

UNIVERSIDADE DO EXTREMO SUL CATARINENSE – UNESC

CURSO DE CIÊNCIA DA COMPUTAÇÃO

RAFAEL CESA COPETI

UTILIZAÇÃO DAS REDES DE PETRI NOS SISTEMAS DE GERENCIAMENTO

WORKFLOW NA FASE DE CONCEPÇÃO DO PROJETO UNIFICADO

CRICIÚMA, JULHO DE 2009

RAFAEL CESA COPETI

**UTILIZAÇÃO DAS REDES DE PETRI NOS SISTEMAS DE GERENCIAMENTO
WORKFLOW NA FASE DE CONCEPÇÃO DO PROJETO UNIFICADO**

Trabalho de Conclusão de Curso apresentado
para obtenção do Grau de Bacharel em Ciência
da Computação da Universidade do Extremo
Sul Catarinense

Orientador: Profa. MSc. Ana Cláudia Garcia
Barbosa

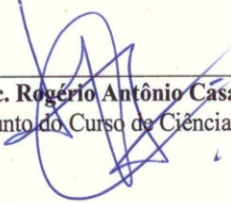
Co-orientador: Prof. Esp. Kristian Madeira

CRICIÚMA, JULHO DE 2009

RAFAEL CESA COPETI

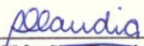
**Utilização das Redes de Petri nos Sistemas de Gerenciamento Workflow na Fase de
Concepção do Projeto Unificado**

Submetido ao corpo docente do Curso de Ciência da Computação da Universidade do Extremo Sul Catarinense como um dos requisitos para obtenção do grau de Bacharel em Ciência da Computação.

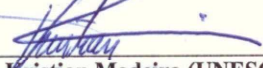


Prof. MSc. Rogério Antônio Casagrande
Coordenador Adjunto do Curso de Ciência da Computação

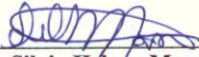
Banca Examinadora:



Profa. MSc. Ana Cláudia Garcia Barbosa (UNESC)
Orientador



Prof. Esp. Kristian Madeira (UNESC)
Co-Orientador



Profa. MSc. Silvia Helena Mangili (Faculdade Energia/ESUCRI)



Prof. MSc. Gustavo Bisognin (UNESC)

Dedico esta pesquisa aos meus pais que
sempre me incentivaram a continuar
meus estudos, minha irmã, e minha
namorada por me ajudar no cotidiano.

AGRADECIMENTOS

Agradeço a Deus por ter iluminado meu caminho e permitido a conclusão deste trabalho, pois sem ele nada disso seria possível.

A minha família que me deu força e coragem nos momentos difíceis da minha vida e que sempre acreditaram em mim.

A minha namorada Juliane que sempre ofereceu ajuda para esta pesquisa mesmo não conhecendo o assunto contribui dentro do que sabia.

A minha orientadora Ana Cláudia por ter me aceitado como orientando e por toda paciência e atenção, mesmo após ganhar gêmeos, dedicadas a esta pesquisa.

Ao meu co-orientador Kristian por ter acreditado no meu projeto e proporcionado o intercâmbio de estudo nos levando até outra universidade, sendo isto fundamental para a conclusão do trabalho.

Ao professor e pesquisador da Universidade Federal de Santa Catarina (UFSC), Jean Marie Farines, por ser acessível e estar, sempre que solicitado, disposto a ajudar nesta pesquisa.

Aos meus amigos, principalmente ao Thiago que conseguiu um livro, disponível na Universidade Federal de Santa Catarina (UFSC), para esta pesquisa.

“A persistência é o caminho do êxito.”

(Charles Chaplin)

RESUMO

A presente pesquisa relata sobre conceitos de Engenharia de *Software*, sua importância na fase inicial de um sistema, estudo sobre gerência de processos, processo unificado, fluxo de trabalho de requisitos, redes de Petri e suas propriedades, compreensão de um projeto modelo. Aborda-se a aplicação das redes de Petri nos sistemas de gerenciamento *workflow* como ferramenta de modelagem e análise do comportamento de um projeto pronto, como exemplo, na busca de melhorar a visualização e a coerência dos relacionamentos entre os requisitos, com isso, permitindo a correção de erros, conflitos e bloqueios.

Utiliza-se a estrutura gráfica das redes de Petri na fase de concepção do projeto e suas atividades de requisitos para demonstrar aos desenvolvedores de *software* uma modelagem unificada servindo como pesquisa para fases futuras do projeto.

Os resultados obtidos foram satisfatórios, pois evidenciaram que o método sugerido com interatividade proporciona subsídios fundamentais para a viabilização de um projeto menos propício a erros e conseqüentemente confiável.

Palavras-chave: Engenharia de *Software*, Gerência de Processos, Processo Unificado, Sistemas de Gerenciamento *Workflow*, Redes de Petri.

ABSTRACT

This research presents concepts of software Engineering, its importance in the initial phase of a system, study about processes management, unified process, work flow requirements, Petri nets and their properties, understanding of a model project. It is approached the application of Petri nets in the management system of workflow as modeling tool and analysis of a finished project as an example, with the purpose of improving the visualization and the coherence of the relationships among the requirements, so it lets to correct errors, conflicts and blockades.

The graphic structure of the Petri nets is used in the conception phase of the Project and their activities of requirements to demonstrate to the software creators an unified modeling serving as research for future phases of the project.

The obtained results were satisfactory, because they evidenced that the suggested method with interactivity, provides fundamental information's for the viability of a project which is less favorable to mistakes and consequently reliable.

Word-key: Engineering of Software, Management of Processes, Unified Process, Workflow Management Systems, Petri Nets.

LISTA DE ILUSTRAÇÕES

Figura 1.Ciclo de Desenvolvimento de Software	24
Figura 2.Ciclo de Vida do Processo Unificado	25
Figura 3.Foco de Cada Fase do Processo Unificado	26
Figura 4.Modelo de Referência da WPMC	28
Figura 5.Elementos da Rede de Petri.....	45
Figura 6.Disparo de Uma Transição.....	47
Figura 7.Redes de Petri Ordinária Limitada, Viva e Reiniciável.....	48
Figura 8.Exemplo de Invariantes.....	49
Figura 9.Estado Sair_Desligado	55
Figura 10.Programa em Execução.....	56
Figura 11.Estado de Espera	57
Figura 12.Modelagem do Projeto em Redes de Petri	58
Figura 13.Tela Principal do Sistema	69
Figura 14.Função Recall.....	69
Figura 15.Tela Inicial do Sistema.....	70
Figura 16.Operações Geométricas.....	70
Figura 17.Desenho do Item Ferramentas do Menu	70
Figura 18.Modelagem da Barra de Operação.....	71
Figura 19.Grafo Conexo	75
Figura 20.Grafo Não Conexo	75
Figura 21.Grafo Fortemente Conexo.....	76
Figura 22.Grafo Fracamente Conexo	76
Figura 23.Foto na UFSC.....	77

LISTA DE TABELAS

Tabela1.Critérios de Avaliação	37
Tabela2.Interpretações Típicas de Transições e Estados.....	46

LISTA DE SIGLAS

RUP *Rational Unified Process*

UFSC Universidade Federal de Santa Catarina

UML *Unified Modeling Language*

WfMC *Workflow Management Coalition*

WFMS *Workflow Management System*

SUMÁRIO

1 INTRODUÇÃO	14
1.1 OBJETIVO GERAL.....	15
1.2 OBJETIVOS ESPECÍFICOS	16
1.3 JUSTIFICATIVA	16
1.4 ESTRUTURA DO TRABALHO	18
2 ENGENHARIA DE SOFTWARE	21
2.1 MODELOS DE PROCESSOS DE SOFTWARE	22
2.2 MODELAGEM DE SISTEMAS.....	23
2.3 CICLO DE VIDA DE SOFTWARE.....	24
2.4 IMPORTÂNCIA DA ENGENHARIA DE SOFTWARE NA FASE INICIAL DO PROJETO	26
3 WORKFLOW – FLUXO DE TRABALHO	28
3.1 SISTEMAS DE GERENCIAMENTO WORKFLOW	29
3.2 TIPOS DE WORKFLOW	30
3.3 MODELAGEM WORKFLOW.....	31
4 PROCESSO UNIFICADO	33
4.1 FASES DO PROCESSO UNIFICADO	34
4.1.1 FASE DE CONCEPÇÃO.....	35
4.1.2 PREPARAÇÃO	37
4.1.3 ATIVIDADES DE REQUISITOS.....	38
4.2 WORKFLOW DE REQUISITOS	38
4.2.1 REQUISITOS FUNCIONAIS	39
4.2.2 REQUISITOS NÃO FUNCIONAIS	40

4.2.3 ARTEFATOS.....	40
4.2.4 TRABALHADORES.....	41
4.2.5 ATIVIDADES.....	41
5 REDES DE PETRI.....	43
5.1 DEFINIÇÕES BÁSICAS.....	43
5.2 ESTRUTURA DAS REDES DE PETRI.....	44
5.3 COMPORTAMENTO DINÂMICO DAS REDES DE PETRI.....	46
5.4 PROPRIEDADES DAS REDES DE PETRI.....	47
5.4.1 PROPRIEDADES COMPORTAMENTAIS.....	48
5.4.2 PROPRIEDADES ESTRUTURAIS.....	49
5.5 WORKFLOW-NET.....	50
6 TRABALHOS CORRELATOS.....	51
6.1 CICLO DE VIDA DE SISTEMAS BASEADO EM REDES DE PETRI.....	51
6.2 ESTUDO E ANÁLISE DE MODELAGEM DE SISTEMAS DE WORKFLOW BASEADOS EM REDES DE PETRI.....	51
6.3 O POTENCIAL DAS REDES DE PETRI EM MODELAGEM E ANÁLISE DE PROCESSOS DE NEGÓCIO.....	52
7 A UTILIZAÇÃO DAS REDES DE PETRI NA FASE DE CONCEPÇÃO DO PROJETO.....	53
7.1 REDES DE PETRI NO CONTEXTO DE WORKFLOW.....	53
7.2 APLICAÇÃO DAS REDES DE PETRI NA MODELAGEM DE SISTEMAS GERENCIAS WORKFLOW.....	54
7.3 MODELAGEM DO PROJETO UTILIZANDO REDES DE PETRI.....	55
7.4 METODOLOGIA.....	59
7.4.1 REALIZAR LEVANTAMENTO BIBLIOGRÁFICO.....	59

7.4.2 RELATAR CONCEITOS RELEVANTES SOBRE ENGENHARIA DE SOFTWARE	59
7.4.3 EFETUAR ESTUDO VOLTADO À SISTEMAS DE GERENCIAMENTO WORKFLOW, REDES DE PETRI E PROCESSO UNIFICADO	60
7.4.4 COMPREENSÃO DA FASE DE CONCEPÇÃO DO PROJETO	60
7.4.5 ENTENDIMENTO DO WORKFLOW DE REQUISITOS DA FASE DE CONCEPÇÃO DO PROJETO	60
7.4.6 REPRESENTAÇÃO DAS REDES DE PETRI COMO FERRAMENTA DE MODELAGEM GRÁFICA	61
7.4.7 REALIZAÇÃO DE ESTUDO DO SOFTWARE DE APOIO AO ENSINO COM O USO DE COMPUTAÇÃO GRÁFICA	61
7.4.8 APLICAÇÃO DAS REDES DE PETRI NA FASE INICIAL DO PROJETO..	61
7.4.9 DEMONSTRAÇÃO DE UMA MODELAGEM EM REDES DE PETRI DIRECIONADA À FASE DE CONCEPÇÃO DO PROJETO.....	62
7.5 RESULTADOS OBTIDOS	62
CONCLUSÃO.....	64
REFERÊNCIAS	66
APÊNDICE A – PROJETO EXEMPLO PARA UTILIZAÇÃO DAS REDES DE PETRI	68
APÊNDICE B – TEORIA DOS GRAFOS.....	72
ANEXO A – FOTO COM JEAN MARIE FARINES NA UFSC.....	77

1 INTRODUÇÃO

Hoje em dia as organizações precisam de sistemas para acompanhar as constantes mudanças no ambiente empresarial. Os desenvolvedores de *software* necessitam ter uma compreensão maior sobre a organização, suas metas, objetivos e estratégias de mercado para garantir que os sistemas cumpram com seus objetivos. Entretanto, a falta de modelagem específica dificulta o trabalho destes profissionais (PRESSMAN, 2006).

Utiliza-se redes de Petri como ferramenta gráfica para modelar e analisar diversos tipos de sistemas. Sua estrutura define um grafo bipartido, dirigido e conexo, possuindo quatro elementos: estado, marca, ação e relação de fluxo. Elas também podem servir como método para descrever e analisar o comportamento do fluxo de informações e controle de sistemas gerenciais. Sendo assim, pode-se usá-la na modelagem de sistemas de gerenciamento de fluxo de trabalho, ou seja, de *workflow* (MARRANGHELLO, 2005a).

A tecnologia de gerenciamento *workflow* não oferece somente ferramentas para modelar, mas também para executar e gerenciar o seu fluxo de trabalho que funciona por meio de suporte de um *software*, cuja ordem de atividades é conduzida por uma representação lógica de um *workflow* no computador. Entretanto, estes sistemas têm falta de definição formal no que se refere à sintaxe, à semântica e funcionalidades de tais técnicas dificultando análises mais complexas dos modelos. A falta de flexibilidade nestes sistemas dificulta a realização da sua modelagem (BAEYENS, 2004).

Pretende-se modelar o fluxo de trabalho na fase de concepção do processo unificado. Este processo consiste na repetição de uma série de ciclos durante a vida de um sistema. Cada ciclo possui quatro fases: concepção, elaboração, construção e transição. Cada fase é subdividida em iterações que passam pelos cinco fluxos de trabalho (*workflow*) do processo: requisitos, análise, projeto, implementação e teste (SCOTT, 2003).

Na fase de concepção, o foco está na captação de requisitos tendo menos ênfase em análise e projeto, portanto nesta fase utiliza-se o *workflow* de requisitos. Durante o fluxo de trabalho, as tarefas do sistema de requisitos são especificadas por meio da identificação das necessidades de usuários e clientes. Geralmente utilizam-se ferramentas de caso de uso em *Unified Modeling Language* (UML) para desenhar estes requisitos, que podem ser funcionais ou não funcionais, de sistemas gerenciais financeiros, ou seja, referente à administração no ambiente organizacional.

Neste trabalho faz-se a utilização das redes de Petri, por ser de natureza uma modelagem flexível do tipo predicado/transição, como ferramenta para obter o desenho gráfico da fase de concepção do projeto unificado e seu *workflow*, visando suprir as necessidades dos desenvolvedores de *software* na fase inicial do projeto, sendo estas a análise dos requisitos e suas atividades.

Apresentam-se também alguns conceitos relevantes sobre Engenharia de *Software* para identificar qual a sua verdadeira função no início de um projeto de *software*.

Utiliza-se o projeto “*Software* de Apoio à Aprendizagem com o Uso de Computação Gráfica”, que está relacionado com educação matemática, em específico, geometria, para demonstrar sua modelagem e análise por meio das redes de Petri na fase inicial do sistema, juntamente com seu fluxo de trabalho de atividades, possibilitando maior visualização e descrição dos seus requisitos com a aplicação das propriedades na qual a ferramenta proporciona.

1.1 OBJETIVO GERAL

Utilizar a estrutura das redes de Petri nos sistemas de gerenciamento *workflow*.

1.2 OBJETIVOS ESPECIFICOS

- a) Relatar conceitos relevantes sobre Engenharia de *Software*;
- b) Compreender os sistemas de gerenciamento *workflow*;
- c) Realizar estudo da modelagem gráfica das redes de Petri e de suas propriedades;
- d) Aplicar as redes de Petri como ferramenta gráfica para modelar e analisar os sistemas de gerenciamento *workflow*;
- e) Entender o projeto unificado, a fase de concepção e *workflow* de requisitos;
- f) Optar por um sistema computacional para exemplificá-lo em redes de Petri;
- g) Proporcionar aos desenvolvedores de *software* a modelagem gráfica utilizando redes de Petri na fase de concepção do projeto.

1.3 JUSTIFICATIVA

Atualmente a Engenharia de *Software* realiza um importante papel na viabilização de produtos de *software* por meio da utilização de métodos, procedimentos e ferramentas. Ela se ocupa de todos os aspectos na geração do *software*, desde os estágios iniciais da concepção do produto até a manutenção do mesmo, após entrar em operação. Abrange também uma área do conhecimento voltada para a especificação, desenvolvimento e manutenção de sistemas de *software* aplicando tecnologias e práticas de ciência da computação, gerência de projetos em outras áreas, objetivando produtividade e qualidade (SOMMERVILLE, 2003).

Os sistemas de gerenciamento *workflow* fazem parte da gerência de processos de *software*. Este processo é um conjunto coerente de práticas que objetiva o desenvolvimento de sistemas de *software*. Estas práticas englobam as atividades de requisitos, análise, projeto,

implementação, testes e caracterizam-se pela interação de ferramentas, pessoas e métodos. Portanto, estes sistemas gerenciam todo o fluxo de trabalho do projeto e são de grande complexidade, dificultando a realização de uma modelagem padrão e específica para tal gerência (ARAÚJO; BORGES 2001).

Faz-se necessário realizar um estudo aprimorado por uma ferramenta que dê a condição de desenhar a modelagem de um sistema gerencial de fluxo de informação. Uma ferramenta que possui modelagem apta para suprir tal necessidade chama-se redes de Petri. Portanto, é importante aprender sobre esta ferramenta para poder explorar todo o seu potencial em modelagem gráfica.

Uma vantagem das redes de Petri é a sua versatilidade tornando-a apropriada para representar sistemas de gerência de processos. Esta ferramenta serve para detectar situações relacionadas ao mau funcionamento de um sistema, neste caso de *workflow*. A aplicação desta ferramenta nos sistemas de gerenciamento de fluxo de trabalho será relevante por apresentar uma modelagem gráfica bem definida e ser de modo interativo.

O início de um *software* por meio do projeto unificado não possui uma ferramenta específica para realizar a modelagem dos seus processos, contudo utiliza-se a UML como ferramenta padrão. Portanto, precisa-se entender a sua fase de concepção e o seu fluxo de trabalho que recebe a maior parte de atenção nesta fase, no caso *workflow* de requisitos, para se fazer a modelagem de um sistema computacional específico com o auxílio de uma ferramenta capaz de preencher tais necessidades.

O benefício de proporcionar a modelagem gráfica no início do projeto de *software* torna-se importante para os seus desenvolvedores, pois irá servir de pesquisa nas fases futuras do projeto. Portanto, os engenheiros de *software* se preocuparam menos em relembrar o esboço feito na fase de concepção do projeto unificado, por ter como objeto de pesquisa a modelagem mencionada, feita por meio de uma ferramenta, no caso redes de Petri.

Optou-se por um *software* computacional já desenvolvido por causa da falta de tempo hábil para se realizar a análise dos requisitos funcionais e não funcionais de sistema, modelagem e verificação por meio das redes de Petri, programação em ambiente de desenvolvimento específico e por não se tratar do objetivo desta pesquisa fazer todas as etapas de desenvolvimento do projeto.

1.4 ESTRUTURA DO TRABALHO

A pesquisa está dividida em sete capítulos, iniciando em introdução, na seqüência Engenharia de *Software*, *workflow*, processo unificado, redes de Petri, trabalhos correlatos, a utilização das redes de Petri na fase de concepção do projeto, finalizando com conclusão, apêndices e referências.

O primeiro capítulo retrata a introdução do projeto, definindo o problema, descrevendo o que a pesquisa pretende resolver e a possível solução para tal. Ele possui os subitens, objetivo geral que consiste no que se deve atingir com o desenvolvimento da pesquisa, objetivos específicos que detalham o objetivo geral refletindo o que se espera do trabalho em relação aos aspectos computacionais e ao domínio de aplicação, justificativa que descreve os motivos que determinaram a escolha do tema da pesquisa, o porquê, vantagens e benefícios, relevância computacional e de domínio de aplicação e por fim a estrutura do trabalho que descreve os capítulos e faz breve explicação do conteúdo de cada um.

O capítulo dois representa a grande área da pesquisa, a Engenharia de *Software*, possuindo os subitens modelos de processos de *software*, modelagem de sistemas, ciclo de vida do *software* e importância da Engenharia de *Software* na fase inicial do projeto.

No capítulo seguinte trata-se sobre fluxo de trabalho (*workflow*), os sistemas de gerenciamento *workflow*, seus tipos e a modelagem destes, conceitos e explicações sobre o seu funcionamento.

O quarto capítulo representa o processo unificado, abordando suas fases, dando ênfase para a fase de concepção do projeto e seu *workflow* de requisitos, toda a preparação, as atividades de requisitos, os requisitos funcionais e não funcionais, artefatos, trabalhadores e suas tarefas.

No capítulo seguinte aborda-se as redes de Petri, seus conceitos, elementos, seu funcionamento, comportamento dinâmico, suas propriedades comportamentais e estruturais, sua estrutura e a modelagem *workflow-net*.

O capítulo seis retrata três trabalhos correlatos a esta pesquisa, sendo estes o ciclo de vida de sistemas baseado em redes de Petri, estudo e análise de modelagem de sistemas *workflow* baseados em redes de Petri e o potencial das redes de Petri em modelagem e análise de processos de negócio.

O sétimo capítulo representa o trabalho desenvolvido, ou seja, a utilização das redes de Petri na fase de concepção do projeto, as redes de Petri no contexto de *workflow*, aplicação das redes de Petri na modelagem de sistemas gerenciais *workflow* e aplicação das redes de Petri na modelagem e análise de um projeto.

A metodologia aborda os subitens: realização de pesquisa bibliográfica, relatar conceitos relevantes sobre Engenharia de *Software*, estudo direcionado à sistemas de gerenciamento *workflow*, redes de Petri e processo unificado, compreensão da fase de concepção do processo unificado, entendimento do *workflow* de requisitos da fase de concepção do projeto, representação das redes de Petri como ferramenta de modelagem gráfica, aplicação das redes de Petri na fase de concepção do projeto, realização de estudo do

“*Software* de Apoio ao Ensino com o Uso de Computação Gráfica”, demonstração de uma modelagem em redes de Petri voltada à fase de concepção do projeto.

Logo após as etapas metodológicas utilizadas no desenvolvimento da pesquisa está o sub-capítulo “resultados obtidos”. O próximo item aborda a conclusão do trabalho, as referências bibliográficas, apêndices e anexo.

2 ENGENHARIA DE SOFTWARE

Engenharia de *Software* é uma área do conhecimento voltada para a especificação, desenvolvimento e manutenção de sistemas de *software* aplicando tecnologias e práticas de ciência da computação, gerência de projetos e outras disciplinas, objetivando produtividade e qualidade (SOMMERVILLE, 2003).

A Engenharia de *Software* proporciona ao engenheiro o controle sobre os processos de desenvolvimento de *software*, oferecendo a oportunidade de criação de sistemas de qualidade, fornecendo as bases nas quais os programas devem ser desenvolvidos, ou seja, a área que atua na criação de *software*, três elementos são indispensáveis, sendo estes, métodos, ferramentas e processos (INTHURN, 2001).

Os métodos fornecem a técnica de como construir *softwares*. Eles compreendem um vasto grupo de tarefas que incluem comunicação, análise de requisitos, modelagem de projeto, criação de programas, testes e manutenção. Os métodos de Engenharia de *Software* flutuam num conjunto de princípios básicos que administram cada área da tecnologia, fazem inserção de atividades de modelagem e outras técnicas descritivas (PRESSMAN, 2006).

As ferramentas de Engenharia de *Software* procuram auxílio para o processo e para os métodos usando apoio automatizado ou semi-automatizado. Há ocorrência de ferramentas que possibilitam esses suportes aos métodos separadamente, mas quando elas são unidas de maneira que o resultado de uma possa ser utilizado em outra, então se alcança o seu objetivo principal, um sistema de suporte para a criação de *software* (SOMMERVILLE, 2003).

Processo de *software* é um conjunto coerente de práticas que objetiva o desenvolvimento ou evolução de sistemas de *software*. Estas práticas englobam as atividades

de especificação, projeto, implementação, testes e caracterizam-se pela interação de ferramentas, pessoas e métodos (PRESSMAN, 2006).

2.1 MODELOS DE PROCESSO DE SOFTWARE

O modelo de processo de *software* é a representação abstrata do mesmo. Cada desenho figura um processo a partir de uma perspectiva privada, de modo que proporcione apenas informações parciais sobre o procedimento. Existe uma série de modelos de processo muito genéricos que não possuem descrições definitivas, mas são usuais para explicar diferentes temas quanto ao desenvolvimento de sistemas. A importância destes modelos está na estabilidade, no controle e na organização a uma atividade, entretanto, cada modelo de processo deve ser adaptado para que seja utilizado permanentemente em um projeto de *software* específico (SOMMERVILLE, 2003).

Há vários modelos prescritivos de processo de *software*, pois prescrevem um grupo de elementos de procedimentos sendo atividades de arcabouço, ações de Engenharia de *Software*, tarefas, produtos de trabalho, que são os programas, documentos e dados produzidos em consequência das atividades e tarefas definidas pelo processo, e mecanismos de segurança de qualidade e de controle de modificações para cada projeto. O modelo de processo também prescreve um fluxo de trabalho (*workflow*), isto é, o modo com que os elementos do processo se interagem em uma relação de uns com os outros (PRESSMAN, 2006).

Os engenheiros de *software* e seus gerentes adaptam um modelo de processo prescritivo a suas necessidades e depois a seguem. Além disso, o pessoal que solicitou o projeto também participa à medida que o modelo de processo é seguido (INTHURN, 2001).

Os modelos de processo de *software* mais conhecidos são modelo em cascata,

modelos incrementais de processo, modelos evolucionários de processo, modelo espiral, modelo de desenvolvimento concorrente, desenvolvimento orientado a reuso e processo unificado que desenvolve o projeto por meio de iteração de processo (PRESSMAN, 2006; SOMMERVILLE, 2003).

2.2 MODELAGEM DE SISTEMAS

A modelagem de sistemas é um elemento importante do processo de Engenharia de *Software*. O seu foco está na visão de um todo ou detalhada, portanto o engenheiro cria modelos que definem os processos, representam o seu comportamento e todos os relacionamentos que permitirão ao profissional da área entender melhor a visão. Na construção de um modelo de sistema, o engenheiro deve considerar os seguintes fatores de restrição (SOMMERVILLE, 2003):

- a) pressupostos: que reduzem a quantidade de permutações e variações possíveis, permitindo assim a um modelo refletir o problema de modo razoável;
- b) simplificações: que habilitam o modelo na criação de um prazo adequado;
- c) limitações: que auxiliam a delimitar o *software*;
- d) restrições: que guiarão a maneira pelo qual o modelo é criado e o método adotado quando o modelo está sendo implementado;
- e) preferências: que indicam a arquitetura pretendida para todos os dados, funções e tecnologia.

O resultado deste desenho adota ou uma solução completamente automatizada, ou semi-automatizada, ou não automatizada. Pode-se caracterizar um modelo de cada tipo, fazendo com que sirvam como soluções alternativas para o problema. O engenheiro de

software modifica a influência relativa de diferentes componentes do sistema (*hardware*, *software*, pessoal) para mudar os modelos de cada tipo (INTHURN, 2001).

Os requisitos de sistema e atividade de projeto precisam ser modelados como um conjunto de componentes e de relações entre estes, isto é, geralmente, demonstrado graficamente em um modelo de arquitetura de sistema, que proporciona uma visão geral da organização do projeto (PRESSMAN, 2006).

O *software* em desenvolvimento pode ser representado em diferentes níveis de escolha, os modelos de sistemas tendem a ser de natureza hierárquica ou em camadas. No topo da hierarquia, um modelo completo do sistema é apresentado sendo este a visão de mundo do projeto. Os principais objetos de dados, funções de processamento e comportamentos são representados sem considerar os componentes do sistema que programarão os elementos do modelo da perspectiva global. À medida que a hierarquia é refinada em camadas, detalhes dos componentes são modelados. Após isso, os modelos de sistema evoluem para desenhos de Engenharia de *Software* (SOMMERVILLE, 2003).

2.3 CICLO DE VIDA DE SOFTWARE

O processo de desenvolvimento de *software* elaborado com eficácia acarreta uma boa qualidade do produto final, para a realização de uma estrutura de desenvolvimento satisfatória é preciso seguir as etapas de criação de *software* conforme a Figura 1 (INTHURN, 2001).

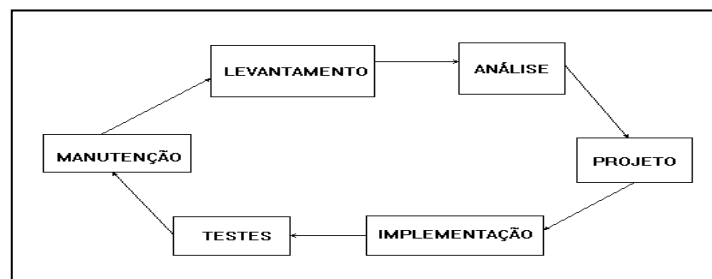


Figura 1. Ciclo de desenvolvimento de *software*
Fonte: INTHURN, C. (2001).

O ciclo de vida de um *software* possui variações tanto num ciclo longo quanto num curto, para que um sistema tenha tempo de vida longo faz-se a necessidade de uma boa estruturação das fases que estão presentes em seu ciclo. No Processo Unificado, cada ciclo possui quatro fases, sendo que uma fase é o tempo percorrido entre dois marcos principais, no qual desenvolvedores tomam decisões sobre ir adiante com o projeto ou não. As fases do Processo Unificado são (SCOTT, 2003):

- a) concepção: abrange toda a etapa de criação do *software*, sua modelagem e análise de requisitos;
- b) elaboração: estabelece a capacidade para a construção do novo sistema, dadas as restrições financeiras e outras que o desenvolvedor se defronta;
- c) construção: é onde o projeto terá atingido o marco de capacidade operacional inicial, ou seja, na construção o sistema deve ser capaz de operar em ambientes organizacionais;
- d) transição: se concentra em corrigir defeitos e modificar o sistema afim de eliminar problemas não identificados previamente, nesta fase entrega-se o sistema em funcionamento para o usuário.

A Figura 2 demonstra as fases e os principais marcos do processo unificado da *Rational Unified Process* (RUP) que representa um ciclo de vida, de desenvolvimento de *software*, incremental e evolutivo (FALBO, 2000).

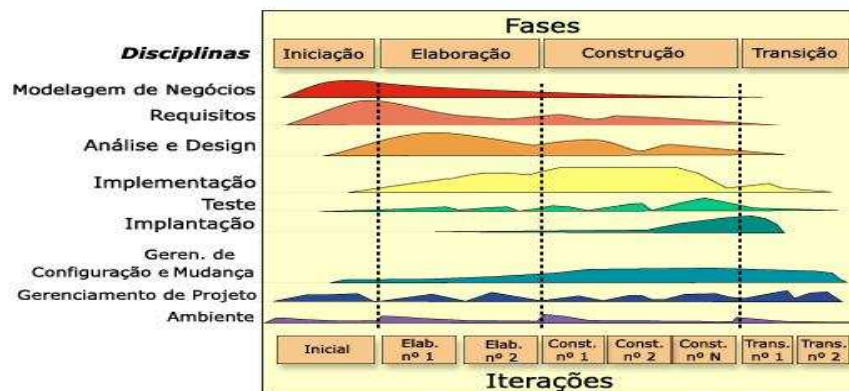


Figura 2. Ciclo de vida do Processo Unificado
Fonte: FALBO, R. (2000).

Na Figura 2 pode-se ver a ênfase de cada disciplina nas fases do processo unificado e de suas iterações. Portanto, observa-se que no início do projeto o enfoque está direcionado para a modelagem em questão, desenho, arquitetura, análise dos requisitos e também os princípios de gerenciamento de projeto e ambiente.

2.4 IMPORTÂNCIA DA ENGENHARIA DE SOFTWARE NA FASE INICIAL DO PROJETO

É relevante o estudo de Engenharia de *Software* no começo de um projeto de sistema, pois há necessidade de profundo conhecimento dos seus principais conceitos para obter-se um *software* eficaz e com qualidade, para isso a fase de concepção do projeto de *software* torna-se fundamental.

Nesta fase precisa-se compreender o produto de trabalho com o intuito de estabelecer uma visão global do projeto, englobando tanto a comunicação com o cliente, quanto suas atividades de planejamento, enfatizando o desenvolvimento e refinando a modelagem como peça principal.

Na Figura 3 pode-se ver o destaque de cada fase do processo unificado, ou seja, a importância principal de cada uma delas. Nota-se que na fase de concepção o enfoque crucial está na atividade de levantamento dos requisitos (FALBO, 2000).

	Levantamento de Requisitos	Análise	Projeto	Implementação	Testes
Concepção					
Elaboração					
Construção					
Transição					

Figura 3: Foco de cada fase do Processo Unificado
Fonte: FALBO, R. (2000).

A tarefa de levantamento e análise de requisitos é o estágio onde o grupo técnico do desenvolvimento de *software* trabalha junto com o usuário final do sistema para descobrirem maiores informações sobre o domínio da aplicação por meio do estudo do seu fluxo de trabalho (*workflow*), quais serviços o projeto deve fornecer, desempenho exigido e as suas restrições de *hardware* (SOMMERVILLE, 2003).

Para cada fase do projeto existe um ou mais *workflows* específicos. Na fase de concepção utiliza-se o *workflow* de requisitos e muito pouco de análise, já na fase de elaboração são importantes os dois fluxos de trabalho. Na fase de construção o enfoque está nos *workflows* de projeto e implementação e na fase de transição o fluxo de trabalho predominante é o de testes. Sabendo disso, o estudo de *workflow* torna-se relevante para a pesquisa.

3 WORKFLOW – FLUXO DE TRABALHO

Workflow é definido pelo *Workflow Management Coalition* (WfMC) como a automação total ou parcial de um processo de negócio, durante a qual documentos, informações e tarefas são passadas entre os participantes do processo. A Figura 4 mostra o modelo de referência usado pelos desenvolvedores *workflow*, este modelo propõe um sistema formado por seis componentes (WfMC, 2006).

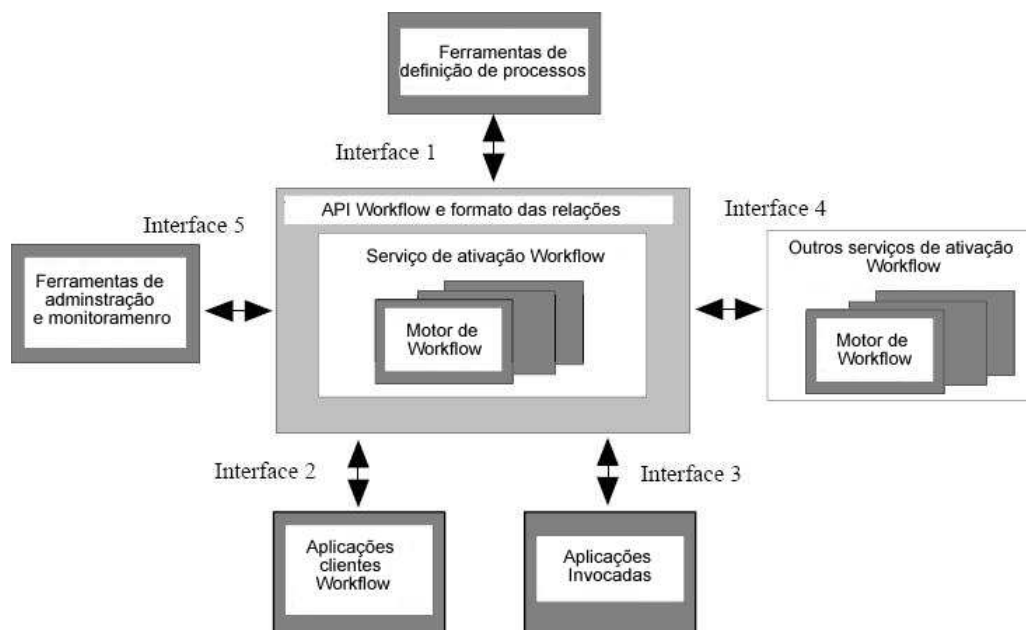


Figura 4. Modelo de referência da WfMC
Fonte: WfMC(2006).

O termo *workflow* é utilizado na literatura representando processos de negócio. Também chamado de fluxo de trabalho, pode ser definido como um conjunto de atividades que podem ou não ser executadas simultaneamente, com alguma especificação de controle e fluxo de dados entre as atividades. Sendo assim, podendo representar vários processos e seus relacionamentos (PÁDUA, 2003).

Os sistemas de *workflow* consistem de dois componentes: modelagem e representação do sistema. Na área de modelagem não é possível ainda descrever um conjunto de modelos e técnicas que formalizem seu desenvolvimento (LEYMANN; ROLLER, 1997).

A idéia de *workflow* não é nova, nem está associada especificamente para área de tecnologia da informação. Pelo contrário, no passado até hoje, o processo é semelhante: supervisores designam trabalhos, provavelmente baseado em treinamentos, habilidades e experiência, para as pessoas (PÁDUA, 2003).

A tecnologia *workflow* é a automação e gerência de processos, ou seja, um sistema onde uma tarefa pode ser transferida automaticamente de uma pessoa para outra, por meio da rede de computadores, de acordo com um grupo de regras definidas.

Os processos são representados por meio de fluxos de trabalho, ou seja, modelos que especificam as atividades que são compostas por ordem, condições que as tarefas devem ser executadas e as ferramentas a serem usadas durante a sua execução. As representações de processos na forma de fluxos de trabalho podem ser automatizadas por meio de sistemas de gerenciamento de *workflow* (ARAÚJO; BORGES, 2001).

3.1 SISTEMAS DE GERENCIAMENTO WORKFLOW

Os Sistemas de Gerência de *Workflow*, *Workflow Management System* (WFMS), dispõe de um conjunto de ferramentas de *software* para apoio a definição e execução de *workflows*. Os WFMS surgiram primeiro na indústria e só depois se tornaram alvos de pesquisa. Em função disto, podemos observar um grande número de produtos nessa área que resolvem problemas concretos, mas que, em muitas situações, não apresentam uma base científica forte que permita resolver certas limitações. Além disso, existem diversos tipos de WFMS para vários tipos de negócio (WfMC, 2006).

Um WFMS é um sistema que define e organiza a execução de fluxo de trabalho pela utilização de *software*, sendo capaz de interpretar a definição de um processo, interagir

com os participantes e invocar o uso de ferramentas e aplicações quando necessário (CRUZ, 2000).

Pode-se observar que este sistema desempenha duas tarefas distintas relacionadas entre si. A primeira tarefa é a especificação do *workflow*. Nesta tarefa é produzida a sua definição representando o processo de negócio. A outra tarefa é a animação ou execução do mesmo.

O *workflow* não se limita ao roteamento de documentos em empresas, pois atualmente torna-se uma ferramenta para solução de apoio automatizado às atividades de comércio eletrônico e outras formas de negócio em grande escala (PÁDUA, 2003).

Os sistemas de *workflow* propõem uma associação entre máquinas, sistemas computacionais e redes de telecomunicações, que podem controlar e interligar empresas permitindo que trabalhos manuais sejam transferidos por esses meios. Portanto, a tecnologia de fluxo de trabalho torna-se uma relevante oportunidade para melhoria de processos gerenciais dentro de uma organização (BAEYENS, 2004).

3.2 TIPOS DE WORKFLOW

O crescimento dos WFMS proporcionou aos desenvolvedores a criação de diversos sistemas, com isso houve a necessidade de obter uma classificação dos existentes. A classificação dos sistemas de *workflow* é definida quanto à abordagem de comunicação e ao grau de estruturação de processos (BAEYENS, 2004).

A abordagem de comunicação está relacionada com a sua ocorrência entre os processos. Os WFMS quanto a esta abordagem compreendem o uso de mensagens, os sistemas que utilizam esta classificação realizam a tarefa de enviar mensagens que contenham documentos ou formulários requisitados (ARAÚJO; BORGES, 2001).

Os processos estruturados podem ser considerados aqueles cujas atividades podem ser definidas *a priori*. Para essa classificação nomeiam-se três categorias: *ad-hoc*, administrativos e de produção (CRUZ, 2000).

- a) *ad-hoc*: estes sistemas possuem a capacidade de automatizar processos não estruturados, geralmente não possuem capacidades de segurança e tratamento de um grande volume de dados, por isso não são recomendados para automação de processos críticos;
- b) administrativos: estes sistemas tem capacidade de automatizar processos com um maior grau de estruturação em relação aos sistemas *ad-hoc*, pois envolve processos repetitivos com regras de coordenação de atividades simples e também existe uma maior previsão no encadeamento das tarefas;
- c) produção: ao contrário das outras classificações, os sistemas de produção possuem alto grau de estruturação por ter tarefas bem definidas, englobam um processamento de dados complexos envolvendo muitos sistemas, por isso a automatização se torna mais complexa.

3.3 MODELAGEM WORKFLOW

A modelagem *workflow* é complexa e não flexível. Ela consiste na descrição do processo por meio de desenhos que possam representar as suas principais características, além de contribuir para o registro de processos, também incentiva o raciocínio lógico sobre o melhor modo de reorganizá-lo nesta fase.

Considera-se alguns critérios na modelagem de sistemas *workflow* (NICOLAO, 1996):

- a) formalismo apresentado pelo modelo, ou seja, apresentação formal do desenho

- dos processos, o que assegura a consistência dos dados e garantia no processo;
- b) formalismo de tarefas automáticas, ou seja, representar as atividades automaticamente e não precisar da intervenção manual;
 - c) representação formal das atividades, sendo que permite-se o monitoramento do processo de fluxo de trabalho com relação as características que habilitam o término e andamento das tarefas;
 - d) representação da estrutura do fluxo, definindo as dependências das atividades envolvendo a sucessão de caminhos alternativos;
 - e) formalismo do tratamento de exceções, garantir a especificação de quais ações tomar caso uma tarefa falhe ou que um *workflow* fique incompleto.

Portanto, para assegurar que um modelo permita identificar e corrigir problemas eventuais da implementação necessita-se que o método de modelagem escolhido consiga representar todas as características apresentadas. O projeto unificado se identifica com esse tipo de abordagem, pois trata de metodologia iterativa e de atribuição de fluxo de trabalho para cada uma de suas fases.

4 PROCESSO UNIFICADO

A idéia de processo unificado teve início no trabalho de Ivar Jacobson e seus colegas na década de 1960. Eles modelaram um enorme sistema de telecomunicações utilizando camadas de componentes, onde as camadas inferiores serviam como alicerces para subsistemas nas camadas superiores. Anos depois, na década de 1990, eles começaram a trabalhar em um método unificado que combinaria as melhores características de cada método individual e adotaria características adicionais indicadas por outros especialistas no ramo de orientação à objetos, esse resultado foi a UML (PRESSMAN, 2006; SCOTT, 2003).

O processo unificado é praticamente uma definição generalizada de processo. Entende-se por processo unificado um conjunto de atividades executadas e necessárias para transformar requisitos do usuário em um sistema de *software*. Ele é baseado em componentes, isto significa o sistema ser construído a partir de componentes de *software* interconectados por meio de interfaces bem definidas (SCOTT, 2003).

O processo unificado apóia-se nos bons recursos e características dos modelos convencionais de processo de *software*. Ele reconhece a importância da comunicação com o cliente e dos métodos imediatos para descrever a visão do usuário para com o sistema. Ele proporciona um fluxo de processo iterativo e incremental, dando a sensação evolucionária que é essencial na atualidade para o desenvolvimento de *software* (SOMMERVILLE, 2003).

A diferença do processo unificado para outros tipos de processo tais como processo em cascata e processos ágeis, está no modo em que os desenvolvedores seguem os passos de um projeto. Por exemplo, no método em cascata os engenheiros de *software* utilizam a ordem respectiva, primeiramente captam os requisitos, os analisam, abordam uma solução, arquitetam um esboço do *software*, fazem a sua implementação, testam, implantam e o mantêm. Já os processos de desenvolvimento ágeis existem inúmeros métodos para o seu

crescimento, seus períodos de tempo são medidos em semanas, ao contrário do método iterativo que é dividido em meses, e o progresso é efetuado de maneira colaborativa (PRESSMAN, 2006).

A essência dos processos iterativos está no desenvolvimento concorrente da especificação e do *software*. Contudo, pode-se entrar em conflito quando a especificação completa do sistema faz parte do contrato para desenvolvimento do sistema, isto requer um novo tipo de contrato difícil de ser aceito, por exemplo, para órgãos do governo (SOMMERVILLE, 2003).

4.1 FASES DO PROCESSO UNIFICADO

Um sistema de *software* possui sua vida normalmente representada por ciclos. A liberação de uma versão do sistema para o usuário acontece após o término de cada ciclo. No processo unificado cada ciclo possui quatro fases: concepção, elaboração, construção e transição (SCOTT, 2003).

Na fase de elaboração o objetivo principal é estabelecer a capacidade para a construção do novo projeto e os tipos de restrições com que o desenvolvimento do projeto se defronta. Nesta fase realiza-se a expansão da representação arquitetural para incluir visões diferentes do *software* e o aperfeiçoamento da modelagem que foi desenvolvida na fase de concepção (PRESSMAN, 2006).

A fase de construção do projeto tem o objetivo de construir um sistema que desempenhe bem os seus deveres em ambientes teste de clientes. Nesta fase faz-se uso da modelagem arquitetural para desenvolver ou adquirir os componentes de *software* que vão tornar a ferramenta operacional para o usuário final. Devido a este fato, o principal marco desta fase é chamado de capacidade operacional inicial (SOMMERVILLE, 2003).

A última fase do processo unificado chama-se transição, no qual seu objetivo crucial é entregar o sistema em total funcionamento para o cliente. Nesta fase busca-se modificar o sistema conforme relatório de erros apontados pelo usuário efetuando a correção de defeitos previamente não identificados anteriormente (PRESSMAN, 2006).

Neste trabalho o foco está voltado para a fase de concepção e suas atividades de requisitos, os quais serão descritos detalhadamente nos subcapítulos seguintes.

4.1.1 Fase de Concepção

A fase inicial do processo unificado é chamada de concepção. O grande objetivo desta fase é estabelecer a viabilidade do sistema proposto, ou seja, realizar um estudo da qualidade viável deste pré-sistema e deve-se também justificar o comprometimento com o projeto de desenvolvimento, cujo andamento posterior deseja-se argumentar. As atividades executadas nesta fase para alcançar esse objetivo atendem (SCOTT, 2003):

- a) definir o escopo do sistema: todos devem chegar a um acordo sobre o conteúdo do sistema e como ele interagirá com seu ambiente externo;
- b) desenhar a arquitetura pretendida: adotar uma arquitetura como base viável sobre a qual o sistema será construído;
- c) explorar os requisitos funcionais e não funcionais de alto nível: fazer a modelagem para captura dos requisitos mais importantes, que ajudarão no direcionamento do desenvolvimento;
- d) identificar os riscos principais e determinar quando e como o projeto os abordará, ou seja, aqueles que prejudicam a viabilidade do projeto e especificar como o grupo se comportará perante eles;

- e) iniciar uma análise do projeto, indicando se ele é viável, apontando as estimativas de custos, esforço, tempo e qualidade do produto. Especificar valores, demonstrando se há ganhos na produtividade e se existe razões para continuação do projeto no ponto de vista econômico.

A fase de concepção abrange as tarefas de comunicação e planejamento. Em cooperação com o cliente e usuários finais, as atividades de requisitos para o sistema são identificadas, um esboço da modelagem do projeto é proposto e um planejamento para a sua natureza iterativa e incremental é desenvolvido (SOMMERVILLE, 2003).

O principal ponto de partida relacionado com a fase de concepção é definido como objetivos do ciclo de vida. As designações de que o projeto o concluiu tornam-se evidente se os principais interessados concordam sobre o escopo do sistema proposto, a arquitetura candidata equaciona visivelmente um grupo de requisitos críticos de alto nível e se a análise econômica do projeto é concreta o suficiente para justificar a continuação do seu desenvolvimento.

Na concepção toda atenção está voltada para as atividades de requisitos, existem quatro tarefas fundamentais para realizar o capturamento dos mesmos. Estas sendo construir uma lista de características do sistema, os desenvolvedores devem concordar entre si sobre o contexto do projeto, capturar os requisitos funcionais e os não-funcionais na forma de modelagem específica (SCOTT, 2003).

Os requisitos para um projeto de *software* relatam o que o sistema deve fazer e definem suas restrições sobre sua operação e implementação. O termo requisito não é usado de modo consistente pelos desenvolvedores. Em alguns casos é visto como declaração abstrata, de uma função que o sistema deve fornecer ou de alto nível, no outro extremo, ele é uma definição detalhada formalmente de uma função do *software* (SOMMERVILLE, 2003).

4.1.2 Preparação

As tarefas de iniciação do projeto feitas pelo gerente antes da equipe de desenvolvimento começar com as atividades específicas de fluxo de trabalho compreendem em planejar a fase de concepção, expandir a visão do sistema e estabelecer os critérios de avaliação.

O objetivo de planejamento da fase de concepção é de fazer um esboço descrevendo como o trabalho acontecerá nesta fase e, em termos mais abstratos, nas outras fases. A sua organização deve incluir cronograma e orçamento para uma iteração, não podendo haver mais de duas iterações (SCOTT, 2003).

No início do projeto deve-se construir uma visão que todos entendam e concordem. Uma declaração de visão apropriada serve como base para o desenvolvimento da análise econômica do projeto. Esta idéia deve ser expressa também na análise de requisitos, pois é fundamental para o entendimento do grupo de desenvolvedores (SOMMERVILLE, 2003).

Os principais critérios de avaliação durante a fase de concepção podem ser classificados em cinco áreas, conforme a Tabela 1 (SCOTT, 2003).

Tabela 1. Critérios de avaliação

Área	Justificativa
Escopo	Todos concordar com o que está dentro e com o que está fora do sistema.
Requisitos de alto nível	Os requisitos devem ser não-ambíguos, livres de redundância e contradições.
Arquitetura candidata	Definir uma arquitetura satisfatória às necessidades prioritárias dos usuários.
Riscos críticos	Todos os riscos são identificados e especificados para o grupo saber como devem tratá-los.
Análise Econômica	É necessário satisfazer apenas um critério, ser atrativa o suficiente para o progresso do projeto.

Fonte: SCOTT, K. (2003).

4.1.3 Atividades de Requisitos

A construção de um modelo de domínio durante a fase inicial ajuda os desenvolvedores a chegarem num acordo sobre o contexto global do projeto, ou seja, o que está dentro e o que está fora dele. Na modelagem de negócio o objetivo é auxiliar na identificação dos principais atores e casos de uso e, também, a ter uma visão dos processos em fase de modelagem (SCOTT, 2003).

A descoberta dos casos de uso é a principal meta nesta fase, pois eles ajudarão o arquiteto a imaginar a arquitetura e os atores que efetuaram associações com os mesmos.

A próxima tarefa será priorizar os casos de uso sendo decidido qual deles necessitaram ser mais vistos durante esta fase. Estes normalmente serão os referidos às funcionalidades críticas do sistema e aos riscos mais significativos. Os casos de uso que ocupam esses critérios tornam-se o conteúdo inicial da descrição da arquitetura (SOMMERVILLE, 2003).

O detalhamento de um caso de uso faz parte da atividade final de requisitos dentro da fase inicial do projeto que consiste em escrever um texto para sua maior prioridade. Geralmente, o foco está em cursos de ação durante esta fase, sendo que o especificador de casos de uso realiza esta tarefa (SCOTT, 2003).

4.2 WORKFLOW DE REQUISITOS

Um fluxo de trabalho de Engenharia de *Software* é repartido para todas as fases do processo unificado. Neste contexto um *workflow* é análogo a um grupo de tarefas, ou seja, ele busca as atividades exigidas para efetuar uma ação relevante de Engenharia de *Software* e os produtos de trabalho feito em consequência da conclusão bem-sucedida dessas tarefas,

entretanto, nem todo fluxo de trabalho é conduzido para qualquer projeto de *software*, o processo é adaptado conforme as necessidades dos desenvolvedores (PRESSMAN, 2006).

As principais tarefas do fluxo de trabalho de requisitos visam na construção da modelagem em uma ferramenta gráfica que capture os requisitos funcionais do sistema que está sendo definido. Este modelo serve de auxílio aos desenvolvedores do projeto com o intuito de fazerem um acordo sobre as capacidades do sistema e as condições que ele deve satisfazer (SCOTT, 2003).

O levantamento e análise de requisitos compõem um processo complexo, uma das razões normalmente é que os desenvolvedores não sabem na realidade o que querem do sistema computacional, eles podem fazer pedidos irrealis por não saberem do custo de suas solicitações, outra questão, a mente de cada projetista é diferente, por isso eles precisam descobrir todos os requisitos e encontrar os pontos comuns e os conflitos do sistema (SOMMERVILLE, 2003).

A ação de trabalhar para construir o sistema certo envolve em capturar os requisitos dos interessados no projeto e entrar em negociação com eles, até que fique claro que a equipe de desenvolvimento concorda sobre as características do novo sistema, entretanto, dentro do grupo que está desenvolvendo o projeto podem existir pessoas com idéias diferentes em relação à este contexto (PRESSMAN, 2006).

4.2.1 Requisitos Funcionais

As declarações de funções que o sistema fornece, sendo estas, como elas devem reagir a entradas específicas e como tem que se comportar em determinadas situações chama-se requisitos funcionais. Em alguns casos, estes requisitos podem também dizer o que o *software* não deve fazer (SCOTT, 2003).

A especificação de requisitos funcionais de um sistema deve ser completa, pois isto significa que todas as funções requeridas pelo usuário devem estar definidas e consistentes, pois indica que os requisitos não possuem contradição em sua conceituação (SOMMERVILLE, 2003).

Os casos de uso quando usados com protótipos de interface do usuário, também servem como apoio para negociar requisitos com clientes, pois se utiliza a modelagem como mecanismo para reduzir a ambigüidade, assim eliminando suposições (SCOTT, 2003).

4.2.2 Requisitos Não Funcionais

Os requisitos não funcionais são restrições sobre os serviços oferecidos pelo sistema, sendo estas, restrições de tempo, sobre o desenvolvimento e entre outras. O seu surgimento dá-se conforme as necessidades do usuário. Eles não dizem respeito às funções específicas ditas pelo sistema, mas sim como um todo (SOMMERVILLE, 2003).

Os requisitos não funcionais estão associados a questões como desempenho, segurança, escalabilidade e confiabilidade. Os requisitos podem ser capturados dentro de um modelo de domínio, atribuindo valores permitidos a predicados em classes (SCOTT, 2003).

4.2.3 Artefatos

A modelagem de domínio considerada artefato efetua o capturamento das entidades e dos conceitos relevantes ao qual o problema pertence, assim há definição do que se deve solucionar dentro do projeto em construção (PRESSMAN, 2006).

O ator representa o papel que um usuário desempenha em relação ao *software* e também uma entidade, como, por exemplo, um banco de dados que está localizado fora do sistema (SOMMERVILLE, 2003).

O caso de uso é uma seqüência de ações que um ator executa junto com um sistema para atingir um objetivo particular. Ele descreve um aspecto de uso do *software* sem supor qualquer projeto ou implementação específica (PRESSMAN, 2006).

Um texto de caso de uso discrimina caminhos possíveis por meio do caso de uso e inclui as ações que o ator executa e as suas respectivas respostas perante o sistema. Estes caminhos são capturados como fluxos de eventos (SCOTT, 2003).

4.2.4 Trabalhadores

O analista de sistema tem como foco obter os requisitos de casos de uso, isso envolve direcionar esforços para construção da modelagem, achar os atores e casos de uso, assegurar a consistência da modelagem como um todo por completo (SOMMERVILLE, 2003).

O arquiteto prioriza o conjunto de casos de uso proposto e inclui os mais significativos na descrição da arquitetura, enquanto o projetista de interface determina a forma visual dos elementos da interface que um ou mais atores utilizarão (SCOTT, 2003).

4.2.5 Atividades

A construção da modelagem de domínio envolve entender e descrever as classes que representam o problema e também os seus inter-relacionamentos, em nível de abstração. Os analistas de sistema decidem quais classes e associações devem se tornar reais, neste

momento, os modeladores tomam decisões sobre os relacionamentos de generalização e agregados entre as classes (PRESSMAN, 2006).

Os projetistas ajudam todos os membros do grupo a chegar num acordo comum sobre os nomes dos objetos e os conceitos de relevância para o sistema em desenvolvimento, produzem um dicionário de nomes de classes que podem servir como auxílio nos textos de caso de uso e constroem um conjunto de diagramas de classes que trabalham juntos (SCOTT, 2003).

O analista de sistema comanda a tarefa de descoberta dos atores humanos e não humanos que estarão interagindo com o sistema e reúne um conjunto de casos de uso que realize a tradução do comportamento destes atores agregados com o sistema (INTHURN, 2001).

O especificador de casos de uso realiza a tarefa de escrever o texto para o curso normal e alternativo de ação. A estruturação do modelo de caso de uso é feita pelo analista de sistemas que compreende em quebrar os casos de uso na busca por fatos mais simples (SCOTT, 2003).

A rede de Petri servirá como ferramenta de análise e comportamento do fluxo de trabalho de requisitos na fase de concepção do projeto, sendo os seus elementos na forma de artefatos que realizarão as atividades necessárias dos trabalhadores por meio da captação dos requisitos do sistema.

5 REDES DE PETRI

A rede de Petri surgiu a partir da tese de doutorado, *Kommunikation TT Automaten* (Comunicação entre autômatos) defendida por Carl Adam Petri em 1962, submetida à Faculdade de Matemática e Física da Universidade Técnica de Darmstadt localizada na Alemanha, porém, foi concebida durante seu trabalho como cientista na Universidade de Bonn na Alemanha (FERNANDES, 2006).

O objetivo de Carl era desenvolver um modelo em que às máquinas de estado pudessem se comunicar entre si, assim ele apresentou um grafo bipartido, ou seja, um grafo cujos vértices pudessem ser divididos em dois conjuntos, nos quais não existem arestas entre si num mesmo conjunto (OLIVEIRA, 2007). Mais informações sobre grafos vide Apêndice B.

A característica principal atingida por este modelo foi de representar a concorrência. Este modelo ganhou o nome de rede de Petri em homenagem ao autor.

Alguns anos mais tarde, devido ao fato de se adaptar a modelagem de vários tipos de sistema, as redes de Petri alcançaram várias áreas dentro da tecnologia da informação como a modelagem de componentes de *hardware*, controle de processos, linguagens de programação, sistemas computacionais, protocolos de comunicação, entre outros (MARRANGHELLO, 2005a).

5.1 DEFINIÇÕES BÁSICAS

A rede de Petri apresenta-se como uma linguagem com forte apelo gráfico e matemático, relacionando estados e ações de sistemas, sendo utilizadas para especificação, projeto, simulação e verificação de sistemas de processamento de informação que se caracterizam como concorrentes, assíncronos, distribuídos, paralelos, não determinísticos e estocásticos (MARRANGHELLO, 2005b).

As redes de Petri podem ser aplicadas em diversos tipos de sistemas, inclusive computacionais, apresentando um bom nível de abstração em comparação com outros modelos gráficos. Além disso, dá suporte à verificação quanto a erros, bloqueios e conflitos do sistema especificado, um recurso muito interessante quando se trata de modelagem de sistemas (PETERSON, 1981).

Sua modelagem é do tipo estado-evento, onde cada evento possui pré-condições que permitirão sua ocorrência e pós-condições decorrentes desta. A ferramenta possui características interessantes, como um formato intuitivo e facilidade na representação gráfica, além de sua natureza formal e rigorosa, na qual permite um raciocínio formalizado e lógico, chegando assim a ser comparada com linguagem de nível similar ao Prolog (JANOUSEK, 1995).

5.2 ESTRUTURA DAS REDES DE PETRI

Os quatro elementos básicos que formam a estrutura de uma rede de Petri, ilustrados na Figura 5, são (MARRANGHELLO, 2005a).

- a) lugar ou estado: armazena dados correspondentes as condições para que os eventos ocorram, modelando a atividade de um agente ou componente de *hardware* ou *software*;
- b) transição ou ação: elemento responsável por efetuar a caracterização das mudanças de estado, representa a modelagem de eventos, operações e estruturas de decisão;
- c) marca ou ficha: representa o estado ou recurso de um sistema, ou seja, uma característica habilitada. Sua posição no grafo constitui a marcação, a sua evolução permite modelar o comportamento dinâmico da ferramenta;

- d) arco ou flecha: representa a ligação entre estado/transição e transição/estado que é feita através de arcos direcionados usados para especificar como se dá a transformação de um estado em outro pela ocorrência das transições no sistema.

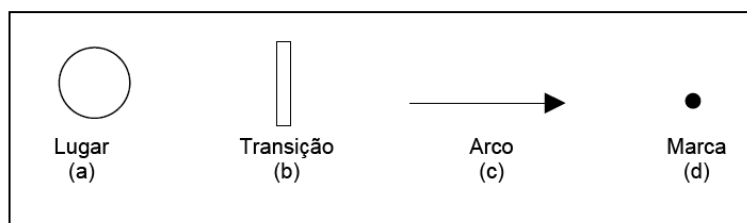


Figura 5. Elementos da rede de Petri
Fonte: MARRANGHELLO, N. (2005a).

As flechas podem possuir valor, caso o peso seja igual a um a rede de Petri é classificada como rede de Petri Ordinária, nos casos em que são permitidos múltiplos arcos ela é dita como rede de Petri Generalizada. Existem outros tipos de rede de Petri, uns são chamados de alto nível possuindo alto grau de complexidade, exemplo rede de Petri colorida, e outros de rede de Petri temporizada associando estados e transições com intervalo de tempo (CARDOSO; VALETTE, 1997).

A rede de Petri Ordinária é considerada como modelo básico desta ferramenta por ser de baixo-nível, pois as marcas utilizadas nesta rede representam apenas os seus estados. (OLIVEIRA, 2007).

Os estados e as marcas podem ser usados para discriminar entidades abstratas como condições ou atividades e também entidades físicas como objetos (CARDOSO; VALETTE, 1997).

Cada estado em uma rede de Petri representa um lugar parcial do sistema modelado, quando ocorre um evento o sistema muda do estado atual para o próximo. Um evento somente pode acontecer se todas as suas pré-condições forem satisfeitas. Após este fato, será gerada uma saída que é representada pela pós-condição (MARRANGHELLO, 2005a, 2005b).

Um evento está associado a uma transição, assim a representação de sua ocorrência é dada pelo disparo da transição ao qual está representado. Para disparar uma transição, faz-se necessário retirar as marcas dos lugares de entrada da ação e em seguida botá-las em cada estado de saída. A ocorrência da mudança de marcas dos estados corresponde ao comportamento dinâmico do sistema (CARDOSO; VALETTE, 1997).

Pode-se visualizar as interpretações mais usadas sobre os elementos de uma rede de Petri na Tabela 2 (MURATA, 1989).

Tabela 2. Interpretações típicas de transições e estados

Estados de entrada	Transições	Estados de saída
Pré-condição	Evento	Pós-condições
Dado de entrada	Passo computacional	Dado de saída
Sinal de entrada	Processador de sinal	Sinal de saída
Recursos necessários	Tarefa ou trabalho	Recurso liberado
Condições	Cláusulas lógicas	Conclusões
<i>Buffers</i>	Processador	<i>Buffers</i>

Fonte: MURATA, T. (1989).

5.3 COMPORTAMENTO DINÂMICO DAS REDES DE PETRI

O comportamento dinâmico de um sistema dá-se por seus estados e pela mudança para outros estados. Na ferramenta rede de Petri a ocorrência de um evento perante estes estados é denominado como disparo de transição (PALOMINO, 1995).

O controle pelo número e distribuição de marcas na rede é gerado pela execução de uma rede de Petri. As marcas ficam nos lugares e fazem o controle de execução das transições. A ferramenta é executada por meio do disparo de transições, assim que isto acontece, as marcas dos lugares de entrada são removidas, criando-se novas marcas que serão atribuídas para os lugares de saída (FERNANDES, 2006).

Uma transição dispara somente se estiver habilitada, isto ocorre quando cada um de seus lugares de entrada tem pelo menos uma quantia de marcas maior ou igual ao peso dos

arcos que conectam esses lugares nesta transição, portanto para uma transição disparar o lugar de entrada tem que estar marcado (CARDOSO; VALETTE, 1997).

As marcas são informações atribuídas aos estados, para representar a situação da rede em um determinado momento, assim, para simular o comportamento dinâmico dos sistemas, a marcação da rede de Petri é modificada a cada ação realizada (MARRANGHELLO, 2005b).

A Figura 6 ilustra o disparo de uma transição, pode-se notar que os lugares de entrada estão marcados e a quantia das fichas é maior ou igual ao peso dos arcos, portanto a transição está habilitada à disparar (PALOMINO, 1995).

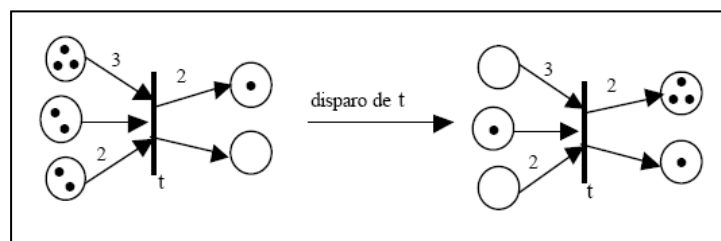


Figura 6. Disparo de uma transição
Fonte: PALOMINO, R. C. (1995).

5.4 PROPRIEDADES DAS REDES DE PETRI

As propriedades das redes de Petri podem ser divididas em dois tipos, sendo eles propriedades comportamentais ou dinâmicas e propriedades estruturais ou estáticas (OLIVEIRA, 2007).

As propriedades comportamentais mudam durante o funcionamento dinâmico da rede de Petri e são totalmente dependentes da marcação inicial e conseqüentemente do estágio de execução em que se encontra o sistema (FERNANDES, 2006).

As propriedades estruturais são definidas por meio dos invariantes, ou seja, são independentes da marcação inicial e não se modificam no decorrer do seu funcionamento, podendo até aplicar, incluindo a rede de Petri, sem qualquer marcação (MURATA, 1989).

5.4.1 Propriedades Comportamentais

Dentre as propriedades dinâmicas temos a limitação. Uma rede de Petri é dita limitada se, e somente se, todos os estados são n-limitados. Isto significa que se em cada lugar da rede o número total de marcas nunca exceder a um inteiro “n” ela será limitada. Se a rede for limitada ao inteiro um significa que todos os estados também são, então se diz que a rede é segura (MARRANGHELLO, 2005b).

Outra propriedade comportamental é a vivacidade. Uma rede de Petri é dita viva quando é possível executar todas as suas transições, disparos, a partir de qualquer uma marcação da rede, sendo assim, as transições devem estar habilitadas para qualquer uma das marcações que se pode alcançar. Esta propriedade é relevante na análise de sistemas, pois existindo uma verificação pode-se apontar a inexistência de bloqueios no sistema modelado (FERNANDES, 2006).

Ainda nas propriedades comportamentais tem-se a reiniciabilidade. É dita reiniciável toda rede de Petri que oferecer a possibilidade de voltar para a marcação inicial de qualquer marca alcançável. Uma marcação é alcançável se existir uma seqüência finita de disparos que conduza de volta a ela a partir da marcação inicial, sendo assim, se todas as marcas forem decorrentes da inicial, a rede é dita alcançável (MARRANGHELLO, 2005b).

A Figura 7 representa um exemplo de rede de Petri limitada, viva e reiniciável.

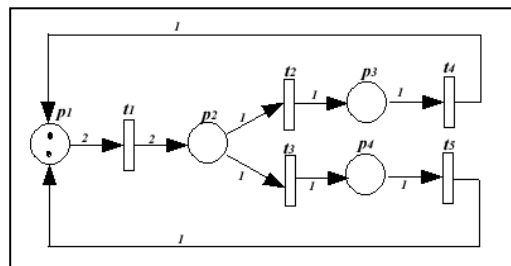


Figura 7. Rede de Petri Ordinária limitada, viva e reiniciável
Fonte: MARRANGHELLO, N. (2005b).

Observa-se na figura que a rede é limitada, pois as marcas nos estados da rede nunca excedem o valor inteiro “2”, entende-se então que é uma rede 2-limitada. Diz-se rede de Petri viva, pois todas as transições podem ser percebidas a partir de qualquer marcação da rede. Fala-se reiniciável, pois se pode chegar à marcação inicial por meio de qualquer seqüência de disparo (MARRANGHELLO, 2005b).

5.4.2 Propriedades Estruturais

A conservação é uma propriedade estática. Um componente conservativo é o conjunto de estados para os quais a soma de suas marcas permanece constante independentemente de qualquer seqüência de disparo. Uma rede de Petri é estritamente conservativa se todas as suas transições forem estáticas (OLIVEIRA, 2007).

É considerada consistente uma rede de Petri que depois de uma seqüência de disparos, a partir de uma marcação inicial, for possível retornar ao estado inicial após o disparo de pelo menos uma vez cada transição (PALOMINO, 2001).

Outra propriedade estrutural é a repetitividade. Uma rede de Petri, quando associada a uma dada marcação, torna-se repetitiva por existir uma seqüência de disparos onde todas as transições são feitas infinitamente (MARRANGHELLO, 2005b).

A Figura 8 apresenta uma invariação notória, determinando as propriedades estruturais de conservação, consistência e repetitividade (MARRANGHELLO, 2005b).

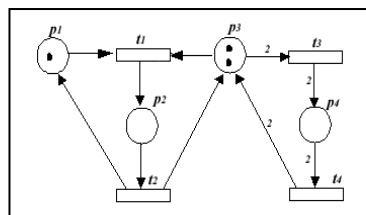


Figura 8. Exemplo de invariantes
Fonte: MARRANGHELLO, N. (2005b).

5.5 WORKFLOW-NET

A rede de Petri que representa um processo de *workflow* é chamada de *workflow-net*, esta tem apenas um lugar fonte e uma absorção. Todo nó é um caminho do estado fonte para o lugar absorção (PÁDUA, 2003).

A tarefa primária do sistema de gerenciamento de processos de negócio é representar tais processos direcionados a determinados casos por meio da junção de várias perspectivas. Na modelagem e execução de processos de negócio são relevantes as seguintes perspectivas (PÁDUA, 2003).

- a) fluxo de controle ou processo: são perspectivas na qual são criadas as definições dos processos para especificar quais atividades necessitam ser executadas e em que seqüência;
- b) recurso: representa as especificações da estrutura organizacional;
- c) dados: trata do controle e produção das informações;
- d) tarefas: discrimina o conteúdo das etapas de um processo, unidade lógica de trabalho com características como grupos de operações que precisam ser realizadas, descrição, duração, prioridades e classes de recursos requeridas;
- e) operação: descreve ações elementares, uma atividade pode envolver muitas operações, que são executadas por aplicações que empregam editores de texto ou até *softwares* construídos para fazer cálculos complexos, estes sistemas criam, lêem ou modificam o controle e a produção de dados na perspectiva destes.

Utiliza-se um projeto já desenvolvido para ser analisado por meio dos recursos das redes de Petri, tais como sua estrutura, dinamismo e propriedades. Há outros estudos correlatos sobre utilizar redes de Petri, como por exemplo, sobre seu potencial em modelagem de negócio.

6 TRABALHOS CORRELATOS

Os trabalhos correlatos são trabalhos que já foram desenvolvidos e que possuem alguma ligação com o projeto de pesquisa que está sendo desenvolvido.

Os trabalhos possuem ligação entre redes de Petri aplicada para sistema de gerenciamento *workflow*, seu potencial em modelagem e trabalhos que falam sobre ciclos de vida de sistemas.

6.1 CICLO DE VIDA DE SISTEMAS BASEADO EM REDES DE PETRI

O objetivo deste trabalho é propor um modelo, combinando técnicas descritas no modelo de ciclo de vida do método Fusion, que foi criado com a pretensão de abranger o processo de desenvolvimento de *software* nas fases iniciais do projeto (FERNANDES, 2006).

Este método é capaz de descrever o comportamento do sistema com a perspectiva de como ele se comunica com o ambiente desde a sua criação até o seu fim, com as vantagens comportamentais e dinâmicas garantidas pelas propriedades de redes de Petri, obtendo um modelo com maior poder de validação na obtenção do ciclo de vida de sistemas e minimizando a ocorrência de erros (FERNANDES, 2006).

6.2 ESTUDO E ANÁLISE DE MODELAGEM DE SISTEMAS DE WORKFLOW BASEADOS EM REDES DE PETRI

As organizações estão buscando cada vez mais por maiores índices de qualidade e eficácia no gerenciamento de seus processos de negócio. Para alcançar esses índices o *workflow* (fluxo de trabalho) tem sido bastante utilizado, proporcionando maior produção e

desempenho das organizações. Um dos principais obstáculos enfrentados é a falta de métodos e ferramentas que possam permitir a especificação dos processos que fazem parte desse sistema (OLIVEIRA, 2007).

Neste contexto, o presente trabalho procura realizar um estudo da utilização das redes de Petri para a modelagem de *workflow*. Para isso, são apresentados os seus principais conceitos. Posteriormente, a aplicação da modelagem de *workflow* usando as redes de Petri no cenário do processo de Colação de Grau da UFG/Campus Catalão (OLIVEIRA, 2007).

6.3 O POTENCIAL DAS REDES DE PETRI EM MODELAGEM E ANÁLISE DE PROCESSOS DE NEGÓCIO

A tecnologia de gerenciamento *workflow* procura oferecer uma solução flexível em apoio aos processos de negócios, por meio da facilitação de modificações e da criação de novos processos. Entretanto, a falta de definição no que se refere à sintaxe e à semântica de tais técnicas dificulta análises mais complexas dos modelos. Nesse caso, a ferramenta chamada de redes de Petri atua com excelência em potencial, por possuir representação gráfica, funcionar como linguagem de comunicação, possibilitar descrever aspectos estáticos e dinâmicos do sistema a ser representado (PÁDUA, 2003).

O trabalho tem por objetivo oferecer uma visão atualizada do estado da arte na área de modelagem de *workflow* baseada em redes de Petri e expor um exemplo de modelo de *workflow* com parâmetros de tempo e custo associados às transições da rede. (PÁDUA, 2003).

Os trabalhos correlatos estão relacionados com estudo e análise de processos, modelagem *workflow* e comportamento de sistemas utilizando as redes de Petri na fase inicial do projeto.

7 A UTILIZAÇÃO DAS REDES DE PETRI NA FASE DE CONCEPÇÃO DO PROJETO

A presente pesquisa consiste no estudo de conceitos de Engenharia de *Software* e de sua importância referente à fase de concepção de um projeto por meio do processo unificado e de seu *workflow* de requisitos, construindo sua modelagem e efetuando análise de seu comportamento por meio das redes de Petri verificando possíveis erros, conflitos e bloqueios.

O trabalho se propõe a realizar a transformação das atividades de requisitos, fluxo de trabalho, da fase de concepção do processo unificado em uma modelagem específica utilizando a ferramenta chamada redes de Petri, junto com a sua estrutura e suas propriedades, por causa da sua flexibilidade, que irá auxiliar na análise do seu comportamento e na sua representação gráfica, proporcionando aos desenvolvedores um objeto de pesquisa para fases futuras do projeto.

A fase seguinte do projeto também fará a abordagem das redes de Petri no contexto de *workflow*, na sua aplicação em modelagem de sistemas gerenciais de fluxo de trabalho e na sua representação como ferramenta de análise e desenho gráfico, demonstrando sua modelagem direcionada para fase inicial de um projeto.

7.1 REDES DE PETRI NO CONTEXTO DE WORKFLOW

As redes de Petri permitem visualizar os processos e a comunicação entre eles, tornando-se assim uma ferramenta útil para realização desta tarefa. O seu uso na modelagem de *workflow* é dado pelo seguinte modo, cada tarefa é representada por uma transição correspondente, os estados figuram as pré e pós-condições, estas sendo os recursos requeridos

para execução de determinada atividade. Os arcos desempenham relações lógicas entre o fluxo de trabalho. As marcas dos estados podem ser abstraídas representando um lugar de determinado evento.

Não há um consenso sobre quais tipos de redes de Petri adaptam-se melhor às necessidades específicas da modelagem dos sistemas de *workflow*. Entretanto, os processos simples podem ser desenhados diretamente como sistemas de transição, já os processos mais avançados tornam-se difícil por causa do grande número de estados. O uso de ferramenta computacional é um fator importante no auxílio da construção de modelagem e métodos de análise relacionados.

No contexto de *workflow* as redes de Petri permitem uma representação mais concisa e com naturalismo dos processos, sendo que a tarefa ocorre quando certos objetos como recursos ou permissões estão presentes. A idéia consiste em dar início à atividade, assim os objetos são consumidos e por último, novos objetos são produzidos para possibilitar que novas tarefas iniciem.

7.2 APLICAÇÃO DAS REDES DE PETRI NA MODELAGEM DE SISTEMAS GERENCIAIS WORKFLOW

As três razões principais para aplicação das redes de Petri na modelagem de *workflow* são identificadas primeiramente, por elas possuírem tanto semântica formal quanto natureza gráfica, segundo, por poderem modelar com notoriedade e flexibilidade os estados do sistema e terceiro, por existir grande variedade e disponibilidade de técnicas de análise que são utilizadas para verificar a correção de erros de processo do fluxo de trabalho.

Outros motivos para escolher esta ferramenta nesse contexto podem ser vistos como integração de dados e aspectos comportamentais e apoio concorrente e cooperativo.

A principal vantagem em empregar redes de Petri na modelagem *workflow* é a combinação de fundamentação matemática, representação gráfica entendível e possibilidade de simulações e verificações.

A rede de Petri demonstra vantagens no uso em fluxo de trabalho por possibilitar que ocorra rastreamento minucioso e sem ambigüidade a cada etapa da operação e por permitir uma representação gráfica acoplada a mecanismos de análise para verificar propriedades do sistema e se este está correto. Porém, uma deficiência do uso desta técnica é a falta de uma apresentação mais descritiva, o que pode ser compensado pela possibilidade de implementação de *software*, criando interfaces mais amigáveis ao usuário.

7.3 MODELAGEM E ANÁLISE DO PROJETO UTILIZANDO REDES DE PETRI

Utiliza-se o projeto do bacharel em Ciência da Computação, Thiago Casagrande Darós, finalizado no ano de 2003, conforme citado no apêndice “A” deste trabalho como sistema modelo para obter-se a modelagem da fase de concepção, conforme captura dos seus requisitos, e análise do seu comportamento por meio das redes de Petri.

A Figura 9 retrata a ocorrência do sistema em modo “Sair_Desligado”, sendo que no momento em que a marca se encontra nele, somente uma transição está sensibilizada na rede inteira, vide Figura 12, ou seja, quando a rede está marcada no estado “Sair_Desligado” significa que o sistema não está operando e sua única saída é entrar em modo de execução.

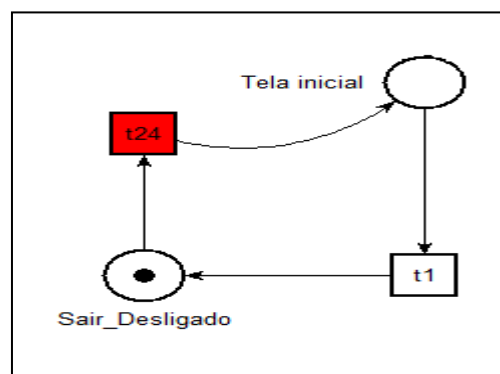


Figura 9. Estado sair_desligado.

Quando o sistema for executado por meio de requisição do usuário, a transição “t24” dispara a ficha do estado atual, “Sair_Desligado”, removendo-a, para o estado alcançável posterior, “Tela Inicial”, criando-se uma nova marca, por meio da única ligação de fluxo de trabalho entre a transição e o lugar de saída.

Após este acontecimento, diz-se que o sistema está em operação, ou seja, neste caso, na tela inicial do aplicativo, como se pode ver na Figura 10. Nota-se que neste momento há três transições habilitadas, “t1”, “t21” e “t12”, isto quer dizer que existem três opções para o usuário, sendo estas fechar o programa e retornar ao estado de desligado, ou então, ir para barra de ferramentas e iniciar uma função, ou ainda, escolher o menu e realizar uma tarefa.

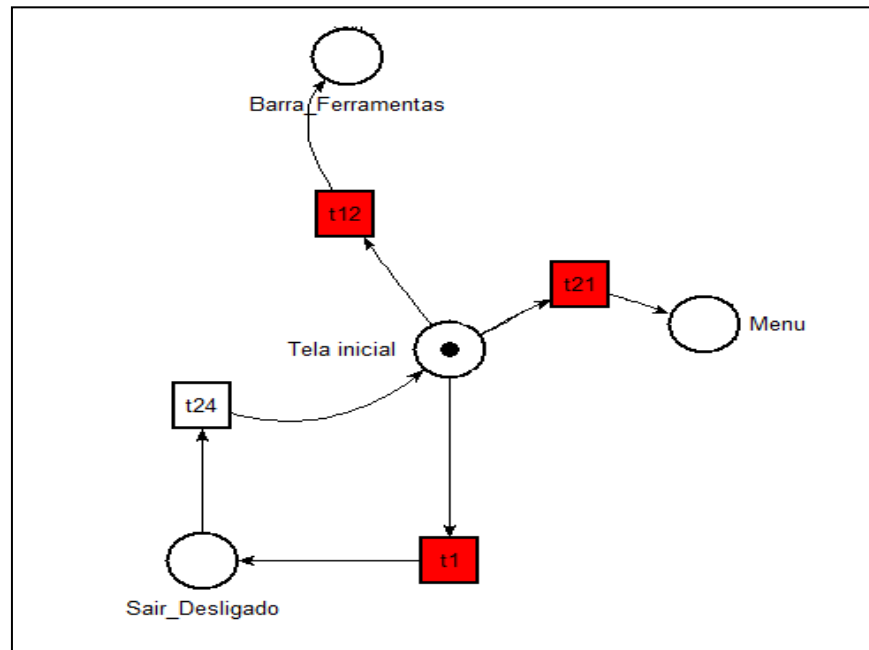


Figura 10. Programa em execução.

Na escolha por qualquer uma das transições a ficha será removida do estado “Tela inicial”, conseqüentemente um caminho será percorrido e ao final da operação uma transição fará com que a marca retorne para este lugar (Tela inicial), pois como dois corpos não ocupam o mesmo lugar no espaço fisicamente, também não há como realizar duas ou mais operações no mesmo instante, como por exemplo, salvar e abrir arquivo.

No caso de o usuário optar por fazer um texto ou qualquer outro item tanto da barra de ferramentas, quanto da barra de operações, sendo que esta é opção estendida da

citada anterior, e também do menu, há necessidade de tratamento em redes de Petri, criando-se um estado de espera, sempre marcado, ligado às suas respectivas transições, conforme o exemplo da Figura 11.

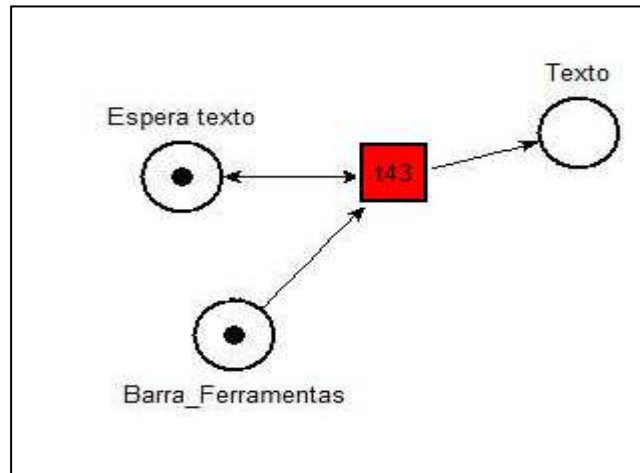


Figura 11. Estado de espera

Nota-se que para a transição “t43” estar habilitada é necessário que a barra de ferramentas seja escolhida, pois sempre haverá uma espera pela ação, possibilitando que haja o disparo da transição, onde serão removidas as fichas dos estados “Espera texto” e “Barra_Ferramentas”, gerando uma nova marca no lugar “Texto” e outra retornando para o estado de espera, aguardando a próxima vez que o usuário irá requisitar esta ação.

A Figura 12 demonstra a modelagem unificada do projeto na fase de concepção e seus requisitos ligados por meio de fluxo de trabalho utilizando a estrutura gráfica das redes de Petri. Também foi analisado, por meio desta, que o sistema possui 67 estados e 83 transições, sendo de propriedade comportamental invariante e consistente por possibilitar que depois de uma seqüência de disparos, retorne-se ao estado inicial por meio de qualquer caminho na rede.

Pode-se concluir que a representação da fase de concepção do projeto por meio de desenho baseado em redes de Petri auxilia os desenvolvedores antes ou depois de implementar o sistema, pois visualiza-se a coerência entre os requisitos por meio do fluxo de trabalho.

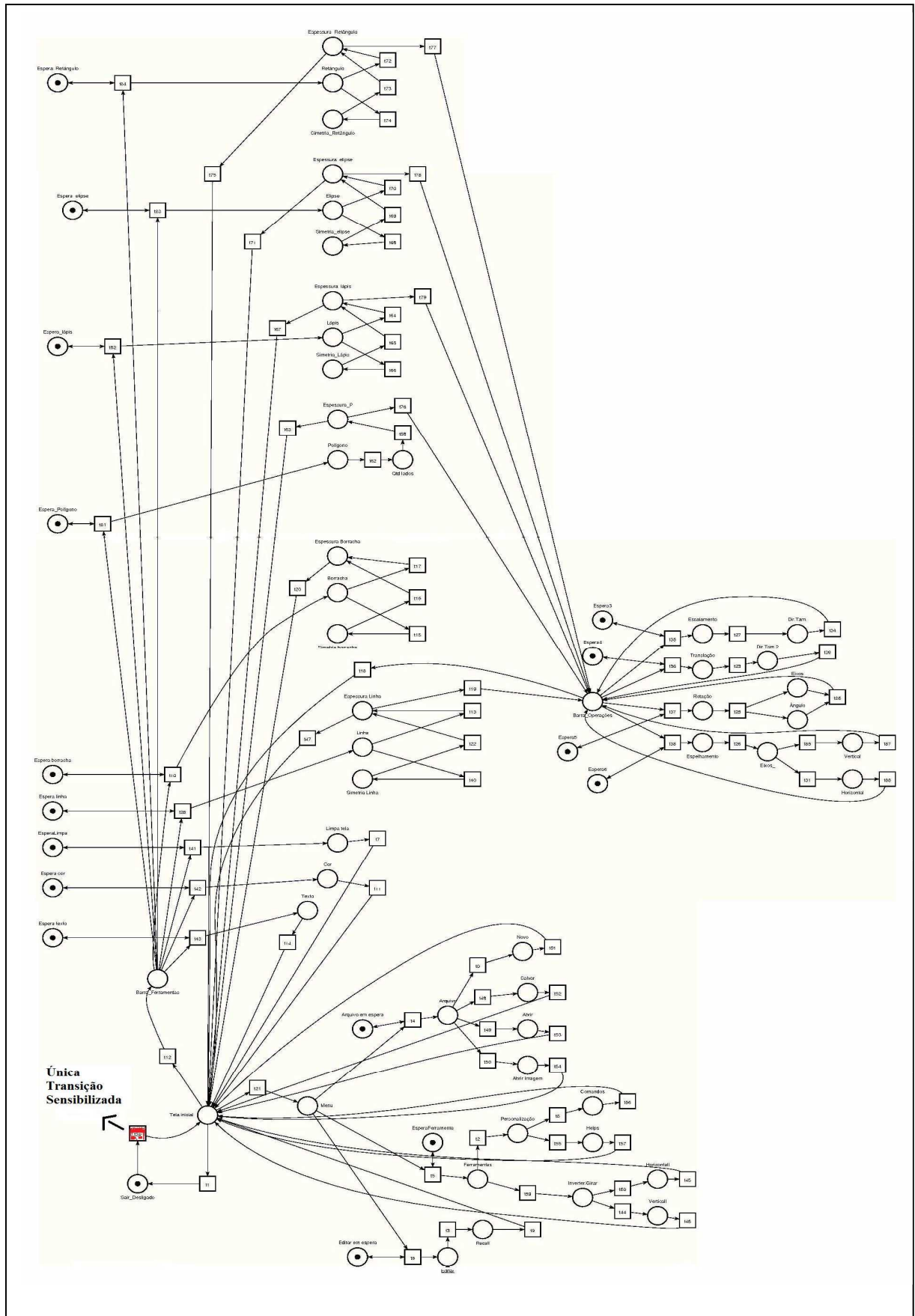


Figura 12. Modelagem do projeto em redes de Petri

7.4 METODOLOGIA

Na criação deste trabalho foram realizadas as seguintes etapas metodológicas: realizar levantamento bibliográfico, relatar conceitos relevantes sobre Engenharia de *Software*, efetuar estudo voltado à sistemas de gerenciamento *workflow*, redes de Petri e processo unificado, compreensão da fase de concepção do projeto, entendimento do *workflow* de requisitos da fase de concepção do projeto, representação das redes de Petri como ferramenta de modelagem gráfica, aplicação das redes de Petri na fase inicial do projeto, realização de estudo do *software* de apoio ao ensino com o uso de computação gráfica e demonstração de uma modelagem em redes de Petri direcionada à fase de concepção do projeto.

7.4.1 Realizar Levantamento Bibliográfico

Realiza-se o levantamento bibliográfico para esta pesquisa por meio da rede mundial dos computadores, a internet, e principalmente de livros da biblioteca da universidade, livros e apostilas dos professores e colaboradores.

7.4.2 Relatar Conceitos Relevantes sobre Engenharia de Software

Os conceitos sobre Engenharia de *Software* são selecionados por meio de levantamento bibliográfico específico, ou seja, somente os conceitos relevantes sobre a área global da pesquisa são considerados.

7.4.3 Efetuar Estudo Voltado à Sistemas de Gerenciamento Workflow, Redes de Petri e Processo Unificado

Realizam-se estudos sobre sistemas de gerenciamento *workflow*, redes de Petri e processo unificado por meio da leitura de apostilas, livros e internet, interpretação de exemplos e explicações de professores destas áreas.

7.4.4 Compreensão da Fase de Concepção do Projeto

A compreensão da fase de concepção do processo unificado torna-se determinante para poder entender sua preparação e suas atividades de requisitos, sendo assim a leitura de textos e visualização de figuras e exemplos que abordam o assunto se faz necessário.

7.4.5 Entendimento do Workflow de Requisitos da Fase de Concepção do Projeto

O entendimento do fluxo de trabalho de requisitos se faz necessário, pois ele é o enfoque da fase de concepção do projeto, sendo importante analisar o seu conceito, seus artefatos, trabalhadores, atividades e requisitos funcionais e não funcionais por meio de exemplos e leitura de material.

Portanto é importante compreender este fluxo de trabalho, pois é ele que realiza a captação dos requisitos da fase de concepção do projeto unificado.

7.4.6 Representação das Redes de Petri como Ferramenta de Modelagem Gráfica

Representa-se as redes de Petri como ferramenta de modelagem gráfica pelos elementos da sua estrutura, tais como estados, marcas, transições, fluxos de relacionamento, por meio de exemplos e análise de seu funcionamento e comportamento.

7.4.7 Realização de Estudo do Software de Apoio ao Ensino com o Uso de Computação Gráfica

Realiza-se o estudo do “projeto de apoio ao ensino com o uso de computação gráfica” por meio de leitura do trabalho de conclusão de curso do bacharel em Ciência da Computação, Thiago Casagrande Darós, no ano de 2003, e também por análise e utilização do *software* em funcionamento, capturando os seus requisitos.

7.4.8 Aplicação das Redes de Petri na Fase Inicial do Projeto

A aplicação das redes de Petri na fase inicial do projeto deu-se com auxílio do *software* gratuito “Tina-2.9.4”, disponível no link <http://www.laas.fr/tina/distribution.php>, para modelar e analisar o comportamento dinâmico do sistema por meio desta ferramenta. Também serviu para relatar a quantidade de estados e transições que a rede do projeto possui e definição de sua propriedade adequada.

7.4.9 Demonstração de uma Modelagem em Redes de Petri Direcionada à Fase de Concepção do Projeto

A modelagem está representada de forma gráfica, sendo que desta maneira os engenheiros de *software* possam analisá-la e utilizá-la para detectar possíveis erros, conflitos e bloqueios irreversíveis do sistema futuramente.

7.5 RESULTADOS OBTIDOS

A presente pesquisa resultou na compreensão de metodologias voltadas para a fase de concepção do desenvolvimento de sistemas, abstraídas da Engenharia de Software, fluxo de trabalho, processo unificado, *workflow* de requisitos, utilização e aplicação das redes de Petri na modelagem e análise do comportamento de um projeto, e por fim, demonstração do desenho de um projeto específico na fase inicial do seu desenvolvimento baseado em redes de Petri.

O estudo do processo unificado e da tecnologia *workflow* possibilitou a compreensão da fase de concepção do projeto e o entendimento de seu fluxo de trabalho de atividades e requisitos tornando a modelagem de um projeto com coesão nos seus ligamentos.

A utilização das redes de Petri serve para visualização adequada dos relacionamentos entre as tarefas de um projeto computacional. O auxílio de sua estrutura proporcionou uma modelagem aderente que possibilitou analisar o comportamento do sistema prevenindo-o ou corrigindo-o de conflitos e bloqueios presentes ou futuros.

As ferramentas de caso de uso nem sempre possibilitam ao desenvolvedor de

projeto ter interatividade com o desenho pretendido, já as redes de Petri permitem ao Engenheiro de Software interagir com o sistema mesmo antes de ser implementado, obtendo como resultado, em vez da demora na procura em várias linhas de código na fase de programação, rapidez na correção de possíveis problemas.

Pode-se concluir que este método vai direto ao ponto, no caso na fase de concepção de um projeto, resultando em *softwares* de qualidade, menos propícios a falhas e complicações futuras, enquanto outros projetos tratam as redes de Petri generalizando-as.

CONCLUSÃO

O presente trabalho evidenciou a relevância da Engenharia de Software na fase de concepção de um projeto, a gerência de processos *workflow* por meio do processo unificado, a captação dos requisitos de um sistema, como exemplo, já terminado, a aplicação das redes de Petri como ferramenta de desenho para modelar e de interatividade para analisar o seu comportamento e a demonstração desta para os desenvolvedores de *software*.

Neste contexto, a pesquisa alcançou os seus objetivos baseando-se no estudo de metodologias referentes a modelagem de *software*, sistemas de gerenciamento *workflow*, processo unificado, fase inicial do sistema, fluxo de trabalho de atividades, redes de Petri e um projeto exemplo como modelo.

No entanto, conclui-se que a UML, ferramenta padrão de caso de uso, possui função importante dentro do desenvolvimento de sistemas computacionais, principalmente no ramo de gestão empresarial por representar melhor uma unidade funcional coerente provida pelo sistema.

Quanto às metodologias abordadas nesta pesquisa, pode-se observar que existe diversidade entre os processos expostos pelo nível de material requerido e no modo como as etapas de desenvolvimento são encadeadas. Desta forma, os engenheiros de *software* devem adotar o método que melhor fique adequado para as características do sistema requisitado, sendo auxiliadas sempre que possível por ferramentas de análise de comportamento, independente do procedimento adotado.

Sugere-se para trabalhos futuros a aplicação da rede de Petri colorida na fase de concepção de um projeto, podendo assim obter-se a condensação ideal para sistemas gerenciais *workflow*, desenhando e analisando o comportamento de um sistema pronto ou em fase de andamento.

Outra recomendação proposta trata-se de utilizar as redes de Petri temporizada para a análise de intervalos de tempo do funcionamento das atividades de um sistema pronto, obtendo êxito quanto a duração de certa tarefa requisitada pelo usuário, por exemplo, pode-se acessar determinada função enquanto estiver disponível e quando desabilitada, aguarda-se um determinado evento ocorrer para que esta possa ser habilitada novamente.

Ao longo do desenvolvimento desta pesquisa foram encontradas diversas dificuldades e problemas, como por exemplo, sendo custoso encontrar a ferramenta certa para desenhar e analisar o projeto modelo em redes de Petri e realizar a ligação entre os seus requisitos da fase de concepção, para isso, buscou-se auxílio em outra universidade, conforme foto no Anexo “A”, mais precisamente na UFSC com o professor e pesquisador, Sr. Jean Marie Farines, doutor em Engenharia Elétrica, que atua na área de modelagem e verificação de sistemas, sistemas de tempo real, qualidade de serviços em sistemas distribuídos e redes, publicou diversos artigos e mais de cem trabalhos completos em anais de congresso.

REFERÊNCIAS

- ARAUJO, R. M. d.; BORGES, M. R. d. S. **Sistemas de workflow**. Publicado na XX Jornada de Atualização em Informática – Congresso da Sociedade Brasileira de Computação. Fortaleza, Ceará, Brasil. 2001. Disponível em: <<http://chord.nce.ufrj.br/cursos/teesi/textos/apostilaJai2001div.pdf>>. Acesso em: 02 set. 2008.
- BAEYENS, T. **The state of workflow**. 2004. Disponível em: <<http://www.jboss.com/products/jbpm/stateofworkflow>>. Acesso em: 11 set. 2008.
- CARDOSO, J.; VALETTE, R. **Redes de petri**. Florianópolis: Editora da UFSC, 1997.
- CRUZ, A. **Workflow: a tecnologia que vai revolucionar processos**. São Paulo: Editora Atlas S.A., 2000.
- DARÓS, Thiago Casagrande. **Software de apoio à aprendizagem com o uso de computação gráfica**. 2003. 9 v. Tese (Bacharel em Ciência da Computação) – Unesc, Criciúma, 2003.
- FALBO, Ricardo de Almeida. **A experiência na definição de um processo padrão baseado no processo unificado**. 2000. Disponível em: <<http://www.inf.ufes.br/~falbo/download/pub/Simpros2000.pdf>>. Acesso em: 28 out. 2008.
- FERNANDES, Danilo Douradinho. **Ciclo de vida de sistemas baseado em redes de petri**. 2006. Disponível em: <<http://www2.dc.uel.br/nourau/document/?down=381>>. Acesso em: 20 de out. de 2008.
- INTHURN, Cândida. **Qualidade e teste de software**. Florianópolis: Visual Books, 2001.
- JANOUSEK, V. **Pntalk: object orientation in petri nets. In: TTP ing simulation multicoference ems '95**. Prague, Czech Republic: Czech Technical University, 1995. P.196–200. Disponível em: <[HTTP://www.citeseer.ist.psu.edu/janousek95pntalk.html](http://www.citeseer.ist.psu.edu/janousek95pntalk.html)>. Acesso em: 24 out. 2008.
- LEYMANN, F.; ROLLER, D. **Workflow-based TTP ing ons**. IBM Systems Journal, v. 36, n. 1, 1997. Disponível em: <<http://www.researchweb.watson.ibm.com/journal/sj/361/leymann.html>>. Acesso em: 15 out. 2008.
- MARRANGHELLO, N. DCCE/IBILCE/UNESP. **Redes de petri: conceitos e aplicações**. Mar. 2005a. Disponível em: <<http://www.dcce.ibilce.unesp.br/~norian/cursos/mds/ApostilaRdP-CA.pdf>>. Acesso em: 04 set. 2008.

MARRANGHELLO, N. DCCE/IBILCE/UNESP. **Redes de petri: propriedades e análise.** Ago. 2005b. Disponível em:
<<http://www.dcce.ibilce.unesp.br/~norian/cursos/tds/ApostilaRdP-PA.pdf>>. Acesso em: 27 out. 2008.

MURATA, T. **Petri nets: properties, analysis and applications.** Proceedings of the IEEE 77(4), 1989. Disponível em: <<http://www.ieeexplore.ieee.org>>. Acesso em: 21 out. 2008.

NICOLAO, M. & OLIVEIRA, J. P. M. **Caracterizando sistemas de workflow.** Revista READ, ISSN 1413-2311 (3), 1996. Disponível em:
<<http://www.inf.ufrgs.br/~palazzo/docs/read/workflow.html>>. Acesso em: 21 out. 2008.

OLIVEIRA, Kênia Santos de. **Estudo e análise de modelagem de sistemas de workflow baseada em redes de petri.** 2007. Disponível em:
<<http://www.catalao.ufg.br/cc/disc/pfc/mono/Kenia2007.pdf>>. Acesso em: 17 out. 2008.

PÁDUA, Sílvia Inês Dallavalle de T. **O potencial das redes de petri em modelagem e análise de processos de negócio.** 2003. Disponível em:
<<http://www.scielo.br/pdf/gp/v11n1/a10v11n1.pdf>>. Acesso em: 10 out. 2008.

PALOMINO, Reynaldo Chile. . **Uma abordagem para a modelagem, análise e controle de sistemas de produção utilizando redes de Petri.** 1995. Disponível em:
<<http://www.eps.ufsc.br/disserta/palomino/indice/index.html>>. Acesso em: 18 out. 2008.

PETERSON, James L. **Petri net theory and the TTP ing of systems.** New Jersey, USA: Prentice-Hall, 1981. <<http://www.informatik.uni-hamburg.de/TGI/PetriNets>>. Acesso em: 25 out. 2008.

PRESSMAN, Roger S. **Engenharia de software.** São Paulo: Pearson Makron Books, 2006.

SCOTT, Kendall. **O Processo unificado explicado – UML.** Porto Alegre: Ed Bookman, 2003.

SOMMERVILLE, Ian. **Engenharia de software.** 6.ed. São Paulo: Addison Wesley, 2003.

WFMC. **The workflow reference model.** 2006. Disponível em: <<http://www.wfmc.org>>. Acesso em: 06 nov. 2008.

APÊNDICE A – Projeto Exemplo para Utilização das Redes de Petri

SOFTWARE DE APOIO AO ENSINO COM O USO DE COMPUTAÇÃO GRÁFICA

O software em questão foi desenvolvido pelo bacharel em computação, Thiago Casagrande Darós, no ano de 2003, o qual permite ao usuário criar e manipular geometricamente figuras e objetos, no qual são desenhados na tela do sistema com o tamanho de 550x350 pixels, por meio de comandos ou pela utilização de botões disponibilizados na interface, proporcionando o entendimento de alguns tópicos que possam utilizar a computação gráfica como ferramenta de apoio. O enfoque deste sistema está em construir algumas formas geométricas, aplicando suas propriedades e quatro operações sobre elas: translação, rotação, escalamento e espelhamento.

As formas geométricas que podem ser desenhadas são: linha, retângulo, elipse, polígonos regulares com três lados no mínimo e dez no máximo. Pode-se também utilizar o lápis para desenho livre, a borracha para apagar em parte ou todo o esboço, a função limpa tela que apaga tudo o que foi feito com rapidez, dando início a um novo desenho, a opção mudar de cor para preto ou branco ou vice-versa e a função caixa de texto a qual permite escrever um texto com um número limitado de vinte caracteres.

As funções citadas como retângulo, elipse, linha, lápis e borracha, contam com recursos adicionais para desenho sendo estes a simetria e a espessura. No caso da última há possibilidade de optar pelo seu tamanho com um limite, inclusive no polígono regular.

A Figura 13 demonstra a tela principal do sistema imediatamente após sua execução, onde se pode notar que a linha, no canto superior esquerdo da figura, já inicia habilitada para o uso, sendo assim esta operação, mesmo não a utilizando, precisa ser concluída antes de o usuário optar por outra.

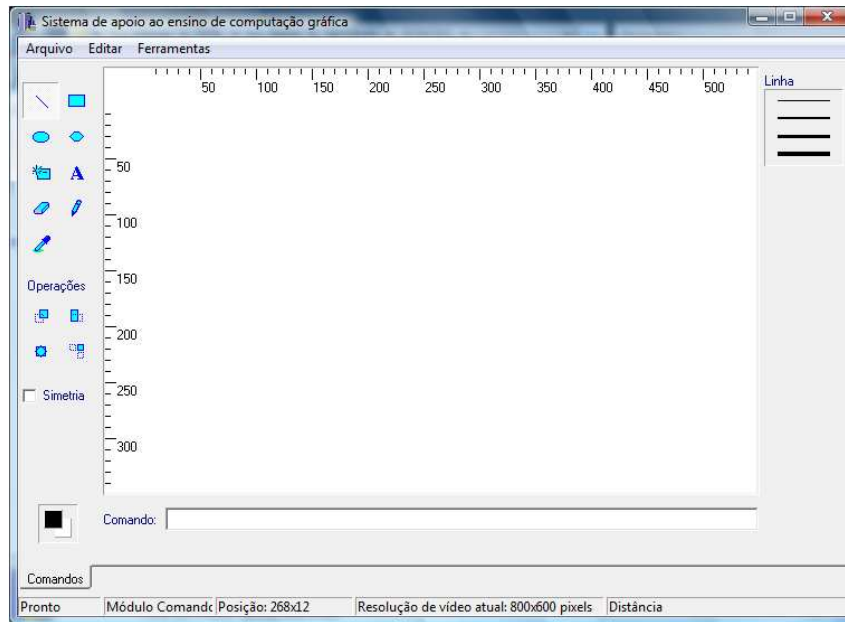


Figura 13. Tela principal do sistema
Fonte: DARÓS, T. C. (2003).

O Recall é uma função interessante deste sistema, pois possibilita retornar para situações anteriores, ou seja, cada comando executado pelo usuário é armazenado temporariamente pelo software para que seja possível, por exemplo, refazer um comando anterior ou uma seqüência deles, conforme pode-se ver na Figura 14.

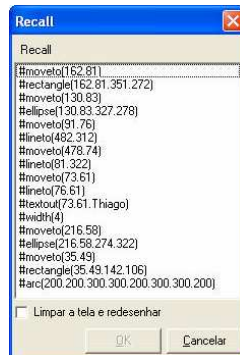


Figura 14. Função recall
Fonte: DARÓS, T. C. (2003).

Modelagem do Sistema

O software de apoio à aprendizagem com o uso de computação gráfica foi modelado em forma de diagrama de classes, no qual o seu desenho torna-se simples e entendível. A Figura 15 mostra a tela principal do sistema.

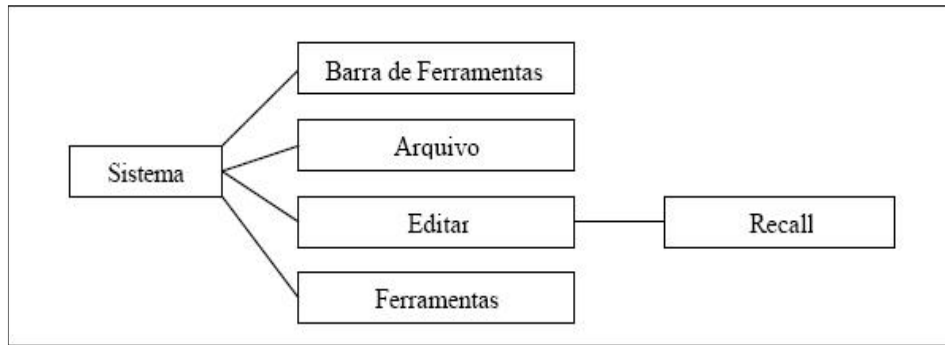


Figura 15. Tela inicial do sistema
Fonte: DARÓS, T. C. (2003).

A barra de ferramentas possui as operações geométricas que podem vir a serem aplicadas nos desenhos criados pelo usuário. A figura 16 demonstra o diagrama de classes com a localização destes recursos no sistema.

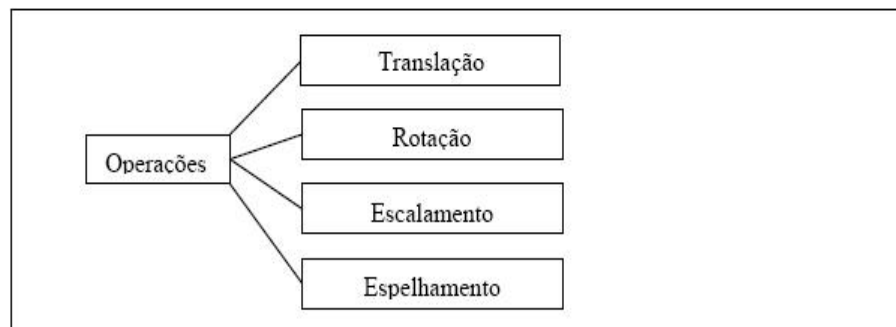


Figura 16. Operações geométricas
Fonte: DARÓS, T. C. (2003).

A opção ferramentas do menu principal, representado na Figura 17, possui itens, relacionados a preferência do usuário, como visualizar a régua, exibir ajuda, posição da caneta virtual e o item inverter ou girar um objeto na horizontal ou vertical.

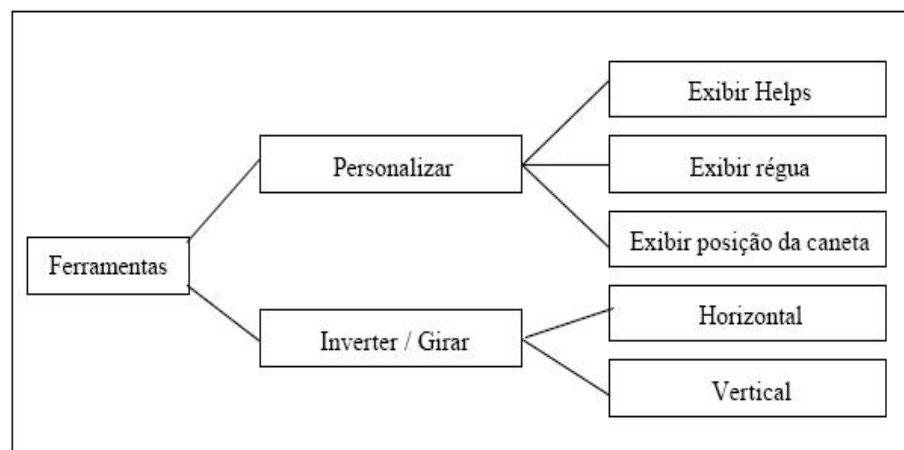


Figura 17. Desenho do item ferramentas do menu
Fonte: DARÓS, T. C. (2003).

As ferramentas de visualização da construção dos objetos estão localizadas na barra de ferramentas, conforme a Figura 18. Com elas o usuário é possibilitado a desenhar dentro do software utilizando o mouse do computador. Algumas ferramentas possuem o recurso adicional da simetria.

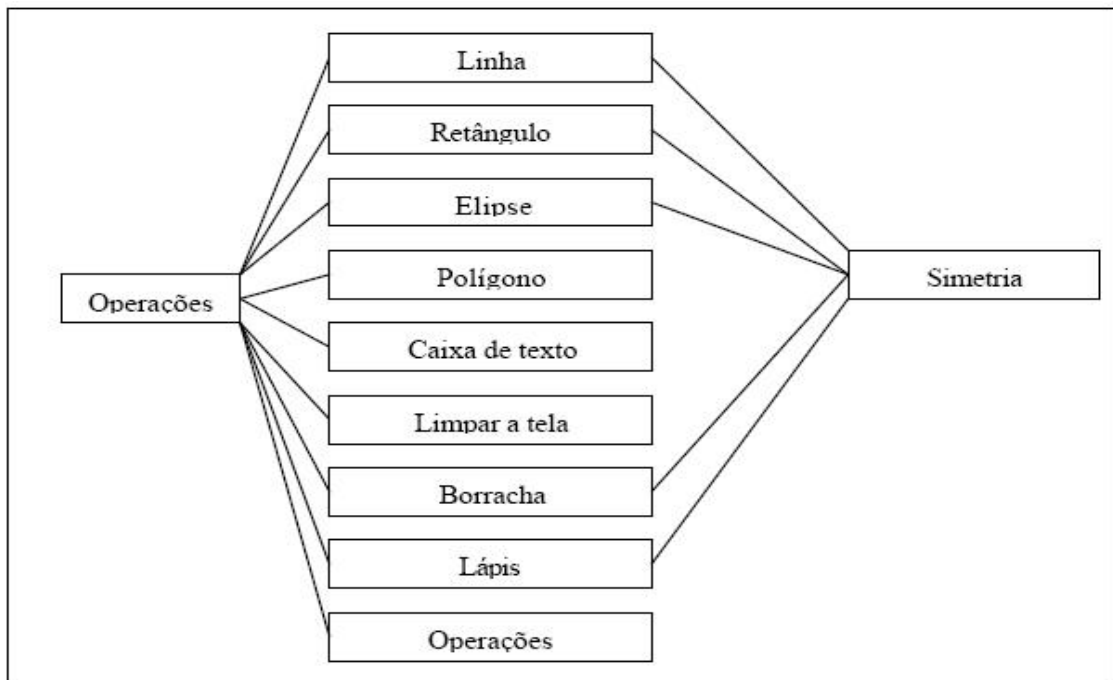


Figura 18. Modelagem da barra de operações
Fonte: DARÓS, T. C. (2003).

APÊNDICE B – Teoria dos Grafos

GRAFOS

A teoria dos grafos, conforme Cardoso e Valette o livro “Redes de Petri” da editora UFSC no ano de 1997, é um tema antigo com aplicações modernas. O primeiro resultado foi dado por Euler em 1736 com a solução do problema da cidade de Königsberg: como atravessar as sete pontes da cidade passando apenas uma única vez por cada uma.

Até 1946 a teoria dos grafos se restringe à Matemática; as contribuições mais importantes foram as dos matemáticos Kirchoff, Hamilton, Sylvester, Kempe, Lucas, Petersen e Tarry no século XIX e Poincaré, Sainte-Lagu, Kurutowski, Hall, Polya, Köning, Whitney, Tutte na primeira metade do século XX.

A *pesquisa operacional*, surgida das pesquisas militares ligadas à Segunda Guerra, provocou um desenvolvimento da teoria de grafos como modelo de sistemas concretos. Entre os grandes especialistas desta nova orientação pode-se citar Kühn, Dantzig, Roy, Faure, Kaufmann. Um novo salto foi dado nos anos 60 com o desenvolvimento das Ciências da Informação e da Comunicação, com contribuições de informáticos como Dijkstra, Knuth, Wirth, Sakharovitch, Gondran e outros (tanto no desenvolvimento de algoritmos quanto na modelagem de problemas).

No estudo de grafos, é importante conhecer os algoritmos adaptados aos principais tipos de problemas de grafos: procura do caminho de menor custo, passar uma única vez por todos os arcos do grafo, etc.

Definições Formais e Notação

Definição de grafo: um grafo ou grafo não ordenado $G = (V, E)$ consiste em um conjunto V de vértices (ou nós) e um conjunto E de arcos tal que cada arco $e \in E$ é associado com um par não ordenado de nós.

Definição de grafo direcionado: um grafo direcionado ou ordenado $G = (V, E)$ consiste em um conjunto V de vértices (ou nós) e um conjunto E de arcos tal que cada arco $e \in E$ é associado com um par ordenado de nós.

Definição de arco incidente: um arco (direcionado ou não) associado com um par de nós v e w é dito incidente em v e w , e os nós são ditos adjacentes.

Definição de arcos paralelos: os arcos são ditos paralelos se estão associados com os mesmos pares de vértices.

Definição de malha: uma malha é um arco incidindo num único vértice.

Definição de vértice isolado: considera-se um vértice isolado quando não existe nenhum arco incidente.

Definição de vértice pendente: o vértice é dito pendente quando existe apenas um arco incidente.

Definição de grafo simples: um grafo é simples quando se apresenta sem malhas nem nós paralelos.

Definição de grau: o grau de um nó é o número de arcos incidindo no nó. No caso de uma malha, contam-se dois.

Definição de aplicação multívoca: a aplicação multívoca “ T ” é definida por $T : V \rightarrow P(V)$, onde T_i é o conjunto de sucessores do nó i , e T_i^{-1} é o conjunto de predecessores do nó i .

Conectividade (Caminhos e Circuitos)

Caminhos e circuitos são definidos num grafo orientado.

Definição de caminho: considere v_0 a v_n de comprimento n é uma seqüência alternada de $n+1$ nós e n arcos começando em v_0 e terminando em v_n .

Definição de caminho elementar: um caminho elementar de v a w é um caminho que passa uma única vez por cada nó.

Definição de circuito: um circuito é um caminho de comprimento não-nulo de v a v que passa uma única vez por cada arco.

Definição de circuito elementar: um circuito elementar é um ciclo de v a v passando uma única vez por cada nó (exceto $v_0 = v_n = v$).

Cadeias e Ciclos

Cadeias e ciclos são definidos para grafos não orientados; cadeias correspondem aos caminhos e ciclos correspondem aos circuitos para grafos orientados nas definições acima.

Definição de fechamento transitivo: o fechamento transitivo da aplicação T é a aplicação T' definida por $T'_i = i \cup T_i \cup T_i^2 \dots \cup T_i^{n-1}$, sendo que T_i^k é o conjunto de nós que se pode atingir a partir do nó i por caminhos tendo exatamente k arcos. T'_i é o conjunto de sucessores de i . Do mesmo modo, T'^{-1}_i representa o conjunto de predecessores de i .

Definição de grafo conexo: um grafo é conexo se para qualquer par de nós v e w de G existe uma cadeia de v a w . Por exemplo, é possível construir um grafo representando os estados brasileiros, com um arco ligando os estados que possuem uma fronteira comum. Um

grafo conexo é único, enquanto que um grafo não conexo possui vários pedaços, chamados de subgrafos ou componentes.

Definição de subgrafo: $G' = (V', E')$ é um subgrafo de $G = (V, E)$ se: a) $V' \subseteq V$ e $E' \subseteq E$; b) $\forall e' \in E'$, se e' incide em v' e w' , então v' e $w' \in V'$ ((a restrição b) garante que G' é um grafo).

Definição de componente: dado um grafo G e um nó v de G , o subgrafo G' de G formado por todos os arcos e nós de G contidos nas cadeias começando em v é chamado componente de G contendo v . Um grafo é conexo se tiver um só componente. O grafo da Figura 19 é conexo, pois dado qualquer par de vértices v e w , existe um caminho de v a w . Já o da Figura 20 não é conexo; por exemplo, não há um caminho de v_2 a v_5 .

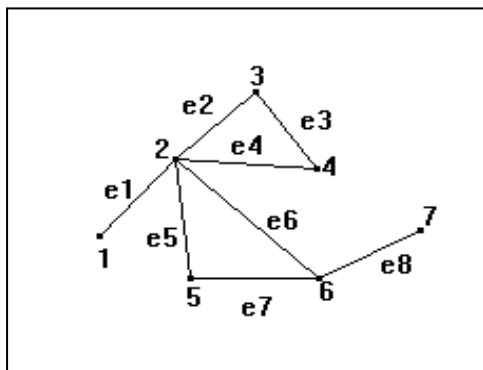


Figura 19. Grafo conexo
Fonte: Cardoso,J.: Valette,R. (1997) .

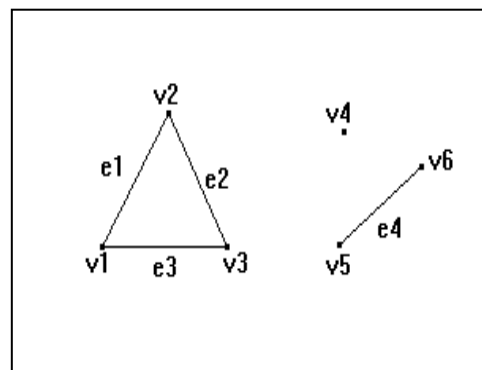


Figura 20. Grafo não conexo
Fonte: Cardoso,J.: Valette,R. (1997) .

Definição de grafo fortemente conexo: um grafo direcionado é fortemente conexo se para qualquer par de nós v e w do grafo subjacente existe uma cadeia de v e w . Na Figura 21, o grafo G é fortemente conexo (existe um caminho entre quaisquer pares de vértices). G também é fracamente conexo. Já na Figura 22, o grafo H não é fortemente conexo, pois não existe caminho entre a e b ; entretanto, é fracamente conexo, pois existe uma cadeia entre quaisquer pares de vértices no grafo subjacente.

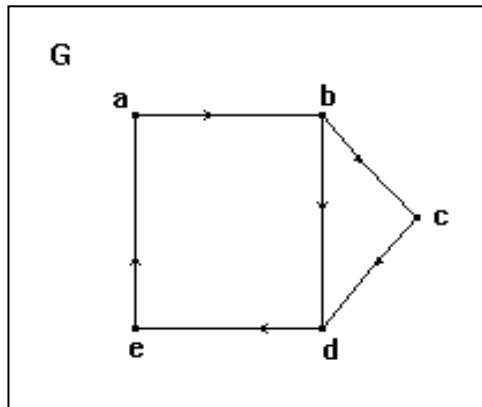


Figura 21. Grafo fortemente conexo
Fonte: Cardoso,J.: Valette,R. (1997) .

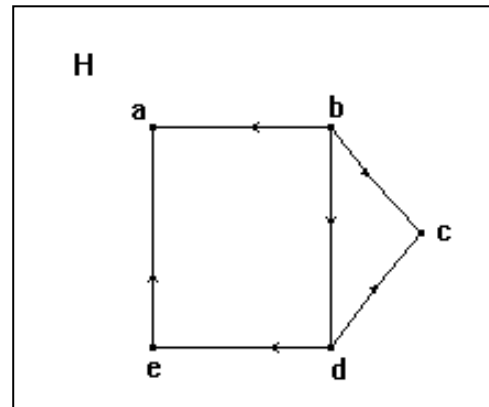


Figura 22. Grafo fracamente conexo
Fonte: Cardoso,J.: Valette,R. (1997) .

Definição do ciclo de Euler: um ciclo em G que inclui todos os arcos e todos os nós é um ciclo de Euler. Se G consiste em um nó v , sem arcos, o caminho (v) é chamado um ciclo euleriano em G . Se um grafo G possui um ciclo de Euler, então G é conexo e todos os nós têm grau par. Se o nó possui número ímpar de vértices incidentes, é impossível sair do nó sem passar pelo arco duas vezes. Se G é um grafo conexo e todos os nós têm grau par, então G tem um ciclo euleriano.

Em qualquer grafo, existe um número par de vértices com grau ímpar. Um grafo G possui um caminho sem arcos repetidos de v a w ($v \neq w$) contendo todos os arcos e nós se e somente se é conexo e v e w são os únicos nós com grau ímpar. Se G contém um ciclo de v a v , G contém um ciclo elementar de v a v .

Algoritmo de Construção de Ciclos de Euler

Procedimento

Euler (G : grafo conexo, $\forall v \in V, \text{grau}(v) = \text{par}$)

Ciclo := ciclo em G começando num nó qualquer

H := G com os arcos deste ciclo removidos

Enquanto H tem arcos

Início

Subciclo := ciclo em H começando num nó em H que também é um nó de ciclo

H := H com arcos de subciclo e nós isolados removidos

Ciclo := ciclo com subciclo inserido no nó apropriado

Fim { *ciclo* é um ciclo de Euler

ANEXO A – Foto com Jean Marie Farines na UFSC

Figura 23. Foto na UFSC