UNIVERSIDADE FEDERAL DE GOIÁS – UFG

CAMPUS CATALÃO – CaC

DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO – DCC

Bacharelado em Ciência da Computação

Projeto Final de Curso

Análise e Simulação do Mecanismo de Alocação de Recursos Humanos em Sistemas de Workflow Combinado com Algoritmos de Controle de Acesso Baseado em Papéis

Autora: Elaine Aires de Oliveira

Orientadora: Profª. Ms. Liliane do Nascimento Vale

Catalão - 2012

Elaine Aires de Oliveira

Análise e Simulação do Mecanismo de Alocação de Recursos Humanos em Sistemas de Workflow Combinado com Algoritmos de Controle de Acesso Baseado em Papéis

Monografia apresentada ao Curso de
Bacharelado em Ciência da Computação da
Universidade Federal de Goiás – Campus Catalão
como requisito parcial para obtenção do título de
Bacharel em Ciência da Computação

Área de Concentração: Engenharia de Software
Orientadora: Profª. Ms. Liliane do Nascimento Vale

Catalão - 2012

O., Elaine Aires

Análise e Simulação do Mecanismo de Alocação de Recursos Humanos em Sistemas de Workflow Combinado com Algoritmos de Controle de Acesso Baseado em Papéis/Elaine Aires de Oliveira- Catalão - 2012

Número de páginas: 70

Projeto Final de Curso (Bacharelado) Universidade Federal de Goiás, Campus Catalão, Curso de Bacharelado em Ciência da Computação, 2012.

Palavras-Chave: 1. Sistemas de *Workflow*. 2. Redes de Petri. 3. Alocação de Recursos Humanos. 4. Controle de Acesso Baseado em Papéis

Elaine Aires de Oliveira

Análise e Simulação do Mecanismo de Alocação de Recursos Humanos em Sistemas de Workflow Combinado com Algoritmos de Controle de Acesso Baseado em Papéis

Monografia apresentada e aprovada em _____ de _____
Pela Banca Examinadora constituída pelos professores.

Profª. Ms. Liliane do Nascimento Vale – Presidente da Banca

Prof. Dr. Vaston Gonçalves da Costa
Universidade Federal de Goiás

Profª. Ms. Valquíria Duarte
Universidade Federal de Goiás

Aos meus pais, Tarciso e Marleide, pessoas essencialmente importantes em minha vida

AGRADECIMENTOS

Em primeiro lugar agradeço a Deus, pela dádiva da vida, pela oportunidade dessa existência e pela proteção.

Aos meus amados pais, Tarciso e Marleide, pelo amor, dedicação, confiança, e cujo apoio incondicional se constituem para mim em motivação para continuar trilhando o caminho do conhecimento. Por me entenderem todas as vezes que fui passarinho e quis voar... Essa vitória é de vocês!

Às minhas queridas irmãs, Aline e Gabriella, pelo incentivo, paciência e carinho.

Ao meu lindo sobrinho, Gabriel, pelos momentos de profunda alegria.

Ao meu amigo fiel, Locke, pelo companheirismo.

Ao meu namorado Rafael, pelo amor, paciência, amizade, carinho, companheirismo e principalmente, por ser o melhor namorado do mundo!

Às minhas lindas amigas, Carla e Laísa, companheiras nesse ~~longo~~ período de graduação, dividindo frustrações e momentos de alegria. Ao meu amigo Salviano, pela colaboração no desenvolvimento deste trabalho. Ao Nélio, e aos demais colegas nerds do curso. Vocês fizeram parte de uma fase muito importante na minha vida e estarão sempre em minhas lembranças!

À minha professora e orientadora Liliane, por ter me auxiliado no desenvolvimento desse trabalho, e acima de tudo pelo profissionalismo, paciência e compreensão desempenhados a mim. Aos demais professores que contribuíram para a minha formação, por estimularem meu desejo de aprender.

Aos meus familiares e amigos pelo incentivo contínuo.

Sem mais, agradeço a todas as pessoas que me constituíram e me constituem no que fui, sou e serei. Muito Obrigada!

NEXT LEVEL!

"Você não consegue ligar os pontos olhando pra frente; você só consegue ligá-los olhando pra trás. Então você tem que confiar que os pontos se ligarão algum dia no futuro. Você tem que confiar em algo – seu instinto, destino, vida, carma, o que for. Esta abordagem nunca me desapontou, e fez toda diferença na minha vida."

Steve Jobs

RESUMO

Sistemas de Gerência de *Workflow* são empregados na coordenação e dinamização de processos de negócio, sendo utilizados em aplicações críticas e estratégicas no âmbito organizacional. Assim, a segurança tem se tornado um fator primordial em *Workflows*, visto que envolve a execução de mecanismos de segurança de controle de acesso objetivando garantir que as tarefas sejam executadas exclusivamente por usuários autorizados. Neste contexto, o controle de acesso baseado na distribuição de papéis aos recursos com a correspondente atribuição de privilégios, maximiza a segurança no desempenho de atividades, contribuindo para que as organizações possam alcançar seus objetivos. Além disso, colabora para a solução do problema de alocação de recursos, pois as atividades podem ser melhor distribuídas, conforme a característica de hierarquia existente entre os papéis. Para a formalização desta problemática é abordado o conceito de redes de Petri, que incorporam características temporais para a alocação de recursos no contexto de *Workflow* (as *Workflow-Nets*), e o algoritmo *RBAC (Role-Based Acess Control)* no critério de particionamento de papéis em sistemas de *Workflow*. Por fim, a validação do modelo é apresentada através da implementação do *software RdP Simulation*.

Palavras-Chaves: Sistemas de *Workflow*, Redes de Petri, Alocação de Recursos Humanos, Controle de Acesso Baseado em Papéis

ABSTRACT

Workflow Management Systems are used in coordinating and streamlining business processes and are used in critical applications and in organizational strategic. Thus, security has become a major factor in workflows, since it involves the implementation of security mechanisms designed to ensure access control tasks to be performed only by authorized users. In this context, the access control based on the distribution of roles to resources with a awarding of privileges, maximizing the performance of activities, helping organizations to achieve their goals. It helps to solve the problem of resource allocation, because the activities can be better distributed, as the characteristic hierarchy between roles. For the formalization of this problem is addressed the concept of Petri nets that incorporate temporal characteristics for resource allocation in the context of Workflow (the Workflow-Nets), and the algorithm RBAC (Role-Based Access Control) on the criterion of partitioning of roles Workflow systems. Finally, model validation is presented through the implementation of RdP Simulation software.

Keywords: Workflow Management System, Petri Net, Allocation of Human Resources, Role Based Acess Control

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Algoritmos

Lista de Siglas

API	*Application Programming Interface*
CSCW	*Computer-Supported Cooperative Work*
ISO	*International Standards Organization*
MIT	*Massachusetts Institute of Technology*
OMG	*Object Management Group*
RdPs	Redes de Petri
SGW	Sistema de Gerenciamento de *Workflow*
UML	*Unified Modeling Language*
WAPI	*Workflow API and Interchange Formats*
WfMC	*Workflow Management Coalition*

Capítulo 1

Introdução

Com o advento da globalização, as organizações deparam-se com novas oportunidades e desafios, adaptando-se para satisfazer as novas necessidades do mercado, fazendo o uso estratégico da tecnologia para o alcance de níveis de competitividade adequados. Assim, as empresas empregam recursos de computação, sistemas e ferramentas tecnológicas, que as auxiliam no processo de lidar com a informação de forma exata e com qualidade.

Frente a este cenário, o desenvolvimento de pacotes de *softwares* genéricos para gerenciamento de processos de negócios - denominados Sistemas de Gerenciamento de *Workflows* (SGWs) - são particularmente importantes, pois provê suporte computadorizado às automações procedimentais das organizações garantindo maiores índices de qualidade e eficiência no gerenciamento de seus processos de negócio. Sistemas de Workflow, ou de gerenciamento de fluxo de trabalho, correspondem a um conjunto de ferramentas que tem como finalidade a automação e gestão de processos fornecendo auxílio no alcance das metas empresariais.

A tecnologia da informação é concretizada pelas funcionalidades implementadas no sistema computacional. Entretanto, problemas decorrentes do processo de desenvolvimento de *software* podem provocar sérios impactos em uma empresa. No intuito de minimizar esses problemas são empregados princípios da Engenharia de *Software*, que visam disciplinar a produção de *softwares* a partir da introdução de atividades que permitem o gerenciamento de projetos e o emprego de ferramentas e métodos teóricos que auxiliem o desenvolvimento de sistemas computacionais [Sommerville, 2003]. Uma das técnicas de Engenharia de *Software* fundamentais durante o processo de desenvolvimento é a modelagem, através desta, é possível construir modelos que representam a estrutura e o comportamento desejado para o sistema.

O formalismo das redes de Petri (RdPs) é empregado neste trabalho para modelagem de um Sistema de *Workflow*. As redes de Petri são consideradas ferramentas gráficas e matemáticas de representação formal que possibilitam a modelagem, a análise e o controle de sistemas de processamento de informação. [Aalst and Hee, 2002] identificam três razões

principais para aplicação de redes de Petri na modelagem de *Workflow*: (1) as redes de Petri possuem semântica formal além da sua natureza gráfica; (2) permitem modelar explicitamente os estados do sistema e (3) existência de variedade e disponibilidade de técnicas de análise. Segundo [Merz, 1995], a principal vantagem de empregar RdP na modelagem de *Workflow* é a combinação de fundamentação matemática, representação gráfica compreensiva e possibilidade de simulações e verificações.

Além da modelagem baseada em rede de Petri, outro quesito importante no contexto de *Workflows* é a segurança. O serviço de segurança de autorização (controle de acesso) é de relevância primordial no contexto destes sistemas, pois assegura que a tarefa seja executada exclusivamente por usuários autorizados. Um modelo de autorização deve ser capaz de impedir a modificação desautorizada dos dados e também fornecer meios de reforçar o padrão legítimo das operações nos dados acessados para a execução de uma tarefa.

Em sistemas de *Workflow* são verificadas algumas deficiências referentes a ferramentas que promovem a segurança nestes sistemas e dessa forma, neste trabalho, o modelo de segurança utilizado como referência é o modelo RBAC (Role-Based Access Control - Controle de Acesso Baseado em Papéis), no critério de particionamento de papéis proposto em [Nyanchama and Osborn, 1994]. Assim, o modelo de autorização é aplicado em sistemas de *Workflows*, visando melhorias na administração do acesso de usuários nestes sistemas.

Em modelos de Sistemas de Gerenciamento de *Workflows* consideram-se, entre outros, mecanismos de alocação de recursos cumulativos e restrições temporais (intervalos de datas de execução das atividades). Nesse contexto, os recursos eventualmente precisam ser alocados a duas ou mais tarefas de casos diferentes em um mesmo instante de tempo para que se possa cumprir com o cronograma instituído. Considerando a inconsistência entre restrições de tempo e os períodos disponíveis dos recursos, a má alocação de recursos apropriados a determinadas atividades apresentam-se como um problema em SGWs, o que implica em atrasos no cronograma estabelecido e conseqüente prejuízo financeiro para a organização.

Em [Jeske, 2006] foi proposto um modelo de Rede de Petri p-temporal associado a conjuntos híbridos *fuzzy* como solução para o problema de alocação de recursos humanos em Sistemas de Gerenciamento de *Workflow*. Para expressar de forma realista o mecanismo de alocação de recursos onde o comportamento humano é considerado, foram definidos conjuntos *fuzzy* delimitados pela distribuição de possibilidade na forma triangular e associados com as marcações dos lugares para representar a disponibilidade humana.

O objetivo do presente trabalho é propor uma solução para o problema de alocação de recursos humanos e distribuição adequada de papéis aos recursos disponíveis baseado em um modelo de *Workflow-Net*, considerando o acoplamento de mecanismos de RBAC

disponíveis a este critério, no intuito de evitar que a sobrecarga de recursos humanos influencie no desempenho de suas atividades, garantindo que a distribuição de papéis e o controle de acesso dos mesmos sejam cumpridos. Dessa forma, é possível garantir um maior controle na qualidade e agilidade na execução de atividades. Assim, inclui-se a capacidade de expressar e impor uma política de segurança específica e simplificar o processo de gerenciamento de segurança em sistemas de *Workflow*.

No Capítulo 2 é apresentada uma revisão dos principais conceitos de Sistemas de *Workflow*.

No Capítulo 3 são apresentados os principais conceitos sobre redes de Petri (RdPs), abordando extensões como as redes de Petri com representação do tempo, e as *Workflow-Nets*, que consistem em RdPs especializadas na representação, validação e verificação de *Workflow*.

No Capítulo 4 são abordados os principais conceitos sobre o modelo de segurança RBAC, bem como o particionamento de papéis proposto em [Nyanchama and Osborn, 1994].

O Capítulo 5 descreve o estudo de caso, apresentando um protótipo para gerência de papéis e atividades com a finalidade de obter um cenário admissível para a validação do modelo de rede de Petri apresentado, além disso, considera a aplicação do algoritmo de particionamento de papéis a fim de verificar a aplicabilidade do mesmo em sistemas de *Workflow*.

Para concluir este trabalho, são apresentadas as considerações finais, assim como sugestões de trabalhos futuros no capítulo 6.

Capítulo 2

Sistemas de Gerenciamento de *Workflow*

O impacto do dinamismo decorrente do surgimento de novas tecnologias resulta em constantes variações no ambiente de negócios, em conseqüência exige-se uma maior capacidade de gerência das atividades nas organizações. Frente a este cenário, os Sistemas de *Workflows* apresentam-se como uma importante tecnologia, por oferecer um conjunto de soluções para atender a demanda das organizações por maiores índices de qualidade e eficiência no gerenciamento de seus processos de negócio.

Sistemas de *Workflow* objetivam a automação e gerência de processos. Estes sistemas correspondem a um conjunto de ferramentas que permitem o projeto e a definição de fluxos de trabalho, visando a eficiência do gerenciamento e do controle do trabalho, e a minimização do problema da coordenação das tarefas nos processos de negócios.

Este capítulo aborda uma revisão dos principais conceitos de sistemas de *Workflow*.

2.1 Contexto Histórico

Historicamente, Sistemas de Gerência de *Workflows* (SGWs) têm sua origem por volta dos anos setenta a partir de pesquisas em automação de escritório. Visando o fornecimento de soluções sobre como direcionar e compartilhar documentos, os processos passaram a ser, parcial ou totalmente, automatizados por sistemas computacionais, objetivando a redução na manipulação física de documentos em papel [Fischer and Moore, 1997].

Os sistemas de *Workflows* precursores não obtiveram aceitação. Grande parte dos sistemas projetados tornou-se inadequado aos objetivos das empresas devido à falta de flexibilidade, além disso, não havia tecnologia disponível como redes de computadores e profissionais qualificados, resultando no insucesso desses sistemas [Nicolao, 1998].

Na década de oitenta, os conceitos e paradigmas de trabalho em grupo antecedido pelas

pesquisas em *groupware*[1] e em CSCW (*Computer-Supported Cooperative Work* - Trabalho Colaborativo Suportado por Computador), exerceram influência na definição de sistemas de *Workflows* como ferramentas para a coordenação do trabalho em equipes. Na década de noventa, a tecnologia de sistemas de *Workflows* passou por uma crescente evolução devido ao desenvolvimento acelerado das infra-estruturas de redes de computadores. As atuais questões relacionadas ao processamento distribuído trouxeram novos desafios à definição de arquiteturas para sistemas de *Workflows*, levando às pesquisas em *Workflow* a um novo patamar voltado para a definição de arquiteturas distribuídas de execução de processos [Nicolao, 1998].

Assim, a tecnologia de *Workflow* tem evoluído no intuito de apoiar as novas necessidades de relacionamento e execução de atividades em organizações meio aos efeitos da revolução tecnológica, apresentando-se como uma solução capaz de aperfeiçoar a eficiência e a gestão dos processos organizacionais, fornecendo apoio automatizado às atividades no campo bancário e de seguros, departamentos governamentais, empresas de telecomunicações, de atendimento ao usuário e de gerência de documentos.

Sistemas de *Workflow* constituem-se em um componente padrão para sistemas de informação empresariais, visando atender a demanda das organizações por maiores índices de qualidade e eficiência no gerenciamento de seus processos organizacionais [Cruz, 2000].

2.2 Conceitos Sobre *Workflow*

Sistemas de *Workflow* são ferramentas que objetivam a automação e gestão de processos e que possibilitam o acompanhamento e distribuição de atividades que compõe o fluxo de trabalho ao longo de sua execução.

Diversos conceitos atribuídos a Sistemas de *Workflows* são identificados na literatura, mas, em geral, todos contemplam a idéia de que *Workflow* é direcionado a processos. As tecnologias de *Workflow* têm sido utilizadas para modelar e automatizar processos de negócio.

Processos de negócio consistem em um conjunto de atividades relacionadas que, coletivamente, visam atingir um objetivo, dentro do contexto de uma estrutura organizacional [Hollingsworth, 1985]. Nesse contexto, *Workflow* pode ser visto como a automação total ou parcial de um processo de negócio e consiste na representação do *Workflow* em um formato compreensível por uma máquina.

Conforme a Coalizão para Especificações de Gerenciamento de *Workflow* (WfMC), *Workflow* é a automação de um processo de negócio, por inteiro ou por partes, durante o qual documentos, informações e atividades são passadas de um participante para outro

[1]Conjunto de ferramentas cooperativas/colaborativas que possibilitam a interação entre múltiplos participantes.

para que estes desenvolvam ações respeitando um conjunto de regras [WfMC, 2010].

Nesse contexto apresentam-se os Sistemas de Gerenciamento de *Workflows* (*Workflow Management System* - SGWs) que são, em geral, ferramentas colaborativas que provêem a automação procedimental do gerenciamento de processos de negócios [Aalst and Hee, 2002], [Hollingsworth, 1985]. Estes sistemas apresentam-se como uma solução capaz de melhorar a eficiência e a gestão dos processos, permitindo o gerenciamento da sequência de atividades de trabalho, realizando chamadas ou invocando os recursos humanos e/ou eletrônicos apropriados, que são associados com as várias atividades que compõem os processos organizacionais.

Embora a definição de *Workflow* esteja tradicionalmente ligada a processos de negócios, seus princípios e ferramentas podem ser aplicados a múltiplas atividades onde seja exigida a coordenação do trabalho. A distribuição das atividades para pessoas, equipamentos ou sistemas computacionais pode ser feita automaticamente. Sistemas que manejam *Workflows* constituem-se em poderosas ferramentas para a organização de tarefas complexas e podem ser empregados em pesquisas científicas para comprovação dos princípios da teoria da relatividade, como a detecção de ondas gravitacionais [Gil, 2004], e também no seqüenciamento de DNA [Meidanis and Weske, 1996].

Segundo a [WfMC, 2010], um processo de negócio consiste em um conjunto de atividades relacionadas que visam atingir um objetivo de negócio, no contexto de uma estrutura organizacional. A descrição (modelo) de um processo a ser automatizado em um sistema de *Workflow* deve conter todas as informações necessárias sobre os processos a serem executados pelo sistema. Estas informações incluem dados sobre as atividades que compõe os processos, suas condições de início e finalização, regras para sua efetivação, usuários encarregados, documentos manipulados em cada atividade, aplicações a serem empregadas etc. [Cruz, 2000]. Apresentamos, a seguir, uma breve descrição dos componentes do sistema de *Workflow*.

Atividade

Em sistemas de *Workflow* uma tarefa é uma unidade lógica de trabalho que é executada integralmente por um recurso (humano e/ou máquina), o que denota que uma tarefa é indivisível e não pode ser interrompida. Assim, uma atividade consiste na execução de uma tarefa realizada por um recurso. Uma atividade corresponde a um conjunto de procedimentos que colaboram para que o processo alcance determinado objetivo [Araujo e Borges, 2001]. Uma Instância de atividade consiste na representação de uma única ocorrência de um processo, ou uma atividade em um processo. A execução de uma atividade pode ser desempenhada de forma manual, quando depende do manuseio do participante, ou automática, executada por uma aplicação ou dispositivo eletrônico.

Participante

Igualmente conhecido como ator, agente ou usuário de *Workflow*, o participante tem a capacidade de assumir um papel e realizar uma atividade durante a execução de uma instância do *Workflow*. Um papel consiste no conjunto de características e habilidades necessárias para executar determinada tarefa ou tarefas pertencentes a uma atividade. O participante pode ser tanto um recurso humano como um recurso computacional. Para recursos do tipo humano, um papel pode se referir a habilidades ou responsabilidades, por exemplo. Para equipamentos, um papel pode se referir a capacidades computacionais.

Alguns recursos utilizados em SGW podem ser considerados do tipo discreto, sendo representados por simples fichas. Por exemplo, uma impressora utilizada para tratar uma classe específica de documentos será representada como um recurso disjuntivo e pode ser alocada para um único documento em um mesmo momento. No entanto, alguns recursos não podem ser representados por uma simples ficha, como é o caso da maioria dos recursos do tipo humano, onde um funcionário pode tratar vários casos simultaneamente e não necessariamente de uma maneira puramente sequencial. Os mecanismos de alocação de recurso discreto e contínuo são apresentados formalmente no próximo capítulo, por envolverem o conceito de redes de Petri.

Regra

O desempenho de qualquer atividade em um ambiente organizacional implica na concordância e subordinação às regras pré-estabelecidas. A definição de um fluxo de trabalho adota um conjunto de regras para sua execução. As regras determinam quais informações transitarão pelo fluxo de trabalho e sob quais condições, ou seja, são atributos que definem como os dados que trafegam no fluxo de trabalho devem ser roteados, processados e controlados pelo sistema de *Workflow* [Cruz, 2000]. Regras descrevem restrições e diretrizes impostas por um negócio e/ou pela cultura de uma organização onde o processo será executado e se refletem no processo implementado no sistema de *Workflow*. As regras devem ser informadas ao SGW durante a modelagem do processo.

Rota

A rota consiste no caminho lógico que, determinado sob regras específicas, possui a função de transferência da informação dentro do processo. O encadeamento de atividades do processo (roteamento) acontece de um modo geral, segundo um grafo orientado onde os nodos representam atividades e as setas indicam a precedência de execução [Pereira e Casanova, 2003]. No roteamento são considerados quatro tipos básicos de rotas:

- rota serial: apresenta fluxo linear e direto onde uma tarefa é executada após a outra, havendo uma dependência entre as mesmas. Logo que uma atividade é fina-

lizada, a atividade subsequente é ativada e roteada para execução pelo participante responsável;

- rota paralela: grupo de atividades que possue execução simultânea;
- rota condicional ou alternativa: múltiplas rotas podem ser utilizadas e a escolha é feita por meio de determinada regra, procedimento ou variável;
- rota iterativa: a mesma atividade é executada várias vezes.

2.3 Caracterização de *Workflow*

Workflow pode ser classificado de diferentes formas. Destaca-se a classificação quanto à estruturação dos processos. Esta abordagem propõe três categorias para os diferentes tipos de aplicações baseados no grau de frequência e fluxos de trabalho [Cruz, 2000], [Georgakopoulos and Shet, 2004].

Workflow Ad hoc[2] *(Ad Hoc Workflow)*

Sistemas de *Workflow Ad hoc* apresentam um fluxo de trabalho pouco estruturado. As tarefas e o fluxo de interação entre elas são geralmente imprevisíveis ou desconhecidas até o momento da execução. Segundo [Georgakopoulos and Shet, 2004], estes sistemas estão voltados para grupos dinâmicos que desempenham processos únicos e altamente individualizados e não possuem capacidades de segurança e tratamento de grandes volumes de dados, portanto, não são recomendados para automatização de processos críticos[3] [Silva, 2001]. A ordenação e a coordenação de tarefas em um *Workflow* do tipo *Ad hoc* não são automatizadas, portanto necessitam da coordenação ou co-decisão humana. Exemplos destes tipos de processos são aqueles que abrangem a produção de conhecimento como o processo de desenvolvimento de um *software*, a elaboração de um relatório ou a revisão de um livro ou artigo onde não se conhece previamente o revisor [Nicolao, 1998].

Workflow Administrativo (Administrative Workflow)

Sistemas de *Workflow* Administrativo gerenciam processos com maior grau de estruturação. Essa categoria de *Workflow* trata tarefas administrativas rotineiras, estabelecendo fluxos de informação e realizando o roteamento inteligente de formulários por meio da corporação. *Workflows* do tipo administrativo apóiam processos administrativos que alteram bastante de uma organização para outra e a ordenação e organização de tarefas

[2]Ad Hoc é uma expressão em latim que significa "para isto", "para um fim específico".

[3]Sistemas críticos correspondem a sistemas particulares e com exigências muito elevadas em termos de confiabilidade e segurança, cujas características possuem riscos inerentes a danos físicos, pessoais e financeiros.

podem ser facilmente automatizadas. Esta classe de *Workflow* não requer acesso à sistemas de informação múltiplos e não engloba um processamento complexo de informações [Georgakopoulos and Shet, 2004]. Como exemplo tem-se um pedido de compra de materiais, que ordinariamente é requerido por uma pessoa uma vez por semana e tramita por diferentes áreas que vão gradualmente preenchendo as informações no documento eletrônico até o término do processo.

Workflow de Produção (Production Workflow)

Sistemas de *Workflow* de Produção envolvem processos de negócios repetitivos e previsíveis e que seguem uma estruturação rígida e bem definida. Destacam-se por serem bastante estruturados. Tipicamente estão envolvidos com processos de informações complexas, onde se verifica a necessidade de acesso a múltiplos sistemas de informação. Sistemas de *Workflow* de produção automatizam processos de negócios complexos que toleram elevado volume de dados, assim, esses sistemas tendem a ser executados em grandes corporações, envolvendo ambientes e aplicativos heterogêneos [Georgakopoulos and Shet, 2004]. Como exemplo tem-se a avaliação de empréstimos e o processo de requisição de seguros.

2.4 *Workflow* Management Coalition - WfMC

A *Workflow Management Coalition (WfMC)* é uma organização internacional sem fins lucrativos, formada por um conjunto de empresas e grupos de pesquisa, que objetiva a padronização da tecnologia de *Workflow*. Fundada em 1993, visa promover o uso de sistemas de *Workflow* através do estabelecimento de terminologias e padrões de *software*, de interoperabilidade[4] e de conectividade entre os sistemas de *Workflow*.

Em 1995, a WfMC publicou seu Modelo de Referência que vem sendo adotado por grande parte dos desenvolvedores da tecnologia de *Workflow*. Esse Modelo de Referência identifica os componentes que podem se comunicar e as respectivas interfaces e formatos de intercâmbio imprescindíveis para alcançar a interoperabilidade. O Modelo de Referência é apresentado na Figura 2.1.

O Modelo define cinco interfaces entre componentes e uma interface sobre o SGW, denominada *WAPI (Workflow API and Interchange Formats)*. Esta interface trata da série de construções pelas quais os serviços de gerência de *Workflow* podem ser empregados. Assim, os serviços de *Workflow* podem ser implementados de diversas maneiras, contanto que sejam fornecidas interfaces que traduzam a implementação particular de cada produto de *Workflow* para a interface padronizada pela WfMC. Onde:

[4]Capacidade de um sistema (informatizado ou não) se comunicar de forma transparente (ou o mais próximo disso) com outro sistema (semelhante ou não).

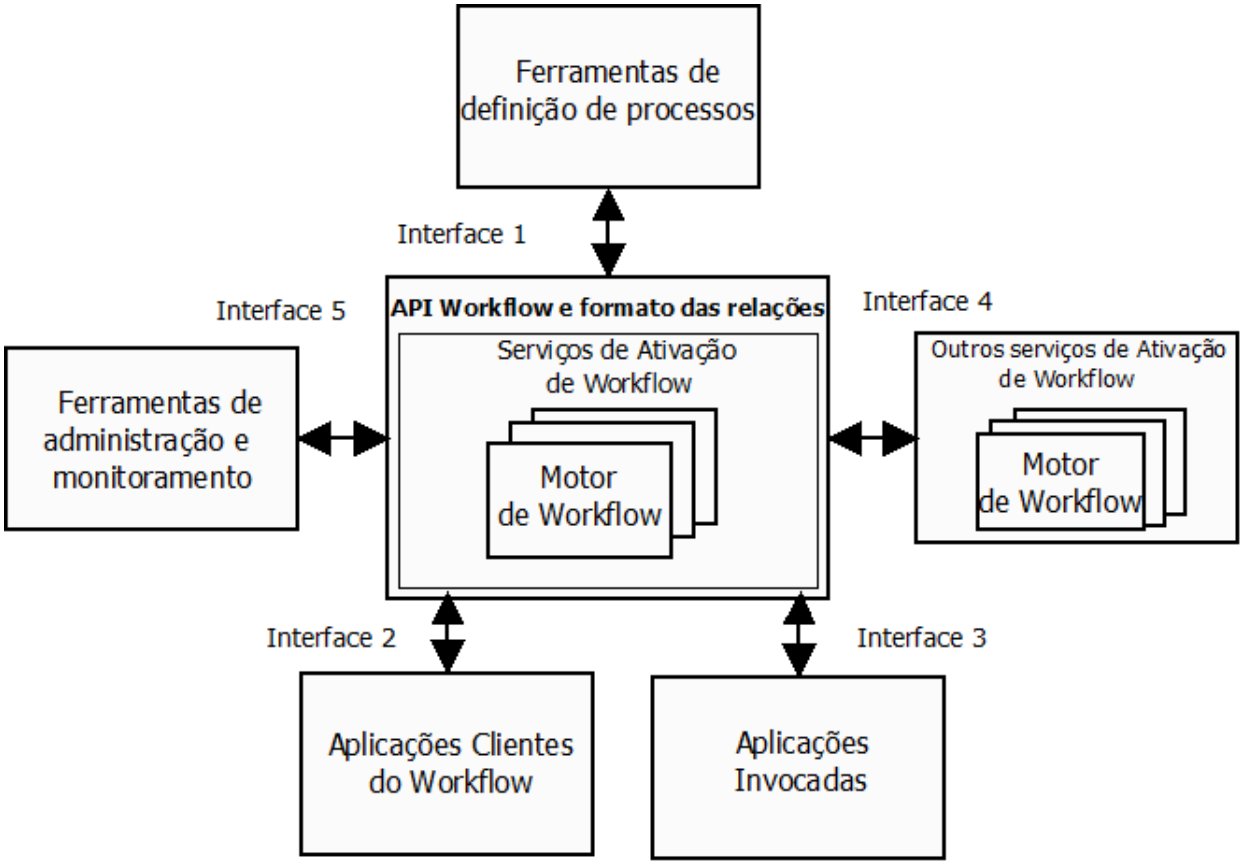

Figura 2.1: Modelo de Referência de *Workflow* - Componentes e Interfaces [WfMC, 2010]

- a Interface 1 (Ferramentas de Definição de Processos - *Process Definition Tools*) abrange a definição de processos através da WPDL (*Workflow Process Definition Language*), que consiste na linguagem padrão para a definição de processos e que possibilita exportar uma definição de processo, criada em uma determinada ferramenta de modelagem para qualquer SGW que suporte este padrão [WfMC, 2010].

- a Interface 2 (Aplicações Clientes do *Workflow* - *Workflow Client Applications*) define uma API que permite que aplicativos cliente de *Workflow* possam invocar serviços de gerência de *Workflow* de forma padrão. Assim, um aplicativo construído por terceiros pode empregar os serviços de gerência de *Workflow* de qualquer SGW que suporte este padrão. Atualmente, esta interface inclui chamada a serviços como recuperação de itens de trabalho e declaração de início e conclusão de atividades [WfMC, 2010].

- a Interface 3 (Aplicações Invocadas - *Invoked Applications*) trata da comunicação com aplicativos externos invocados pelo serviço de gerência de *Workflow* para a execução de atividades específicas. Em uma situação onde determinada atividade necessita acessar um sistema de gerência de banco de dados ou um sistema de correio eletrônico, por exemplo, essa interface promoveria a padronização da interface

com esses sistemas sistemas, envolvendo serviços de notificação e aspectos como a passagem e o retorno de parâmetros [WfMC, 2010].

- a Interface 4 (Outros serviços de ativação de *Workflow - Other Workflow Enactment Service(s)*) aborda a comunicação entre múltiplos sistemas de gerência de *Workflow* quando os mesmos estão envolvidos na administração de partes de um mesmo processo. Através desta interface é possível executar um processo por meio de diferentes SGWs. Assim, várias corporações poderiam fazer parte de um único processo, por exemplo, maximizando o grau de coordenação e levando os benefícios da tecnologia de *Workflow* para o campo inter-organizacional [WfMC, 2010].

- a Interface 5 (Ferramentas de Administração e Monitoramento - *Administration and Monitoring Tools*) envolve ferramentas de monitoramento e administração dos processos executados. Essa interface explicita quais dados devem ser gerados pelo SGW durante a execução dos processos, possibilitando a consulta por meio de ferramentas e a realização de análises [WfMC, 2010].

O SGW caracteriza-se por suportar aplicações em três áreas funcionais [Hollingsworth, 1985]:

- Funções de tempo de construção (*build-time functions*): relacionadas com a definição e modelagem do processo de negócio e das atividades que constituem este processo;

- Funções de controle em tempo de execução (*run-time control functions*): abordam o gerenciamento de processos de *Workflow* em um ambiente operacional e a sequenciação apropriada das várias atividades a serem manipuladas como itens de cada processo;

- Funções de interação em tempo de execução (*run-time interaction functions*): tratam do gerenciamento da interação de usuários com diferentes sistemas de tecnologia da informação, .

SGWs determinam que os projetos referentes à sua adoção e utilização apresentem uma coordenação perfeita entre os processos de negócios e a tecnologia. No entanto, para que o projeto seja bem sucedido, toda a metodologia de implementação deve ser conduzida levando em consideração diferentes aspectos de ordem institucional. Portanto, as organizações devem apostar não só no desenvolvimento e implementação, mas também na modelagem dos processos de negócios.

2.5 Técnicas de Modelagem de *Workflow*

A técnica de modelagem consiste na descrição do processo através de modelos que possam representar suas principais características. Um modelo de *Workflow* consiste na representação gráfica ou textual de um conjunto de atividades e o relacionamento existente entre estas [Araujo e Borges, 2001]. Na literatura são encontrados diversos trabalhos relacionados à modelagem de sistemas de *Workflow*, porém, cada qual emprega sua própria técnica de modelagem, isto é, não se verifica a existência de um modelo conceitual comum aceito e empregado pelos desenvolvedores da área.

Um dos padrões utilizados para representação gráfica de *Workflow* é o modelo Casati [Casati et al., 1995], que emprega gráficos, símbolos e textos para descrever as tarefas envolvidas e especificar os mecanismos de disparo e término das ações previstas. Diagramas de atividades da UML também podem ser empregados para modelagem de sistemas de *Workflow* [OMG, 2010]. Tanto o modelo Casati quanto os diagramas de atividades da UML apresentam a vantagem de mostrar os principais roteiros existentes em um *Workflow*. Contudo, não existe a possibilidade de representação dos mecanismos de alocação de recursos e restrições temporais quando são utilizados para a especificação de características de tempo real nos SGWs.

As Redes de Petri são apropriadas para modelar sistemas de *Workflow* uma vez que possibilitam uma boa representação de estados de conflito, compartilhamento de recursos, exceções de precedência, comunicações síncronas e assíncronas e explícitas restrições temporais, no caso das Redes de Petri temporais. [Aalst and Hee, 2002] apresentam diferentes razões para o emprego de Redes de Petri na modelagem de processos de *Workflow*, as quais são: existência de uma semântica formal, natureza gráfica, poder de expressividade, existência de propriedades, técnicas de análise, etc. Além disso, acrescem que a utilização de tal formalismo apresenta importantes benefícios, como o fato de forçar uma definição precisa dos processos de negócios, além de processar quais características (como ambiguidade, incerteza e contradições) são prevenidas em contraste com outras técnicas de diagramação informais. Segundo [Merz, 1995], a principal vantagem de empregar RdPs na modelagem de *Workflow* é a combinação de fundamentação matemática, representação gráfica compreensiva e possibilidade de simulações e verificações.

Modelos fundamentados em redes de Petri foram definidos exclusivamente para a representação de *Workflow*: as *Workflow-Nets (WF-Nets)*, com um lugar de entrada (*Start*) e um lugar de saída (*End*) sinalizando o início e o fim do processo de negócio modelado. Verifica-se que essas redes são apropriadas para representação, validação e verificação de *Workflow* [Aalst and Hee, 2002]. O modelo *Workflow-Net* será utilizado neste trabalho para a modelagem de um processo de *Workflow*. Tais redes são apresentadas em detalhes no próximo capítulo.

Capítulo 3

Redes de Petri

As redes de Petri (RdPs) são consideradas ferramentas gráficas e matemáticas de representação formal que possibilitam a modelagem, a análise e o controle de sistemas de processamento de informação que se caracterizam como concorrentes, assíncronos, paralelos e não determinísticos [Murata, 1989].

As redes de Petri oferecem suporte à verificação da corretude do sistema especificado, além disso, proporcionam um bom nível de abstração em relação a outros modelos gráficos. O emprego das redes de Petri para modelagem e especificação do comportamento dinâmico de sistemas é efetivado por diferentes razões: fornecimento de técnicas de descrição gráfica de simples entendimento, disponibilidade de ferramentas (*softwares*) que possibilitam a análise de redes de Petri, e a existência de extensões, como redes de Petri Coloridas, Hierárquicas e Temporais, que comportam um ganho de flexibilidade e características adicionais [Aalst and Hee, 2002].

Este capítulo contempla os fundamentos teóricos referentes à redes de Petri.

3.1 Contexto Histórico

O conceito de redes de Petri foi introduzido em 1962 por Carl Adam Petri em sua tese de doutorado, *Kommunikation mit Automaten* (Comunicação entre autômatos), submetida à Faculdade de Matemática e Física da Universidade Técnica de Darmstadt na Alemanha [Petri, 1962]. Desde o princípio, RdPs tinham como escopo a modelagem de sistemas com componentes concorrentes.

As aplicações iniciais em RdPs ocorreram em 1968, no projeto norte-americano *Information System Theory* (Teoria de Sistemas de Informação), da A.D.R. (*Applied Data Research, Inc.*). Este trabalho apresentou grande parte da teoria inicial, da notação e da representação de RdPs e destacou como as mesmas poderiam ser empregadas na modelagem e na análise de sistemas com componentes concorrentes.

A década de 70 marcou o progresso da teoria de RdPs e a ampliação de sua área

de aplicação. O Grupo de Estruturas Computacionais do Projeto MAC, do MIT (*Massachusetts Institute of Technology*), sob influência dos trabalhos de Petri, produziram consideráveis pesquisas e publicações sobre RdPs. Nessa década, surgiram RdPs apropriadas para a modelagem de características temporais determinísticas, como as RdPs temporal introduzidas por [Merlin, 1974], RdPs temporizadas [Ramchandani, 1974] e o modelo de RdPs p-temporizadas [Sifakis, 1977].

Na década de 80 houve um avanço considerável nas aplicações de RdPs devido ao surgimento das redes de Petri de alto nível, como por exemplo, as RdPs Coloridas definidas por [Jensen, 1990] e as RdPs Predicado-Transição definidas por [Genrich et al., 2000]. Esses avanços acrescentaram uma grande força descritiva ao processo de modelagem através do uso de marcas com identidade para representação da dinâmica dos sistemas modelados. Na década de 90, devido à variedade de extensões que estavam sendo propostas, houve o esforço para a padronização das redes de alto nível junto à ISO, bem como a definição de uma sintaxe de transferência comum às diferentes extensões de RdPs.

Atualmente, as redes de Petri se mostram adaptadas a um amplo número de aplicações em que as noções de eventos e de evoluções simultâneas são necessárias. No contexto das aplicações pode-se citar: avaliação de desempenho, análise e verificação formal em sistemas discretos, protocolos de comunicação, concepção de software tempo real e/ou distribuído, sistemas de transporte, logística, gerenciamento de base de dados, interface homem-computador e multimídia [Cardoso et al., 1997].

Informações atualizadas sobre assuntos relativos a redes de Petri e outros projetos podem ser obtidos no site "Petri Nets World" [Petri Nets World, 2011].

3.2 Conceitos Fundamentais das Redes de Petri

Rede de Petri ou rede de transição constitui formalmente em uma ferramenta gráfica e matemática de representação formal que permite a modelagem, a análise e o controle de sistemas a eventos discretos[1] que comportam atividades paralelas, concorrentes e assíncronas.

As redes de Petri permitem modelar o estado do sistema num nível maior de granularidade do que é fornecido pela teoria convencional de autômatos, possibilitando a representação facilitada de paralelismo e concorrência, e também às relações de causalidade entre partes do sistema.

Sua estrutura é composta por dois elementos estruturais: os lugares e as transições. Arcos direcionados e ponderados ligam os lugares às transições e às transições aos lugares,

[1] "Sistema a eventos discretos são sistemas modelados de tal sorte que as variáveis de estado variam bruscamente em instantes determinados e que os valores das variáveis nos estados seguintes podem ser calculados diretamente a partir dos valores precedentes e sem ter que considerar o tempo entre estes dois instantes" [Cardoso et al., 1997].

associados em cadeia para criar o modelo. Estes arcos podem ser rotulados com um valor inteiro positivo, enumerando o peso do arco. Entretanto, não existe a necessidade de se rotular um arco unitário. Um arco de peso k pode ser interpretado como k arcos paralelos de peso unitário [Murata, 1989]. Uma rede de Petri é definida pelos componentes que se seguem, conforme a Figura 3.1.

- Lugar (representado graficamente por círculos ou elipses): corresponde a uma variável de estado do sistema. Representa uma condição, uma atividade ou recurso;
- Transição (representada graficamente por barras ou retângulos): representa um evento, ou seja, é responsável pela realização de mudanças de estado do sistema;
- Arco (representado graficamente por setas): elemento que conecta lugares à transições, ou vice-versa, encadeando condições e eventos;

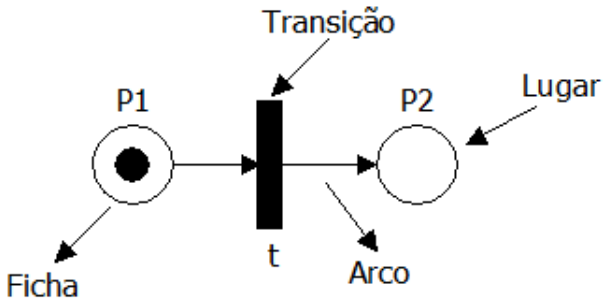

Figura 3.1: Elementos que Representam Graficamente uma RdP

- Marca ou Ficha (representado graficamente por um ponto em um lugar): representa o estado do sistema. O posicionamento dessas fichas em alguns lugares do grafo constitui a marcação. Sua evolução permite modelar o comportamento dinâmico do sistema. Pode representar um objeto, recurso ou peça em uma posição.

A definição de uma rede de Petri autônoma é dada por [Cardoso et al., 1997]:

Definição 2.1: *Formalmente uma rede de Petri pode ser definida como uma quádrupla,*

$$R = (P,\ T,\ Pre,\ Pos)$$

Onde:

- *P é um conjunto finito de lugares de dimensão n,*
- *T é um conjunto finito de transições de dimensão m,*

- *Pre:* $(P \times T) \rightarrow N$ *é a aplicação de entrada (lugares precedentes ou incidência anterior), indica o peso do arco, ligando um lugar P a uma transição t, sendo N o conjunto dos números naturais, os quais representam o número de fichas associados aos lugares,*

- *Pos:* $(T \times P) \rightarrow N$ *é a aplicação de saída (lugares seguintes ou incidência posterior), indica o peso do arco, ligando uma transição t a um lugar P.*

A partir dos elementos $a_{ij} = Pre(p_i, t_j)$ que indica o peso do arco ligando o lugar de entrada p_i a transição t_j, define-se uma matriz de incidência anterior *Pre*, de dimensão nxm, onde o número de linhas é igual ao número de lugares e o número de colunas é igual ao número de transições. Do mesmo modo, a matriz de incidência posterior *Pos* é definida a partir dos elementos $b_{ij} = Pos(p_i, t_j)$ [Cardoso et al., 1997].

Para as redes de Petri marcadas a seguinte definição é apresentada:

Definição 2.2: *Uma rede marcada N é uma dupla* $N =< R, M >$ *onde:*

- *R é uma rede de Petri,*

- *M é a marcação inicial dada pela aplicação M:* $P \rightarrow N$.

M representa a distribuição das fichas nos lugares e pode ser representado por um vetor de inteiros positivos ou nulos, cuja dimensão é o número de lugares da rede. Um exemplo de rede de Petri marcada é dado na Figura 3.2, onde a marcação é dada por M^T = [1 0 3 0 1] (M^T é o vetor transposto).

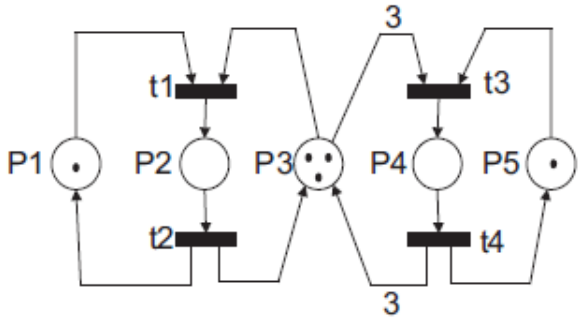

Figura 3.2: Exemplo de uma Rede de Petri Marcada

Formalmente, a evolução dinâmica de uma rede de Petri é dada pelas definições que se seguem:

Definição 2.3: *Uma transição t está sensibilizada ou habilitada se e somente se:*

$$\forall p \in P, M(p) \geq Pre(p,t) \tag{3.1}$$

Uma transição está sensibilizada, se o número de fichas em cada um dos seus lugares de entrada for maior (ou igual) ao peso do arco que liga este lugar à transição.

Definição 2.4: *Se t está sensibilizada por uma marcação M, uma nova marcação M' pode ser obtida através do disparo de t de maneira que:*

$$\forall p \in P, M'(p) = M(p) - Pre(p,t) + Pos(p,t) \tag{3.2}$$

O disparo de uma transição t consiste em remover as fichas dos lugares de entrada (Pre(p,t)), e depositar fichas em cada lugar de saída (Pos(p,t)). Um exemplo de disparo de transição é dado na Figura 3.3. A notação matricial dessa rede é dada por:

$Pre(p_1, t_1) = 1$
$Pre(p_2, t_1) = 1$
$Pre(p_3, t_1) = 0$
$Pre(p_4, t_1) = 0$

$Pos(p_1, t_1) = 0$
$Pos(p_2, t_1) = 0$
$Pos(p_3, t_1) = 1$
$Pos(p_4, t_1) = 1$

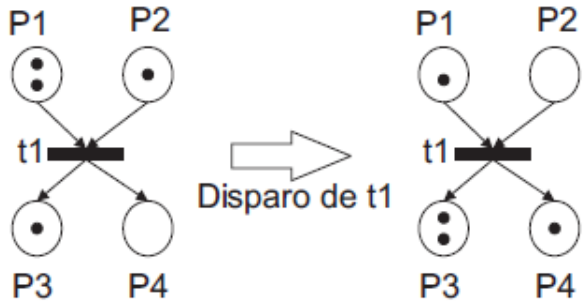

Figura 3.3: Exemplo de Disparo de uma Transição

As variações de lugares das fichas, que sucedem na Figura 3.3, representam o comportamento dinâmico do sistema modelado. Vale ressaltar que as interpretações dos lugares e fichas são variadas, ou seja, dependem do contexto em que são criados. Segundo [Cardoso et al., 1997] os lugares e as fichas podem ser empregados para descrever entidades abstratas como condições ou estados, e também entidades físicas como peças ou depósitos.

O disparo em sequência de transições é chamado de sequência de disparo. O vetor s é

denominado vetor característico da sequência s, onde cada componente $s(t)$ representa o número de ocorrências da transição t. A dimensão do vetor característico s é dada pelo número de transições da rede de Petri.

A utilização de meios analíticos possibilitam a verificação das propriedades de uma rede de Petri autônoma [Cardoso et al., 1997]. Algumas propriedades são dependentes da marcação inicial e são agrupadas sob o nome genérico de *boas propriedades*, que são:

Definição 2.5 ***Alcançabilidade****: uma marcação M_n é dita alcançável se existe uma sequência finita de disparo de transições que possibilita a chegada a M_n a partir da marcação inicial M_0 . Essa propriedade garante que determinados estados sempre serão atingidos.*

Definição 2.6 ***Limitabilidade****: uma rede é limitada ou k-limitada se o número de fichas em cada lugar não excede um número finito k para qualquer marcação alcançável a partir da marcação inicial.*

Definição 2.7 ***Vivacidade****: uma rede é considerada viva se toda transição t pode ser sensibilizada a partir de qualquer marcação M' do grafo de marcações alcançáveis. Esse conceito, na prática, garante que o sistema será livre de bloqueios (deadlock free).*

Definição 2.8: ***Reiniciabilidade****: uma rede de Petri é reiniciável se a partir de qualquer marcação acessível, existe uma sequência de disparo que leva à marcação inicial M_0.*

Existem também *propriedades estruturais* [Cardoso et al., 1997], que são dependentes da estrutura da rede e não da marcação inicial. Estas propriedades são definidas por meio dos componentes conservativos de lugar e dos componentes repetitivos estacionários. Esses componentes possibilitam a definição de invariantes de lugar, que fornecem informações sobre a dinâmica da rede. Uma apresentação detalhada das propriedades e dos algoritmos de verificação das propriedades das redes de Petri encontra-se em [Murata, 1989].

Na seção seguinte é apresentado um breve resumo de algumas extensões de redes de Petri de alto nível.

3.3 Classes de Redes de Petri

O formalismo das redes de Petri têm sido empregadas no domínio das mais variadas aplicações. A grande diversidade de sistemas existentes apresentam características distintas que necessitam ser representadas de forma clara e sem ambiguidades. Conforme a complexidade do sistema a ser modelado se verifica a necessidade da utilização de outros

modelos de redes de Petri que forneçam mais recursos do que o simples modelo autônomo.

As RdPs podem ser classificadas em: redes de baixo nível e redes de alto nível. As RdPs de baixo nível são caracterizadas pelo tipo de marcação que possuem. As marcas (fichas) que se encontram nos lugares da rede não possuem significado, indicam apenas o estado do sistema. As RdPs de alto nível são aquelas cujas marcas incorporam alguma semântica, viabilizando sua diferenciação. Algumas extensões visam à inclusão de hierarquias e de aspectos temporais às redes de Petri. São apresentados, a seguir, alguns tipos de redes de Petri de alto nível:

- Redes de Petri Coloridas: são associadas cores às fichas com o objetivo de diferenciá-las, o que permite representar diferentes processos e recursos [David and Alla, 2004], [Jensen, 1990]. Em [Moncelet et al., 1998], por exemplo, foi utilizado um modelo de rede de Petri Colorida para a modelagem de sistemas de produção.

- Redes de Petri Predicado/Transição: descrevem de maneira estruturada o conjunto *controle* e *dados* [Cardoso et al., 1997], [Genrich et al., 2000]. Assim, os lugares são chamados de predicados, as marcas representam condições válidas do predicado e as transições são consideradas como regras da lógica de primeira ordem, isto é, regras como variáveis. Em [Champagnat et al., 1998], por exemplo, as redes de Petri predicado/transição são integradas a uma sequência de equações diferenciais e algébricas para a modelagem de uma estocagem de gás.

- Redes de Petri Estocásticas: é incluído um tempo aleatório associado ao disparo de uma transição [Florin and Natkin, 1984]. Estas redes podem ser aplicadas em sistemas cujo tempo para ocorrência de eventos não são bem definidos, como por exemplo, o tempo entre a falha de uma máquina e outra.

- Redes de Petri Híbridas e Redes de Petri Contínuas: em uma rede de Petri contínua, a marcação de um lugar e a taxa de disparo corresponde a uma variável contínua (número real não negativo). A marcação contínua é transferida de um lugar para outro respeitando certa velocidade de disparo (fluxo de marcas contínuo transferido de um lugar para outro). As redes de Petri híbridas são modelos que apresentam tanto uma parte discreta quanto uma parte contínua. Uma rede de Petri híbrida pode conter lugares discretos e contínuos e transições discretas e contínuas [David and Alla, 2004].

- Redes de Petri a Objetos: podem ser consideradas como uma extensão da rede de Petri predicado/transição no contexto de uma abordagem a objetos [Silbertin, 1985]. A principal motivação dessa modelagem se deve à capacidade das redes de Petri para representar concorrência, controle de fluxo e restrições, e ao mesmo tempo beneficiar-se da modularidade da orientação a objetos.

- Redes de Petri temporais/temporizadas: são empregadas em sistemas onde o tempo é um fator importante. Suas principais aplicações são nas simulações, no diagnóstico, na supervisão, e na análise de desempenho de sistemas. Nestas redes, o tempo é representado por durações associadas ao lugar (p-temporizadas) ou a transição (t-temporizadas) [Sifakis, 1977], [Ramchandani, 1974]. As redes de Petri temporais/-temporizadas são apresentadas em detalhes na próxima seção.

3.4 Redes de Petri com Representação do Tempo

A inclusão da variável tempo no comportamento dinâmico das redes de Petri permite modelar e analisar sistemas com restrições temporais. A informação temporal possibilita sequencializar eventos, comparar suas durações, assim como determinar o intervalo existente entre eles. Assim, é possível representar e analisar problemas referentes às várias atividades onde a avaliação do tempo é um critério de fundamental importância.

Existem duas grandes classes de modelo: rede de Petri temporal e rede de Petri temporizada. Na Rede de Petri temporal é associado um par de datas (θ_{min}; θ_{max}) a cada transição. Onde, θ_{min} indica a duração mínima de sensibilização da transição anterior ao disparo, enquanto θ_{max} possibilita calcular a duração máxima de sensibilização. A transição necessita disparar neste intervalo de tempo. Na rede de Petri temporizada, é associada uma duração de tiro às transições. A rede é denominada rede de Petri com transição temporizada. Existem ainda outros modelos da mesma família: as redes com lugares temporizados e as redes com arcos temporizados [Cardoso et al., 1997].

3.4.1 Redes de Petri Temporizada (Timed Petri Nets)

Na RdP temporizada o tempo pode estar associado ao lugar ou a transição:

- o tempo associado ao lugar - nesse caso é associada a cada lugar uma duração, considerando que um lugar representa uma atividade A, atribui-se θ como a duração desta atividade. Quando uma ficha chega ao lugar p na data τ, a mesma só pode deixá-lo após $\tau' = \tau + \theta$ instantes. Portanto, enquanto não transcorrer o tempo as fichas não podem ser usadas para disparar a transição [Cardoso et al., 1997]. A Figura 3.4 apresenta um exemplo de RdP temporizada com associação aos lugares.

- o tempo associado à transição - nesse caso é associada a cada transição da rede uma duração de tiro (T). As fichas usadas para disparar a transição não estão disponíveis ou visíveis em nenhum lugar. Assim que a ficha se torna visível em um lugar, a mesma poderá ser utilizada por qualquer transição de saída deste lugar. Logo, o

disparo de uma transição possui uma duração [Cardoso et al., 1997]. A Figura 3.5 apresenta um exemplo de RdP temporizada com associação às transições.

Considerando as redes de Petri temporizadas apresentadas a seguir, suponha que o tempo inicial é zero e que $\theta_1 = 3$, $\theta_2 = 5$, $\theta_3 = 7$.

A transição t_1 da Figura 3.4 só poderá ser disparada no tempo τ' = $\tau + \theta_1$, ou seja, no tempo 3 pois $\tau = 0$, a transição t_2 só poderá ser disparada no tempo τ" = τ' + θ_2, ou seja, no tempo 8 pois τ' = 3, posteriormente ao disparo da transição t_2 haverá mais um tempo para que a atividade finalize (o lugar p_3 está marcado com um tempo θ_3) portanto, o tempo total será 15.

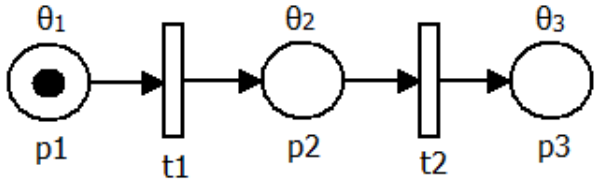

Figura 3.4: Rede de Petri temporizada com tempo associado aos lugares.

No lugar p_2 da Figura 3.5 a ficha só será visível no tempo $T_2 = T_1 + \theta_1$, ou seja, no tempo 3 pois $T_1 = 0$, e no lugar p_3 no tempo $T_3 = T_2 + \theta_2$, ou seja, no tempo 8, pois T_2 = 3.

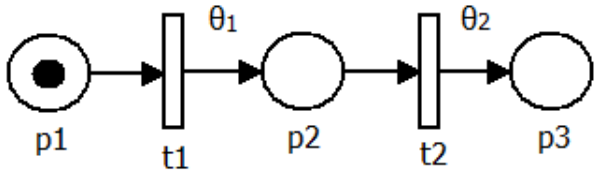

Figura 3.5: RdP temporizada com tempo associado as transições.

As abordagens apresentadas acima, RdP temporizada associada ao lugar e RdP temporizada associada à transição, são equivalentes.

3.4.2 Redes de Petri Temporal (Time Petri Nets)

A restrição temporal é um intervalo de tempo associado a toda transição. A rede de Petri temporal foi inicialmente empregada na descrição de protocolos de comunicação, contudo, seu campo de aplicação tem se expandido para áreas como: validação e verificação de sistemas, manufatura e sistemas em tempo real.

Na RdP temporal é associado a cada transição um intervalo (θ_{min}; θ_{max}). O tiro das transições é instantâneo, porém a transição deve estar sensibilizada durante o intervalo de tempo especificado, assim, a duração de sensibilização deve ser maior que θ_{min} e menor

que θ_{max}.

Definição 2.9: *Uma rede de Petri temporal é um par* $N_{tl} = < N; I >$ *onde:*

- *N é uma rede de Petri* $< P; T; Pre; Pos >$ *com uma marcação inicial* M_0*;*
- $\theta(t) = [\theta_{min}(t), \theta_{max}(t)]$ *é uma função que associa um intervalo fechado racional a cada transição t para descrever a duração de sensibilização.*

Considere a rede de Petri temporal representada na Figura 3.6, com o lugar p_1 marcado no tempo $\tau = 0$. A cada transição t_i é associada uma duração de sensibilização, dada pelo intervalo θ(i) = $(\theta_{min}(t_i), \theta_{max}(t_i))$.

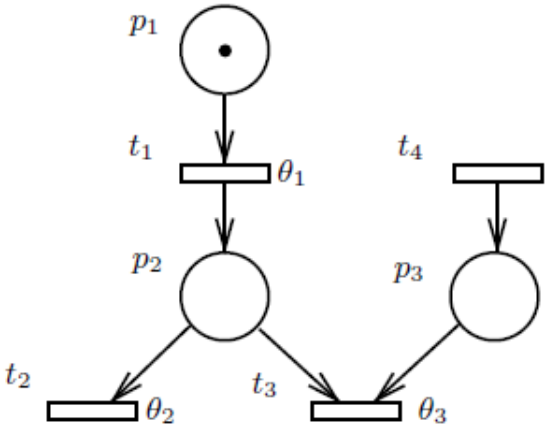

Figura 3.6: O tempo e a rede de Petri.

Assim, se a transição t_1 disparar no tempo τ_1 e $\tau_2 \in (\theta_{min}(t_1), \theta_{max}(t_1))$, o lugar p_2 estará marcado nesta data. Então, a transição t_2 pode disparar em $\tau_1 + \theta_2$. Se a transição t_4 for disparada no tempo τ_2, marcando o lugar p_3, a transição t_3 só poderá disparar no tempo $max(\tau_1, \tau_2) + \theta_3$.

Nesse cenário, existirá um conflito entre t_2 e t_3 em função dos valores de θ_2 e θ_3 e da relação entre as datas em que os lugares p_2 e p_3 são marcados. Se $\theta_3 = 0$, $\theta_2 \neq 0$ e $\tau_2 < \tau_1$, t_3 estará sensibilizada a partir da data τ_1, enquanto nesta data, t_2 está sensibilizada apenas pela marcação. Assim, t_3 dispara no tempo τ_1 e não há conflito entre t_2 e t_3. Contudo, se $\theta_3 \neq 0$, existirá um conflito durante a intersecção dos intervalos $(\tau_1 + \theta_2)$ e $max(\tau_1, \tau_2) + \theta_3$ [Cardoso et al., 1997].

3.5 *Workflow-Net*

O modelo *Workflow-Net (WF-Net)* foi desenvolvido por [Aalst et al., 1998] com a finalidade de permitir a verificação de propriedades qualitativas de *Workflows* através da

aplicação de redes de Petri. Verifica-se que essas redes são apropriadas para representação, validação e verificação de *Workflows*.

Uma *Workflow-Net* satisfaz as seguintes propriedades [Aalst et al., 1998]:

- Apresenta apenas um lugar de entrada (denominado *Start*) e apenas um lugar de saída (denominado *End*), sendo estes dois lugares tratados como lugares especiais; o lugar *Start* têm apenas arcos de saída e o lugar *End* apenas arcos de entrada.
- Uma ficha em *Start* representa um caso que precisa ser tratado e uma ficha em *End* representa um caso que já foi tratado.
- Toda tarefa t (transição) e condição p (lugar) deve estar em um caminho que se encontra entre o lugar *Start* e o lugar *End*.

Definição 2.10: *Um processo é considerado logicamente correto (Sound) se o mesmo não contém nenhuma tarefa desnecessária e se todo Case iniciado pelo processo é concluído integralmente em algum momento, não apresentando nenhum token remanescente no sistema.*

Uma *Workflow-net* é considerada logicamente correta *(Sound)* através do critério de verificação da correção *Soundness*, quando satisfaz as seguintes condições [Aalst and Hee, 2002]:

- Para cada ficha colocada no lugar *Start*, exatamente uma única ficha alcança o lugar *End*;
- Quando uma ficha é colocada no lugar *End*, os demais lugares estão vazios;
- Não deve haver nenhuma transição não-viva. Para toda transição (tarefa), é possível evoluir da marcação inicial até a marcação que sensibiliza tal transição.

Um processo determina quais tarefas necessitam ser executadas e em qual ordem a execução deve suceder. Modelar um processo de *Workflow* em termos de uma *Workflow-Net* é direto: transições são componentes ativos e modelam às tarefas, lugares são componentes passivos e modelam às condições (*Pre e Pos*), e as fichas modelam os casos [Aalst et al., 1998]. A Figura 3.7, a seguir, apresenta um exemplo de *Workflow* modelado pela *Workflow-Net*.

No contexto das *WF-Nets* um acionamento é uma condição externa que orienta a execução de uma tarefa sensibilizada [Aalst et al., 1998]. Existem quatro tipos distintos de tarefas, conforme mostra a Figura 3.7, as quais seguem:

- Usuário: a tarefa é acionada por um recurso humano e este acionamento é mostrado em uma *WF-Net* através do símbolo **seta para baixo** nas transições;

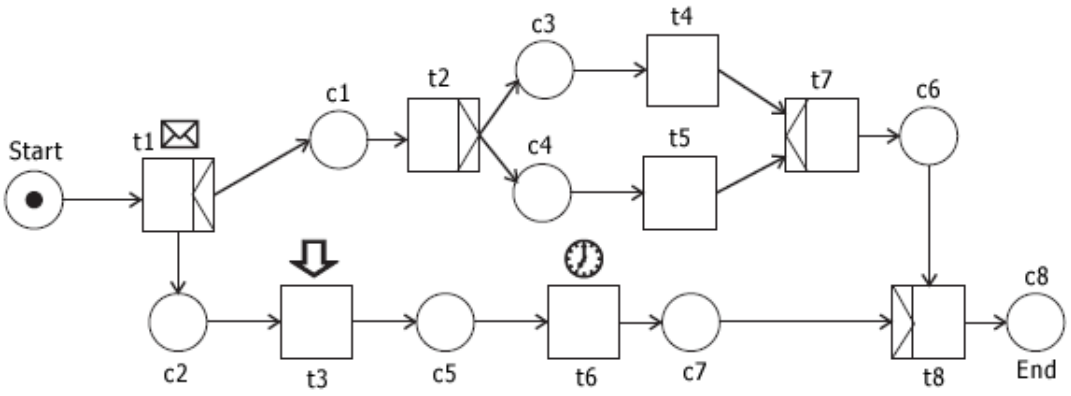

Figura 3.7: Exemplo de uma *Workflow-Net*

- Mensagem: um evento externo aciona uma tarefa sensibilizada através do símbolo **envelope** nas transições;
- Tempo: a tarefa é executada em um tempo pré-definido. Isto é mostrado em uma *WF-Net* através do símbolo **relógio** nas transições;
- Automática: uma tarefa é acionada no instante em que é sensibilizada e não requer interação humana. Para este tipo de tarefa não há nenhuma representação na *Workflow-Net*.

Verifica-se que quando uma tarefa do tipo usuário é considerada, a mesma é acionada por um recurso humano, ou seja, existe uma alocação de recurso associada à esta tarefa. Nos demais tipos de tarefa não há alocação de recursos associada.

A Figura 3.8 apresenta os elementos que constituem a *WF-Net* e suas redes de Petri equivalentes. Verifica-se que uma *WF-Net* pode ser convertida com facilidade em uma rede de Petri comum, conhecendo-se a semântica de cada transição.
Onde:

- Or-Split: consiste em um apontamento dentro do *Workflow* onde uma única linha de controle decide sobre qual ramificação escolher quando encontrar múltiplas alternativas em um *Workflow*.
- And-Split: é um apontamento dentro do *Workflow* onde uma única linha de controle se divide em duas ou mais tarefas, as quais são executadas em paralelo dentro do *Workflow*.
- Or-Join: consiste em um apontamento dentro do *Workflow* onde duas ou mais ramificações de atividades alternativas se reconvergem para uma única atividade comum como a próxima etapa dentro do *Workflow*.

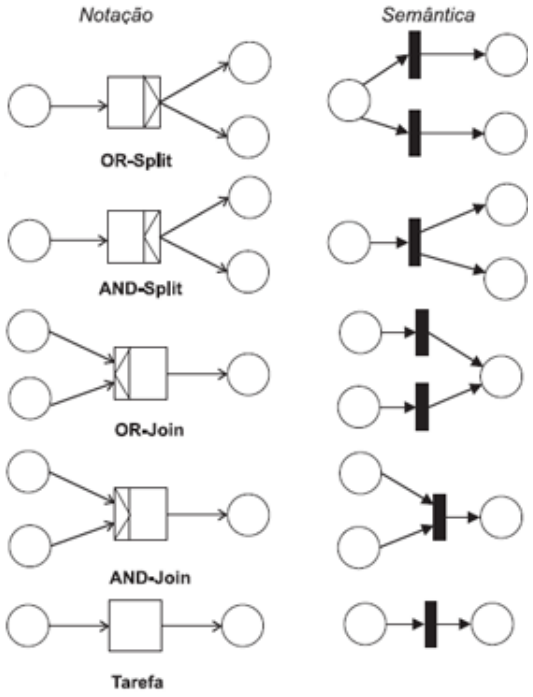

Figura 3.8: Elementos básicos da *WF-Net* e sua rede de Petri equivalente.

- And-Join: é um apontamento no *Workflow* onde duas ou mais atividades executando paralelamente se convergem em uma única tarefa de controle.

A seguir, é apresentada a definição formal dos tipos de mecanismos de alocação de recursos considerados em sistemas de gerenciamento de *Workflow*.

Definição 2.11: *Formalmente, o mecanismo de alocação de recurso discreto pode ser definido através de um modelo de rede de Petri ordinária marcada [David and Alla, 2004]* C_{DR}=<A_{DR}; T_{DR}; Pre_{DR}; Pos_{DR}; M_{DR}> *com:*

- $A_{DR} = \bigcup_{\alpha=1}^{N_{DR}} A_{(DR)_\alpha} \cup R_D$ *onde* R_D *representa o lugar do recurso discreto,* $A_{(DR)\alpha}$ *o lugar da atividade* α *e* N_{DR} *o número das atividades que estão conectadas com o lugar do recurso discreto* R_D.

- $T_{DR} = \bigcup_{\alpha=1}^{N_{DR}} T_{in_\alpha} \cup \bigcup_{\alpha=1}^{N_{DR}} T_{out_\alpha}$ *onde* T_{in_α} *representa a transição de entrada da atividade* $A_{(DR)\alpha}$ *e* T_{out_α} *representa a transição de saída da atividade* $A_{(DR)\alpha}$.

- Pre_{DR}: $A_{DR}\ x\ T_{DR} \rightarrow \{0,1\}$ *é a aplicação de incidência anterior tal que:* $Pre_{DR}(R_D, T_{in_\alpha}) = 1$ *e* $Pre_{DR}(A_{(DR)_\alpha}, T_{out_\alpha}) = 1$. *(Outras combinações de lugar/-transição são iguais a zero).*

- Pos_{DR}: $A_{DR}\ x\ T_{DR} \rightarrow \{0,1\}$ *é a aplicação de incidência posterior tal que:* $Pos_{DR}(R_D, T_{out_\alpha}) = 1$ *e* $Pos_{DR}(A_{(DR)_\alpha}, T_{in_\alpha}) = 1$. *(Outras combinações de lugar/-transição são iguais a zero).*

- *M_{DR}: $A_{DR} \rightarrow N$ é a marcação inicial tal que: $M_{DR}(R_D) = m_D$ e $M_{DR}(A_{(DR)_\alpha}) = 0$ para $\alpha = 1$ até N_{DR} que representa o número de recursos discretos do mesmo tipo.*

Definição 2.12: *Formalmente, o mecanismo de alocação de recurso contínuo pode ser definido por um modelo de rede de Petri híbrida marcada com transições discretas [David and Alla, 2004] C_{CR}=<A_{CR}; T_{CR}; Pre_{CR}; Pos_{CR}; M_{CR}> com:*

- *$A_{CR} = \bigcup_{\alpha=1}^{N_{CR}} A_{(CR)_\alpha} \cup R_C$ onde R_C representa o lugar do recurso contínuo, $A_{(CR)\alpha}$ o lugar da atividade α e N_{CR} o número de atividades que estão conectadas com o lugar do recurso contínuo R_C.*

- *$T_{CR} = \bigcup_{\alpha=1}^{N_{CR}} T_{in_\alpha} \cup \bigcup_{\alpha=1}^{N_{CR}} T_{out_\alpha}$ onde T_{in_α} representa a transição de entrada discreta da atividade $A_{(CR)\alpha}$ e T_{out_α} representa a transição de saída discreta da atividade $A_{(CR)\alpha}$.*

- *Pre_{CR}: A_{CR} x $T_{CR} \rightarrow R^+$ é a aplicação de incidência anterior tal que: $Pre_{CR}(R_C, T_{in_\alpha}) = X_\alpha$ com $X_\alpha \in R^+$ e $Pre_{CR}(A_{(CR)_\alpha}, T_{out_\alpha}) = 1$. (Outras combinações de lugar/transição são iguais a zero).*

- *Pos_{CR}: A_{CR} x $T_{CR} \rightarrow R^+$ é a aplicação de incidência posterior tal que: $Pos_{CR}(R_C, T_{out_\alpha}) = X_\alpha$ e $Pos_{CR}(A_{(CR)_\alpha}, T_{in_\alpha}) = 1$. (Outras combinações de lugar/-transição são iguais a zero).*

- *M_{CR}: $A_{CR} \rightarrow R^+$ é a marcação inicial tal que: $M_{CR}(R_C) = m_C$ e $M_{CR}(A_{(CR)_\alpha}) = 0$ para $\alpha = 1$ até N_{CR} que representa o número de recursos discretos do mesmo tipo.*

Nas redes de Petri t-temporais [Merlin, 1974], apresentada na seção 3.4.2, o tempo é representado por um intervalo (θ_{min}; θ_{max}) associado a cada transição, que equivale a uma duração de sensibilização. No contexto das redes de Petri t-temporais, uma t-Time *Workflow-Net* consiste em uma *Workflow-Net* estendida com intervalos de tempo associados às transições, pois a execução das tarefas ficará associada às transições do modelo.

Definição 2.13: *Formalmente, uma t-Time Workflow-Net N é uma quádrupla (P, T, F, I) tal que:*

- *(P, T, F) é uma Workflow-Net, onde:*
  - *P é um conjunto finito de lugares;*
  - *T é um conjunto finito de transições;*
  - *F $\subseteq$ (P x T)$\cup$ (T x P) é um conjunto de arcos (relação de fluxos).*

- *I é uma aplicação que associa a cada transição $t \in T$ um intervalo de sensibilização I(t)= ($\theta_{min}(t)$; $\theta_{max}(t)$), onde $\theta_{min}(t)$ representa o tempo mínimo de disparo da transição e $\theta_{max}(t)$ representa o tempo máximo de disparo da transição t.*

Capítulo 4

Modelo de Segurança RBAC

Mecanismos de segurança de controle de acesso necessitam assegurar que a tarefa seja executada somente por usuários autorizados. Um modelo de autorização deve ser capaz de impedir a modificação desautorizada dos dados, e também fornecer meios de reforçar o padrão legítimo das operações nos dados acessados para a execução de uma tarefa.

O controle de acesso baseado em papéis (RBAC - Role Based Acess Control) regula o acesso dos usuários para executar operações em objetos (aplicações), com base nos papéis que os mesmos exercem numa organização. Os papéis denotam funções que descrevem a autoridade e a responsabilidade concedidas aos usuários [Ferraiolo and Kuhn, 1995].

Este capítulo aborda os conceitos referentes ao modelo de segurança RBAC no critério de particionamento de papéis proposto em [Nyanchama and Osborn, 1994].

4.1 Mecanismo de Segurança de Controle de Acesso

O desenvolvimento de mecanismos de controle de acesso ao longo da história se deu de várias maneiras, porém, sempre com a finalidade de determinar quais indivíduos apresentam o direito de permanecer em determinado local e de delimitar a autoridade de atuação dos mesmos nesses locais.

O controle de acesso é um item de segurança ligado à confidencialidade das informações e acesso a operações. Conforme [Sandhu et al., 1996], o controle de acesso objetiva limitar as operações que podem ser realizadas por uma entidade sobre um determinado recurso, alcançando objetivos da segurança da informação, como sigilo e integridade, e prevenindo a exposição e a modificação não autorizada da informação.

O controle de acesso geralmente estabelece a autenticação prévia do usuário para que o mesmo possa usufruir das funcionalidades de um sistema. Através da autenticação se verifica a legitimidade da identidade do usuário, utilizando um login de identificação e senhas de acesso, ou biometria, por exemplo. Posteriormente os dados de identificação são submetidos a uma verificação por meio do sistema, geralmente empregando-se um

banco de dados, retornando uma validação positiva ou negativa referente à identidade do usuário.

O alcance de um bom nível de segurança não é garantido pelo controle de acesso, ou mesmo pela atribuição de permissões de acesso corretas a cada indivíduo que tenha a necessidade de tê-la. Muitas vezes, a concentração excessiva de poderes em usuários individuais pode levar a situações de conflitos de interesses, nas quais o mesmo indivíduo tem o poder de auditar as próprias ações, o que representa uma grave falha de segurança. Logo, existe a necessidade de separação das responsabilidades, que objetiva garantir que fraudes ou danos acidentais não ocorram como consequência da demasiada concentração de poder em uma única pessoa.

Questões relacionadas à segurança geralmente remetem a proteção contra ataques externos. Entretanto, verifica-se a existência de problemas de segurança internos, onde um funcionário considerado confiável comete atos que contrariam as expectativas, pelos mais variados motivos, como por exemplo: problemas financeiros, distúrbios comportamentais, influências das mais variadas fontes, etc. Nesse contexto, um modelo de segurança deve ser capaz de expressar políticas que reduzam as chances das ocorrências de conflitos de interesses que possam levar um indivíduo a cometer fraudes ou erros devido à concentração excessiva de poderes sobre as etapas da execução de uma tarefa crítica.

Uma política de acesso determina diretrizes de alto nível que definem como o acesso é controlado e como as decisões de autorização de acesso são estabelecidas [Sandhu and Samarati, 1994]. Os modelos de controle de acesso determinam uma linguagem para expressar técnicas que representem uma política de acesso de alto nível. Assim sendo, o acesso a qualquer sistema deverá satisfazer a sua política de acesso. Dentre os modelos existentes, destacam-se o controle de acesso discricionário, o controle de acesso compulsório e o controle de acesso baseado em papéis (RBAC). Esse último modelo vem se tornado uma importante alternativa aos modelos tradicionais e será adotado como base neste trabalho.

4.2 Controle de Acesso Baseado em Papéis

O controle de acesso baseado em papel descreve mecanismos de segurança que controlam o acesso de usuários a recursos computacionais. Assim, permite que privilégios sejam atribuídos aos papéis arbitrários, os quais podem então ser distribuídos aos usuários reais [Koch and Parrisi-Presicce, 2002], [Nyanchama and Osborn, 1994], [Sandhu et al., 1996]. O controle de acesso baseado em papel provê um modo de controlar autorizações e executar tarefas em sistemas complexos com muitos usuários e recursos [Sandhu et al., 1996].

[Ferraiolo and Kuhn, 1995] acredita que as principais motivações do RBAC são, em primeiro lugar, a capacidade para expressar e impor uma política de segurança específica

para uma organização e, em segundo lugar, simplificar o oneroso processo de gerenciamento de segurança.

A política de controle de acesso baseada em papéis surgiu da necessidade das organizações associarem permissões e recursos a funcionários de acordo com sua função ou tarefa exercida na organização. Nesta política a análise dos requisitos e das regras de negócio é que determinarão [Koch and Parrisi-Presicce 2002], [Nyanchama and Osborn 1994]:

- Papéis necessários e suficientes para cobrir o domínio da informação;
- Permissões atribuídas a cada papel;
- Usuários aos quais serão atribuídos os papéis;
- Relação hierárquica entre papéis;
- Restrições de uso do mecanismo RBAC a fim de mantê-lo no escopo da política de segurança da organização.

Os papéis são criados e são associados aos mesmos as autorizações do sistema e os usuários. Assim sendo, o acesso às funcionalidades do sistema é dado conforme os papéis que o usuário tenha associado. Um usuário apresentará mais de um papel associado quando desempenhar funções distintas na própria organização.

O modelo de segurança RBAC apresenta maior facilidade na administração de políticas de acesso. No caso em que um usuário tenha sua função alterada na organização, basta que ele seja desassociado do papel ao qual pertencia e associado ao novo papel. Da mesma maneira, quando as atribuições de uma função são alteradas, basta que sejam associadas ou desassociadas as autorizações daquele papel, assim, todos os usuários associados ao papel terão suas funções alteradas conforme a nova norma da empresa [Sandhu et al., 1996].

O emprego do modelo RBAC aproxima a política de controle de acesso à própria estrutura organizacional das empresas. Nas seções seguintes serão detalhados os elementos que compõe o modelo RBAC e o modelo proposto pelo NIST, que objetiva formalizar o escopo, os conceitos e a terminologia do RBAC [Ferraiolo and Sandhu, 2001].

4.2.1 Modelo de Referência do Controle de Acesso Baseado em Papéis

O padrão ANSI [ANSI/INCITS, 2004] para o controle de acesso baseado em papéis foi proposto em 2001, pelo NIST (National Institute of Standards and Technology) [Ferraiolo and Sandhu, 2001]. O modelo de referência do padrão de RBAC consiste num

instrumento conceitual abstrato que aborda as relações, restrições e funções administrativas entre entidades do modelo, como usuários, papéis, sessões e autorizações. O padrão para controle de acesso baseado em papéis está organizado em três partes: RBAC Básico, RBAC Hierárquico e RBAC com Restrição (que especifica as restrições de separação de responsabilidade estática e dinâmica). Utilizaremos uma notação baseada no trabalho de [Ferraiolo and Sandhu, 2001] para a descrição formal dos modelos RBAC.

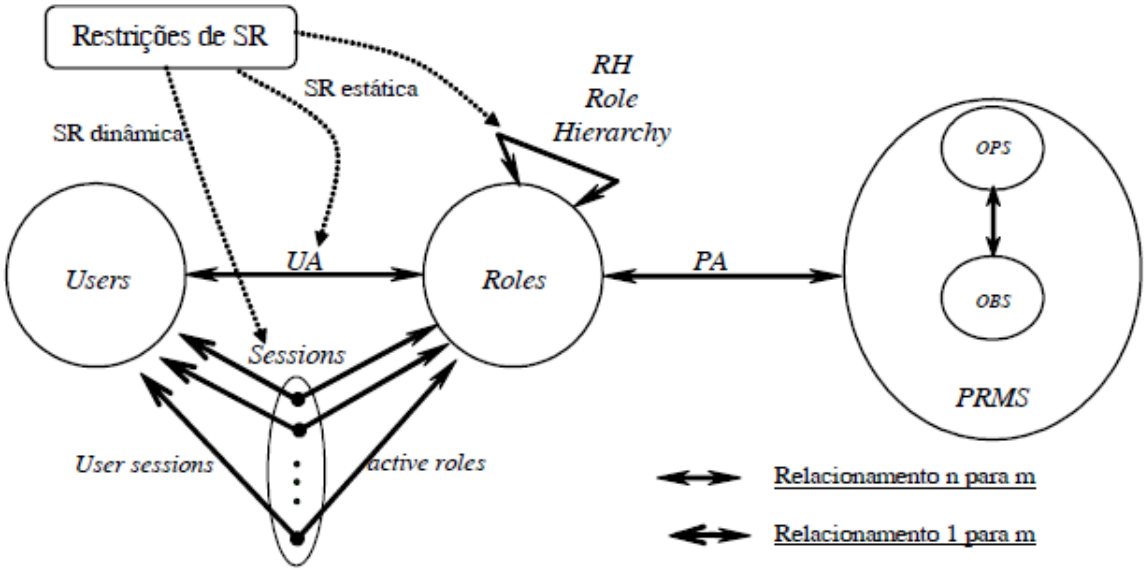

Figura 4.1: Esquema do modelo de referência do padrão de controle de acesso baseado em papéis

RBAC Básico

O RBAC Básico estabelece os aspectos essenciais e fundamentais do RBAC. Especifica as relações usuário-papel (UA) e papel-autorização (PA) (Figura 4.1), define funções básicas e introduz o conceito de sessão, onde um usuário poderá ativar ou desativar um ou mais papéis. Este modelo exige que as associações usuário-papel e papel-autorização possam ser de muitos para muitos.

O conceito do RBAC básico é a associação de usuários a papéis, a associação de permissões a papéis e a aquisição de permissões do usuário pelo papel que o mesmo exerce [Koch and Parrisi-Presicce, 2002]. A seguir é apresentada a especificação das entidades e de seus relacionamentos.

- *Roles*(papéis): coleção de papéis. Papéis representam uma função ou cargo existente em uma organização. Aos papéis serão associados os usuários e as permissões de utilização de um sistema [Ferraiolo and Sandhu, 2001];

- *Users*(usuários): coleção de nomes que representam os agentes humanos. Usuários são associados aos papéis conforme a sua função na organização. Caso essa função seja modificada, o usuário deverá ser associado a outro papel que reproduza as suas

novas atribuições. Ainda assim é possível que um mesmo usuário seja associado a mais de um papel;

- *OBS*(objetos): coleção de nomes que representam os objetos protegidos. Os Objetos podem ser difinidos como uma entidade que tem operações associadas, onde as operações determinam o modo de utilização do objeto em um sistema.

- *OPS*(operações): coleção de nomes das operações associadas aos objetos protegidos.

- *Sessions*(sessões): coleção de nomes que representam as sessões de trabalho.

- *Usuário_da_sessão(s: Sessions)* $\rightarrow$ *Users*: Operação que mapeia a sessão s para o usuário que a inicializou;

- $PRMS \subseteq 2^{(OPS \times OBS)}$: coleção de permissões. A permissão (autorização) se refere ao consentimento para executar uma operação em um objeto de uma aplicação [Ferraiolo and Sandhu, 2001];

- $UA \subseteq Users \times Roles$: coleção estabelecendo as associações muitos-para-muitos entre usuários e papéis. Essa associação possibilia que um mesmo usuário seja associado a um ou mais papéis e vice-versa;

- $PA \subseteq Roles \times PRMS$: coleção estabelecendo as associações muitos-para-muitos entre papéis e permissões. Essa associação estabelece que uma mesma permissão de acesso seja associada a um ou mais papéis;

- *Permissões_associadas(r: Roles)* $\rightarrow 2^{PRMS}$: mapeamento de um papel r para a coleção de permissões associadas. Formalmente, *Permissões_associadas(r)*= $\{p \in PRMS | (r,p) \in PA\}$;

- *Papéis_da_sessão(s: Session)* $\rightarrow 2^{Roles}$: mapeia a sessão s para a coleção dos papéis ativados na sessão por um usuário, que consiste no subconjunto dos papéis que o usuário tem associado. Formalmente, *Papéis_da_sessão(s)* $\subseteq \{r \in Roles | (Usuário\_da\_sessão(s), r) \in UA\}$.

RBAC Hierárquico (Grafo de Papéis)

No contexto organizacional o conceito de pessoas/papéis é prevalecente. Pessoas são uma ou mais unidades tal como departamentos, divisões, ou grupos onde os mesmos têm diferentes cargos em diferentes níveis de hierarquias. Assim, é comum, por exemplo, que em um processo de negócio que reimplanta um pedido feito por um empregado necessita ser aprovado pelo chefe da unidade em que o empregado está estaticamente atribuído [Koch and Parrisi-Presicce, 2002], [Nyanchama and Osborn, 1999].

Os papéis organizados de forma hierárquica são uma maneira natural de organizar papéis para refletir a autoridade, responsabilidade e competência. A hierarquia é uma ordem parcial que define uma relação de precedência de papéis, por meio da qual, papéis de categoria superior adquirem as permissões dos seus subordinados. A hierarquia estrutura papéis permitindo a reflexão das linhas de autoridade e responsabilidade conforme em uma organização [Sandhu et al., 1996].

O RBAC Hierárquico incorpora requisitos ao RBAC básico para permitir a hierarquia de papéis. Matematicamente, uma hierarquia é uma ordem parcial que determina uma relação de responsabilidades entre papéis. Este modelo apresenta a inclusão da hierarquia de papéis (relação RH na Figura 4.1), o que consiste na principal diferença do RBAC Hierárquico com o RBAC Básico. Segue a definição da hierarquia de papéis, conforme [Ferraiolo and Sandhu, 2001]:

- RH $\subseteq$ Roles x Roles. Relação de ordem parcial sobre a coleção de papéis, denominada de herança, designado por $\succeq$, no qual, $p_1 \succeq p_2$ somente se todas as permissões de p_2 também forem permissões de p_1, e todos os usuários de p_1 também forem usuários de p_2. Formalmente, $p_1 \succeq p_2 \Rightarrow$ *Permissões_associadas_rh*(p_2) $\subseteq$ *Permissões_associadas_rh*(p_1) $\wedge$ *Usuários_associados_rh*(p_1) $\subseteq$ *Usuários_associados_rh*(p_2);

- Permissões_associadas_rh(r: Roles) $\subseteq 2^{PRMS}$: Mapeamento de um papel r para uma coleção de permissões direta ou indiretamente associadas a esse papel. Formalmente, *Permissões_associadas_rh(r)*= $\{p \in PRMS | \ r \succeq r' \wedge (r', p) \in PA\}$.

- Usuários_associados_rh(r: Roles) $\subseteq 2^{Users}$: Mapeamento de um papel para uma coleção de usuários associados ao mesmo. Formalmente, *Usuários_associados_rh(r)*= $\{u \in Users | \ r' \succeq r \wedge (u, r') \in UA\}$.

A Figura 4.2 ilustra um grafo de papéis. Os papéis com mais permissões, ocupam posições mais altas, os quais por sua vez, herdam as permissões dos papéis localizados abaixo deles no grafo de papéis. Analisando a Figura 4.2, temos que o Diretor, além de suas próprias permissões, tem as permissões herdadas do Líder de Projeto 1 e do Líder de Projeto 2. O Líder do Projeto 1 tem as suas permissões e mais as do Engenheiro de Produção 1 e Engenheiro de Qualidade 1. Esta hierarquia apesar de permitir a agregação de permissões de diferentes papéis em outro não permite o compartilhamento de permissões entre eles.

O RBAC hierárquico apresenta vantagens como a racionalização do uso dos recursos computacionais e otimização da administração do controle de acesso. Esses benefícios são aparentes quando se tem a necessidade de incorporar ou excluir uma permissão de uso a um grupo grande de papéis, por exemplo. Essa alteração deverá ser reproduzida a todos os papéis que tenham a permissão a ser alterada, assim sendo, essa distribuição

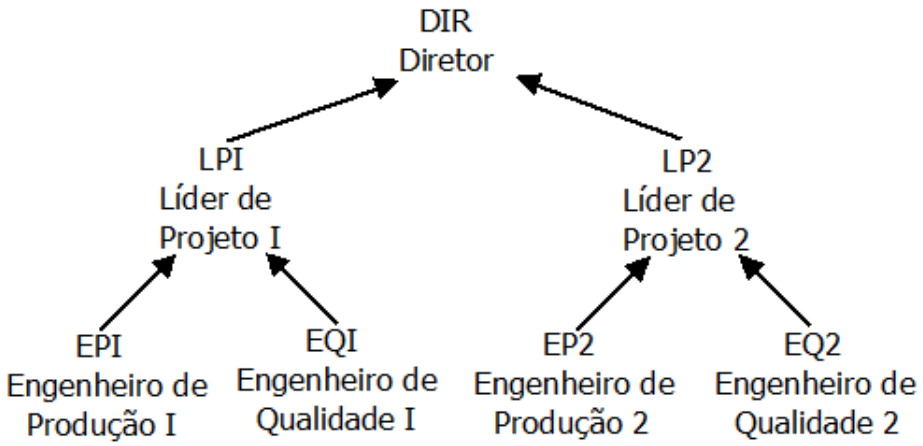

Figura 4.2: Grafo de Papéis Hierárquico [Koch and Parrisi-Presicce, 2002].

de permissões poderá ser dispendiosa caso não se empregue o RBAC hierárquico. Caso a alteração seja comum a papéis que pertençam à mesma hierarquia, ou ramo da árvore, empregando-se o RBAC hierárquico, basta que essa alteração seja realizada em um único papel e a mesma será herdada por todos os seus papéis descendentes.

O RBAC hierárquico possibilita a emprego de árvores normais ou invertidas, que se diferem quanto ao modo de interpretação das informações nela contidos. Nas árvores normais cada papel poderá apresentar mais de um ancestral direto, contudo, poderá apresentar somente um descendente imediato. Logo numa árvore invertida (Figura 4.2), os papéis podem apresentar diversos descendentes diretos, entretanto, podem apresentar apenas um ancestral imediato. A especificação completa da hierarquia de papéis em estrutura de árvore pode ser encontrada na proposta do NIST para RBAC [Ferraiolo and Sandhu, 2001].

RBAC com Restrições

O RBAC com restrição estende o modelo hierárquico tratando a questão dos conflitos de interesses. Esse modelo inclui relações de separação de responsabilidade para estabelecer restrições que auxiliem a evitar que um usuário seja autorizado a executar operações associadas a papéis com conflitos de interesses, colaborando dessa maneira com a redução do número de fraudes e erros acidentais [Koch and Parrisi-Presicce, 2002]. Nesse modelo, a separação de responsabilidade pode ser estática ou dinâmica.

A separação de responsabilidades estática restringe os papéis que poderão ser associados a usuários como forma de coerção dos conflitos de interesses. Atua na relação usuário-papel (UA) e na hierarquia de papéis (RH) (Figura 4.1). A SR dinâmica atua durante a ativação de papéis numa sessão de um usuário (Figura 4.1). Na separação dinâmica de tarefas há uma exclusão mútua entre dois papéis R1 e R2, ou seja, um mesmo usuário pode ser associado a papéis conflitantes, desde que apenas um dos papéis (R1 ou R2) esteja ativo em um dado momento, a fim de evitar que o usuário exerça a

totalidade dos poderes desses papéis simultaneamente [Sandhu and Kuhn, 2000].

Dentre os modelos RBAC abordados acima, o RBAC Hierárquico será acoplado ao modelo de alocação de recursos, no estudo de caso, por categorizar os papéis associados aos recursos de maneira hierárquica. Dessa forma verifica-se que existe uma menor concentração de privilégios dos papéis associados aos recursos. Na seção seguinte é abordado o particionamento de papéis no grafo de papéis hierárquico.

4.3 Partição de Papéis no Grafo de Papéis Hierárquico

No contexto do modelo RBAC Hierárquico em [Nyanchama and Osborn, 1994] foi proposto a implementação deste modelo incluindo a partição de papéis. Verifica-se que um papel pode ser particionado em dois ou mais papéis. As operações básicas de particionamento podem ser desempenhadas de dois modos: verticais ou horizontais e podem naturalmente ser combinadas [Nyanchama and Osborn, 1994].

A principal vantagem desta aplicação consiste na redução significativa dos privilégios que até então permaneciam concentrados em posse de apenas um usuário, o que favorecia o aumento da ocorrência de fraudes e demais erros na execução de tarefas. Neste trabalho, por conveniência, foi abordado o modelo de particionamento vertical.

4.3.1 Partição Vertical do Papel

O particionamento vertical permite a quebra do papel em dois ou mais papéis. Neste modelo, um papel é selecionado para ser quebrado, posteriormente ao particionamento são gerados novos papéis que passam a adquirir os privilégios do papel alvo. Considere por exemplo o papel X, este será particionado em X_1,..., X_n. Estes novos papéis serão criados de forma ordenada, de tal forma que em X_1 deve existir um caminho que leve a X_2 assim por diante, ou seja, (X_1,..., X_n). Assim que os novos papéis forem gerados e após a distribuição dos privilégios, o papel alvo se tornará extinto. A seção a seguir apresenta o algoritmo de particionamento de papéis

4.3.2 Algoritmo de Particionamento de Papéis

O algoritmo de particionamento de papéis proposto neste trabalho foi desenvolvido baseando-se no algoritmo proposto em [Nyanchama and Osborn, 1994]. Os dados de entrada do algoritmo implementados são um grafo de papéis da organização, o papel alvo a ser particionado e o número de novos papéis a serem criados. Já os dados de saída são formados pelo grafo de papéis da organização contendo os novos papéis originados do particionamento do papel alvo.

A idéia essencial do algoritmo proposto é apresentar o grafo de papéis da organização com os novos papéis criados. Após a entrada dos dados citados, o papel alvo a ser particionado será localizado no grafo de papéis da organização através de procedimentos recursivos. Em seguida, este papel sofrerá o particionamento de acordo com o número de entrada indicado pelo usuário. Após o processamento destas informações tem-se como saída o grafo de papéis da organização com a exclusão do papel alvo e, inserido nele, os novos papéis criados. O código 4.1 apresenta o algoritmo em linguagem C para o particionamento de papéis no grafo de papéis hierárquicos.

Código 4.1 Algoritmo para o particionamento de papéis

```
// PARTICIONA PAPEL
int particionaNo( struct grafo **g, struct grafo **ng, int v )
{
    int i;

    // CRIANDO PAPEL
    novoItemGrafo( ng );
    (*ng)->papel = v;

    // TRANSFERE PREDECESSORES DE G PARA NG
    for( i=0; i<=(*g)->qtdepred; i++ )
    {
        inserePred( ng, &(*g)->pred[i] );
        removePred( g, &(*g)->pred[i] );
    }
    inserePred( g, ng ); // NG SE TORNA PREDECESSOR DE G
    insereAnt( ng, g );

    // ANTECESSORES QUE APONTAVAM PARA G, IRÃO APONTAR PARA NG
    for( i=0; i<=(*ng)->qtdepred; i++ )
    {
        removeAnt( &(*ng)->pred[i], g );
        insereAnt( &(*ng)->pred[i], ng );
    }
    return 0;
}
```

Capítulo 5

Estudo de caso

As redes de Petri têm sido amplamente estudadas e aplicadas com sucesso na área de sistemas de *Workflow*. O emprego desse formalismo para modelagem e especificação do comportamento dinâmico de sistemas é efetivado pela combinação de fundamentação matemática, representação gráfica compreensiva, disponibilidade de ferramentas de análise e possibilidade de simulações e verificações.

No contexto de sistemas de *Workflow*, verifica-se que a segurança consiste em um componente crítico e essencial. No intuito de impor uma política de segurança são empregados mecanismos de alocação de recursos e modelos de autorização que objetivam melhorias na administração do acesso de usuários nestes sistemas.

Sendo assim, este capítulo tem como objetivo verificar o formalismo das redes de Petri para a modelagem de um sistema de *Workflow*, considerando o modelo de segurança RBAC no critério de particionamento de papéis proposto em [Nyanchama and Osborn, 1994], no intuito de simplificar o processo de gerenciamento de segurança nestes sistemas. Este capítulo apresenta ainda um protótipo para verificação e validação do modelo de rede de Petri em estudo. Para ilustrar o mapeamento de processos em *Workflow-Nets*, consideraremos o processo para o "Gerenciamento de Reclamações", que será adotado como objeto do estudo de caso.

5.1 Metodologia

Neste estudo de caso será apresentada, primeiramente, a análise e descrição do objeto de estudo de caso, assim como o modelo de alocação de recursos associados às atividades do processo. O processo para o "Gerenciamento de Reclamações" é modelado através de uma rede de Petri t-temporal, embutindo conceitos de *WorkFlow-Net* (*t-Time WorkFlow-Net*), possibilitando a correta alocação de recursos conforme o cálculo de datas de utilização do mesmo para o tratamento de atividades do referido processo.

Posteriormente, será apresentada uma simulação do algoritmo para o particionamento

de papéis em sistemas de *Workflow*. O algoritmo proposto neste trabalho foi baseado no trabalho de [Nyanchama and Osborn, 1994]. Foram realizados os devidos testes de execução a fim verificar se o mesmo poderia ser aplicado eficientemente em sistemas de gerenciamento de *Workflow*.

Por fim, será apresentado o protótipo "RdP Simulation" para simulação dos papéis e tarefas de um modelo de rede de Petri, que possibilita obter um cenário admissível para validação do modelo. Assim, este sistema permite a inserção de dados quaisquer sobre um modelo de rede de Petri e retorna um modelo conceitual ilustrando o roteamento entre as tarefas, e os papéis e usuários vinculados a cada tarefa. O protótipo permite ainda o particionamento de papéis.

O presente estudo de caso aborda a simulação do processo de "Gerenciamento de Reclamações" através do protótipo "RdP Simulation" no intuito de validar o modelo proposto, considerando o particionamento de papéis.

5.2 Estudo de Caso: Modelagem do Processo de "Gerenciamento de Reclamações"

Consideremos um sistema de "Gerenciamento de Reclamações" de uma empresa que forneça serviços quaisquer. Num primeiro tempo, a reclamação feita pelo cliente da empresa é registrada por um atendente de *call center*. Posteriormente, o departamento da empresa que é alvo da reclamação é informado sobre a mesma (por um dos funcionários responsáveis pelas reclamações) e deve apresentar uma justificativa. Enquanto isso, o cliente é contactado (por um dos funcionários responsáveis pelas reclamações) para fornecer mais informações sobre o caso. Essas duas atividades podem ser realizadas simultaneamente, dependendo da disponibilidade dos funcionários responsáveis pelas reclamações. Em seguida, os dados finais sobre a reclamação são registrados no sistema e uma decisão é tomada. Dependendo do resultado da decisão (tomada por um dos funcionários encarregados), uma indenização é paga ao cliente, ou uma carta de recusa da reclamação lhe é enviada. No final, todo o processo é arquivado no sistema.

Cada tarefa do processo (*registrar_reclamação*, *contactar_cliente*, *informar_departamento*, *coletar_dados*, *analisar/tomar_decisão*, *pagar_indenização*, *enviar_carta* e *arquivar_processo*) é representada por uma transição na RdP, como mostra a Figura 5.1.

Os casos são representados pelas fichas presentes nas redes. No lugar *Start* da Figura 5.1, existe uma ficha indicando a presença de um caso. Se a transição *registrar_reclamação* é disparada, duas fichas (uma em c1 e outra em c2) representam o mesmo caso. Quando um caso é tratado, o número de fichas pode variar. A quantidade de fichas que representa um caso particular é sempre igual ao número de suas condições que devem ser satisfeitas.

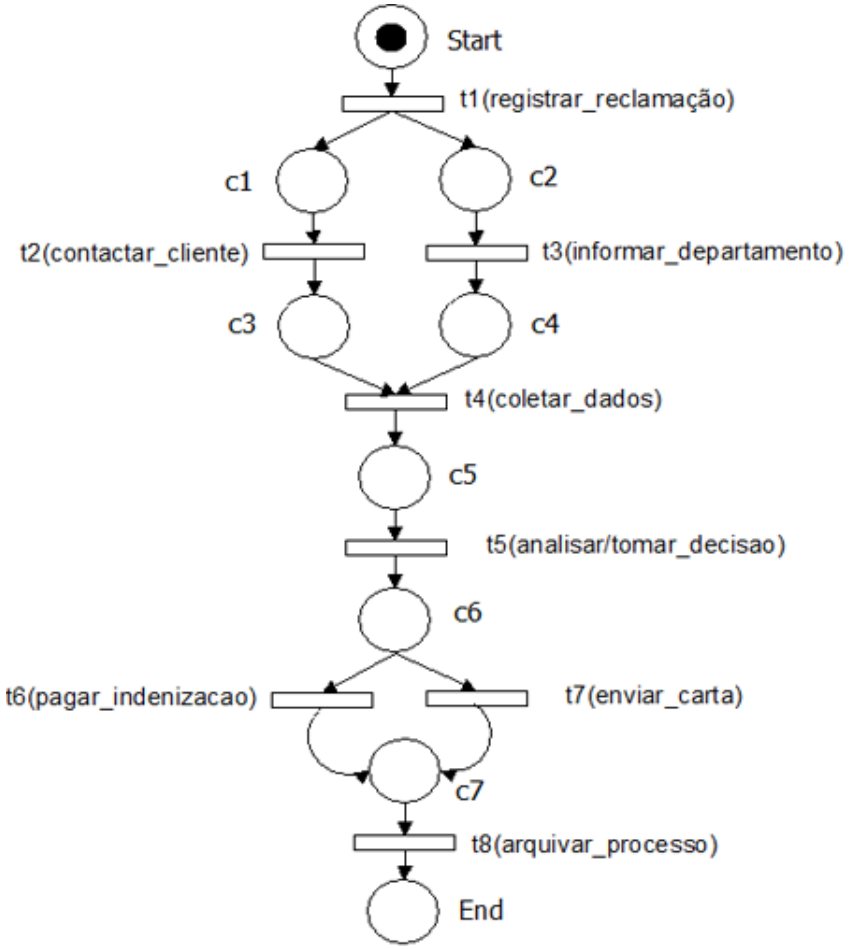

Figura 5.1: *Workflow-Net* para o processo de tratamento de reclamações e os seus acionamentos

No lugar *End* deverá haver uma ficha, quando o caso for concluído.

O processo de tratamento de reclamações mostrado na Figura 5.1 consiste em oito tarefas, das quais duas são automaticamente tratadas, ou seja, não possui intervenção do usuário *(coletar_dados* e *arquivar_processo)* e seis são acionadas por recursos humanos *(registrar_reclamação, contactar_cliente, informar_departamento, analisar/tomar_decisão, pagar_indenização, enviar_carta)*, ou seja, existe uma alocação de recurso associada à estas tarefas.

Um cenário corresponde a uma rota bem definida mapeada na *Workflow-Net*. Na Figura 5.1 há dois cenários diferentes: o primeiro cenário C_1 onde a tarefa *pagar_indenização* será executada, isto é, a transição *t6* será disparada, e o segundo cenário C_2, onde a tarefa *enviar_carta* será executada. Assim, o sequente para o C_1 é dado por:

Start, registrar_reclamação, contactar_cliente, informar_departamento, coletar_dados, analisar/tomar_decisão, pagar_indenização, arquivar_processo $\vdash$ *End*

E o sequente para C_2 é dado por:

Start, registrar_reclamação, contactar_cliente, informar_departamento, coletar_dados, analisar/tomar_decisão, enviar_carta, arquivar_processo $\vdash$ *End*

Modelo de Alocação de Recurso

A alocação de recursos é uma metodologia destinada a distribuição temporal dos recursos disponíveis para a realização de qualquer projeto. O modelo de alocação de recursos humanos para o processo de gerenciamento de segurança é analisado a seguir.

Supondo-se que no "Gerenciamento de Reclamações", apresentado anteriormente, um único funcionário trata as atividades *contactar_cliente*, *informar_departamento* e *enviar_carta*, então o mecanismo de alocação de recurso discreto é dado pela Figura 5.2.

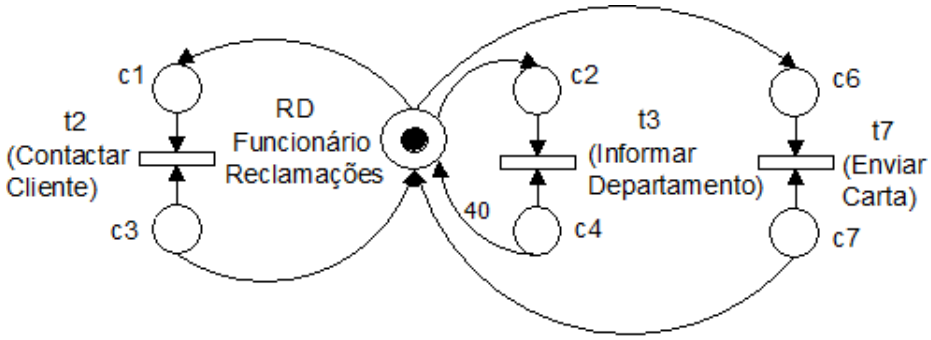

Figura 5.2: Alocação de recurso discreto

Nesse caso, se a ficha em RD (recurso discreto) é usada para realizar a atividade *t2*, nenhuma outra atividade poderá ser realizada. Isto significa que o recurso RD pode ser usado somente de modo disjuntivo. Em particular, se a atividade *t2* é iniciada e o funcionário não pode entrar em contato imediatamente com o cliente, ele não pode usar seu tempo disponível (o tempo de espera da resposta do cliente) para iniciar outra atividade, como enviar uma carta, por exemplo (atividade *t7*). É evidente que na prática, esta situação não ocorre. Se o cliente não está disponível em um dado instante, o funcionário usará sua disponibilidade para executar outra tarefa.

Supondo novamente que no processo de "Gerenciamento de Reclamações" um único funcionário trata as atividades *contactar_cliente*, *informar_departamento* e *enviar_carta*, então o mecanismo de alocação de recurso contínuo é dado pela Figura 5.3.

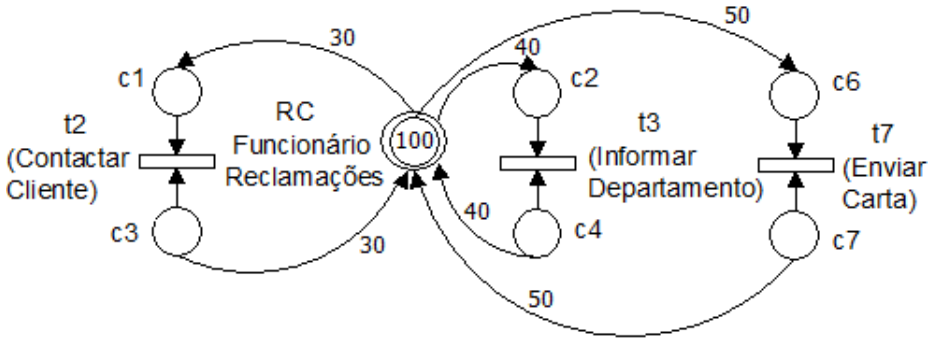

Figura 5.3: Alocação de recurso contínuo

Na Figura 5.3 fica especificado que somente 30% da disponibilidade do funcionário

é necessária para realizar a atividade *contactar_cliente*, 40% para realizar a atividade *informar_departamento* e 50% para realizar a atividade *enviar_carta*. Desta forma, então, é possível que o funcionário trate simultaneamente mais de uma atividade. Por exemplo, se o funcionário realizar a atividade *contactar_cliente* e o cliente não estiver disponível para responder as questões do mesmo, este poderá usar sua disponibilidade de tempo (esperando a resposta do cliente) para iniciar outra atividade, como, por exemplo, enviar uma carta (atividade *t7*). De fato, 50% da disponibilidade do funcionário é necessária para realizar a atividade *enviar_carta*, e depois de iniciada a atividade *contactar_cliente* o funcionário tem ainda 70% de disponibilidade.

A seguir, é apresentado o cálculo de datas de utilização de um recurso humano para o tratamento de atividades do processo de "Gerenciamento de Reclamações". Deste modo, será possível prever a disponibilidade desejada dos recursos humanos envolvidos na execução das tarefas correspondentes.

A Figura 5.4 apresenta uma *t-Time Workflow-Net* para o processo de "Gerenciamento de Reclamações" com intervalos de tempo associado às transições. As transições deste modelo correspondem às transições da Figura 5.1, onde RR corresponde a *registrar_reclamação*, CC *(contactar_cliente)*, ID *(informar_departamento)*, CD *(coletar_dados)*, AT *(analisar/tomar_decisão)*, PI *(pagar_indenização)*, EC *(enviar_carta)* e AP *(arquivar_processo)*.

Na Figura 5.4(a) é ilustrado a associação de intervalos de tempo simbólico às transições enquanto na Figura 5.4(b) são associados intervalos de tempo numérico. Os intervalos estáticos associados às atividades *coletar_dados* em *t4* e *arquivar_processo* em *t8* são iguais a $[0, 0]$, pois a duração destas atividades é desprezível quando comparada com as demais atividades, além disso, são automaticamente tratadas. Os intervalos estáticos associados às atividades foram estipulados apenas a título de exemplificação.

Considerando as tarefas do tipo usuário modelada na *Workflow-Net* para o "Gerenciamento de Reclamação", deve-se definir as datas de produção (Dt_P) e de consumo (Dt_C). Considera-se a data máxima das produções quando há mais de uma pré-condição associada à transição. A data de produção, Dt_P, corresponde ao início da execução da tarefa associada à transição, e a data de consumo Dt_C, corresponde ao término da execução da mesma. Deste modo, é gerado um intervalo de datas $[Dt_P, Dt_C]$, onde o recurso que executará a referida tarefa deverá estar disponível para realizá-la.

As datas de produção e consumo são dependentes de duração de sensibilização (ds_i), cujo valor pertence a um intervalo de tempo $\Delta_i = [\delta_{i_{min}}\ ,\ \delta_{i_{max}}]$, assim, pode-se considerar vários intervalos possíveis de execução das tarefas, conforme um planejamento estratégico. Por exemplo, o intervalo $I_{Ex} = [Dt_{P_{min}}, Dt_{C_{max}}]$ considera que a alocação de recurso para a execução da tarefa poderá ocorrer entre o início ao mais cedo e o término ao mais tarde da atividade. Este intervalo de datas é o mais flexível, pois considera o tempo de

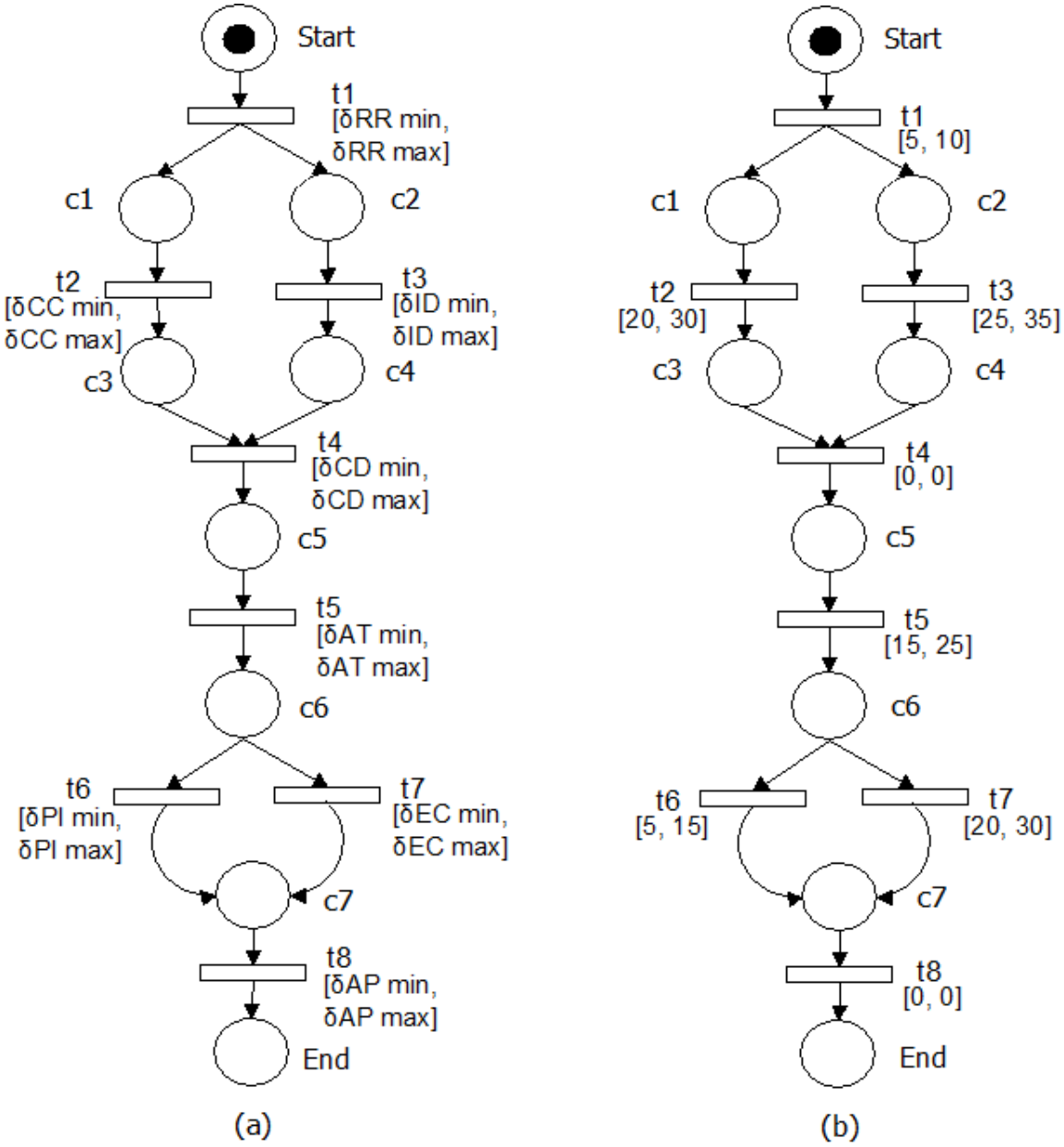

Figura 5.4: *t-Time Workflow-Net* para o "Gerenciamento de Reclamações" (a)intervalos de tempo simbólico associado às transições (b)intervalos de tempo numérico associado às transições.

utilização do recurso o mais amplo possível.

A Tabela 5.1 apresenta os intervalos de datas simbólicas de execução para as tarefas que são do tipo usuário, dos cenários C_1 e C_2, considerando o Intervalo de Execução $I_{Ex} = [Dt_{P_{min}}, Dt_{C_{max}}]$, onde $Dt_{P_{min}}$ corresponde a data de início ao mais cedo da tarefa e $Dt_{C_{max}}$ a data de término ao mais tarde da tarefa. Assim, os intervalos de datas calculados poderão ser utilizados por qualquer caso tratado pelo processo de reclamação modelado pela *Workflow-Net* da Figura 5.1.

Considerando os intervalos de tempo definidos na *t-Time Workflow-Net* (Figura 5.4) e considerando que o processo a ser tratado tem início na data 0, ou seja, D_I=0, os

Tabela 5.1: Intervalo de datas simbólicas para execução de tarefas do tipo usuário dos cenários C_1 e C_2

Transição	**Intervalos de data C_1 e C_2**
t_2=[c1-c3]	$[D_I+\delta RR_{min},\ D_I+\delta RR_{max}+\delta CC_{max}]$
t_3=[c2-c4]	$[D_I+\delta RR_{min},\ D_I+\delta RR_{max}+\delta ID_{max}]$
t_5=[c5-c6]	$[D_I+\delta RR_{min}+max(\delta CC_{min},\delta ID_{min})+\delta CD_{min},$ $D_I+\delta RR_{max}+max(\delta CC_{max},\delta ID_{max})+\delta CD_{max}+\delta AT_{max}]$
Transição	**Intervalos de data C_1**
t_6=[c6-c7]	$[D_I+\delta RR_{min}+max(\delta CC_{min},\delta ID_{min})+\delta CD_{min}+\delta AT_{min},$ $D_I+\delta RR_{max}+max(\delta CC_{max},\delta ID_{max})+\delta CD_{max}+\delta AT_{max}+\delta PI_{max}]$
Transição	**Intervalos de data C_2**
t_7=[c6-c7]	$[D_I+\delta RR_{min}+max(\delta CC_{min},\delta ID_{min})+\delta CD_{min}+\delta AT_{min},$ $D_I+\delta RR_{max}+max(\delta CC_{max},\delta ID_{max})+\delta CD_{max}+\delta AT_{max}+\delta EC_{max}]$

intervalos de datas para execução das tarefas podem ser calculados somente substituindo as datas simbólicas presentes na Tabela 5.1 pelas datas numéricas associadas às transições da *t-Time Workflow-Net* da Figura 5.4(b).

Na tabela 5.2 é apresentado o resultado desse cálculo. Considerando, por exemplo, a tarefa *pagar_indenização* do cenário C_1, temos que o recurso utilizado para executar esta tarefa deverá ser alocado ao mais cedo na data 45 e deverá ser liberado ao mais tarde na data 85. O recurso que executará determinada tarefa poderá ser alocado considerando este intervalo de datas.

Tabela 5.2: Intervalo de datas numéricas para execução de tarefas do tipo usuário dos cenários C_1 e C_2

Tarefa	**Intervalos de data C_1**	**Intervalos de data C_2**
contactar_cliente	[5,40]	[5,40]
informar_departamento	[5,45]	[5,45]
analisar/tomar_decisao	[30,70]	[30,70]
pagar_indenizacao	[45,85]	-
enviar_carta	-	[45,100]

Alocação de papéis

Na Figura 5.5 é apresentado o modelo de rede de Petri para o processo de tratamento de reclamações com a alocação de papéis, representados por chaves, associados às transições.

A Figura 5.6 mostra o grafo de papéis para o modelo de rede de Petri para o "Gerenciamento de Reclamação" considerando a alocação de papéis apresentada anteriormente.

No processo de tratamento de reclamações, os papéis com mais permissões, ocupam posições mais altas no grafo, os quais por sua vez, herdam as permissões dos papéis localizados abaixo deles no grafo. O papel "Coordenador de atendimento ao cliente" da Figura 5.6(a) é o responsável por avaliar a reclamação, sinalizando-a como positiva ou

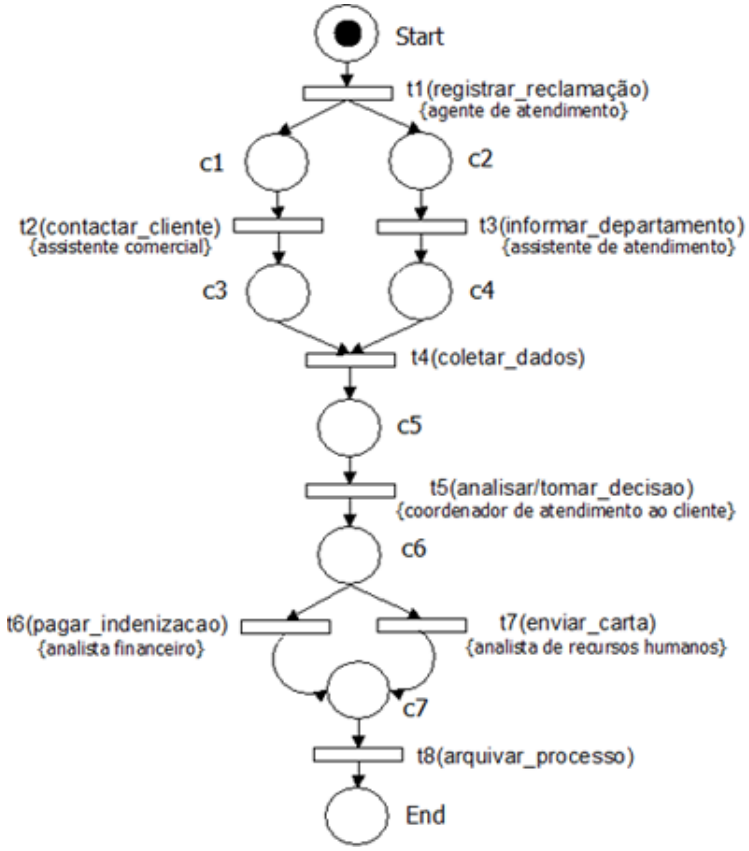

Figura 5.5: *Workflow-Net* para o processo de tratamento de reclamações com a alocação de papéis

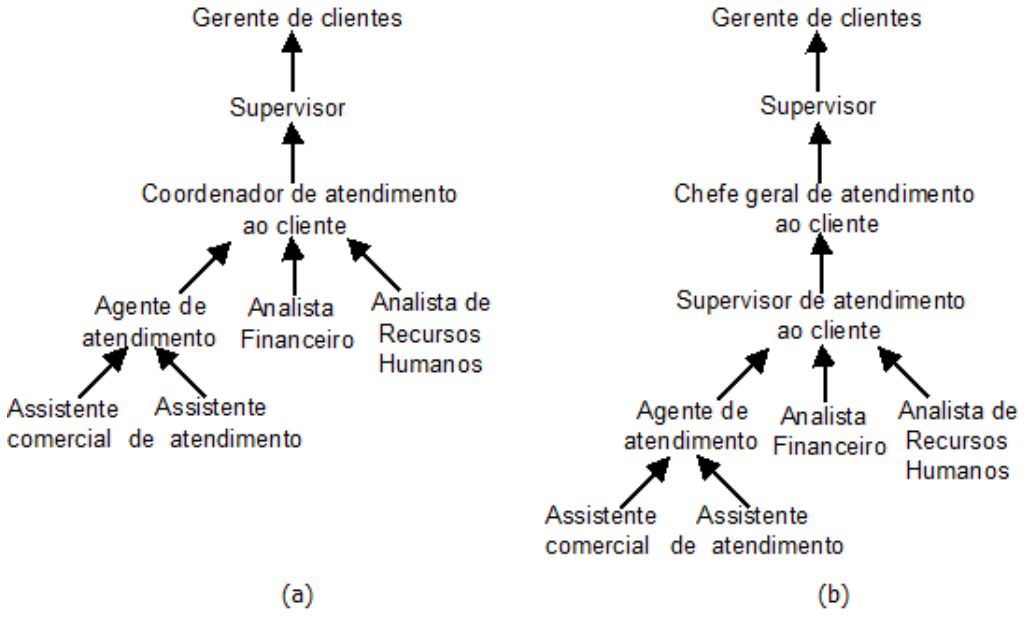

Figura 5.6: (a)Grafo de Papéis Hierárquico para o processo de tratamento de reclamações. (b)Grafo de papéis após o particionamento do papel "Coordenador de atendimento ao cliente".

negativa, além de suas atribuições específicas. Assim, verifica-se uma centralização de privilégios no processo decisório, o que favorece o retardamento nas decisões, sobrecarga

de atividades, aumento da possibilidade de fraudes, erros e avaliações pessoais errôneas.

A concentração excessiva de poderes e a capacidade de auditar os próprios atos consistem num problema de segurança considerado grave no contexto de sistemas de *Workflow*, portanto, é proposta a utilização do algoritmo de particionamento de papéis, no intuito de impor uma política de segurança nestes sistemas.

Portanto, propomos o particionamento do papel "Coordenador de atendimento ao cliente" em dois novos papéis: "Chefe geral de atendimento ao cliente" e "Supervisor de atendimento ao cliente", a fim de tornar mais dinâmica a operação da organização, descentralizando o poder decisorial. O grafo de papéis após o particionamento do papel "Coordenador de atendimento ao cliente" é apresentado na Figura 5.6(b).

A Figura 5.7 apresenta o modelo *Workflow-Net* para o "Gerenciamento de Reclamações" com a transição *analisar/tomar_decisão* diferenciada, indicando que houve o particionamento de papéis e que esta tarefa será executada pelos dois novos papéis: "Chefe geral de atendimento ao cliente" e "Supervisor de atendimento ao cliente".

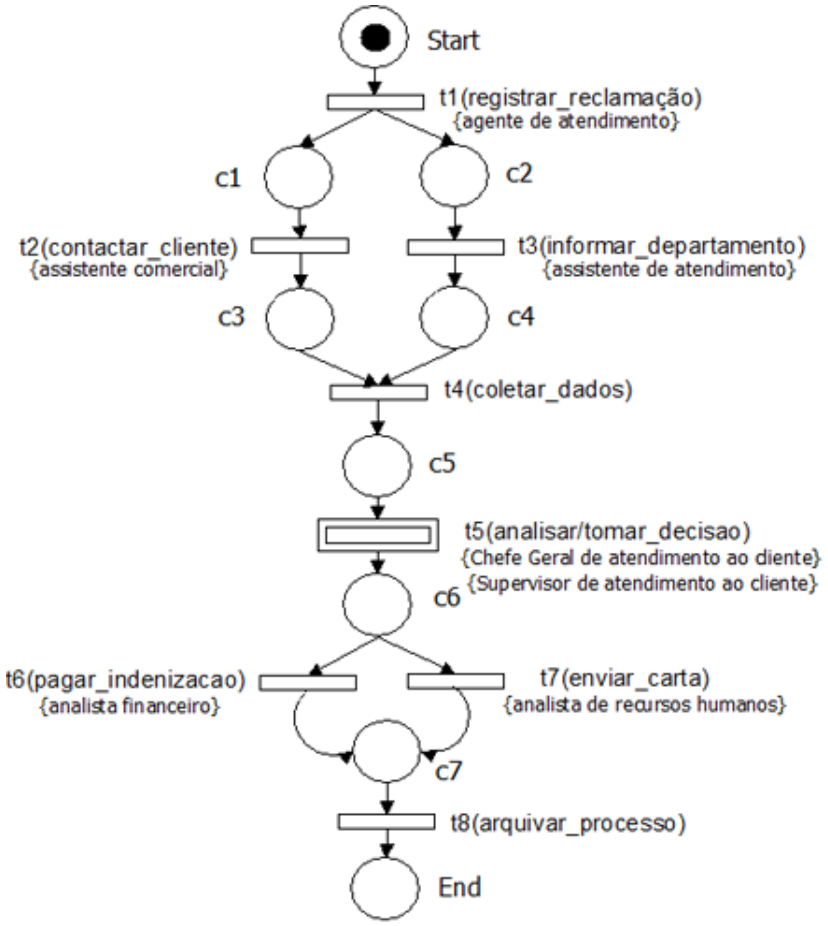

Figura 5.7: *Workflow-Net* para o processo de tratamento de reclamações com transição diferenciada

A seção seguinte apresenta o algoritmo para o particionamento de papéis.

Particionamento de Papéis

O algoritmo para o particionamento de papéis proposto na Seção 4.3.2 foi baseado no trabalho de [Nyanchama and Osborn, 1994] e implementado na linguagem C. Apresenta-se a seguir, os devidos testes de execução a fim de verificar a aplicabilidade do mesmo em sistemas de *Workflow*.

Foi realizada uma simulação do algoritmo para o particionamento de papéis, inicialmente com a criação de cinco papés no grafo de papéis representando assim, o grafo de papéis de uma organização. A estes papéis foram inseridos seus antecessores e seus predecessores, ou seja, os papéis que a eles estão diretamente relacionados hierarquicamente segundo critérios desta mesma organização. Logo em seguida, foi realizada uma busca para encontrar o papel alvo a ser particionado. Houve uma modificação no grafo para serem inseridos os novos papéis criados e a remoção dos seus antecessores e predecessores.

O resultado obtido com a implementação citada, foi um grafo de papéis da organização inseridos em seu contexto os novos papéis criados a partir do particionamento. A Figura 5.8 mostra o resultado de uma das principais funções de particionamento onde apresenta o particionamento do papel 10 nos papéis 12, 14, 16 e 18.

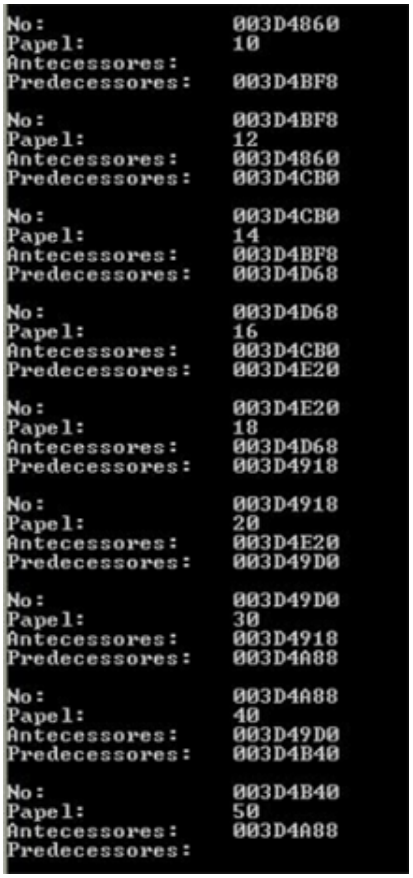

Figura 5.8: Representação Parcial dos Resultados Obtidos

Assim, é verificado que o modelo de particionamento vertical de papéis proposto em

[Nyanchama and Osborn, 1994] pode ser aplicado de modo eficiente em sistemas de gerenciamento de *Workflow*, contribuindo então para maior segurança ao acesso a dados e consequentemente, a descentralização de poderes atribuídos a um papel.

A seção seguinte aborda a simulação do processo de tratamento de reclamação através do protótipo "RdP Simulation" considerando a inclusão do modelo formal para particionamento vertical de papéis.

5.2.1 Validação do Modelo de Rede de Petri

O protótipo *"RdP Simulation"* para simulação de modelos de rede de Petri foi implementada em HTML (abreviação para a expressão inglesa *HyperText Markup Language*, que significa Linguagem de Marcação de Hipertexto) que consiste em uma linguagem de marcação utilizada para produzir páginas na *web*. As demais funcionalidades foram implementadas utilizando *JavaScript*, linguagem de *script* para programação cliente-servidor em navegadores *web* e *Ruby*, linguagem de programação interpretada multiparadigma, de tipagem dinâmica e forte, com gerenciamento de memória automático que suporta programação funcional, orientada a objetos, imperativa e reflexiva. Além disso, foi utilizada a biblioteca Raphaël, em *JavaScript*, para geração de gráficos vetoriais para a *web*.

O protótipo está disponível *on-line* no serviço de hospedagem de site *Heroku*. O *Heroku* é uma plataforma de desenvolvimento de aplicações *web* denotado pela sigla *PaaS (Plataform as a Service)*. Este serviço de hospedagem na nuvem é interessante pela facilidade de escalar uma aplicação tanto em termos de processamento quanto de banco de dados. Outro ponto importante é que a configuração mais simples de aplicação no *Heroku* é gratuita, e à medida que crescem, passam a pagar por mais recursos do servidor. Os dados são armazenados através do banco de dados *PostgreSQL*, disponível por padrão pela plataforma.

O protótipo está disponível no endereço: *http://rdp-simulation.heroku.com/*.

Através do *"RdP Simulation"* é possível obter um cenário admissível para validação do modelo de rede de Petri. Assim, o protótipo permite a inserção de dados quaisquer sobre um modelo de rede de Petri e o mesmo retorna um modelo conceitual ilustrando o roteamento entre as tarefas e os papéis e usuários vinculados a cada tarefa.

"RdP Simulation" é um sistema simples, fácil e intuitivo que possibilita a realização de ações, como inserção, edição e remoção dos dados para simulação dos papéis e tarefas de uma rede de Petri. A seguir são apresentadas as principais telas do protótipo.

Na Figura 5.9 é apresentada a tela inicial do protótipo "RdP Simulation". O acesso ao protótipo se restringe aos usuários cadastrados, para tanto, é solicitado o cadastro do usuário através do campo "Cadastro", onde é necessário a inserção dos dados: "Nome", "Email" e "Senha". No campo "Apresentação", é mostrado uma breve descrição do

sistema e o campo “Entrar” permite que o usuário faça *login* no sistema.

Figura 5.9: Tela inicial do “RdP Simulation”

A Figura 5.10 apresenta a tela para o cadastro de tarefas. As tarefas são representadas por uma transição na rede de Petri. Nesta fase do processo, o usuário do sistema cadastra o nome da tarefa e uma descrição da mesma. No campo “Visualizar tarefas cadastradas” é possível visualizar as tarefas inseridas. Assim, o usuário é direcionado para outra janela onde é possível visualizar cada tarefa individualmente, editar e remover os dados. O campo “Voltar” direciona o usuário para a tela inicial e o campo “Cadastrar Papéis”, direciona o usuário para a tela de cadastro de papéis.

Figura 5.10: Tela para cadastro de tarefas

Na Figura 5.11 é apresentada a tela de cadastro de papéis. Nesta fase do processo, o usuário do sistema cadastra o nome do papel e a permissão de uso. Posteriormente a inserção, o papel e a permissão podem ser visualizados através do campo “Visualizar papéis cadastrados”. Assim, é possível visualizar cada papel individualmente, editar e remover os dados. O campo “Voltar” direciona o usuário para a tela de cadastro de tarefas e o campo “Cadastrar Funcionários” para a tela de cadastro de funcionários.

A Figura 5.12 apresenta a tela para o processo de cadastro de funcionários, onde o

usuário do sistema cadastra o nome e o número de matrícula do mesmo. Nesta tela é possível visualizar os dados inseridos clicando no campo "Visualizar funcionários cadastrados". Os campos "Voltar" e "Associar Informações" direcionam o usuário para a tela de cadastro de papéis e para a tela de associação de informações, respectivamente.

Figura 5.11: Tela para cadastro de papéis

Figura 5.12: Tela para cadastro de funcionários

Após o cadastro dos dados o usuário realiza a associação dos dados, especificando quais papéis são responsáveis por realizar determinada tarefa. A Figura 5.13 apresenta a tela para esta fase, considerada uma das mais importantes do processo. Nesta etapa, o usuário seleciona uma determinada tarefa e associa dados como o roteamento da atividade e as atividades sucessoras. O usuário necessita selecionar também o papel e o funcionário responsável por aquela tarefa.

No campo "Visualizar Associações", da Figura 5.13, é possível visualizar os dados que foram relacionados. Na tela para visualização dos dados associados é possível visualizar cada associação separadamente no campo "Mostrar", e realizar a edição e remoção dos dados através dos campos "Editar" e "Deletar" respectivamente.

Posteriormente ao cadastro e a associação dos dados (Figura 5.13) é possível gerar um modelo conceitual baseado nas informações fornecidas pelo usuário do sistema clicando no campo "Gerar Imagem". O protótipo retorna como resultado, as tarefas apresentadas dentro de blocos, o roteamento entre as mesmas, e os papéis associados são apresentados ao lado dos blocos.

A tela para associação de informações (Figura 5.13) apresenta ainda o campo "Deseja particionar papéis?" que permite o particionamento de papéis. Ao clicar neste campo,

Figura 5.13: Tela que permite a associação de atributos da RdP

Figura 5.14: Tela de particionamento de papéis

o usuário do sistema é direcionado para a tela de particionamento de papéis conforme Figura 5.14.

Na Figura 5.14 são carregados os papéis já inseridos, permitindo que o usuário selecione um papel a ser quebrado. Posteriormente, são inseridos os novos papéis. Assim, o papel alvo que foi particionado é extinto e os novos papéis cadastrados passam a obter as permissões/privilégios desse papel. Após a inserção dos novos papéis é possível voltar para a tela de Associação de informações através do campo "Voltar" e também gerar a imagem a partir das informações fornecidas no campo "Gerar Imagem".

O objetivo deste trabalho é simular a rede de Petri apresentada na Figura 5.1 no intuito de validar a mesma. Para tanto, foram inseridos os dados referentes ao processo de "Gerenciamento de Reclamações", como as tarefas do processo, os papéis responsáveis pelas tarefas e os funcionários associados aos papéis. Posteriormente foi realizada a associação desses dados, conforme é ilustrado na Figura 5.15.

A partir da associação dos dados foi obtido como resultado o modelo apresentado na Figura 5.16. Nessa figura pode-se verificar que a transição *analisar/tomar_decisão* tem como papel associado o papel "Coordenador de atendimento ao cliente".

A Figura 5.17 apresenta o modelo conceitual para o processo de "Gerenciamento de Reclamações" após o particionamento do papel "Coordenador de atendimento ao cliente". A tarefa *analisar/tomar_decisão* mostra os novos papéis associados.

Assim sendo, o papel "Coordenador de atendimento ao cliente" foi particionamento

Figura 5.15: Tela de particionamento de papéis

verticalmente em dois novos papéis: "Chefe geral de atendimento ao cliente" e "Supervisor de atendimento ao cliente", a fim de tornar mais dinâmica a operação da organização, descentralizando o poder decisorial e promovendo melhorias no processo de deliberação e na segurança. Dessa forma, a avaliação da transição *analisar/tomar_decisão* deve agora, ser analisada pelos dois novos papéis.

Verifica-se que as decisões delegadas a dois novos papéis promovem a resolução correta e rápida do processo, provendo um retorno mais rápido aos clientes e avanços na segurança do processo. Assim, essa abordagem atende ao alinhamento das organizações em torno das estratégias para alcançar as metas estabelecidas e mantê-las competitivas no mercado. Além disso, é importante ressaltar que o particionamento vertical de papéis resolve o problema da alocação de recursos, uma vez que as atividades podem ser melhor distribuídas.

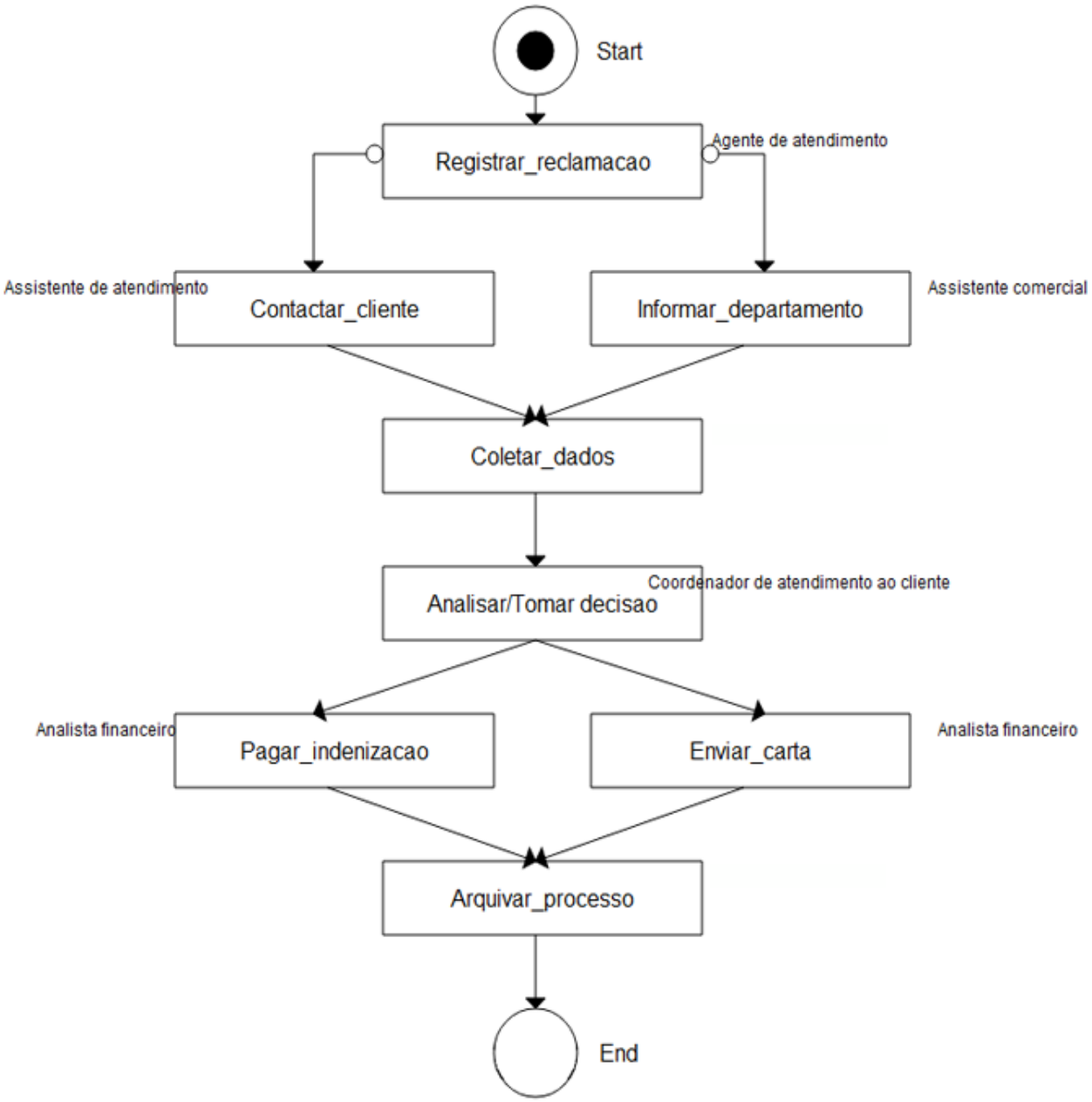

Figura 5.16: Resultado

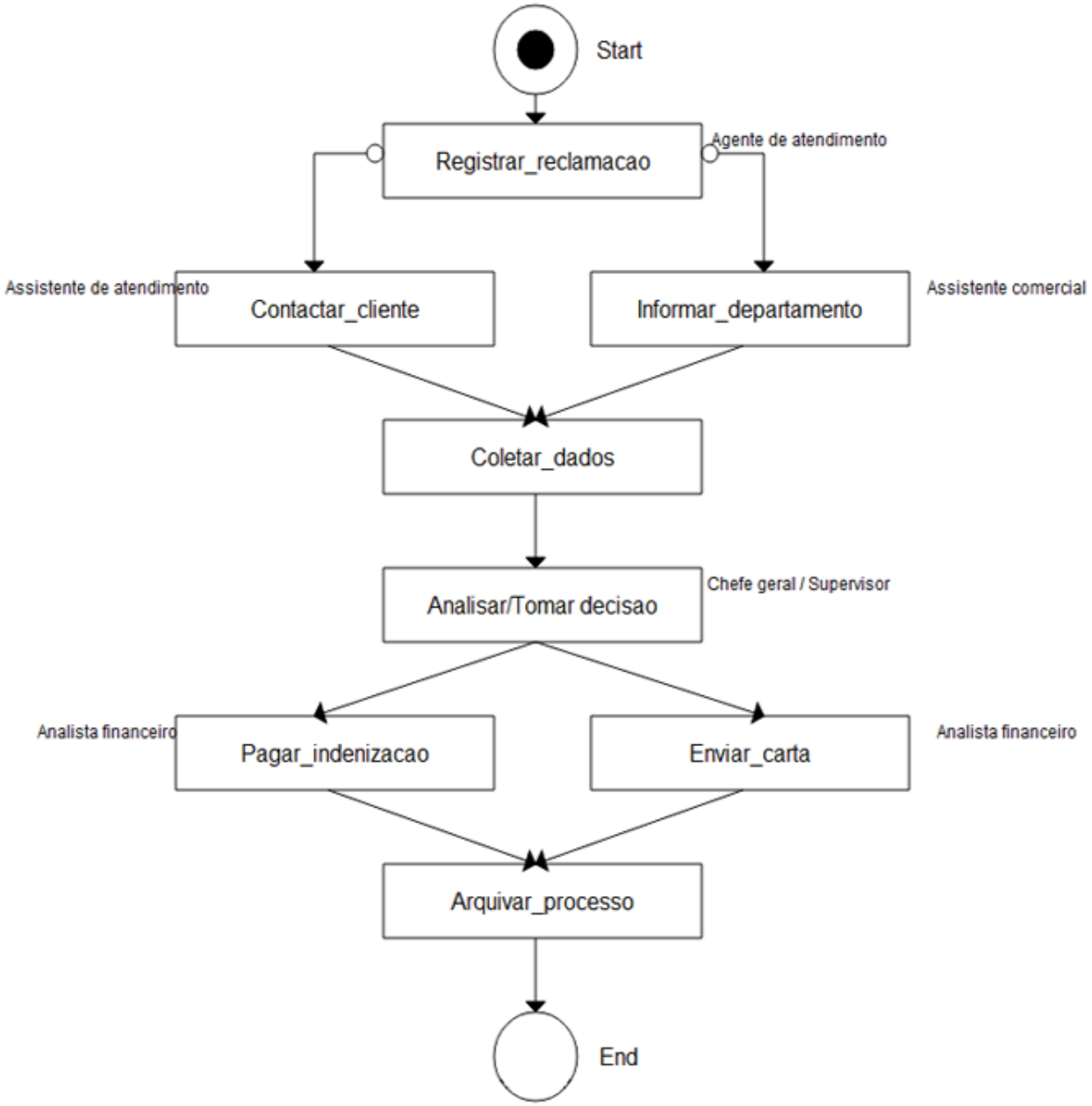

Figura 5.17: Resultado após o particionamento de papéis

Capítulo 6

Conclusão

6.1 Conclusão

Este trabalho está centrado no estudo de questões relativas à segurança em sistemas de *Workflow*. Assim, foi empregado um modelo matemático formal de controle de acesso baseado em papéis visando progressos na administração do acesso de usuários nestes sistemas.

O modelo de controle de acesso baseado em papel (RBAC) descreve mecanismos de segurança que controlam o acesso de usuários a recursos computacionais, fornecendo um modo de controle de autorizações e execução de tarefas em sistemas complexos. O RBAC do tipo Hierárquico foi utilizado por estruturar papéis permitindo a reflexão das linhas de autoridade e responsabilidade em uma organização. Foi realizado o particionamento vertical de papéis no Grafo de Papéis Hierárquico baseado no modelo de rede de Petri para o tratamento de reclamações.

Foi verificado que o modelo de particionamento vertical de papéis proposto em [Nyanchama and Osborn 1994] pode ser aplicado eficientemente em sistemas de gerenciamento de *Workflow*, contribuindo para maior segurança ao acesso a dados e consequentemente a descentralização de poderes atribuídos a um papel, constituindo-se como solução do problema de alocação de recursos humanos nestes sistemas.

A principal contribuição deste trabalho é apresentar uma abordagem de segurança para sistemas de *Workflow* incluindo a capacidade de expressar e impor uma política de segurança e simplificar o oneroso processo de gerenciamento de segurança nestes sistemas. Além disso, o controle de acesso também auxilia na alocação de recurso para o desempenho e distribuição de atividades aos usuários.

Como resultado do presente trabalho se tem o artigo "Análise e Simulação do Mecanismo de Alocação de Recursos Humanos em Sistemas de Workflow Combinado com Algoritmos de Controle de Acesso Baseado em Papéis", publicado no IX ENACOMP -

Departamento de Ciência da Computação do Campus Catalão da Universidade Federal de Goiás - Out/2011.

Como proposta de trabalhos futuros, tem-se:

- a atribuição da alocação de datas para a realização de tarefas no *software* "RdP Simulation";
- a inclusão de lugares no modelo conceitual gerado pelo "RdP Simulation";
- a construção de um compilador que contenha um editor de rede de Petri que permita a edição da rede de Petri, bem como a alocação de datas para a execução de atividades e a atribuição de papéis, que posteriormente serão traduzidos (compilados) para um sistema de gerência de *Workflow*.

Referências

Aalst, W.v.d., et al.. (1998). The Application of Petri Nets to Workflow Management. *The Journal of Circuits, Systems and Computers*.

Aalst, W.v.d., and Hee, K.v. (2002). Workflow Management: Models, Methods, and Systems. *The MIT Press Cambridge*, Massachusetts. Londres, Inglaterra.

ANSI/INCITS 359-2004. (2004) Information Technology: Role Based Access Control. *International Committee for Information Technology Standards*.

Araujo, R. M., e Borges, M. R. S. (2001). Sistemas de Workflow. *Publicado na XX Jornada de Atualização em Informática - Congresso da Sociedade Brasileira de Computação*. Fortaleza, Ceará, Brasil.

Cardoso, J. e Valette, R. (1997). *Redes de Petri*. Editora da UFSC. Florianópolis.

Casati, F.; Ceri, S.; Pernici, B.; Pozzi, G. (1995). Conceptual Modeling of Workflows. *Proceedings of OO-ER Conference*. Gold Coast, Austrália.

Champagnat, R.; Pingaud, H.; Alla, H. et al. (1998). *A gas storage example as a benchmark for hybrid modeling*. APII-JESA, Special Issue on Automation of mixed process and hybrid dynamical systems. Vol. 32.

Cruz, T. (2000) *Workflow: A tecnologia que vai Revolucionar Processos*. Ed. Atlas, São Paulo.

David, R. and Alla, H. (2004). *Discrete, Continuous, and Hybrid Petri Nets*. Springer.

Ferraiolo, D. and Cugini, J. and Kuhn, D. (1995). Role based access control: Features and motivations. *Computer Security Applications Conference*.

Ferraiolo D. and Sandhu, R.and Gravila, S. K. R. C. R. (2001). Proposed nist standard for role-based. *Transactions on Information and System Security*.

Fischer, L. and Moore, C.(1997). *Excellence in Pratice: Innovation and Excellence in Workflow and Imaging*. 1ed. vols. 1 e 2, Future Strategies.

Florin, G., Natkin, S. (1984). *Définition formelle des Réseaux de Petri Stochstiques*. Technical Report CNAM -Paris. Springer-Verlag.

Georgakopoulos, D,; Hornick, M.; Shet, A. (2004). *An Overview of Workflow Management: From Process Modeling to Workflow Automation Infrastructure*. Distributed and Parallel Databases.

Genrich, H. et al. (2000). Executable petri net models for the analysis of metabolic pathways. *21st International Conference on Application and Theory of Petri Nets*. Aarhus, Denmark. *Workshop Proceedings, Practical Use of High-level Petri Nets*

Gil, Y. et al. (2004). *Artificial Intelligence and Grids: Workflow Planning and Beyond*. IEEE Intelligent Systems, v. 19, n. 1.

Hollingsworth, D. (1985). The Workflow Reference Model. *Workflow* Management Coalition Document Number TC00-1003. Document Status - Issue 1.1.

Jensen, K. (1990). *Coloured Petri nets: a high level language or system design and analysis*. Advances in Petri Nets (G. Rozemberg, Ed). Lecture Notes in Computer Science. Springer Verlag.

Jeske, C. J. (2006). Mecanismo de alocação de recurso fuzzy para sistemas de gerenciamento de workflow. *Dissertação de mestrado, Universidade Federal de Uberlândia*.

Koch, M.and Mancini, L. and Parrisi-Presicce, F. (2002). A graph-based formalism for rbac. *ACM Transactions on Information and System Security (TISSEC)*.

Meidanis, J.; Vossen, D.; Weske, M. (1996). Using Workflow Management in DNA Sequencing. *Proceedings: The First International Conference on Cooperative Information Systems*. Los Alamitos, California, USA.

Merlin, P. (1974). *A study of the recoverability of computer systems*. University of California.

Merz, M. et al., (1995). *Workflow modeling and execution with coloured petri net*

Moncelet, G.; Christensen, S.; Paudeto M et al. (1998). Dependedability evaluation a simple mechatronic system using colored Petri Nets. *Workshop on practical use of colored Petri nets and Design CPN*.

Murata, T., (1989). Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, v. 77, n. 4.

Nicolao, M.(1998). *Modelagem de Workflow utilizando um Modelo de Dados Temporal Orientado a Objetos com Papéis*. Dissertação de Mestrado, Universidade Federal do Rio Grande do Sul, Porto Alegre.

Nyanchama, M. and Osborn, S. (1994). Access rights administration in role-based security systems. *Proceedings of the IFIP WG11.3 Working Conference on Database Security*.

Nyanchama, M. and Osborn, S. (1999). The role graph model and conflict of interest. *ACM Transactions on Information and System Security*.

OMG (2010). OMG Unified Modeling Language Specifitication. *Object Management Group*. `http://www.omg.org/spec/UML/2.2/Superstucture/PDF`. Acesso em: 20/04/2010.

Pereira, L. A. M. e Casanova, M. A. (2003). *Sistemas de Gerência de Workflows: Características, Distribuição e Exceções*. INF PUC-Rio.

Petri, C. A. (1962). *Kommunikation mit Automaten*. Tese (Doutorado). Technical University Darmstadt. Germany, 1962.

Petri Nets World (2011). `http://www.informatik.unihamburg.de/TGI/PetriNets/`. Acesso em: 10/02/2011.

Ramchandani, C. (1974). *Analysis of asynchronours concurrent systems by timed Petri nets*. Massachussests Institute of Technology.

Sandhu, R., and Kuhn, D. (2000). The nist model for role-based acess control: Towards a unified standard. *ACM Workshop on Role-Basec Acess Control*.

Sandhu, R. S.; Samarati, P. (1994). Access control: principles and practice. *IEEE Communications Magazine*.

Sandhu, R., Coyne, E., Feinstein, H., and Youman, C. (1996). *Role-based access control models*. IEEE Computer.

Sifakis, J. (1977). Use of Petri nets for Performance Evaluation. *3rd International Symposium on Modeling and Evaluation*. IFIP, North Holland.

Silbertin, C. (1985). *High-Level Petri nets with Data Structures*. In European Workflow on Application and Theory of Petri Nets. Ed. Digital Systems Laboratory. Helsink, Finland.

Silva, A. V. (2001). *Modelagem de Processos para Implementação de Workflow: uma avaliação crítica*. Rio de Janeiro.

Sommerville, I. (2003). *Engenharia de Software*. Editora Addison Wesley, 6th Edition.

WfMC (2010). *Workflow Management Coalition*. `http://www.wfmc.org/standards/XPDL.htm`. Acesso em: 09/04/2010.