais, anuais, de 1998 a 2003 (um total de 91626 tomadas de peso).

A base de dados possui, ainda, uma quantidade e variedade muito grande de informações, uma vez que essa empresa trabalha também com inseminação artificial. Além disso, a base possui outro grande conjunto de dados para trabalhar a parte econômico-financeira.

A seguir é apresentado um exemplo para ilustrar o método adotado para validar o modelo. No caso, foram escolhidos 32 animais do rebanho experimental da EMBRAPA/CPPSE. Esses animais possuíam datas de nascimento próximas. A Figura 6.1 apresenta parte da planilha contendo os dados dos animais, identificados por código, sexo, data de nascimento, peso ao nascimento e as pesagens obtidas mensalmente.



Microsoft Excel - MIAN DADOS CANCHIM

Arquivo Editar Exibir Inserir Formatar Ferramentas Dados Janela Ajuda

AJ17 =

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	NÚMERO	SEXO	NASC.	P0	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12
2	614	F	29/jan	36	55	85	112	128	148	160	180	190	195	210	220	28
3	615	F	29/jan	30	59	80	100	120	135	149	150	148	160	165	175	190
4	616	F	29/jan	34	50	80	105	110	136	150	170	150	165	170	170	220
5	617	M	30/jan	40	79	107	135	150	175	180	200	190	200	208	229	220
6	618	M	30/jan	35	50	73	105	120	140	150	160	160	170	190	207	209
7	619	M	30/jan	40	50	72	97	110	123	134	142	140	152	165	180	200
8	621	M	1/fev	40	59	82	115	135	140	170	180	180	182	200	220	220
9	622	M	2/fev	39	50	70	95	110	120	129	150	140	140	140	165	180
10	623	F	3/fev	35	42	70	98	119	130	140	158	155	149	158	175	160
11	625	F	6/fev	31	45	70	96	120	142	160	170	164	180	182	190	230
12	627	M	7/fev	40	49	70	90	110	123	135	142	140	152	164	177	180
13	631	M	10/fev	42	50	79	100	120	142	159	165	170	178	190	215	220
14	632	F	10/fev	30	40	65	89	107	130	138	145	140	160	160	170	180
15	633	M	11/fev	24	30	50	75	97	110	130	140	134	150	169	183	199
16	634	F	11/fev	36	50	85	116	140	160	166	189	170	190	199	205	230
17	635	F	11/fev	40	50	80	110	130	145	150	163	155	175	180	200	210
18	636	M	12/fev	40	40	58	85	100	110	117	125	120	130	148	161	170
19	637	M	12/fev	37	40	60	107	125	147	165	180	178	182	200	210	240
20	638	F	13/fev	41	44	70	92	112	129	140	155	140	145	140	146	160
21	639	F	13/fev	33	40	70	100	123	145	160	175	163	180	180	200	210
22	640	M	14/fev	30	30	42	70	79	100	120	130	130	145	150	165	180
23	641	F	17/fev	34	37	53	70	96	112	130	140	130	150	152	170	180
24	642	M	18/fev	43	40	43	68	90	110	125	132	120	140	150	150	170
25	643	M	21/fev	35	39	59	85	103	120	140	160	150	160	180	209	220
26	644	M	22/fev	30	35	50	70	90	107	120	130	10	140	158	170	180
27	645	F	26/fev	40	60	80	110	130	130	140	134	148	152	165	180	202
28	646	M	8/mar	47	68	95	119	135	158	165	160	178	190	200	208	210
29	647	M	11/mar	46	70	100	119	145	160	180	176	190	200	228	250	247

Figura 6.1: **Dados do rebanho experimental da Embrapa/CPPSE**

A entrada para o modelo *SanCarlo* se dá através de um arquivo texto. Nele estão contidos os dados dos animais nas mesmas posições ocupadas pelas

correspondentes variáveis no ARENA, e serão lidos através de um comando READ que importa dados no formato texto ou planilha.

A Figura 6.2 mostra uma saída da simulação com os valores simulados para os animais (com entrada dos dados referentes aos animais quando tinham a idade de 12 meses e gerando o peso de cada um no 24º. mês de vida). A partir desses dados é possível fazer comparações animal por animal ou qualquer tipo de estatística.

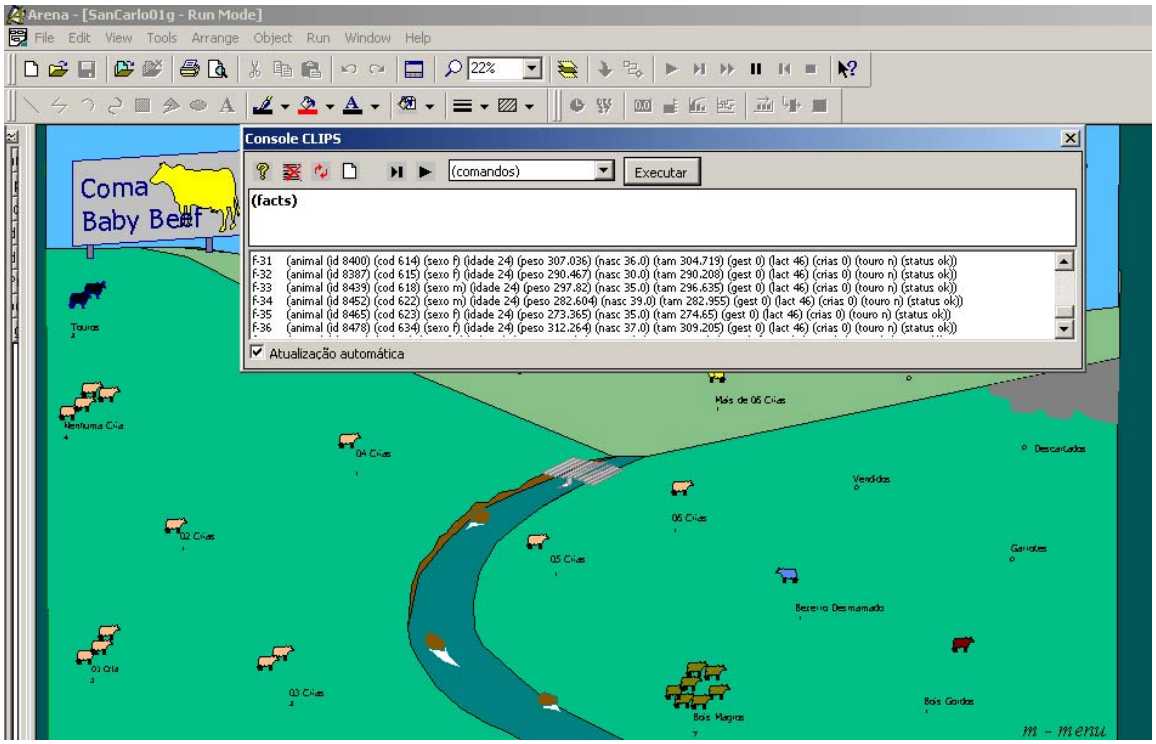


Figura 6.2: Resultados da simulação

A título de ilustração, o rebanho simulado apresentou um resultado bom. Temos que levar em conta que o tamanho da amostra utilizado no exemplo é pequeno pequeno. O conjunto de valores do sistema real (Embrapa/CPPSE) apresentou uma média=297 e desvio padrão= 19.1, enquanto que os valores simulados resultaram em uma média=294 e desvio padrão= 14.8.

Conforme dito anteriormente, agora ocorrerá novamente o processo de validação com uma base de dados bem maior. Após essa nova validação, o modelo deixará de ser um protótipo de pesquisa e poderá ser implementado em sistemas reais.

6.1 Comentários Gerais

A simulação de sistemas procura reproduzir uma realidade em um programa computacional, modelado de forma a representar essa realidade, com o objetivo claro de prever o futuro. A simulação é utilizada quando não é possível, ou não é interessante, fazer experimentações no sistema real, por problemas de complexidade, tempo demandado, recursos necessários, ou outro qualquer. A simulação tem por meta, emular o sistema real em um computador, através de um conjunto de cenários diferentes, cujos resultados são utilizados para auxiliar o tomador de decisão nas suas estratégias, nas suas táticas e nas suas operações.

O processo de simulação tradicional, apesar de viver um crescimento significativo, em termos de aplicação, na última década do século passado, é um processo amarrado, pouco flexível quando se pensa no aproveitamento de um modelo para tratar situações semelhantes, porém não idênticas (reutilização de modelos). Isso ocorre porque o modelo é desenvolvido através de procedimentos programados utilizando linguagens de aplicação geral, ou específica de simulação, que exige nova codificação toda vez que o modelo sofrer alguma alteração devido a uma nova política ou postura do tomador de decisão. Para contornar esse problema é possível combinar conceitos de simulação com conceitos de inteligência artificial, em particular sistemas especialistas.

Tanto a simulação, como a inteligência artificial utilizam metodologias que buscam modelar a realidade, representando os objetos e as relações entre eles. A simulação e os sistemas especialistas são baseados na representação modular de um sistema, com um mecanismo de inferência que dirige essa representação. A simulação assim criada é chamada de “simulação inteligente” ou simulação baseada em conhecimento. Essa abordagem transforma a simulação de uma ferramenta descritiva, em uma ferramenta prescritiva.

Os conceitos de simulação e sistemas baseados em conhecimento, apesar de consistentes, não solucionam problemas se não tiverem como sustentação uma metodologia que estruture e oriente os passos do desenvolvimento do modelo representativo da situação/problema.

No estudo de caso tratado nesse trabalho, a metodologia SimuCAD, já aplicada na solução de diversos problemas envolvendo empresas dos mais variados setores, foi adaptada para atuar no setor primário da economia, na agropecuária, mais especificamente no setor de produção de bovinos de corte. A contribuição mais marcante dessa metodologia diz respeito à introdução do conceito de centro de produção no sistema de produção pecuário. A forma de tratar a atividade e detalhá-la como um processo produtivo, adicionada a um levantamento minucioso dos fatores de produção existentes no sistema estudado, culminando com a definição dos sistemas físico, de fluxos e dinâmico que representam o sistema real, permitiu um bom tratamento de análise e síntese do problema, orientando a construção de um modelo consistente de simulação para auxiliar na tomada de decisões do pecuarista.

7 Conclusões

Atendendo ao objetivo central deste trabalho, foi desenvolvido um modelo de simulação para auxiliar a tomada de decisão do pecuarista no planejamento e controle de rebanhos de bovinos de corte. Trata-se de um modelo de simulação de eventos discretos, criado a partir de uma adaptação do método das três fases, genérico o suficiente para simular, mensalmente, qualquer produção extensiva de bovinos de corte. Os animais são tratados de forma individual, independentemente do tamanho do rebanho. Este simulador foi programado inicialmente na linguagem Pascal e depois, por motivos de vantagens computacionais, foi re-programado no *software* ARENA que possui uma boa interface gráfica, além de facilidades de integração com outras ferramentas computacionais, via Visual Basic.

Para representar a política de manejo do pecuarista, foi atrelado ao modelo de simulação um módulo de decisão baseado em conhecimento, criado a partir de conceitos oriundos da inteligência artificial, em particular da área de sistemas especialistas. O conhecimento do pecuarista foi representado, utilizando regras de produção, no CLIPS, um *software* desenvolvido pela NASA e disponibilizado para domínio público.

Dada a complexidade do tema estudado, foi utilizada para orientar a execução deste trabalho uma metodologia chamada SimuCAD, desenvolvida pelo grupo de estudos e pesquisa SimuCAD, do Departamento de Engenharia de Produção da Universidade Federal de São Carlos. Este grupo é coordenado pelo autor desta tese.

O modelo de simulação resultante foi testado e validado através de dados reais de um rebanho experimental do Centro de Pesquisa da Pecuária do Sudeste , da EMBRAPA**.** No momento, está passando por um novo processo de validação uma vez que foi atualizado.

Durante o desenvolvimento da tese, algumas linhas potenciais de pesquisa, sobre animais e pastagens, foram identificadas. É o caso dos temas digestibilidade e quantidade de pastagens e definição do conteúdo de proteína bruta dos alimentos ingeridos. Esses temas já estão sendo trabalhados em recentes projetos da EMBRAPA/Centro Nacional de Pesquisa de Gado de Corte de Campo Grande,

MS (EUCLIDES & MEDEIROS, 2003) e da EMBRAPA/Centro de Pesquisa da Pecuária do Sudeste de São Carlos, SP (CORRÊA & SANTOS, 2003).

Com os objetivos da tese atendidos, uma proposta de trabalho para um futuro imediato é acoplar ao simulador um módulo para fazer análises econômicas. Com isso, o modelo poderá, a partir de dados do pecuarista, do mercado e da bolsa de mercadorias, ampliar a sua função de apoio à decisão, contemplando as áreas de custos e finanças.

Devido aos bons resultados obtidos, este trabalho incentiva a aplicação, da metodologia e dos conceitos teóricos adotados, em sistemas de produção de animais de outras raças, como caprinos, ovinos e bubalinos, uma vez que a estrutura do modelo seria a mesma, necessitando apenas identificar as variáveis, parâmetros e relações funcionais, associadas ao novo animal em estudo e seu ambiente.

Ainda, pensando de forma mais abrangente, novas aplicações de modelos análogos ao desta tese podem ocorrer em qualquer outro tipo de sistema produtivo, não se prendendo somente à produção animal.

Referências Bibliográficas

ALENCAR, J.A., LIMA, E.C., PEDEAG – Plano Estratégico da Agricultura Capixaba, V.12, Pecuária de Corte, Secretaria de Estado da Agricultura, ES, 2003.

ANNINO,J.S., RUSSEL,E.C., The seven most frequent causes of simulation analysis failure – and how to avoid them, Interfaces, V11, N3, pp. 59-63, 1981.

ANUALPEC, Anuário da Pecuária Brasileira 2004, FNP Consultoria & Agro-informativos, São Paulo, 2004.

ARENA, manuais, Rockwell Softwares Inc., Pittsburgh, 2004.

AUTOMOD, manuais, Brooks Automation Inc. Utah, 2002.

BAKER, J.F., Evaluation of genotype-environment interactions in beef cattle production systems using a computer simulation model, Texas A & M University, College Station, Tx, 1982.

BANKS, J., CARSON II, J., Discrete-event system simulation, Englewood Cliffs, Prentice-Hall, 1984.

BANKS, J., Getting started with AUTOMOD, Brooks Automation, S.L. City, 2000.

BANKS, J., NORMAN, V., Justifying simulation in today's manufacturing environment, IIE Solutions, pp. 16-19, 1995.

BARBOSA, P.F., Análise genética-quantitativa de características de crescimento e reprodução em fêmeas da raça Canchim. Faculdade de Medician/USP, Ribeirão Preto, SP, tese de doutorado, 1991.

BARBOSA, P.F., ALENCAR, M.M., SILVA, A.M., Peso à maturidade, taxa de maturação e eficiência produtiva em fêmeas da raça Canchim, Arq. Brás. Méd. Vet. Zootec. V. 54, No. 5, Belo Horizonte, 2002.

BUCHIARELLI, L. L. Designing engineers. London: MIT Press, 1994.

BULGREN, W.G., Discrete System Simulation, Prentice-Hall, Englewood Cliffs, NJ, 1982.

CAMAROTTO, J.A., COSTA, M.A.B., MENEGON, N.L., OLIVI, A.C.K., TORRES, I., ZANONI, L.C., Simulação e computação gráfica aplicadas ao projeto de instalações industriais, ENEGEP, CD Rom, 1999.

CLEMENTSON,A.T., Extended control and simulation language, CLE.COM Ltda, Birmingham, 1982.

CLIPS, Reference Manual, Volumes 1, 2 e 3, Houston, Tx, 2003.

CLIPS, User's Guide, Houston, Tx, 2002.

CORRÊA, L.A., SANTOS, P.M., Produção de carne em pastagens adubadas, Embrapa Pecuária Sudeste, Sistemas de Produção 2, versão eletrônica, 2003.

COSTA, A., GURGEL, A., Programas que raciocinam, Micro Sistemas, pp. 14-20, Maio, 1984.

COSTA, M.A.B., Uma abordagem sobre inteligência artificial e simulação, com uma aplicação na pecuária de corte nacional, Produção, V.2, N.1, pp. 51-60, Setembro, 1991.

COSTA, M.A.B., MENEGON, N.L., CAMAROTTO, J.A., Simucad: Simulação & CAD – Um sistema de apoio à decisão no planejamento e implantação de sistemas de produção, XVI ENEGEP, CD Rom, Piracicaba, 1996.

COSTA,M.A.B., BARBOSA,P.F., Ferramentas computacionais de apoio à tomada de decisão em sistemas agropecuários e agroindustriais, I Simpósio Nacional de Instrumentação Agropecuária, anais, pp. 180-187, EMBRAPA/CNPDIA, São Carlos, SP, 1997.

COSTA,M.A.B., Pesquisa Operacional Aplicada à Agroindústria, Gestão Agroindustrial, Coord. M.O. Batalha, Atlas, 2001.

COSTA, M.A.B., TORRES, I., Modelos de simulação como ferramentas de auxílio à tomada de decisões em sistemas de produção de gado de corte, Cap. IV do Livro: Modelagem e simulação de sistemas de produção animal, Ed.Tec.: Barbosa, P.F., Assis, A. G. e Costa, M.A.B., EMBRAPA/Pecuária Sudeste, 2002.

COSTA, M.A.B., MENEGON, N.L., CAMAROTTO, J.A., TORRES, I., Aplicações da metodologia Simucad em empresas dos setores agroindustrial e agropecuário, IV Congresso Brasileiro da Sociedade Brasileira de Informática Aplicada à Agropecuária e à Agroindústria, Porto Seguro, 2003.

COSTA, M.A.B., TORRES, I., Study of beef cattle production systems using simulation and knowledge-based systems: the SanCarlo model, em Knowledge Engineering and Management applied for the development of productive systems, published by ABEPRO, 2004.

CROOKES,J.G., BALMER,D.W., CHEW,S.T., PAUL, R.J., A three-phase simulation system written in Pascal, JORS, Great Britain, V37, N6, pp. 603-618, 1986.

DEWHURST, F.W., GWINNETT, E.A., Artificial intelligence and decision analysis, JORS, Great Britain, V.41, N.8, pp. 693-701, 1990.

DOUKIDIS,G.L., An anthology on the homology of simulation with artificial intelligence, JORS, Great Britain, V.38, N.8, pp. 701-712, 1987.

DZAMA, K., WALTER, J.P., RUVUNA, F., SANDERS, J.O., CHIMONYO, M., Index selection of beef cattle for growth and milk production using computer simulation modelling, South African Journal of Animal Science, V. 31, N. 2, pp. 65-75, South African, 2001.

EKLUND, J. et al., Creating usable knowledge through co-operative research: an emerging model. Sweden: Linkoping University, 1998.

EUCLIDES, V.P.B., MEDEIROS,S.R., Valor nutritivo das principais gramíneas cultivadas no Brasil, EMBRAPA/CNPGC, relatório de pesquisa a ser publicado, Campo Grande, 2003.

EVANS, J.B., Simulation and intelligence, _____, Hong Kong, 1984.

FINLAY, P.N., Decision support systems and expert systems: a comparison of their components and design methodologies, Computers & Operations Research, Great Britain, V.17, N.6, pp. 535-543, 1990.

FOX, M.S., AI and expert systems myths, legends and facts, IEEE Expert, pp 08-20, February, 1990.

GANE, C., SARSON, T., Análise Estruturada de Sistemas, Livros Técnicos e Científicos Editora, Rio de Janeiro, 1988.

GIARRATANO, J., RILEY, G., Expert Systems: Principles and Programming, PWS Publ. Co., Boston, 1998.

GORDON, G., System Simulation, Prentice-Hall, NJ, 1978.

HARREL,C.R, MOTT,J.R.A, BATEMAN,R.E., BOWDEN,R.G., GOGG,T.J., Simulação: otimizando os sistemas, IMAM, 2000.

HARREL,C.R., BATEMAN,R.E., GOGG,T.J., MOTT,J.R.A., System improvement using simulation, Orem: Promodel Corporation, 1995.

HARRELL, C., TUMAY, K., Simulation made easy: a manager guide, EMP Books, Georgia, 1995.

HILL, T.R., ROBERTS, S.D., A prototype knowledge-based simulation support system, Simulation, V.48, N. 4, pp. 152-161, April, 1987.

IGNIZIO, J.P., A brief introduction to expert systems, Computers & Operations Research, Great Britain, V.17, N.6, pp.523-533, 1990.

KELTON, W.D., SADOWSKI,R.P., SADOWSKI, D.A., Simulation with ARENA, McGraw-Hill, Boston, 1998.

KERCKOFFS, E.J.H., VANSTEENKISTE, G.C., The impact of advanced information processing on simulation – an illustrative review, Simulation, V.46, N.1, pp. 17-26, 1986.

KNUTH,D.E., The art of computer programming, V1, Fundamental Algorithms, Addison-Wesley, Reading, MS, 1969.

KOHRMANN, C., WIENDAHL, H., International used strategies for availability optimization of assembly systems, Institute of Production Systems (IFA), University of Hanover, 1999.

LAW, A.M., KELTON, W.D., Simulation Modeling & Analysis, McGraw-Hill, New York, 1991.

MARTIN, J., LEBEN, J., Strategic Information Planning Methodologies. Prentice Hall, Englewood, New Jersey, 1989.

MARTIN, N., Máquinas de Pensar, Revista Exame, pp. 39-40, Julho, 1990.

MOSER, J.G., Integration of artificial intelligence and simulation in a comprehensive decision suport system, Simulation V.47, N.6, pp. 223-229, December, 1986.

MURRAY,K.J., SHEPPARD,S.V., Knowledge-based simulation model specification, Simulation, V50, N3, pp. 112-119, 1988.

MUTHER, R., Planejamento do Layout: Sistema SLP. Editora Edigard Blucher Ltda. São Paulo, 1978.

NARAYANAN,S, Web-based modelling and simulation. In: Proceedings of the 2000 winter simulation conference, anais, 2000.

NAYLOR, T.H., BALINTFY, J.L., BURDICK, D.S., CHU, K., Técnicas de Simulação em Computadores, Editora Vozes, 1971.

NOTTER, D.R., SANDERS, J.O., DICKERSON, G.E., SMITH, G.M., CARTWRIGHT, T.C., Simulated efficiency of beef production for a Midwest cow-calf-feedlot management system. II. Mature body size. J. Anim. Sci., 49: 83, 1979.

O'KEEFE, R.M., Simulation and expert systems – a taxonomy and some examples, Simulation, V.46, N.1, pp-10-16, January, 1986a.

O'KEEFE, R.M., BELTON, V., BALL,T., Experiences with using expert systems in OR, JORS, Great Britain, V.37, N.7, 1986b.

OLIVÉRIO, J. L., Projeto da Fábrica. Instituto Brasileiro do Livro Científico Ltda. São Paulo, 1985.

OREN, T.I., Artificial intelligence and simulation: from cognitive simulation toward cognizant simulation, Simulation, V.48, N. 4, pp. 129-130, April, 1987.

OREN, T.I., Simulation: as it has been, is, and should be, Simulation, ____, 1987a.

OREN, T.I., ZEIGLER, B.P., Concepts for advanced simulation methodologies, ____, 1979.

PAHL, G.; BEITZ W., Engeneering Design-a systematic approach, Editora Springer, Glasgow, Scotland, 1995.

PAUL,R.J., CHEW,S.T., Simulation modelling using an interactive simulation program generator, JORS, Great Britain, V38, N8, pp. 735-752, 1987.

PEGDEN, C.D., SHANNON, R.E., SADOWSKI, R.P., Introduction to simulation using SIMAN, McGraw-Hill, NY, 1995.

PIDD, M., Computer simulation in management science, Chichester, John Wiley & Sons, 2001.

PMBOK, Project Management Institute, Manuais (2000).

PROMODEL, manuais, Promodel Solutions, Utah, 2004.

PUGH, S. Total design: integrated methods for successful product engineering. Wokinghan: Addison-Wesley, 1991.

REDDY, R., Epistemology of knowledge-based simulation, Simulation, V.48, N.4, pp. 162-166, April, 1987.

REDDY,Y.V.R., FOX, M.S., HUSAIN N., MCROBERTS,M., The knowledge-based simulation system, IEEE SOFTWARE, pp. 26-37, 1986.

RICH, E., Artificial Intelligence, McGraw-Hill, New York, 1983.

ROBERTS, S., Artificial Intelligence, Mini-Micro Systems, pp. 229-233, December, 1983.

ROCHA, J.L., Analysis of the Texas A&M beef cattle simulation model, Term Report, Texas A&M University, College Station, Tx, 1992.

SALIBY, E., OLIVEIRA, G.C., GERSIMUL: um gerador automático de programas de simulação em Turbo Pascal, Pesquisa Operacional, V11, N1, pp. 41-66, Rio de janeiro, 1991.

SANDERS, J.O., CARTWRIGHT, T.C., A general cattle production systems model I: structure of the model, Agricultural Systems, Great Britain, V.4, N.4, pp. 217-227, 1979 a.

SANDERS, J.O., CARTWRIGHT, T.C., A general cattle production systems model II: simulation of animal performance, Agricultural Systems, Great Britain, V.4, N.4, pp. 289-309, 1979 b.

SCHON, D. A. The reflective practitiones: how professionals think in action. United States of America: Basic Books, 1998.

SHANNON, R.E., Knowledge based simulation techniques for manufacturing, International Journal of Production Researches, V.26, N.5, pp. 953-973, 1988.

SHANNON, R.E., Systems Simulation: The art and Science, Prentice-Hall, NJ, 1975.

SHANNON,R.E., MAYER,R., ADELSBERGER,H.H., Expert systems and simulation, Simulation, V.44, N.6, pp. 275-284, 1985.

SHEARN,D.C.S., PASSIM – A Pascal discrete event simulation program generator, Simulation, pp. 31-38, 1990.

SHTUB,A., BARD,J.F.,GLOBERSON,S., Project Management: engineering, technology and implementation, Prentice-Hall, Englewood Cliffs, NJ, 1994.

SHUBIK,M., Simulation of the industry and the firm, Simulation, N.5, pp. 908-919, 1960.

SIMJAVA, www.dcs.ed.ac.uk/home/hase/simjava, acesso em dezembro de 2003.

SMITH, G.M., The TAMU beef cattle production model, Texas A&M University, College Station, Tx, apostila, 1979.

SULE, D. R., Manufacturing Facilities. PWS-KENT Publishing Co, Boston-EUA, 1988.

THIOLLENT, M, Pesquisa-ação nas Organizações. Atlas, 1997.

THREADTEC, Silk, http://www.threadtec.com/, acesso em dezembro de 2003.

TORRES, I., Integração de ferramentas computacionais aplicadas ao projeto e desenvolvimento de arranjo físico de instalações industriais, Dissertação de Mestrado, PPG-DEP/UFSCar, 2001.

TORSUN, I.S., Foundations of intelligent knowledge-based systems, Academic Press, London, 1995.

TRIPPI, R.R., TURBAN, E., Auto-learning approaches for building expert systems, Computers & Operations Research, Great Britain, V.17, N.6, pp. 553-560, 1990.

WATERMAN, D.A., A guide to expert systems, Addison-Wesley, 1985.

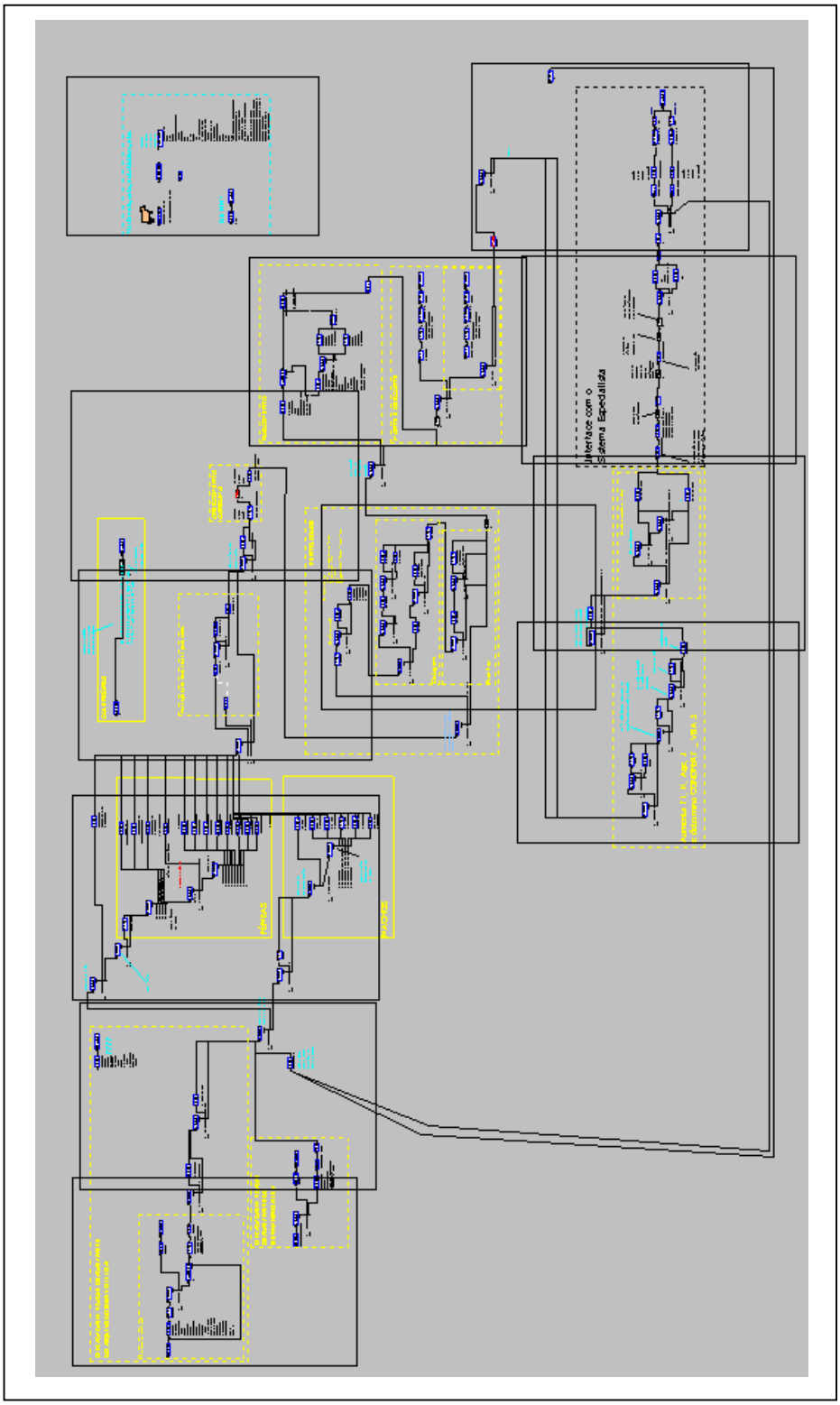
WIDMAN, L.E., LOPARO, K.A., Artificial Intelligence, Simulation and Modelling, Interfaces, V.20, N.2, pp. 48-66, March, 1990.

ZILBOVICIUS, M., Modelos para a produção, produção de modelos. Editora Annablume, São Paulo, 1999.

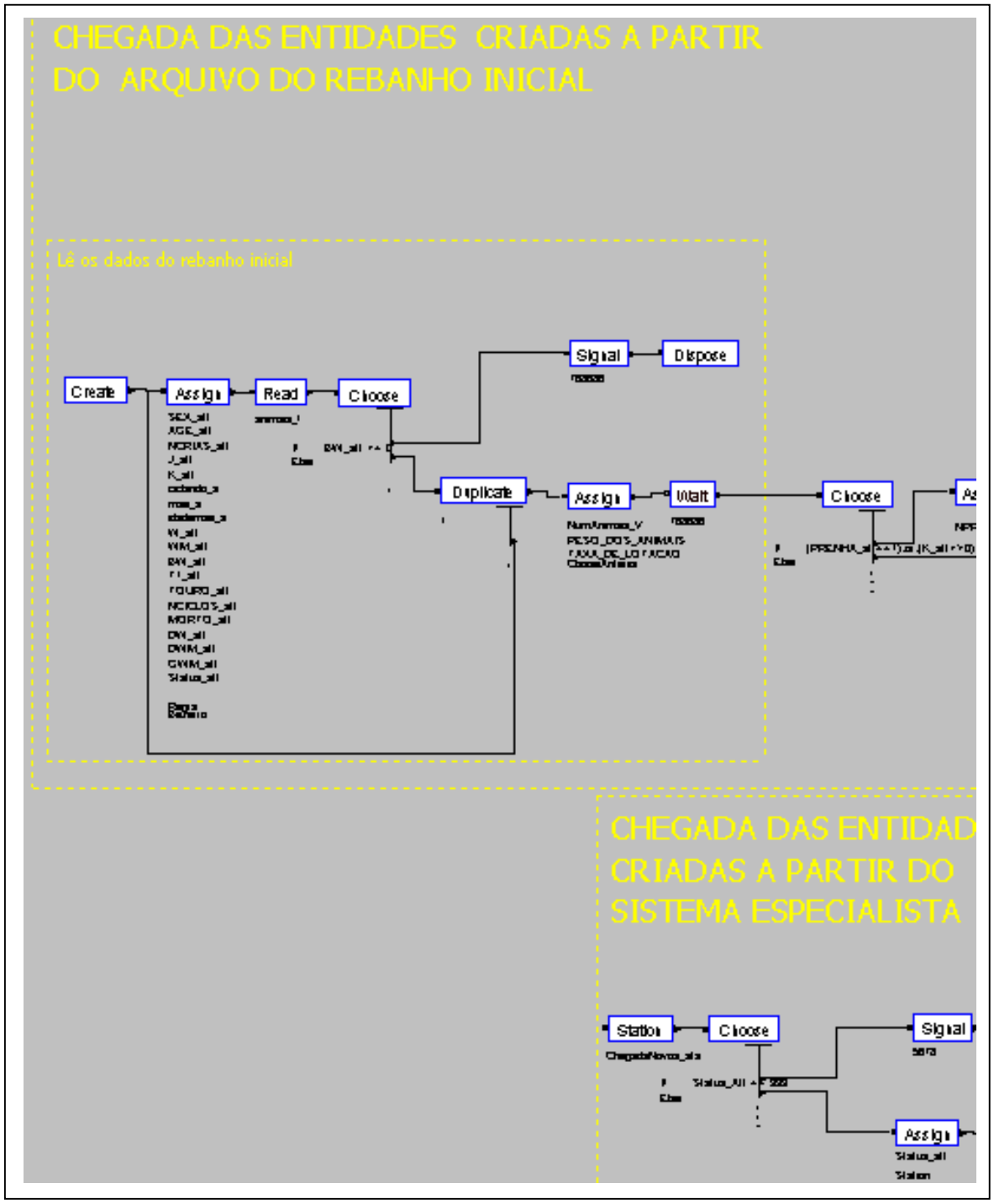
APÊNDICE 01

Lógica do Modelo de Simulação desenvolvida no ARENA (Geral)

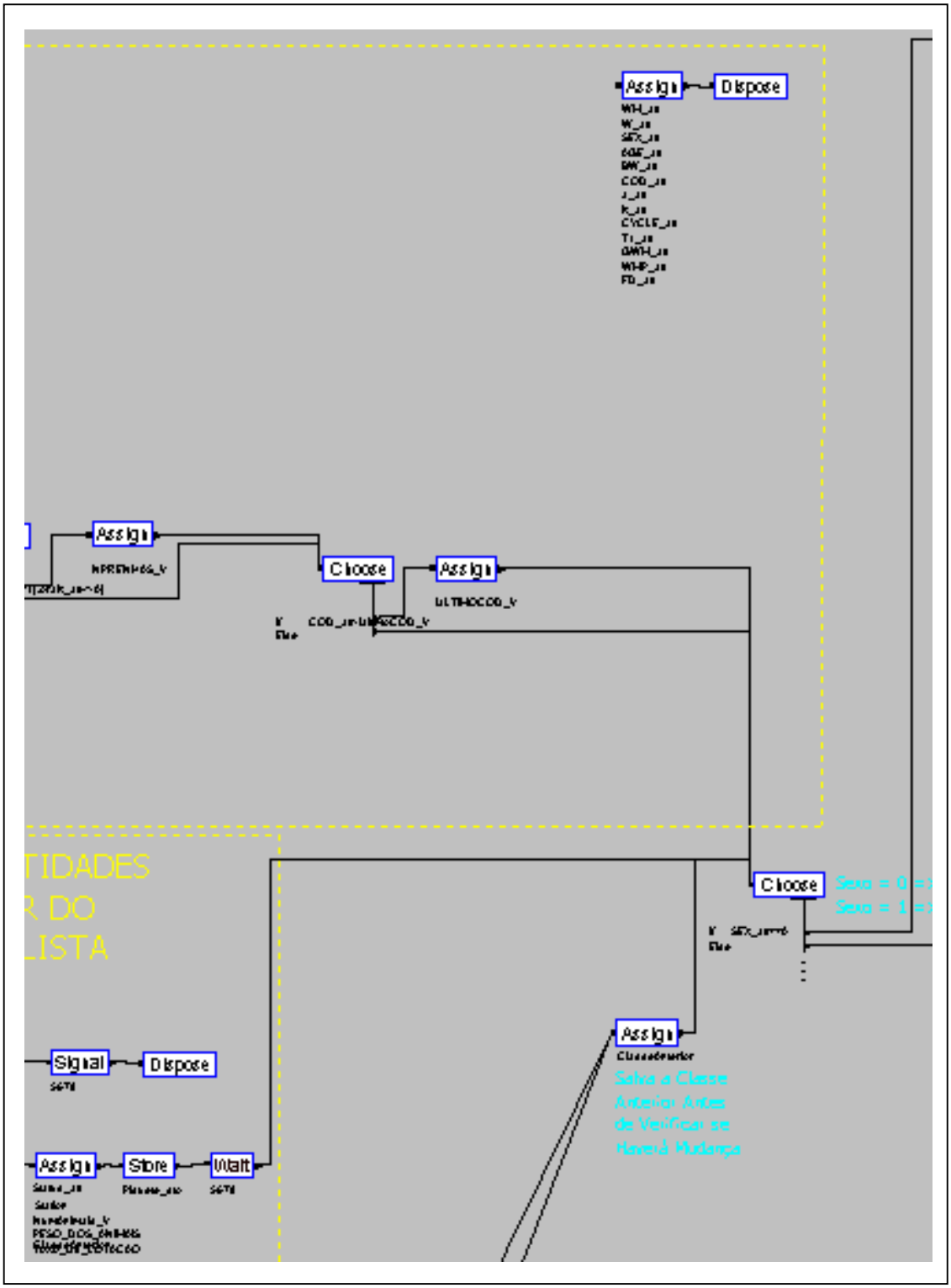
Obs.: as páginas com os detalhes (*zoom* maior) são apresentadas a seguir



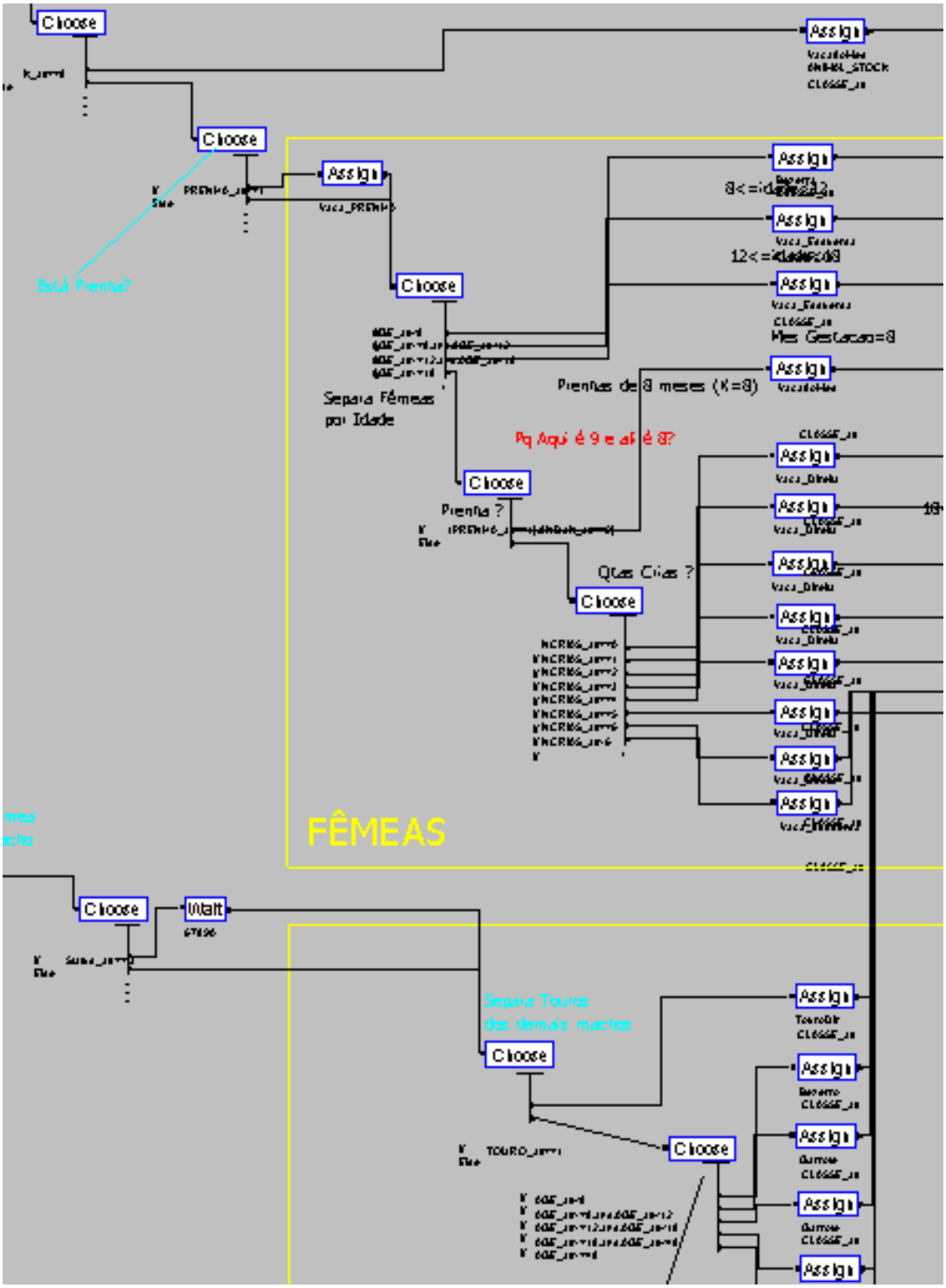
Lógica ARENA (Parte 01)



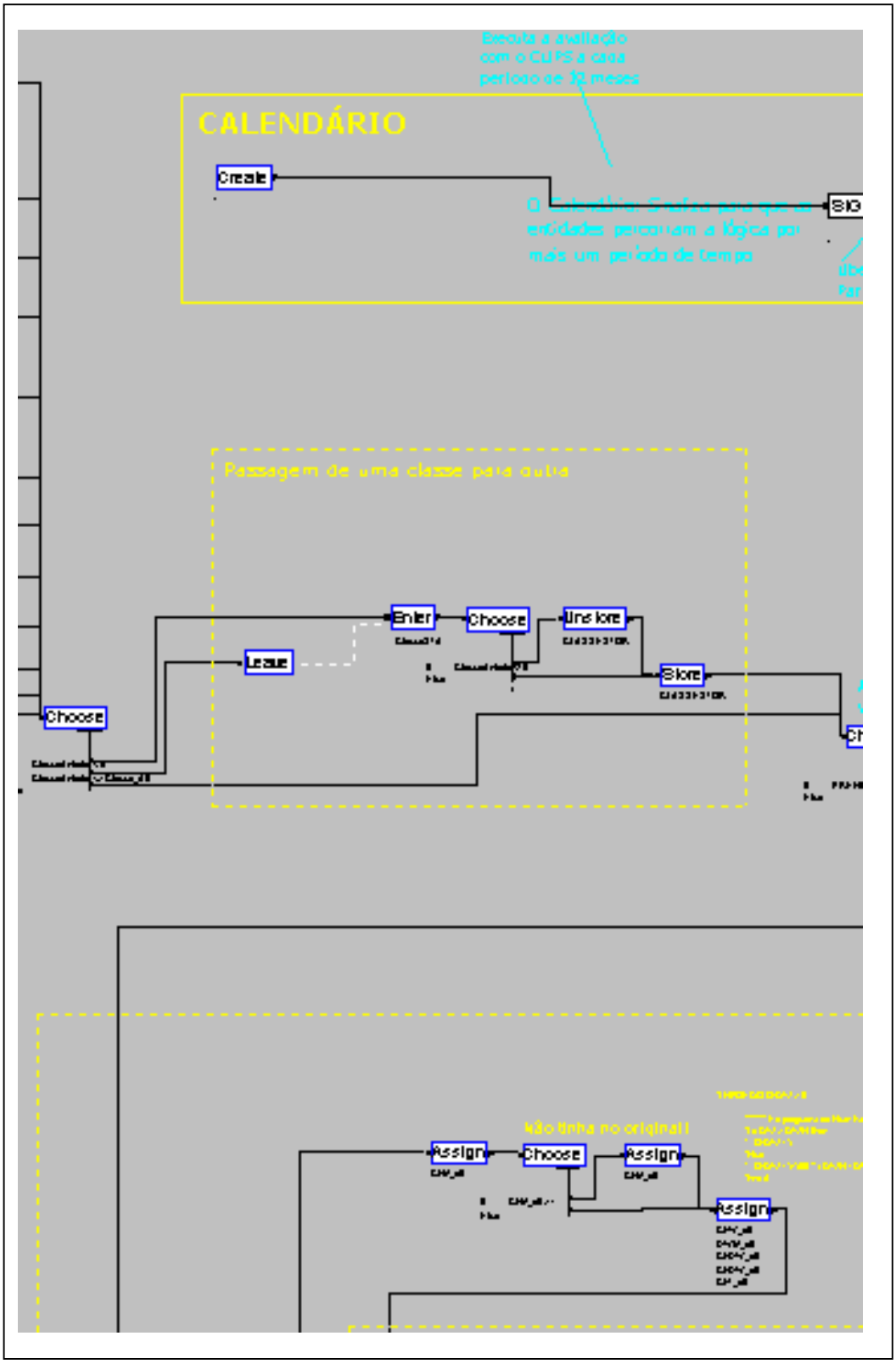
Lógica ARENA (Parte 02)



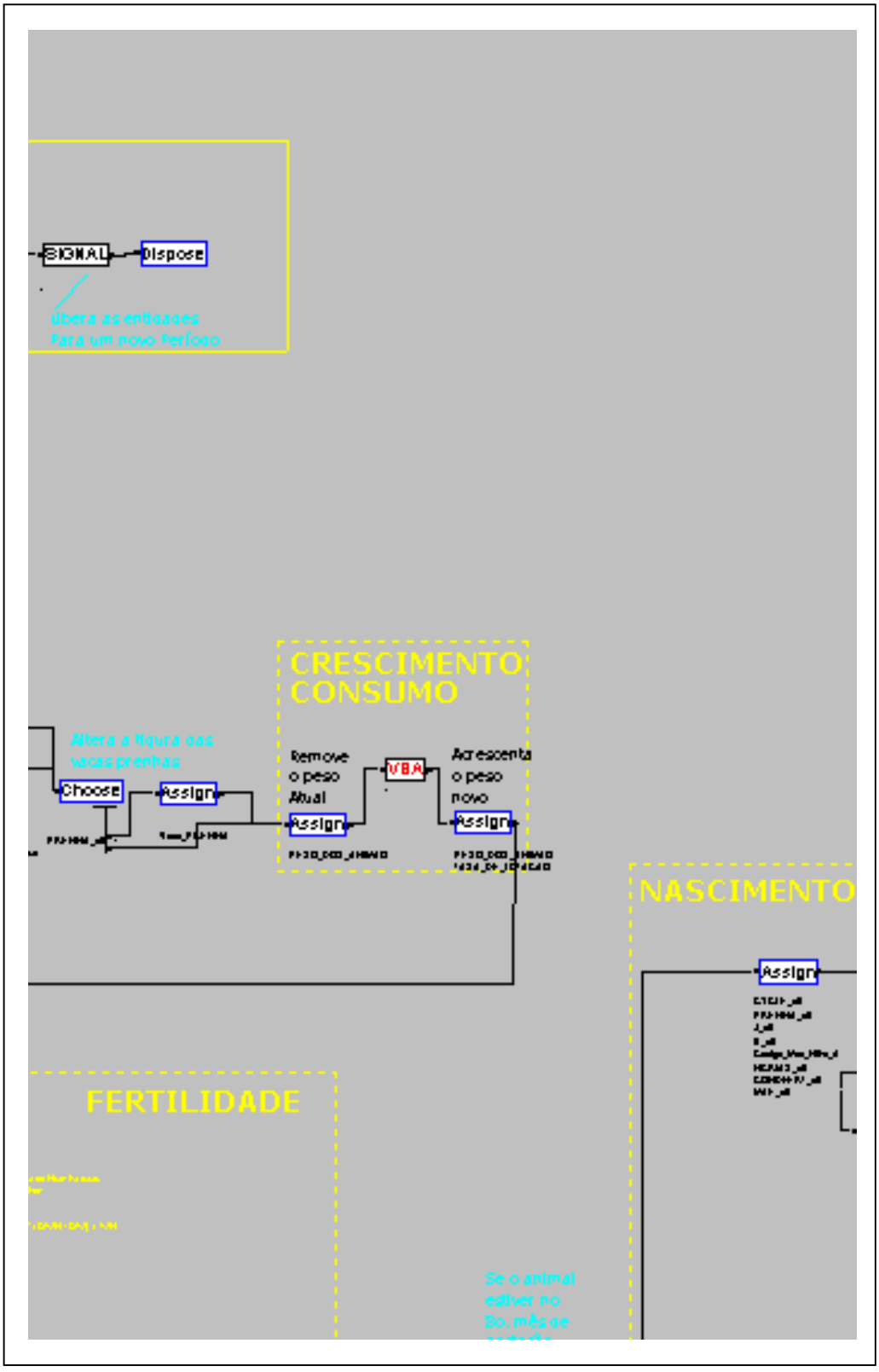
Lógica ARENA (Parte 03)



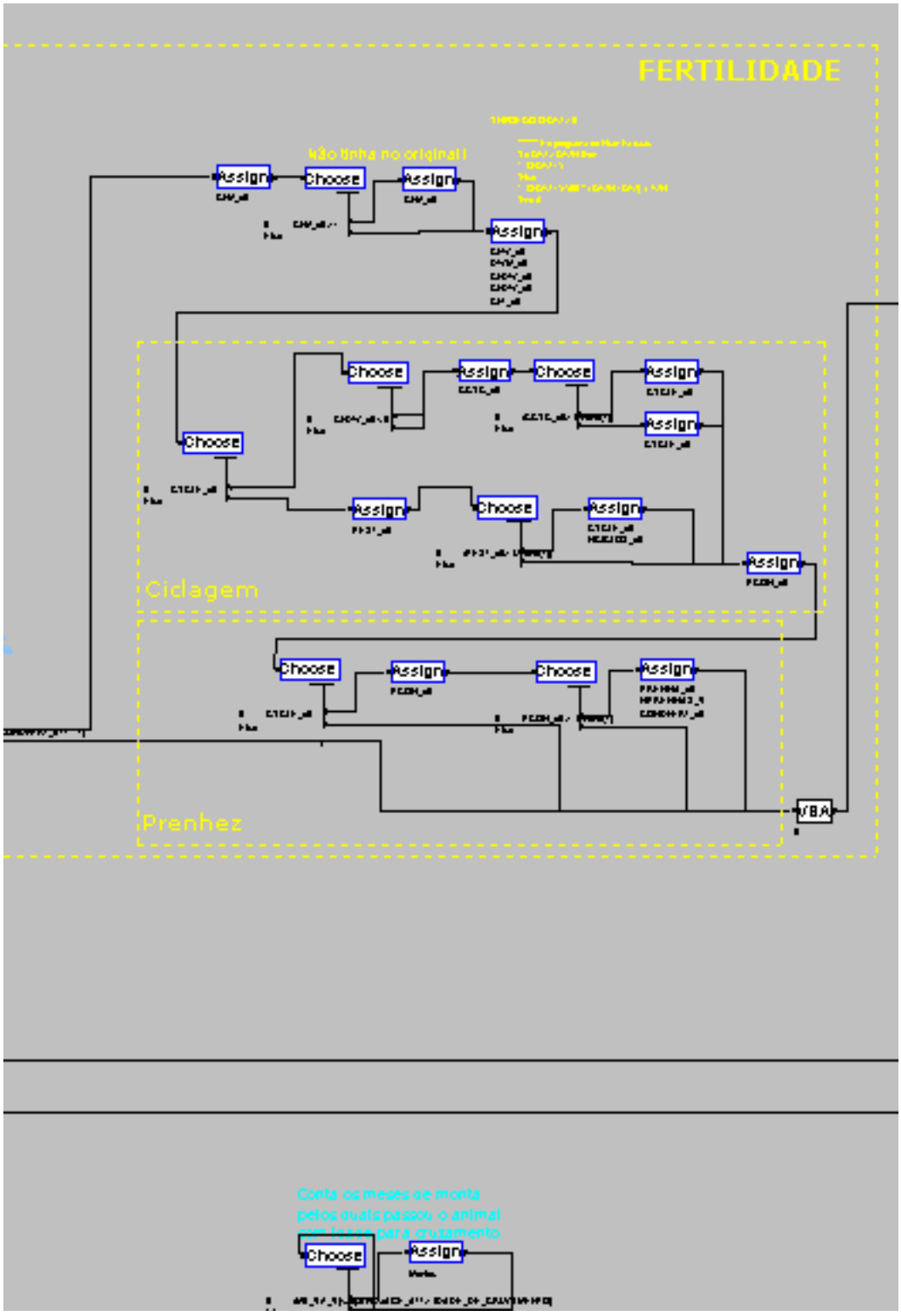
Lógica ARENA (Parte 04)



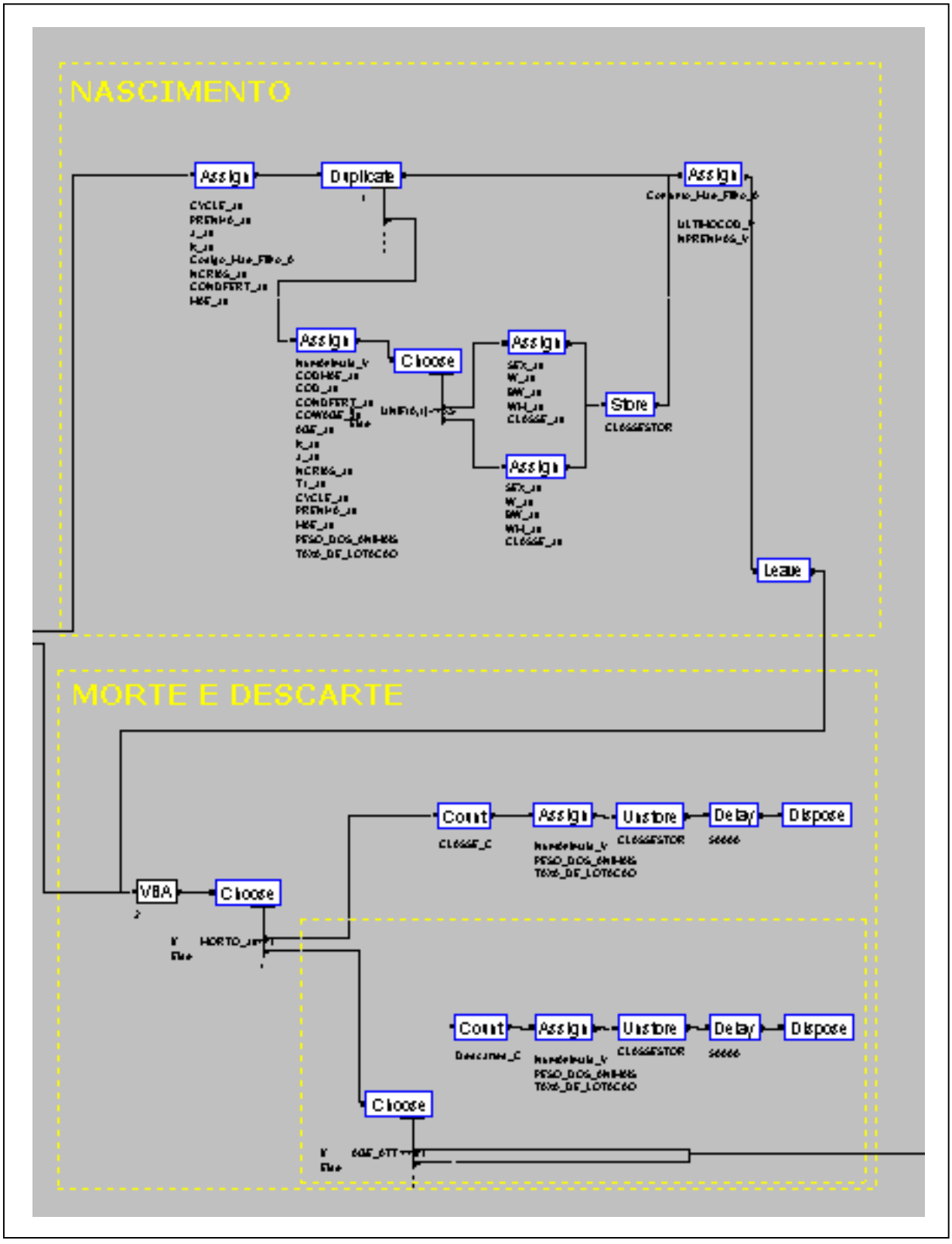
Lógica ARENA (Parte 05)



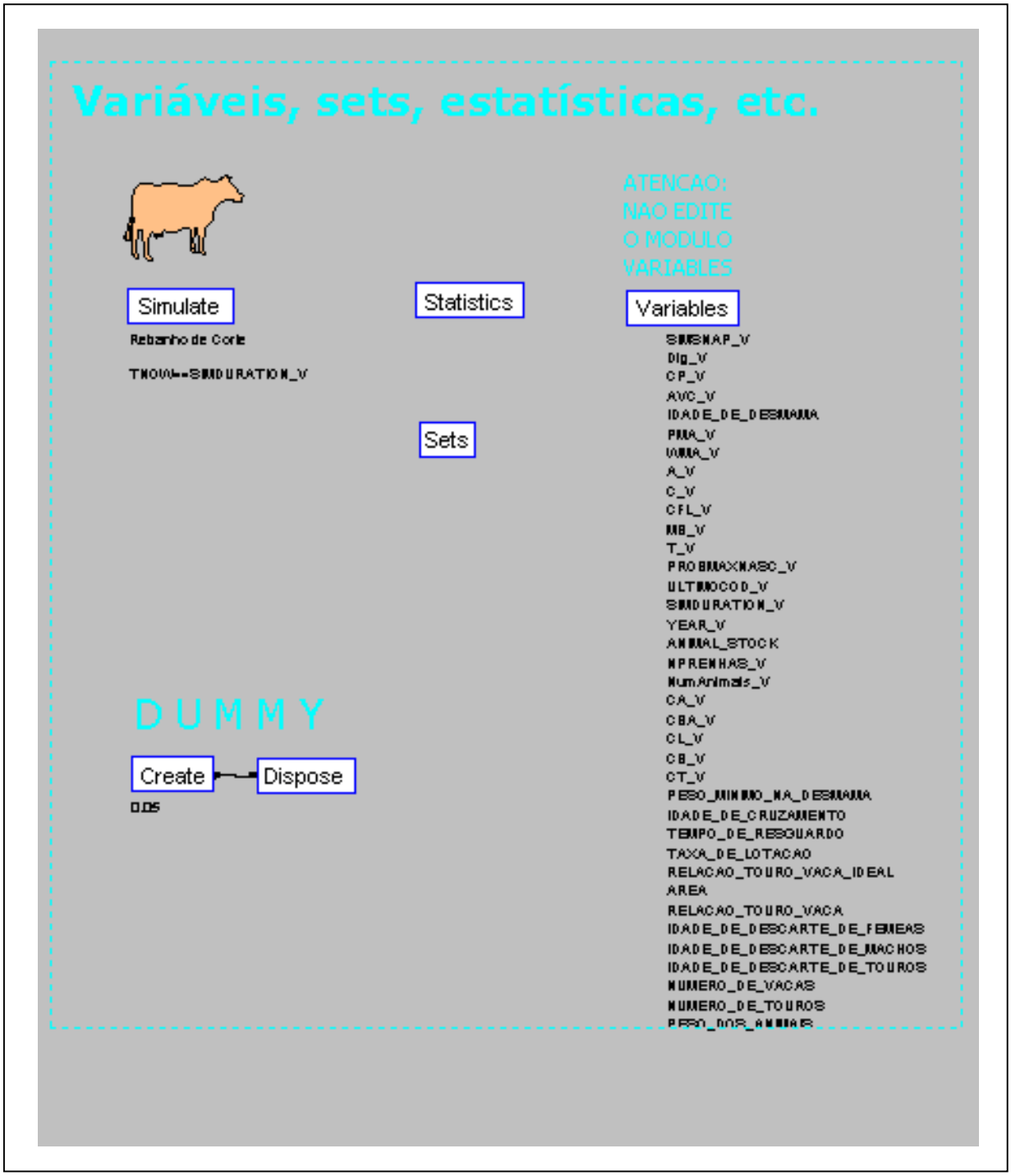
Lógica ARENA (Parte 06)



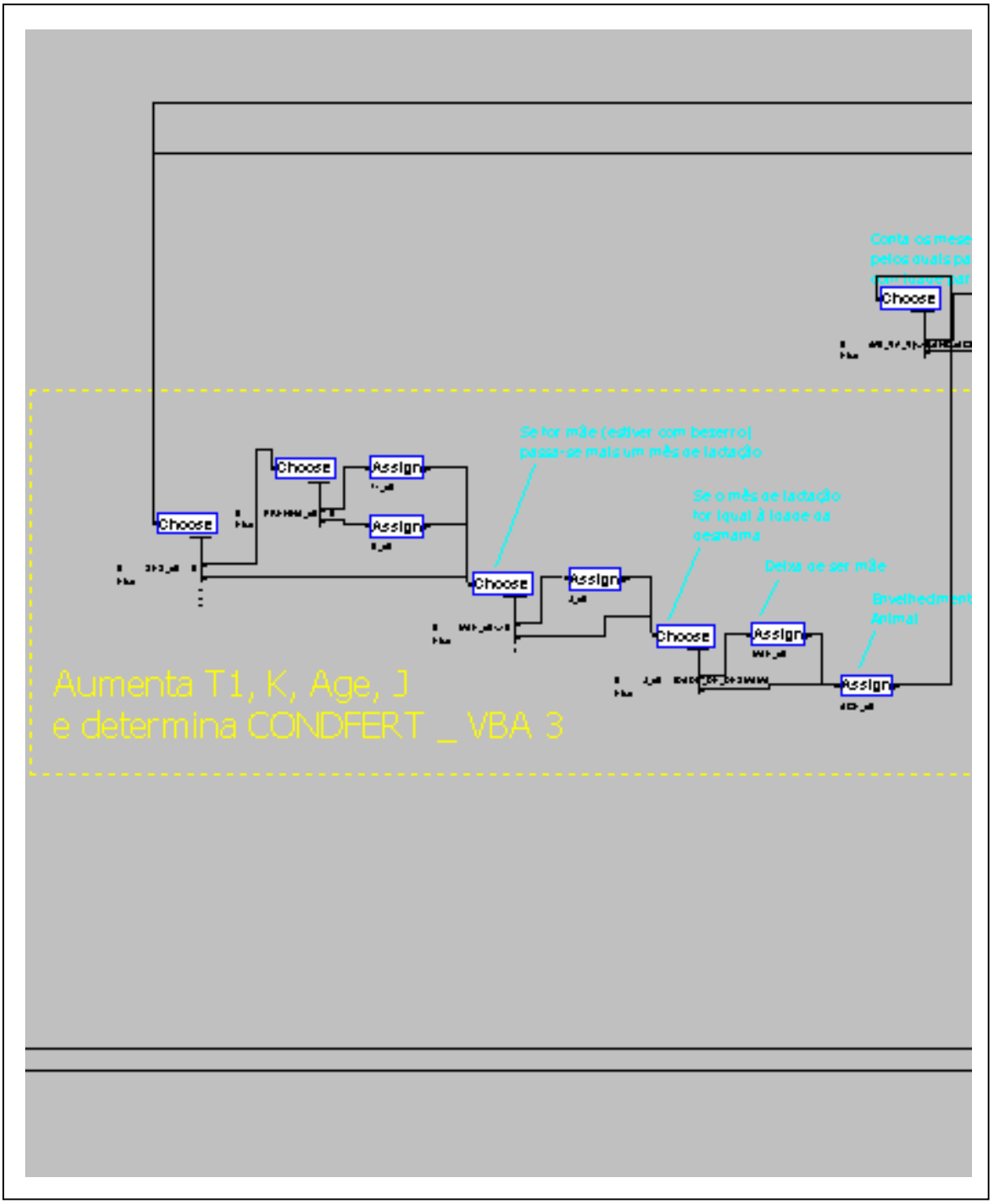
Lógica ARENA (Parte 07)



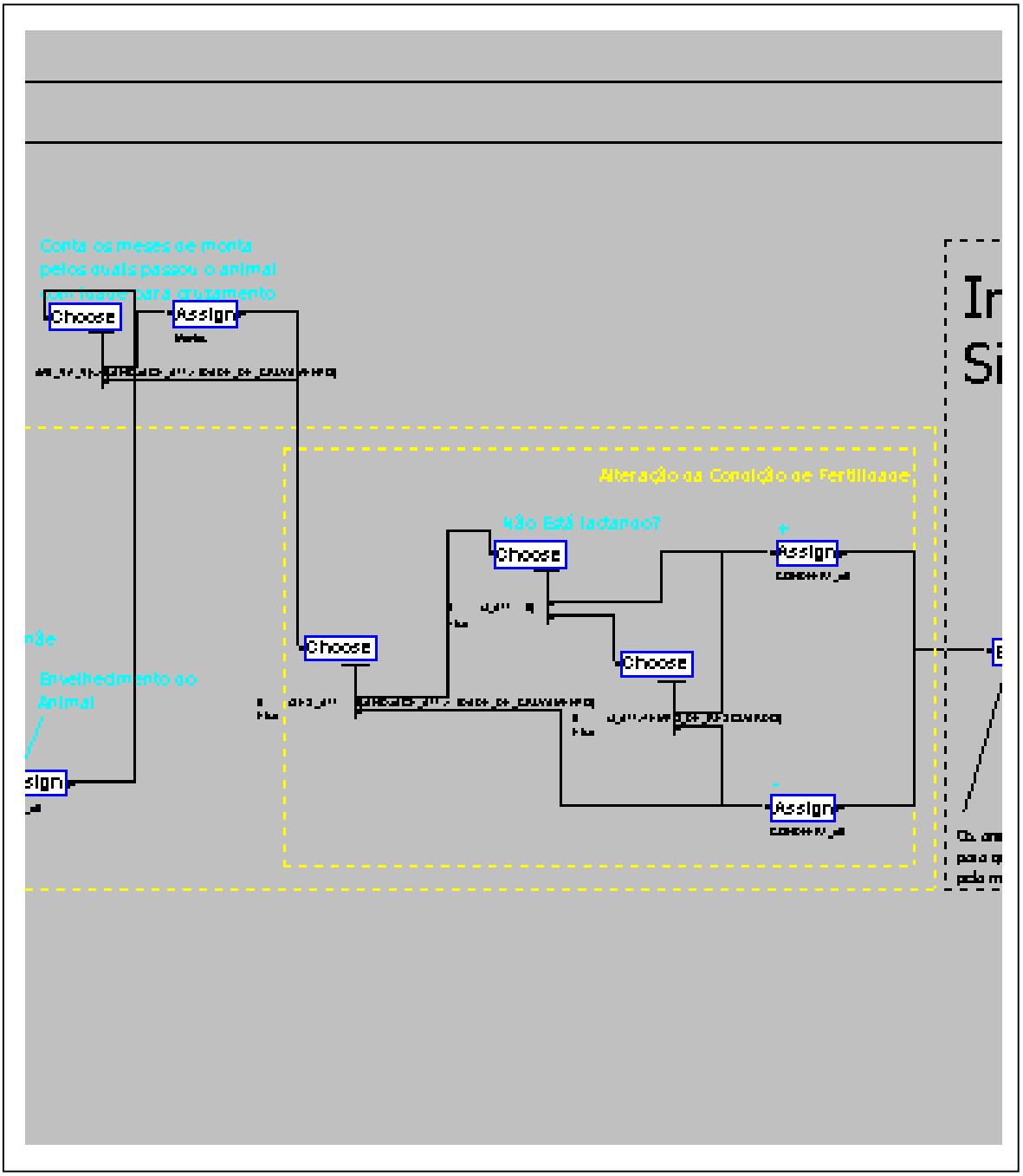
Lógica ARENA (Parte 08)



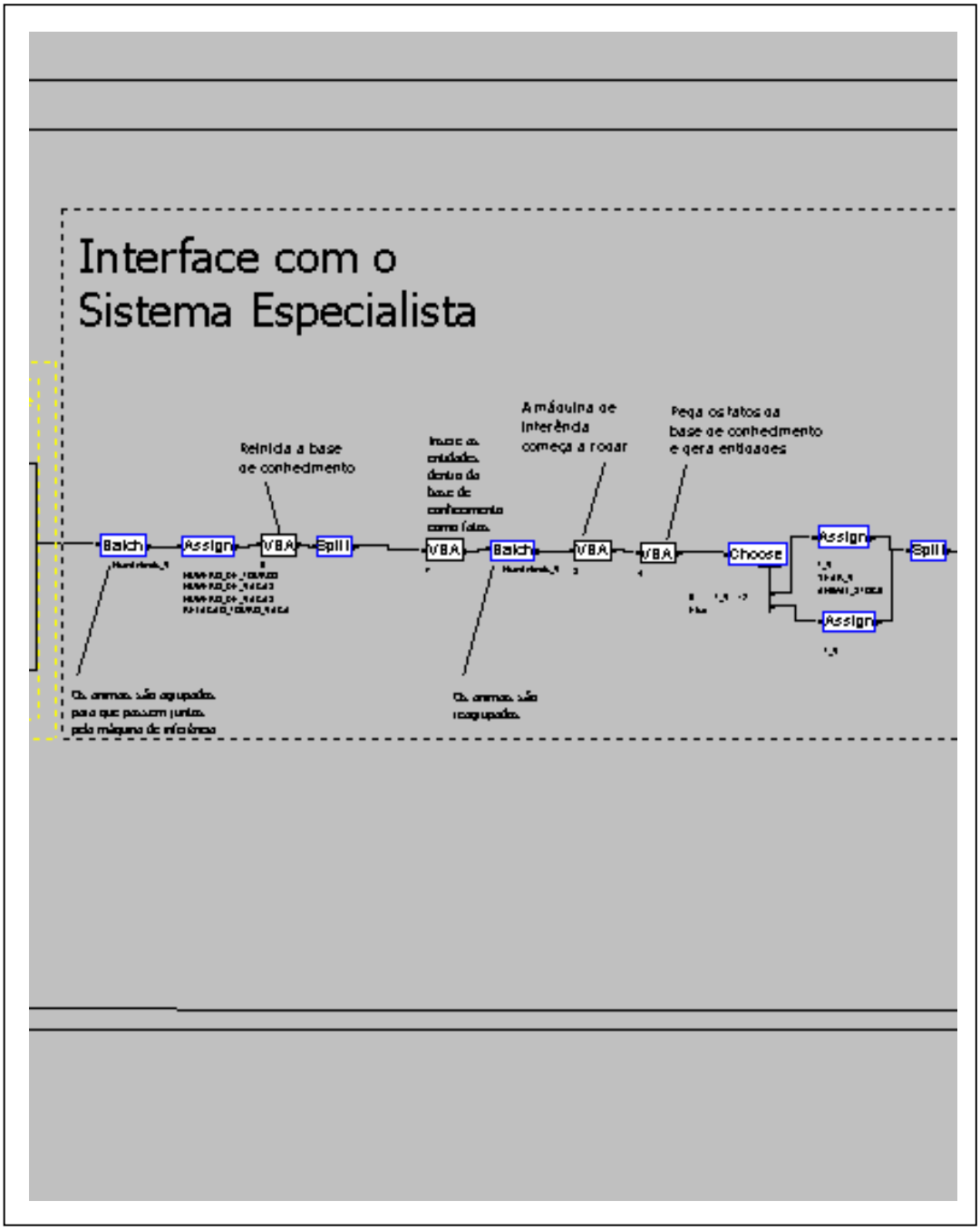
Lógica ARENA (Parte 09)



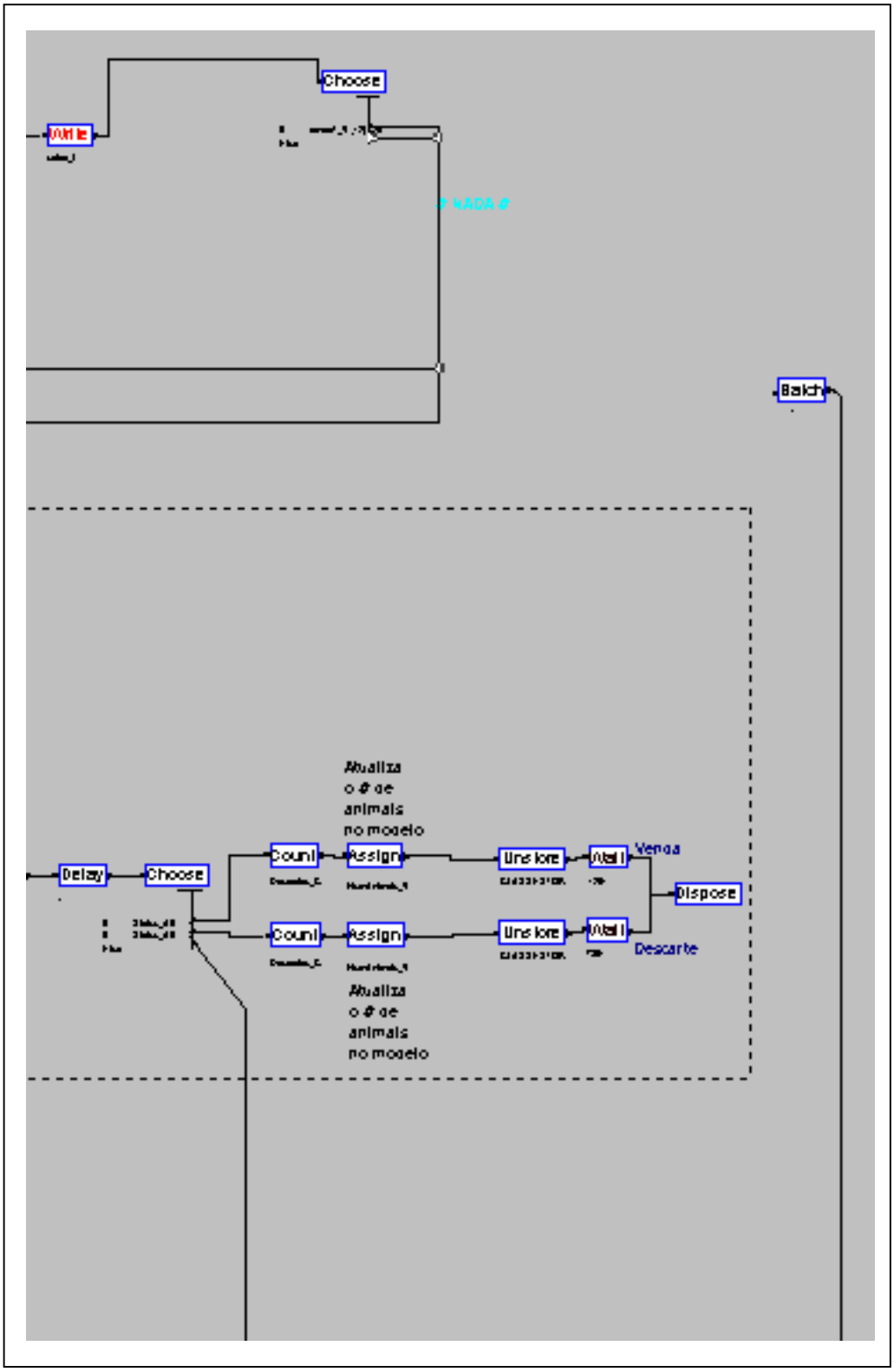
Lógica ARENA (Parte 10)



Lógica ARENA (Parte 11)



Lógica ARENA (Parte 12)



APÊNDICE 02

Lógica do Modelo de Simulação desenvolvida nos módulos VBA – ARENA

```
Option Explicit
'////////////////////////////////////////////////////////////////////////
'// CRESCIMENTO
'//
'//
Public Sub Crescimento()
    Dim cod, sex, Age, AGEY, K, J, T, T1, Cycle, CONDFERT, PRENHA As Integer
    Dim BW, W, WM As Double
    Dim AttNdx As Long
    Dim Q, FG, DW, GWM, DS, FAT, ANF, RL, RP, GR, RG, m, CFX, CFA As
Double
    Dim REQ, R, R1, R2, R3, TM, TN, CN, REQM, ATDN, RXM, PREQ, PE, CEM,
CEG, CFM As Double
    Dim CFT, CFW, CFDW, DWM, CCYC, PEST, PCON, PM, WMP As Double
    Dim NCICLOS, RandN As Integer

    'Set Mdl = ThisDocument.Model
    'Set Sim = Mdl.SIMAN

    '///INTERFACE
    With Sim
        cod = .EntityAttribute(.ActiveEntity, .SymbolNumber("COD_att"))
        sex = .EntityAttribute(.ActiveEntity, .SymbolNumber("SEX_att"))
        BW = .EntityAttribute(.ActiveEntity, .SymbolNumber("BW_att"))
        W = .EntityAttribute(.ActiveEntity, .SymbolNumber("W_att"))
        WM = .EntityAttribute(.ActiveEntity, .SymbolNumber("WM_att"))
        Age = .EntityAttribute(.ActiveEntity, .SymbolNumber("AGE_att"))
        K = .EntityAttribute(.ActiveEntity, .SymbolNumber("K_att"))
```

```
        J = .EntityAttribute(.ActiveEntity, .SymbolNumber("J_att"))
        T = .VariableArrayValue(.SymbolNumber("T_V"))
        'If T > 12 Then
        '    T = 1 'Garante que T varie de 1 a 12
        'End If
        T1 = .EntityAttribute(.ActiveEntity, .SymbolNumber("T1_att"))
        Cycle = .EntityAttribute(.ActiveEntity, .SymbolNumber("CYCLE_att"))
        CONDFERT = .EntityAttribute(.ActiveEntity,
.SymbolNumber("CONDFERT_att"))
        PRENHA = .EntityAttribute(.ActiveEntity, .SymbolNumber("PRENHA_att"))
        GWM = Sim.EntityAttribute(Sim.ActiveEntity, .SymbolNumber("GWM_att"))
        NCICLOS = Sim.EntityAttribute(Sim.ActiveEntity,
.SymbolNumber("NCICLOS_att"))

    End With
    '///INTERFACE

    '----- Inicialização das variáveis
    Q = 0:    FG = 0:   DW = 0:   GWM = 0:  DS = 0:   FAT = 0
    ANF = 0:  RL = 0:   RP = 0:   GR = 0:   RG = 0:   m = 0
    CFX = 0:  CFA = 0:  REQ = 0:  R = 0:    R1 = 0:   R2 = 0
    R3 = 0:   TM = 0:   TN = 0:   CN = 0:   REQM = 0: ATDN = 0
    RXM = 0:  PREQ = 0: PE = 0:   CEM = 0:  CEG = 0:  CFM = 0
    CFT = 0:  CFW = 0:  CFDW = 0: DWM = 0:  CCYC = 0: PEST = 0
    PCON = 0: CCYC = 0: CFM = 0:  CFDW = 0: CFW = 0:  CFT = 0
    'FD = 0
    'EDW = 0
    'EWM = 0
    If Age <= 12 Then
      AGEY = 0
    Else
```

```
    AGEY = Age \ 12
  End If

'----- Manutenção
m = 0.0306 * W ^ 0.75 * (WM / W) ^ 0.5 * (WMA / WM) ^ 0.15
ACTV = 0 ' ACTV=0
m = m + ACTV * W

'----- Se estiver prenha, calcula as necessidades de Prenhez (K em meses)
If K <> 0 Then RP = 0.0000275 * WMA * Exp(0.522 * (K - 1))

'----- Se estiver lactando, calcula as necessidades de Lactação (J em meses)
If J <> 0 Then
    CFA = 1# + 0.01 * (AGEY - 7) - 0.01 * (AGEY - 7) ^ 2
    'CFX = 1# - ((WM - W) / (0.33 * WM)) ^ 3.32
    '----- **** APLICANDO MÓDULO
    CFX = 1# - (Abs((WM - W)) / (0.33 * WM)) ^ 3.32
    PM = CFA * CFX * PMA * Exp(-0.08 * J)
    RL = 0.3 * PM
    m = 1.3 * PM
End If

'----- Calculo do ponto de inflexão
WMP = WMA * (BW + 360 * (0.0026) * 480) / (WMA + 360 * (0.0026) * 480)

'----- Crescimento
If WM <= WMP Then
    GR = (WMP - BW) / 360 '----- GR é taxa linear de crescimento diário (base=360 dias) enquanto o animal não atinge o ponto de inflexão (det. por WMP)
Else
```

```
    GR = 0.0026 * (480 / WMA) * (WMA - WM) '----- A taxa deixa de ser linear depois que W>WMP (além do ponto de inflexão)
  End If
  '----- Se o peso atual for menor do que o tamanho estrutural, corrija a taxa de crescimento
  If (W / WM) < 1 Then
    GR = GR * (1 - (4 - 4 * (W / WM)) ^ 3.5)
    If GR < 0 Then GR = 0 'GR = Max(GR, 0)
  End If
  RG = 0.8 * GR + 5.56 * (0.22 * GR) * (2 * WM + GR - BW) / (WMA - BW)

  '----- Q: Ganho na Condição
  Q = 5.56 * (0.01) * (WM - W)
  If Q < 0 Then Q = 0 'Q = Max(Q, 0)

  REQ = m + RP + RL + RG + Q

  '/////////////// C O N S U M O
  '----- Limite físico
  R2 = (0.0107 * WM) / (1 - DIG(T))                    'Valores de DIG: 0.58,0.58,0.58,0.52,0.52,0.52,0.52,0.52,0.58,0.58,0.58
  If CP(T) < 0.06 Then R2 = R2 * (CP(T) / 0.06) ^ 0.6    'Valores de CP:CP(1) = 0.11,0.11,0.11,0.11,0.09,0.09,0.09,0.09,0.1,0.11,0.11

  'If AGE < IDESM Then 'Se a idade em meses < Idade de desmama então...
  If J <> 0 Then 'Se a vaca estiver amamentando ela ela se alimenta de forma diferente...
    '/// Limite fisiológico
    '----- TM= Consumo de Leite do bezerro
    TM = 0.16 * (1.65) * PM '----- Formula Original: TM = 0.16 * (2) * PM (PM= Produção de Leite)
```

```
        R1 = (0.0225 * WM - TM) / DIG(T)
        If R1 < 0 Then R1 = 0
        If R1 < R2 Then
           R = R1
        Else
           R = R2
        End If
        If R > REQ Then R = REQ / (2 * DIG(T)) + R / 2
        '----- TN = Max(R, 0.01 * WM)
        If R > 0.01 * WM Then
           TN = R
        Else
           TN = 0.01 * WM
        End If
        CN = DIG(T) * TN + TM
     Else
        '/// Limite Fisiológico
        R1 = 0.0225 * WM / DIG(T)
        'R3: Limite da Disponibilidade
        R3 = AVC(T) * (WM / WMA) ^ 0.15 'AVC = Disponib. de mat. seca
        'Valores de AVC:
16.67,16.67,13.89,11.11,10#,9.17,6.28,6.28,6.28,10.89,11.11,13.89

        'R=min(R1,R2,R3)
        If R1 < R2 Then
           R = R1
        Else
           R = R2
        End If
        If R3 < R Then R = R3
```

```
        '----- Vacas têm seu apogeu aos 7/8 anos
        If AGEY > 8 Then
            R = R * (1 - 0.03 * (AGEY - 8)) '(1 - 0.3 * (AGEY - 8)) é linear decrescente
        End If

        If R > REQ Then '----- Se o limite fisiológico for maior do que as necessidades
            TN = REQ / (2 * DIG(T)) + R / 2 '----- consegue ingerir mais do que precisa
        Else
            TN = R
        End If
        CN = DIG(T) * TN
    End If
    '///////////////// F I M  -  C O N S U M O ///////////////
    '----- Se a necessidade é menor do que o consumo de TDN (Ingeriu mais do que precisa)...
    If REQ < CN Then
        FG = (CN - REQ + Q) / 5.56 '----- FG: Mudança em FAT -> O animal está acumulando FAT
        '----- DW: Mudança no peso=Taxa de Crescimento Diária + Mudança em FAT + 0.935 * Necess. de Prenhez
        DW = GR + FG + 0.935 * RP
        GWM = GR '----- GWM: Ganho em WM
    Else
        '------ Reduz a produtividade (Ganho de WM, a Prod. Leite, o Cresc e as necess. Lactação
        GWM = (0.2 + 0.8 * (CN / REQ) ^ 0.6) * GR
        PM = (0.1 + 0.9 * (CN / REQ) ^ 0.5) * PM
        RG = 0.8 * GWM
        RL = 0.3 * PM
        REQ = m + RP + RL + RG
```

```
If REQ < CN Then  '----- Se as necess. São menores do que o que foi ingerido...
    DS = (1 - 0.22 * (2 * WM + GWM - BW) / (WMA - BW)) * GWM
    FG = (CN - REQ) / 5.56
    DW = DS + FG + 0.953 * RP
  Else
    '----- Se emsmo c/ a redução de produtividade as necessidades estão > que o que foi ingerido, mobilize FAT
    FAT = W - WM + 0.22 * WM * (WM - BW) / (WMA - BW) '----- Recalcula FAT: Gordura disponível para mobilização
    ANF = 5.56 * FAT / 30
    If REQ < CN + ANF Then
      DS = (1 - 0.22 * (2 * WM + GWM - BW) / (WMA - BW)) * GWM
      FG = (CN - REQ) / 5.56
      DW = DS + FG + 0.953 * RP
    Else '----- Se mesmo mobilizando FAt não satisfizer, calcule as necessidades mínimas do animal
      REQM = m + RP
      If REQM >= CN + ANF Then
        PM = 0
        GWM = 0
        DW = -1 * FAT / 30 + 0.953 * RP
      Else '----- Se nem as necessidades mínimas forem satisfeitas pelo que se comeu e pela mobilização de FAT...

       '////////////////  3 ////////////////////////
        'J,GWM,ANF,CN,REQM,DW,FAT,RXM,
        'PREQ,PE,ATDN,CEM,PM,DS,BW,WM,WMA

        '----- Lactando?
        If J <> 0 Then
```

```
                REQ = m + RP + 0.25 * RL
                ATDN = CN + ANF - REQ

                If ATDN <= 0 Then
                   PM = (ANF + CN - REQM) / 0.3
                   GWM = 0
                   DW = -1 * FAT / 30 + 0.953 * RP

                Else
                  '----- Divisão entre Leite e Crescimento
                   RXM = 0.75 * RL
                   PREQ = RXM + 0.8 * GWM
                   PE = ATDN / PREQ
                   CEM = ((2 - 0.5 * PE) / 1.5) * (RXM / PREQ) * ATDN
                   CEG = ATDN - CEM
                   PM = (CEM + 0.25 * RL) / 0.3
                   GWM = CEG / 0.8
                   DS = (1 - 0.22 * (2 * WM + GWM - BW) / (WMA - BW)) * GWM
                   DW = DS - FAT / 30 + 0.953 * RP
                End If
             Else
                '----- Não Lactando...
                GWM = (ANF + CN - REQM) / 0.8
                DS = (1 - 0.22 * (2 * WM + GWM - BW) / (WMA - BW)) * GWM
                DW = DS - FAT / 30 + 0.953 * RP
             End If
             '////////////////// 3 ////////////////////////////
       End If 'REQM < CN + ANF
    End If 'REQ < CN + ANF
  End If 'REQ < CN - Reduz a Produtividade
End If
```

```
'----- Mudança no Peso e Ganho de Tamanho Estrutural
W = W + DW
WM = WM + GWM
'***************
'*****************
'*******************
'// TRACING: ATUALIZAÇÃO DO ARQUIVO DE ACOMPANHAMENTO DOS ANIMAIS
'// Info is printed for each year
If (TRACE_CRESCIMENTO) And (T Mod 12 = 0) Then
    Dim v(50) As String

    v(1) = Str(T): v(2) = Str(DIG(T)): v(3) = Str(CP(T)): v(4) = Str(AVC(T)): v(5) = Str(WMA)
    v(6) = Str(PMA): v(7) = Str(WMP): v(8) = Str(cod): v(9) = Str(sex): v(10) = Str(Age)
    v(11) = Str(BW): v(12) = Str(W): v(13) = Str(WM): v(14) = Str(J): v(15) = Str(K)
    v(16) = Str(PM): v(17) = Str(Q): v(18) = Str(FG): v(19) = Str(DW): v(20) = Str(GWM)
    v(21) = Str(DS): v(22) = Str(FAT): v(23) = Str(ANF): v(24) = Str(RL): v(25) = Str(RP)
    v(26) = Str(GR): v(27) = Str(RG): v(28) = Str(m): v(29) = Str(CFX): v(30) = Str(CFA)
    v(31) = Str(REQ): v(32) = Str(R): v(33) = Str(R1): v(34) = Str(R2): v(35) = Str(R3)
    v(36) = Str(TM): v(37) = Str(TN): v(38) = Str(CN): v(39) = Str(REQM): v(40) = Str(ATDN)
    v(41) = Str(RXM): v(42) = Str(PREQ): v(43) = Str(PE): v(44) = Str(CEM): v(45) = Str(CEG)
```

```
    Dim i, difLEN As Integer
    Dim strX, str2insert, strLN As String

    '// Prepares a line with info about the animal
    For J = 1 To 45 'nCOLS                    'Montagem da linha de valores
        strX = Format(v(J), "0.0000000") 'Str(v(J))
        str2insert = ""
        difLEN = 16 - Len(strX) - 1

        If difLEN < 0 Then
            strX = Mid(strX, 1, 16 - 1)
            difLEN = 0
        End If

        For i = 1 To difLEN
            str2insert = str2insert + " "
        Next i
        '--str2insert = str2insert + Space(difLEN)

        strLN = strLN + str2insert + strX + "; "
    Next J
    '--Print #TraceCrescFile, T, DIG(T), CP(T), AVC(T), WMA, PMA, WMP, Cod,
Sex, Age, BW, W, WM, J, K, PM, Q, FG, DW, GWM, DS, FAT, ANF, RL, RP, GR,
RG, m, CFX, CFA, REQ, R, R1, R2, R3, TM, TN, CN, REQM, ATDN, RXM, PREQ,
PE, CEM, CEG
    'strLN = RemoveCommas(strLN)
    '--MsgBox TraceCrescFileName
    Print #TraceCrescFile, strLN
  End If

  '////////////////////////////// INTERFACE //////////////////////
```

```
'// Return the values into arena

    With Sim
        .EntityAttribute(.ActiveEntity, .SymbolNumber("T1_att")) = T1
        .EntityAttribute(.ActiveEntity, .SymbolNumber("CYCLE_att")) = Cycle
        .EntityAttribute(.ActiveEntity,      .SymbolNumber("CONDFERT_att"))      =
CONDFERT
        .EntityAttribute(.ActiveEntity, .SymbolNumber("K_att")) = K
        .EntityAttribute(.ActiveEntity, .SymbolNumber("PRENHA_att")) = PRENHA
        .EntityAttribute(.ActiveEntity, .SymbolNumber("BW_att")) = BW
        .EntityAttribute(.ActiveEntity, .SymbolNumber("W_att")) = W
        .EntityAttribute(.ActiveEntity, .SymbolNumber("WM_att")) = WM
        .EntityAttribute(.ActiveEntity, .SymbolNumber("AGE_att")) = Age
        '.EntityAttribute(.ActiveEntity, .SymbolNumber("Q_att")) = Q
        .EntityAttribute(.ActiveEntity, .SymbolNumber("J_att")) = J
        '.EntityAttribute(.ActiveEntity, .SymbolNumber("FG_att")) = FG
        .EntityAttribute(.ActiveEntity, .SymbolNumber("DW_att")) = DW
        .EntityAttribute(.ActiveEntity, .SymbolNumber("GWM_att")) = GWM
        '.EntityAttribute(.ActiveEntity,.SymbolNumber("DS_att")) = DS
        '.EntityAttribute(.ActiveEntity, .SymbolNumber("FAT_att")) = FAT
        '.EntityAttribute(.ActiveEntity, .SymbolNumber("ANF_att")) = ANF
        '.EntityAttribute(.ActiveEntity, .SymbolNumber("RL_att")) = RL
        '.EntityAttribute(.ActiveEntity, .SymbolNumber("RP_att")) = RP
        '.EntityAttribute(.ActiveEntity, .SymbolNumber("GR_att")) = GR
        '.EntityAttribute(.ActiveEntity, .SymbolNumber("RG_att")) = RG
        '.EntityAttribute(.ActiveEntity, .SymbolNumber("M_att")) = M
        '.EntityAttribute(.ActiveEntity, .SymbolNumber("CFX_att")) = CFX
        '.EntityAttribute(.ActiveEntity, .SymbolNumber("CFA_att")) = CFA
        '.EntityAttribute(.ActiveEntity, .SymbolNumber("REQ_att")) = REQ
        '.EntityAttribute(.ActiveEntity, .SymbolNumber("R1_att")) = R1
        '.EntityAttribute(.ActiveEntity, .SymbolNumber("R2_att")) = R2
```

```
          '.EntityAttribute(.ActiveEntity, .SymbolNumber("R3_att")) = R3
          '.EntityAttribute(.ActiveEntity, .SymbolNumber("TM_att")) = TM
         '.EntityAttribute(.ActiveEntity,.SymbolNumber("TN_att")) = TN
         '.EntityAttribute(.ActiveEntity,.SymbolNumber("CN_att")) = CN
         '.EntityAttribute(.ActiveEntity, .SymbolNumber("REQM_att")) = REQM
         '.EntityAttribute(.ActiveEntity, .SymbolNumber("ATDN_att")) = ATDN
         '.EntityAttribute(.ActiveEntity, .SymbolNumber("RXM_att")) = RXM
         '.EntityAttribute(.ActiveEntity, .SymbolNumber("PREQ_att")) = PREQ
         '.EntityAttribute(.ActiveEntity, .SymbolNumber("CEM_att")) = CEM
         '.EntityAttribute(.ActiveEntity, .SymbolNumber("CEG_att")) = CEG
         '.EntityAttribute(.ActiveEntity, .SymbolNumber("CFM_att")) = CFM
         '.EntityAttribute(.ActiveEntity, .SymbolNumber("PE_att")) = PE
         '.EntityAttribute(.ActiveEntity, .SymbolNumber("CFT_att")) = CFT
         '.EntityAttribute(.ActiveEntity, .SymbolNumber("CFW_att")) = CFW
         '.EntityAttribute(.ActiveEntity, .SymbolNumber("CFDW_att")) = CFDW
         .EntityAttribute(.ActiveEntity, .SymbolNumber("DWM_att")) = DWM
         '.EntityAttribute(.ActiveEntity, .SymbolNumber("CCYC_att")) = CCYC
         '.EntityAttribute(.ActiveEntity, .SymbolNumber("PEST_att")) = PEST
         '.EntityAttribute(.ActiveEntity, .SymbolNumber("PCON_att")) = PCON
         '.EntityAttribute(.ActiveEntity, .SymbolNumber("PM_att")) = PM
         .EntityAttribute(.ActiveEntity, .SymbolNumber("WMP_att")) = WMP
     End With
  "///--------------INTERFACE
 ' '----- TRACING: ATUALIZAÇÃO DO ARQUIVO DE ACOMPANHAMENTO DO FD
 '  If TRACE_CALC_FERT Then
 '     Print #TraceFERTFile, T, MB(T), WMA, COD, SEX, AGE, BW, W, WM, J, K, CFT, CFM, CFW, CFDW, DWM, CCYC, CYCLE, PEST, PCON, CONDFERT, PRENHA
 ' End If
 '
```

```
  '
  ' End If 'checagem se o animal pode entrar em fertilidade
  '
  '
  '
  '
  ' '////////////////// F I M  -  F E R T I L I D A D E ////////////// /
  '

End Sub


Private Sub ModelLogic_RunBegin()

    Dim NL As String
    NL = Chr$(13) + Chr$(10)        '// New line

    Dim str2insert, head As String  '// For trace file header mounting
    Dim difLEN, J, i As Integer
    Dim h(50) As String

    ModelPath = Arena.ActiveModel.Path '// Model directory

    '// Adjust values of global variables. These values will be used to populate
    '// the form variables
    Call SetupAnimalVars  '// Setup the values for the animal-related
                          '// variables: CA , CL, CBA, CB, CT, MB
    Call SetupForage      '// Setup grass variables
    '--AttNdx = 0

    ABORT_SIMULATION = False
```

```
frmUm.Show '// Show the startup form for user input

If Not ABORT_SIMULATION Then

   '////////////////////////////////////////////////////////
   '// Open the files required for tracing
   '//TRACING FILE: Animal Monthly Monitoring
   '--TraceCrescFileName = ModelPath + "FORFAIT_TRACE_CRESC.TXT"
   'TraceCrescFileName = frmUm.txtTraceGrow.Text
   If TRACE_CRESCIMENTO Then
      If Len(TraceCrescFileName) > 0 Then
         TraceCrescFile = FreeFile

         '// TRACING FILE: Growing
         Open TraceCrescFileName For Output As TraceCrescFile
         '// File header:
         'Print #TraceCrescFile, "", "T", "DIG(T)", "CP(T)", "AVC(T)", "WMA", _
                        "PMA", "WMP", "COD", "SEX", _
                        "AGE", "BW", "W", "WM", "J", "K", "PM", "Q", _
                        "FG", "DW", "GWM", "DS", "FAT", "ANF", "RL", _
                        "RP", "GR", "RG", "M", "CFX", "CFA", "REQ", _
                        "R", "R1", "R2", "R3", "TM", "TN", "CN", _
                        "REQM", "ATDN", "RXM", "PREQ", "PE", "CEM", _
                        "CEG", "CFM", "CFT", "CFW", "CFDW", "DWM", _
                        "CCYC", "PEST", "PCON"

         h(1) = "DB_MES(T)"
         h(2) = "DB_DigestPasto(DIG)"
         h(3) = "DB_ProtBrPasto(CP)"
         h(4) = "DB_DispMxPasto(AVC)"
         h(5) = "DB_TamEstrMatu(WMA)"
```

h(6) = "DB_ProdPotLeit(PMA)"

h(7) = "DB_TamEstrInfl(WMP)"

h(8) = "DB_CodigoAnima(COD)"

h(9) = "DB_Sexo_Animal(SEX)"

h(10) = "DB_Idade_Meses(AGE)"

h(11) = "DB_Peso_Nascim(BW)"

h(12) = "DB_Peso_Atual(W)"

h(13) = "DB_Tam_Estrutu(WM)"

h(14) = "DB_Mes_Lactaca(J)"

h(15) = "DB_Mes_Gestaca(K)"

h(16) = "DB_Produ_Leite(PM)"

h(17) = "DB_Ganho_Condi(Q)"

h(18) = "DB_Mudanca_FAT(FG)"

h(19) = "DB_MudancaPeso(DW)"

h(20) = "DB_GanhoTamEst(GWM)"

h(21) = "DB_MudanTamEst(DS)"

h(22) = "DB_GrdDispMobl(FAT)"

h(23) = "DB_VlrEnergFAT(ANF)"

h(24) = "DB_NecLactacao(RL)"

h(25) = "DB_Nec_Prenhez(RP)"

h(26) = "DB_TaxaCrescim(GR)"

h(27) = "DB_NeceCrescim(RG)"

h(28) = "DB_Manututenca(M)"

h(29) = "DB_FC_LactCond(CFX)"

h(30) = "DB_FC_LactIdad(CFA)"

h(31) = "DB_NecesAnimal(REQ)"

h(32) = "DB_R"

h(33) = "DB_R1"

h(34) = "DB_R2"

h(35) = "DB_R3"

h(36) = "DB_ConsumLeite(TM)"

```
h(37) = "DB_Consumo_TDN(TN)"
h(38) = "DB_CN"
h(39) = "DB_Nec_Mininas(REQM)"
h(40) = "DB_ATDN"
h(41) = "DB_RXM"
h(42) = "DB_PREQ"
h(43) = "DB_PE"
h(44) = "DB_CEM"
h(45) = "DB_CEG"

'-- tabulation = 16 'largura da coluna onde será inserido o valor incluindo o separador "|"

'// Setup a header for the growing trace file and print it to the file
For J = 1 To 45
   str2insert = ""
   difLEN = 16 - Len(h(J)) - 1
   If difLEN < 0 Then
      h(J) = Mid(h(J), 1, 16 - 1)
      difLEN = 0
   End If
   For i = 1 To difLEN
      str2insert = str2insert + " "
   Next i
   head = head + str2insert + h(J) + "; "
Next J
'--For J = 1 To Len(head)
'--    sepLN = sepLN + "*"
'--Next J
'--head = head + NL '--+ sepLN
Print #TraceCrescFile, head
```

```
            'Print #TraceCrescFile, "", "Mês", "Digest_Pasto", "Prot_Bruta_Pasto",
"Disp_Máx_Pasto(kg_Dm/hd/day)", "Tam_Estrut_Mat", "Prod_Pot_Leite",
"Tam_Estrut_Inflexão", "Código", "Sexo", "Idade(Meses)", "Peso_Nasc(kg)",
"Peso_Atual(kg)", "Tam_Estrutural(kg)", "Mês_Lactação", "Mês_Gestação",
"Prod_Leite(kg)", "Ganho_Condição", "Mudança_FAT",_
            '                                    "Mudança_Peso", "Ganho_WM",
"Mudança_WM(exc.p/gord.não-essenc.)", "Gord.Disp.p/Mobil",
"Valor_Energ_FAT", "Nec_Lactação(kgATDN)", "Nec_Prenhez(kgATDN)",
"Tx_Cresc", "Nec.p/Cresc", "Manut", "Fat.Corr.Lact.Condição",
"Fat.Corr.Lact.Idade", "Nec.Animal(kgATDN)", _
            '                    "R", "R1", "R2", "R3", "Cons.Leite", "Cons.TDN", "CN
", "Nec.Míminas", "ATDN", "RXM", "PREQ", "PE", "CEM", "CEG"
            'T, DIG(T), CP(T), AVC(T), WMA,
PMA, WMP, COD, SEX, AGE, BW, W,
WM, J, K, PM, Q, FG, DW,
GWM, DS, FAT, ANF, RL,
RP, GR, RG, M, CFX, CFA,
REQ, R, R1, R2, R3, TM, TN, CN, REQM,
ATDN, RXM, PREQ,_
            'PE, CEM, CEG,


        End If
    End If
    '// Arquivo para Tracing do Cálculo do FD
    '--TraceFDFileName = ModelPath + "FORFAIT_TRACE_FD.TXT"
    'TraceFDFileName = frmUm.txtTraceFD.Text
    If TRACE_CALC_FD Then
        If Len(TraceFDFileName) > 0 Then
            TraceFDFile = FreeFile
```

```
            '// Abre o arquivo que ficará assim até o modelo parar
            Open TraceFDFileName For Output As TraceFDFile
             '// Imprime o cabeçalho do arquivo Escolher somente as siglas ou c/ as
explicaç ões
              Print #TraceFDFile, "T", "CB(T)", "CT(T)", "COD", "SEX", "AGE", "FD",
"BW", "W", "WM", "WMP", "J", "K", "COW_AGE", "CL(AGE)", "CBA(COW_AGE)",
"CA(AGEY)", "EWM", "EDW"
         End If
      End If

      '// Arquivo para Tracing do Cálculo do FERT
      '--TraceFERTFileName = ModelPath + "FORFAIT_TRACE_FERT.TXT"
      'TraceFERTFileName = frmUm.txtTraceFert.Text
      If TRACE_CALC_FERT Then
         If Len(TraceFERTFileName) > 0 Then
            TraceFERTFile = FreeFile
            '// Abre o arquivo que ficará assim até o modelo parar
            Open TraceFERTFileName For Output As TraceFERTFile
            '// Imprime o cabeçalho do arquivo Escolher somente as siglas ou c/ as
explicaç ões
            '// --Print #TraceFERTFile, "T",   "MB(T)",        "WMA",          "COD",
"SEX", "AGE",         "BW",         "W",          "WM",          "J",         "K",
"CFT", "CFM", "CFW", "CFDW", "DWM", "CCYC",              "CYCLE",    "PEST",
"PCON",       "CONDFERT", "PRENHA"
            Print #TraceFERTFile, "Mês", "Estação_Monta", "Tam_Estrut_Mat",
"Código", "Sexo", "Idade(Meses)", "Peso_Nasc(kg)", "Peso_Atual(kg)",
"Tam_Estrutural(kg)", "Mês_Lactação", "Mês_Gestação", "CFT", "CFM", "CFW",
"CFDW", "DWM", "Prob.Cont.Ciclando", "Ciclando?", "PEST", "Prob.Conceber",
"Cond.p/Fert", "Prenha?"
         End If
```

```
        End If
        '// Prepara o CLIPS
        Call CLIPSInterface.CLIPS_ModelLogicRunBegin
    Else
       ThisDocument.Model.End
    End If
 End Sub

Private Sub ModelLogic_RunBeginSimulation()

  '//Sim.VariableArrayValue(Sim.SymbolNumber("SIMDURATION_V"))              =
SIMDURATION
  If ABORT_SIMULATION Then
    ThisDocument.Model.End
  End If
 End Sub

Private Sub ModelLogic_RunEnd()

    Close 'Close All Files

    '/// Shut down CLIPS
    Call CLIPSInterface.CLIPS_ModelLogicRunEnd
End Sub

Private Sub ModelLogic_RunEndReplication()

    Close '----- Fecha todos os arquivos

End Sub
```

```
Private Sub ModelLogic_RunPause()

    '// Show CLIPS console
    Call CLIPSInterface.CLIPS_ModelLogicRunPause
 End Sub

Private Sub VBA_Block_1_Fire()

    Call Crescimento '----- A interface com o arena está no próprio crescimento
 End Sub

Private Sub VBA_Block_3_Fire()

   '// Bom, ja limpamos o CLIPS e temos os fatos
   '// oriundos dos animais. Vamos executar agora
   '// a máquina de inferência
   Call ModelLogic_RunPause
End Sub

'////////////////////////////////////////////////////////////////
'// UPDATE VAR FROM FACT
'// Updates a variable from a fact into Arena.
'// The fact and the variable must be equal, i.e., if the variable
'// is called AREA, the fact should be (AREA <value>).
'// This handle only single-valued variables
Private Sub UpdateVarFromFact(factStr As String, varName As String)

   Dim retVal

   retVal = GetFactSlot(factStr, "(" & Trim(varName) & " ", "")
   If Len(retVal) > 0 Then
```

```
    Sim.VariableArrayValue(Sim.SymbolNumber(Trim(varName)))                    =
CSng(Val(retVal))
   End If

End Sub

Private Sub VBA_Block_4_Fire()

  Dim tmpStr As String
  Dim att As TAttribute
  Dim entLoc As Long
  Dim i As Integer

  '// Loop through the facts defined in the CLIPS engine
  tmpStr = CLIPS_GetAFact       '// Get the 1st fact
  While Len(tmpStr) > 0
    If InStr(1, tmpStr, "(animal (") > 0 Then
      'FactListLen = FactListLen + 1
      'ReDim Preserve FactList(FactListLen)
      'FactList(FactListLen) = tmpStr
      '// Extract the attributes from the animal and
      '// set the values in a TAttribute struct. The
      '// ExtractAttributes must be customized to extract
      '// the desired attribute values from a predefined
      '// fact string
      ExtractAttributes tmpStr, att
      '// Now we are ready to setup our new data from CLIPS
      '// into the Arena Model
      '// Opcoes: 1. Destruir todas as entidades e recriá-las
      '//         2. Buscar em algum lugar (aparentemente em filas)
      '//         e alterar os atributos, destruir, etc
```

```
If att.status = 3 Then
  tmpStr = "Animal LoadId:" & Str(att.LoadId) & vbCrLf
  tmpStr = tmpStr & "  fact-id: " & Str(att.FactID) & vbCrLf
  tmpStr = tmpStr & "  code   : " & Str(att.cod) & vbCrLf
  tmpStr = tmpStr & "  age    : " & Str(att.Age) & vbCrLf
  tmpStr = tmpStr & "  sex    : " & Str(att.sex) & vbCrLf
  tmpStr = tmpStr & "  weight : " & Str(att.Weight) & vbCrLf

  tmpStr = tmpStr & "  birthw : " & Str(att.BirthW) & vbCrLf
  tmpStr = tmpStr & "  WM     : " & Str(att.WeightM) & vbCrLf
  tmpStr = tmpStr & "  K      : " & Str(att.K) & vbCrLf
  tmpStr = tmpStr & "  J      : " & Str(att.J) & vbCrLf
  tmpStr = tmpStr & "  NC     : " & Str(att.NC) & vbCrLf
  If att.Bull Then
    tmpStr = tmpStr & "  bull : s" & vbCrLf
  Else
    tmpStr = tmpStr & "  birthw : n" & vbCrLf
  End If
  tmpStr = tmpStr & "  status : " & Str(att.status) & vbCrLf
  tmpStr = tmpStr & "  regra  : " & Str(att.Regra) & vbCrLf
  'MsgBox tmpStr
End If

entLoc = att.LoadId

If att.status = 3 Then
  '// THIS IS A NEW ANIMAL
  entLoc = Sim.EntityCreate '// Creates a new entity

  '// Set the attributes for this load
```

```
att.status = 0 '// status Ok

'// If the user didn't assigned a code for the animal...
If att.cod = -1 Then
   '// Creates a new code for the animal and updates the variable
   att.cod = Sim.VariableArrayValue(Sim.SymbolNumber("UltimoCOD_V")) +
1
   Sim.VariableArrayValue(Sim.SymbolNumber("UltimoCOD_V")) = att.cod
End If
Sim.EntityAttribute(entLoc, Sim.SymbolNumber("COD_att")) = att.cod
Sim.EntityAttribute(entLoc, Sim.SymbolNumber("SEX_att")) = att.sex
Sim.EntityAttribute(entLoc, Sim.SymbolNumber("AGE_att")) = att.Age
Sim.EntityAttribute(entLoc, Sim.SymbolNumber("W_att")) = att.Weight
Sim.EntityAttribute(entLoc, Sim.SymbolNumber("BW_att")) = att.BirthW
Sim.EntityAttribute(entLoc, Sim.SymbolNumber("WM_att")) = att.WeightM
Sim.EntityAttribute(entLoc, Sim.SymbolNumber("K_att")) = att.K
Sim.EntityAttribute(entLoc, Sim.SymbolNumber("J_att")) = att.J
Sim.EntityAttribute(entLoc, Sim.SymbolNumber("NCRIAS_att")) = att.NC
If att.Bull Then
  Sim.EntityAttribute(entLoc, Sim.SymbolNumber("TOURO_att")) = 1
Else
  Sim.EntityAttribute(entLoc, Sim.SymbolNumber("TOURO_att")) = 0
End If
Sim.EntityAttribute(entLoc, Sim.SymbolNumber("Status_att")) = 3 '--att.status
Sim.EntityAttribute(entLoc, Sim.SymbolNumber("REGRA")) = att.Regra

'--Sim.EntityInsertIntoQueue entLoc, Sim.SymbolNumber("Novos_Q")
Sim.EntitySetPicture entLoc, 1234 '// Nao adianta, ele poe a figura que quer...
'--Sim.EntityInsertIntoQueue entLoc, Sim.SymbolNumber("Nouveaux_Q")
Sim.EntitySendToBlockLabel entLoc, 0, "NOVOS_AQUI"
```

```
    '// carga fictícia (dummy) para sinalizar p/ que
    '// a carga anterior prossiga no modelo
    entLoc = Sim.EntityCreate
    Sim.EntityAttribute(entLoc, Sim.SymbolNumber("Status_att")) = 999
    Sim.EntitySetPicture entLoc, 1
    Sim.EntitySendToBlockLabel entLoc, 0, "NOVOS_AQUI"
  Else

    '-- '// NOTE that for the existing animals in the model we
    '-- '// can alter just the status. Of course this feature can be implemented
    '-- '// quite easily

    '// Set the status attribute
    Sim.EntityAttribute(entLoc, Sim.SymbolNumber("Status_att")) = att.status
    Sim.EntityAttribute(entLoc, Sim.SymbolNumber("Regra")) = att.Regra
  End If

Else
  '//
  '// OTHER FACTS THAN ANIMALS
  '//

  '// Currently we use simple facts to handle variables from the
  '// simulation model, e.g., a fact like (AREA <value>) stores
  '// the value of the model variable AREA.

  '// The variables below should be not changed because they are
  '// managed by the simulator or calculated from model variables
  'UpdateVarFromFact tmpStr, "MES"
  'UpdateVarFromFact tmpStr, "ANO"
  'UpdateVarFromFact tmpStr, "TAXA_DE_LOTACAO"
```

```
    'UpdateVarFromFact tmpStr, "RELACAO_TOURO_VACA"

    '// Values that theoretically could be changed by production rules
    UpdateVarFromFact tmpStr, "AREA"
    UpdateVarFromFact tmpStr, "PESO_MINIMO_NA_DESMAMA"
    UpdateVarFromFact tmpStr, "IDADE_DE_CRUZAMENTO"
    UpdateVarFromFact tmpStr, "TEMPO_DE_RESGUARDO"
    UpdateVarFromFact tmpStr, "RELACAO_TOURO_VACA_IDEAL"
    UpdateVarFromFact tmpStr, "IDADE_DE_DESCARTE_DE_FEMEAS"
    UpdateVarFromFact tmpStr, "IDADE_DE_DESCARTE_DE_MACHOS"
    UpdateVarFromFact tmpStr, "IDADE_DE_DESCARTE_DE_TOUROS"

  End If
  tmpStr = CLIPS_GetAFact     '// Get the next fact
 Wend

End Sub

Private Sub VBA_Block_2_Fire()

Dim Age, COW_AGE, J, K, AGEY, sex, cod, T, MORTO As Integer
Dim FD, W, WM, WMP, BW, EDW, EWM As Double

Dim CL_AGE, CA_AGEY As Double 'usadas no tracing

Dim MESES As Integer 'valor para converter de meses p/anos, etc.
                '=1 indica que se usa o mês

  '////////////////////////////////////////////////////
  '/////////////// C A L C U L A  O  F A T O R  D E  M O R T E
  '////////////////////////////////////////////////////
```

```
'////////////// Interface com ARENA //////////////

If Not (Sim Is Nothing) Then

    MORTO = 0
    '//INTERFACE
    With Sim
        cod = .EntityAttribute(.ActiveEntity, .SymbolNumber("COD_att"))
        sex = .EntityAttribute(.ActiveEntity, .SymbolNumber("SEX_att"))
        COW_AGE = .EntityAttribute(.ActiveEntity,
.SymbolNumber("COWAGE_att"))
        Age = .EntityAttribute(.ActiveEntity, .SymbolNumber("AGE_att"))
        W = .EntityAttribute(.ActiveEntity, .SymbolNumber("W_att"))
        WM = .EntityAttribute(.ActiveEntity, .SymbolNumber("WM_att"))
        BW = .EntityAttribute(.ActiveEntity, .SymbolNumber("BW_att"))
        WMP = .EntityAttribute(.ActiveEntity, Sim.SymbolNumber("WMP_att"))
        K = .EntityAttribute(.ActiveEntity, .SymbolNumber("K_att"))
        J = .EntityAttribute(.ActiveEntity, .SymbolNumber("J_att"))
        T = .VariableArrayValue(.SymbolNumber("T_V"))
    End With
    '//INTERFACE
    MESES = 1
    '----- Calcula a idade em anos
    If Age <= 12 Then
      AGEY = 0
    Else
      AGEY = Age \ 12
    End If

    If Age = 0 Then 'Se a idade do animal em meses = 0
        'CB(T) é o efeito do mês do ano nas mortes perinatais
```

```
        'Seus valores são: 1.5,1.5,1.2,1.0,1.0,1.0,1.1,1.2,1.0,1.0,1.2,1.5
        'CBA(COW_AGE) é o efeito da idade da vaca na morte da cria no parto
        'Seus valores são: 0,0.03,0.02,0.01,0.01,0.01,0.01,0.01,0.01,0.01,0.02,
        '                  0.02,0.03,0.03,0.03
        FD = CB(T) * CBA(COW_AGE) 'COW_AGE é a idade da mãe

    Else
        'Se tiver 1 mês de vida ou mais....
        '///// Fator de Morte Básico
        FD = 0.001 * CT(T) 'CT é o efeito do mês do ano nas mortes
                    '**** CT esta com valores 1.0,...1.2
            'Se o peso for menor do que o tamanho estrutural...
        '///// Correção para animal abaixo do peso
        If W < WM Then FD = FD * Exp(140 * ((1 - (W / WM)) ^ 3.3))

        '///////Correção para o Fator de Morte dos Bezerros
        If Age <= 12 * MESES Then 'Se a idade do animal em meses <=12
            FD = FD * CL(Age) 'CL é o efeito do mês de idade do bezerro na sua
própria morte
                        ' **** CL esta com valores 14,7,5,5,2,2,2,2,2,2,2,2
            CL_AGE = CL(Age) '----- tracing
            '"In addition, death rates are increased in calves up to 4
            ' months of age that have reduced structural (WM) growth."
            If Age < 4 * MESES Then
                'EDW: Taxa de Mudança Esperada no Peso por dia
                EDW = (WMP - BW) / 360
                'EWM: Tamanho estrutural esperado
                '(30*EDW) é o ganho por mês
                '(30*EDW)* AGE é o ganho do nascimento à idade atual (meses)
                EWM = BW + (30 * EDW) * Age
                'Se o Tamanho Estrutural for menor do que o esperado...
```

```
            If WM < EWM Then
               FD = FD + (1 - (WM / EWM) ^ 0.64)
            End If
         End If

      Else
         '////////// Correção do Fator de Morte para Vacas que pariram ou que vão parir este mês
         '*****************
         'Calve this month
         'K=9 ou K=8
         If K = 8 * MESES Then
            FD = FD * CA(AGEY) 'CA é o efeito da idade da vaca na
                               'sua própria morte no parto.
                               'CA esta com os valores (15 anos):
                               '0,20,10,5,3,3,3,3,3,3,3,5,5,10,20
            CA_AGEY = CA(AGEY) '----- tracing
         End If
         '////////// Correção do Fator de Morte para Vacas que pariram mês passado
         '*****************
         'Calve Last Month (já faz 1 mês que é lactante)
         If J = 1 * MESES Then
            FD = 1.25 * FD
         End If
      End If
   End If

   Randomize
   RandN = Rnd
   If FD > RandN Then MORTO = 1 'Indica a Morte do Animal
```

```
        '----- Criar uma Lógica de TRACING para o Calcula FD

        ' lógica que dá o animal como morto
        '//INTERFACE
        With Sim
           .EntityAttribute(.ActiveEntity, .SymbolNumber("FD_att")) = FD
           .EntityAttribute(.ActiveEntity, .SymbolNumber("MORTO_att")) = MORTO
        End With
        '//INTERFACE
        '----- TRACING: ATUALIZAÇÃO DO ARQUIVO DE ACOMPANHAMENTO DO
FD
        If TRACE_CALC_FD Then
           Print #TraceFDFile, T, CB(T), CT(T), cod, sex, Age, FD, BW, W, WM,
WMP, J, K, COW_AGE, CL_AGE, CBA(COW_AGE), CA_AGEY, EWM, EDW
        End If

    End If '// if sim is noth nothing
End Sub

Private Sub VBA_Block_9_Fire()

Dim AttNdx As Long
Dim PRENHA, MAE, J, K, T1, Age, T, CONDFERT, sex As Integer

    If Not (Sim Is Nothing) Then

        '///INTERFACE
        With Sim
           T = .VariableArrayValue(.SymbolNumber("T_V"))
```

```
        IDPRIMCRUZ                                                    =
.VariableArrayValue(.SymbolNumber("IDADE_DE_CRUZAMENTO"))
        PRENHA = .EntityAttribute(.ActiveEntity, .SymbolNumber("PRENHA_att"))
        J = .EntityAttribute(.ActiveEntity, .SymbolNumber("J_att"))
        K = .EntityAttribute(.ActiveEntity, .SymbolNumber("K_att"))
        T1 = .EntityAttribute(.ActiveEntity, .SymbolNumber("T1_att"))
        Age = .EntityAttribute(.ActiveEntity, .SymbolNumber("AGE_att"))
        MAE = .EntityAttribute(.ActiveEntity, .SymbolNumber("MAE_att"))
        sex = .EntityAttribute(.ActiveEntity, .SymbolNumber("SEX_att"))
    End With
    '///INTERFACE

    '-----
    If sex = 0 Then
        If PRENHA = 0 Then
            T1 = T1 + 1 '----- Passa-se mais um período sem ficar prenha
        Else
            K = K + 1 '----- Passa-se mais um mês de prenhez
        End If
    End If

    '----- Mae_att indica se a vaca tem bezerro
    If MAE <> 0 Then J = J + 1   '// Passa-se mais um mês de lactação
    If J = IDESM Then MAE = 0    '// Desmama: MAE passa a zero ao se atingir a
idade de desmama

    Age = Age + 1 'Envelhecimento do animal

    '////////////////////////////////////
    '////////////////////////////////////
    '// VERIFICAR
```

```
'----- Condição Para Fertilidade: Vacas com mais de 18 meses,
'----- depois de 4 meses é que podem ser fertilizadas novamente
If ((Age >= IDPRIMCRUZ) Or (J > TEMPO_DE_RESGUARDO)) And (sex =
0) Then
   CONDFERT = 1
Else
   CONDFERT = 0
End If

'////////////////////////////////////
'////////////////////////////////////

'***----- Passado para o Arena
'T = T + 1 'Passa-se mais umm período
'If T > 12 Then
'       T = 1 'Garante que T varie de 1 a 12
'End If
'***-----

'//////// INTERFACE COM ARENA - Valores Modificados neste módulo
With Sim
   .EntityAttribute(.ActiveEntity, .SymbolNumber("T1_att")) = T1
   .EntityAttribute(.ActiveEntity, .SymbolNumber("K_att")) = K
   .EntityAttribute(.ActiveEntity, .SymbolNumber("AGE_att")) = Age
   .EntityAttribute(.ActiveEntity,       .SymbolNumber("CONDFERT_att"))      =
CONDFERT
   .EntityAttribute(.ActiveEntity, .SymbolNumber("J_att")) = J
   .EntityAttribute(.ActiveEntity, .SymbolNumber("MAE_att")) = MAE
   '.VariableArrayValue(.SymbolNumber("T_V")) = T
End With
```

```
    End If '

End Sub

Private Sub VBA_Block_5_Fire()

Dim PEST, PCON, CCYC, CFM, CFDW, CFW, CFT As Double
Dim W, WM, GWM, DWM, DW, BW As Double
Dim CONDFERT, T, T1, Cycle, NCICLOS, PRENHA, NPRENHAS, cod, sex, Age,
J, K As Integer

    If Not (Sim Is Nothing) Then
        '////////////////////////////////////////////////////
        '/////////////// F E R T I L I D A D E
        '////////////////////////////////////////////////////
        '----- Inicialização de variáveis
        PEST = 0
        PCON = 0
        CCYC = 0
        CFM = 0
        CFDW = 0
        CFW = 0
        CFT = 0
        '----- Retira o valor de CONDFERT_att e o coloca na variável CONDFERT
       ' AttNdx =
        With Sim
            CONDFERT                    =                    .EntityAttribute(.ActiveEntity,
.SymbolNumber("CONDFERT_att"))
            W = .EntityAttribute(.ActiveEntity, .SymbolNumber("W_att"))
```

```
        DW = .EntityAttribute(.ActiveEntity, .SymbolNumber("DW_att"))
        WM = .EntityAttribute(.ActiveEntity, .SymbolNumber("WM_att"))
        DWM = .EntityAttribute(.ActiveEntity, .SymbolNumber("DWM_att"))
        GWM = .EntityAttribute(.ActiveEntity, .SymbolNumber("GWM_att"))
        T1 = .EntityAttribute(.ActiveEntity, .SymbolNumber("T1_att"))
        Cycle = .EntityAttribute(.ActiveEntity, .SymbolNumber("CYCLE_att"))
        NCICLOS = .EntityAttribute(.ActiveEntity, .SymbolNumber("NCICLOS_att"))
        PRENHA = .EntityAttribute(.ActiveEntity, .SymbolNumber("PRENHA_att"))
        T = .VariableArrayValue(.SymbolNumber("T_V"))
        NPRENHAS = .VariableArrayValue(.SymbolNumber("NPRENHAS_V"))

        cod = .EntityAttribute(.ActiveEntity, .SymbolNumber("COD_att"))
        sex = .EntityAttribute(.ActiveEntity, .SymbolNumber("SEX_att"))
        Age = .EntityAttribute(.ActiveEntity, .SymbolNumber("AGE_att"))
        J = .EntityAttribute(.ActiveEntity, .SymbolNumber("J_att"))
        K = .EntityAttribute(.ActiveEntity, .SymbolNumber("K_att"))

        'BW, CFT, CFM, CFW, CFDW, CCYC, CYCLE, PEST, PCON, CONDFERT

        BW = .EntityAttribute(.ActiveEntity, .SymbolNumber("BW_att"))
        CFT = .EntityAttribute(.ActiveEntity, .SymbolNumber("CFT_att"))
        CFM = .EntityAttribute(.ActiveEntity, .SymbolNumber("CFM_att"))
        CFW = .EntityAttribute(.ActiveEntity, .SymbolNumber("CFW_att"))
        CFDW = .EntityAttribute(.ActiveEntity, .SymbolNumber("CFDW_att"))
        CCYC = .EntityAttribute(.ActiveEntity, .SymbolNumber("CCYC_att"))
        PEST = .EntityAttribute(.ActiveEntity, .SymbolNumber("PEST_att"))
        PCON = .EntityAttribute(.ActiveEntity, .SymbolNumber("PCON_att"))
        CONDFERT = .EntityAttribute(.ActiveEntity,
.SymbolNumber("CONDFERT_att"))

    End With
```

```
'=== 'MsgBox "Mês T=" + Str(T) + "==> Estação de Monta=" + Str(MB(T))
'=== '----- Se for período de monta (MB(T)<>0) e se o animal tiver condições de ser fertilizado...
'=== If MB(T) <> 0 And CONDFERT = 1 Then
'=== 'If MB(T) <> 0 Then
'=== 'If 1 = 1 Then
'===     '----- Cálculo do Fator de Correção para o Grau de Maturidade
'===     CFM = ((WM / (0.6 * WMA)) - 0.67) / (1 - 0.67)
'===     If CFM > 1 Then CFM = 1 '**** Assegura que CFM<=1
'===     '----- Cálculo do Fator de Correção para a Condição
'===     CFW = (W / WM - 0.75) / (1 - 0.75)
'===     'If CFM > 1 Then CFM = 1 '**** Assegura que CFW<=1
'===
'===      'DWM: Mudança no tamanho estrutural, GWM: Ganho no Tamanho estrutural
'===     DWM = GWM + 0.01 * (WM - W)
'===       CFDW = 100 * (DWM - DW) / WM '******* Este valor dá negativo quando
'===                             'DW>DWM. IMPOR o valor absoluto para
'===                             'CFDW.
'===     '**************** ??????? Este DWM é somado ao WM?
'===     CFDW = Abs(CFDW) 'IMPONDO CFDW >0
'===
'===     'Id DW >DWM then
'===     '   CFDW=1
'===     'Else
'===     '   CFDW=1-100 * (DWM - DW) / WM
'===     'End if
'===
'===     'Cálculo do CFT
```

```
'===    CFT = 1 - Exp(-0.519 * T1 ^ 3.9) '// 1 - EP(-0.519 * EP(3.9*LN(T1_att))

'===    ' Select Case J of
'===    '  Case 0
'===    '      CFT=1
'===    '  Case 1
'===    '      CFT=0
'===    '  Case 2
'===    '      CFT=0.40
'===    '  Case 3
'===    '      CFT=0.99
'===    '  Else
'===    '      CFT=1
'===    'End Select
'===
'===
'===    '////// Controle da Ciclagem
'===    If CYCLE = 1 Then
'===        If CFDW < 0 Then MsgBox
'===        CCYC = CFW ^ 0.1 * CFDW ^ 0.1
'===        '----- Calcula um # aleatório para ver se vai continuar a ciclar
'===        Randomize
'===        RandN = Rnd
'===        If CCYC >= RandN Then
'===            CYCLE = 1 'Continua a ciclar
'===            'MsgBox "Continua a Ciclar"
'===        Else
'===            CYCLE = 0 'Deixa de ciclar
'===            'MsgBox "Pára de Ciclar"
'===        End If
'===    Else
```

```
'===        PEST = A * CFM * CFW * CFL * CFT * CFDW
'===        '----- Calcula um # aleatório para ver se vai começar a ciclar
'===        Randomize
'===        RandN = Rnd
'===        If PEST >= RandN Then
'===           CYCLE = 1 'Começa a ciclar
'===           NCICLOS = NCICLOS + 1
'===        End If
'===
'===     End If
'===
'===     '////// Controle da Concepção
'===     PCON = 0
'===     If CYCLE = 1 Then
'===        If (CFW <= 0) Or (CFDW <= 0) Or (CFT <= 0) Then
'===           PCON = 0
'===        Else
'===           PCON = c * CFW ^ 0.2 * CFDW ^ 0.2 * CFT ^ 0.5 * MB(T)
'===             '// PCON = EP(0.2 * LN(CFW_att)) * EP(0.2 * LN(CFDW_att)) *
EP(0.5 * LN(CFT_att)) * MB_V(T_V)
'===
'===        End If
'===
'===        Randomize
'===        RandN = Sim.SampleUniform(0, 1, 1)
'===        If PCON >= RandN Then '****
'===           'K = 1
'===           PRENHA = 1 'Indica que a vaca prenhou
'===           NPRENHAS = NPRENHAS + 1
'===           CONDFERT = 0
'===           'N_PCON = N_PCON + 1
```

```
'===                'MsgBox " ==> PCON=" + Str(PCON) + "  >> RND=" +
Str(RandN) + "==>PRENHA_ATT=" + Str(PRENHA)
'===       End If
'===    End If
'===
'===    '///INTERFACE
'===    With Sim
'===       .EntityAttribute(.ActiveEntity, .SymbolNumber("T1_att")) = T1
'===            .EntityAttribute(.ActiveEntity, .SymbolNumber("CYCLE_att")) =
CYCLE
'===          .EntityAttribute(.ActiveEntity, .SymbolNumber("NCICLOS_att")) =
NCICLOS
'===        .EntityAttribute(.ActiveEntity, .SymbolNumber("CONDFERT_att")) =
CONDFERT
'===        '.EntityAttribute(.ActiveEntity, .SymbolNumber("K_att")) = K
'===           .EntityAttribute(.ActiveEntity, .SymbolNumber("PRENHA_att")) =
PRENHA
'===        '.EntityAttribute(.ActiveEntity, .SymbolNumber("BW_att")) = BW
'===        .EntityAttribute(.ActiveEntity, .SymbolNumber("W_att")) = W
'===        .EntityAttribute(.ActiveEntity, .SymbolNumber("WM_att")) = WM
'===                .VariableArrayValue(.SymbolNumber("NPRENHAS_V"))  =
NPRENHAS
'===        '.EntityAttribute(.ActiveEntity, .SymbolNumber("AGE_att")) = AGE
'===        '.EntityAttribute(.ActiveEntity, .SymbolNumber("Q_att")) = Q
'===        '.EntityAttribute(.ActiveEntity, .SymbolNumber("J_att")) = J
'===        '.EntityAttribute(.ActiveEntity, .SymbolNumber("FG_att")) = FG
'===        '.EntityAttribute(.ActiveEntity, .SymbolNumber("DW_att")) = DW
'===        '.EntityAttribute(.ActiveEntity, .SymbolNumber("GWM_att")) = GWM
'===        '.EntityAttribute(.ActiveEntity,.SymbolNumber("DS_att")) = DS
'===        '.EntityAttribute(.ActiveEntity, .SymbolNumber("FAT_att")) = FAT
'===        '.EntityAttribute(.ActiveEntity, .SymbolNumber("ANF_att")) = ANF
```

```
'===        '.EntityAttribute(.ActiveEntity, .SymbolNumber("RL_att")) = RL
'===        '.EntityAttribute(.ActiveEntity, .SymbolNumber("RP_att")) = RP
'===        '.EntityAttribute(.ActiveEntity, .SymbolNumber("GR_att")) = GR
'===        '.EntityAttribute(.ActiveEntity, .SymbolNumber("RG_att")) = RG
'===        '.EntityAttribute(.ActiveEntity, .SymbolNumber("M_att")) = M
'===        '.EntityAttribute(.ActiveEntity, .SymbolNumber("CFX_att")) = CFX
'===        '.EntityAttribute(.ActiveEntity, .SymbolNumber("CFA_att")) = CFA
'===        '.EntityAttribute(.ActiveEntity, .SymbolNumber("REQ_att")) = REQ
'===        '.EntityAttribute(.ActiveEntity, .SymbolNumber("R1_att")) = R1
'===        '.EntityAttribute(.ActiveEntity, .SymbolNumber("R2_att")) = R2
'===        '.EntityAttribute(.ActiveEntity, .SymbolNumber("R3_att")) = R3
'===        '.EntityAttribute(.ActiveEntity, .SymbolNumber("TM_att")) = TM
'===        '.EntityAttribute(.ActiveEntity,.SymbolNumber("TN_att")) = TN
'===        '.EntityAttribute(.ActiveEntity,.SymbolNumber("CN_att")) = CN
'===              '.EntityAttribute(.ActiveEntity,  .SymbolNumber("REQM_att"))  =
REQM
'===              '.EntityAttribute(.ActiveEntity,  .SymbolNumber("ATDN_att"))  =
ATDN
'===        '.EntityAttribute(.ActiveEntity, .SymbolNumber("RXM_att")) = RXM
'===              '.EntityAttribute(.ActiveEntity,  .SymbolNumber("PREQ_att"))  =
PREQ
'===        '.EntityAttribute(.ActiveEntity, .SymbolNumber("CEM_att")) = CEM
'===        '.EntityAttribute(.ActiveEntity, .SymbolNumber("CEG_att")) = CEG
'===        '.EntityAttribute(.ActiveEntity, .SymbolNumber("CFM_att")) = CFM
'===        '.EntityAttribute(.ActiveEntity, .SymbolNumber("PE_att")) = PE
'===        '.EntityAttribute(.ActiveEntity, .SymbolNumber("CFT_att")) = CFT
'===        '.EntityAttribute(.ActiveEntity, .SymbolNumber("CFW_att")) = CFW
'===              '.EntityAttribute(.ActiveEntity,  .SymbolNumber("CFDW_att"))  =
CFDW
'===        '.EntityAttribute(.ActiveEntity, .SymbolNumber("DWM_att")) = DWM
```

```
      '===                '.EntityAttribute(.ActiveEntity,  .SymbolNumber("CCYC_att"))  =
CCYC
      '===        '.EntityAttribute(.ActiveEntity, .SymbolNumber("PEST_att")) = PEST
      '===                '.EntityAttribute(.ActiveEntity,  .SymbolNumber("PCON_att"))  =
PCON
      '===        '.EntityAttribute(.ActiveEntity, .SymbolNumber("PM_att")) = PM
      '===        '.EntityAttribute(.ActiveEntity, .SymbolNumber("WMP_att")) = WMP
      '===     End With
      '=== '///--------------INTERFACE

      '----- TRACING: ATUALIZAÇÃO DO ARQUIVO DE ACOMPANHAMENTO
      If TRACE_CALC_FERT Then
         Print #TraceFERTFile, T, MB(T), WMA, cod, sex, Age, BW, W, WM, J, K,
CFT,  CFM,  CFW,  CFDW,  DWM,  CCYC,  Cycle,  PEST,  PCON,  CONDFERT,
PRENHA
      End If

      '=== '; MsgBox "Inside Fertility"
      '=== End If 'checagem se o animal pode entrar em fertilidade

      '////////////////// F I M  -  F E R T I L I D A D E ////////////////

   End If '// Sim is not nothing

End Sub

Private Sub VBA_Block_7_Fire()
  '//
  '// Insere a entidade corrente como um fato no CLIPS            //
  Call CLIPS_InsCurrEntityInCLIPS
```

```
End Sub

Private Sub VBA_Block_6_Fire()

  Dim ano As String
  Dim mes As String
  Dim monta As String
  Dim UseVars As Boolean

  mes = Sim.VariableArrayValue(Sim.SymbolNumber("T_V"))
  ano = Sim.VariableArrayValue(Sim.SymbolNumber("YEAR_V"))
  monta = Sim.VariableArrayValue(Sim.SymbolNumber("MB_V", mes))

  '//CLIPS_Execute "(reset)"
  CLIPS_Reset   '// Limpa os fatos anteriores e ativa os deffacts

  UseVars = False

  If UseVars Then
      '// Setup the ES defglobals using the Arena variables
      CLIPS_Execute "(defglobal ?*PESO_MINIMO_NA_DESMAMA* = " &
Sim.VariableArrayValue(Sim.SymbolNumber("PESO_MINIMO_NA_DESMAMA"))
& ")"
      CLIPS_Execute "(defglobal ?*IDADE_DE_CRUZAMENTO* = " &
Sim.VariableArrayValue(Sim.SymbolNumber("IDADE_DE_CRUZAMENTO")) & ")"
      CLIPS_Execute "(defglobal ?*TEMPO_DE_RESGUARDO* = " &
Sim.VariableArrayValue(Sim.SymbolNumber("TEMPO_DE_RESGUARDO")) & ")"
      CLIPS_Execute "(defglobal ?*MES* = " &
Sim.VariableArrayValue(Sim.SymbolNumber("T_V")) & ")"
      CLIPS_Execute "(defglobal ?*ANO* = " &
Sim.VariableArrayValue(Sim.SymbolNumber("YEAR_V")) & ")"
```

```
        CLIPS_Execute "(defglobal ?*IDADE_DE_DESMAMA* = " &
Sim.VariableArrayValue(Sim.SymbolNumber("IDADE_DE_DESMAMA")) & ")"
        CLIPS_Execute "(defglobal ?*TAXA_DE_LOTACAO* = " &
Sim.VariableArrayValue(Sim.SymbolNumber("TAXA_DE_LOTACAO")) & ")"
        CLIPS_Execute "(defglobal ?*RELACAO_TOURO_VACA_IDEAL* = " &
Sim.VariableArrayValue(Sim.SymbolNumber("RELACAO_TOURO_VACA_IDEAL"
)) & ")"
        CLIPS_Execute "(defglobal ?*RELACAO_TOURO_VACA* = " &
Sim.VariableArrayValue(Sim.SymbolNumber("RELACAO_TOURO_VACA")) & ")"
        CLIPS_Execute "(defglobal ?*AREA* = " &
Sim.VariableArrayValue(Sim.SymbolNumber("AREA")) & ")"

        CLIPS_Execute "(defglobal ?*IDADE_DE_DESCARTE_DE_FEMEAS* = " &
Sim.VariableArrayValue(Sim.SymbolNumber("IDADE_DE_DESCARTE_DE_FEME
AS")) & ")"
        CLIPS_Execute "(defglobal ?*IDADE_DE_DESCARTE_DE_MACHOS* = " &
Sim.VariableArrayValue(Sim.SymbolNumber("IDADE_DE_DESCARTE_DE_MAC
HOS")) & ")"
        CLIPS_Execute "(defglobal ?*IDADE_DE_DESCARTE_DE_TOUROS* = " &
Sim.VariableArrayValue(Sim.SymbolNumber("IDADE_DE_DESCARTE_DE_TOU
ROS")) & ")"
    Else
        '// EM VEZ de usar VARIÁVEIS GLOBAIS, utiliza-se FATOS PARA
        '// PASSAR AS VARIÁVEIS
        CLIPS_Execute "(assert (PESO_MINIMO_NA_DESMAMA " &
RemoveCommas(Sim.VariableArrayValue(Sim.SymbolNumber("PESO_MINIMO_
NA_DESMAMA"))) & "))"
        CLIPS_Execute "(assert (IDADE_DE_CRUZAMENTO " &
Sim.VariableArrayValue(Sim.SymbolNumber("IDADE_DE_CRUZAMENTO")) & "))"
```

```
    CLIPS_Execute "(assert (TEMPO_DE_RESGUARDO " &
Sim.VariableArrayValue(Sim.SymbolNumber("TEMPO_DE_RESGUARDO")) & "))"
    CLIPS_Execute "(assert (MES " &
Sim.VariableArrayValue(Sim.SymbolNumber("T_V")) & "))"
    CLIPS_Execute "(assert (ANO " &
Sim.VariableArrayValue(Sim.SymbolNumber("YEAR_V")) & "))"

    CLIPS_Execute "(assert (IDADE_DE_DESMAMA " &
Sim.VariableArrayValue(Sim.SymbolNumber("IDADE_DE_DESMAMA")) & "))"
    CLIPS_Execute "(assert (TAXA_DE_LOTACAO " &
RemoveCommas(Sim.VariableArrayValue(Sim.SymbolNumber("TAXA_DE_LOTA
CAO"))) & "))"
    CLIPS_Execute "(assert (RELACAO_TOURO_VACA_IDEAL " &
RemoveCommas(Sim.VariableArrayValue(Sim.SymbolNumber("RELACAO_TOUR
O_VACA_IDEAL"))) & "))"
    CLIPS_Execute "(assert (RELACAO_TOURO_VACA " &
RemoveCommas(Sim.VariableArrayValue(Sim.SymbolNumber("RELACAO_TOUR
O_VACA"))) & "))"
    CLIPS_Execute "(assert (AREA " &
RemoveCommas(Sim.VariableArrayValue(Sim.SymbolNumber("AREA"))) & "))"

    CLIPS_Execute "(assert (IDADE_DE_DESCARTE_DE_FEMEAS " &
Sim.VariableArrayValue(Sim.SymbolNumber("IDADE_DE_DESCARTE_DE_FEME
AS")) & "))"
    CLIPS_Execute "(assert (IDADE_DE_DESCARTE_DE_MACHOS " &
Sim.VariableArrayValue(Sim.SymbolNumber("IDADE_DE_DESCARTE_DE_MAC
HOS")) & "))"
    CLIPS_Execute "(assert (IDADE_DE_DESCARTE_DE_TOUROS " &
Sim.VariableArrayValue(Sim.SymbolNumber("IDADE_DE_DESCARTE_DE_TOU
ROS")) & "))"
  End If
```

```
  '-- CLIPS_Execute "(list-deftemplates)"
  '-- CLIPS_Execute "(list-defglobals)"
    '// Adjust the facts
  CLIPS_Execute "(retract 0)" '// Retira o (default-fact)
  '-- CLIPS_Execute "(assert (mes " & Trim(Str(mes)) & "))"
  '-- CLIPS_Execute "(assert (ano " & Trim(Str(ano)) & "))"
  If (monta = 1) Then
    CLIPS_Execute "(assert (estacao-monta))"
  End If

  '// Agora os fatos estão prontos para serem colocados no CLIPS
End Sub

'// Search in the var module for the specified name
Private Function FindVarName(ByVal varName As String) As Integer

    Dim OK As Boolean
    Dim tmpV As String
    Dim NUMVARSINMODULE As Integer
    Dim i As Integer
    Dim m As Arena.Model
    Dim MyMod As Arena.Module

    NUMVARSINMODULE = 24 '// Coloque aqui o # de variaveis no
                         '// Modulo de variaveis

    '// Locate the variables module. In this example, we assigned
    '// the tag to be "VARS"
    Set m = ThisDocument.Model
    i = m.Modules.Find(smFindTag, "VARS")
    Set MyMod = m.Modules.Item(i)
```

```
    '// Set the values for the 2nd variavle: MB_V
    '// ATTENTION: the final index is the row index and the 1st, the col index
    '// Search for the MB_V
    OK = False
    i = 0
    tmpV = " "
    While (Not OK) And (i < NUMVARSINMODULE)
       i = i + 1
       tmpV = Trim(MyMod.Data("var(" & Trim(Str(i)) & ")"))
       If StrComp(tmpV, varName, vbTextCompare) = 0 Then
         '--MsgBox i
         OK = True
       End If
    Wend
    If Not OK Then
      i = 0
    End If

    FindVarName = i '// Return the value

End Function
```