UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
ENGENHARIA DE COMPUTAÇÃO

CASSIANA CHASSOT FÜLBER

SMARTIC: Uma Solução de Correlação de Incidentes Aplicada a um Ambiente Corporativo de Tecnologia da Informação

Trabalho de Conclusão apresentado como
requisito parcial para a obtenção do grau de
Engenheiro de Computação

Prof. Dr. Luciano Paschoal Gaspary
Orientador

Porto Alegre, julho de 2012

CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Fülber, Cassiana Chassot

SMARTIC: Uma Solução de Correlação de Incidentes Aplicada a um Ambiente Corporativo de Tecnologia da Informação / Cassiana Chassot Fülber. – Porto Alegre, 2012.

58 f.: il.

Trabalho de Conclusão – Universidade Federal do Rio Grande do Sul. Engenharia de Computação, Porto Alegre, BR–RS, 2012. Orientador: Luciano Paschoal Gaspary.

1. Gerenciamento de incidentes. 2. Gerenciamento de serviços de tecnologia da informação. 3. Correlação de incidentes. 4. Reincidência. I. Gaspary, Luciano Paschoal

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
Reitor: Profª. Carlos Alexandre Netto
Pró-Reitora de Graduação: Prof. Valquiria Linck Bassani
Diretor do Instituto de Informática: Prof. Luís da Cunha Lamb
Coordenador do curso: Prof. Sérgio L. Cechin
Bibliotecária-chefe do Instituto de Informática: Beatriz Regina Bastos Haro

“If I have seen farther than others,
it is because I stood on the shoulders of giants.”
— Sir Isaac Newton

AGRADECIMENTOS

Após anos de estudo e dedicação, orgulhosamente concluo hoje, com este Trabalho de Conclusão de Curso, minha Graduação. Durante esses anos, a presença e o apoio de familiares, amigos e colegas foi fundamental. Neste momento, agradeço a todos que fizeram parte desta história, em especial:

- A Luciano Paschoal Gaspary, meu orientador, por toda a dedicação e apoio durante a construção desse trabalho. Em nossas reuniões, sempre trouxe críticas construtivas e motivação para desenvolvermos um trabalho altamente qualificado.

- Aos meus Professores, que contribuíram não apenas para o aprendizado técnico, mas principalmente com a construção de caráter e da engenheira que agora me torno.

- Aos meus pais, pelo suporte emocional não apenas durante esse trabalho, mas durante todo o período desta Graduação. Ao meu pai, por me ensinar que o primeiro passo é decidir onde queremos chegar; depois, é preciso ir à luta e seguir o caminho até o fim. À minha mãe, por me ensinar que tudo na vida são amores, e que isso é o que deve determinar os caminhos a serem seguidos. Queridos, graças ao seu apoio e à sua dedicação conquistei mais essa vitória e continuo aceitando desafios, certa de que em vocês sempre terei um porto-seguro.

- A Tyron Scholem, pelo companheirismo, carinho e apoio de todos esses anos. Ninguém vivenciou tanto quanto tu minhas angústias e conquistas ao longo desse caminho. Em todos os momentos, estiveste ao meu lado, certo de que seu apoio faria diferença no resultado final. Com certeza essa trajetória foi muito melhor por ter sido trilhada ao teu lado!

- Às minhas sempre amigas, Vivian Pizzatto e Taís Bassani, por proporcionarem tantos momentos de alegria e sempre me impulsionarem a continuar em busca dos meus sonhos. Desde nossa infância, compartilhamos os momentos mais importantes, e nessa etapa vocês mais uma vez estiveram presentes.

- Às minhas amadas Manoela Trava Dutra e Ana Carolina Trava Dutra, por não me deixarem perder a esperança. As risadas e os choros compartilhados sempre nos moveram para mais uma ação em busca de nosso ideais. Cada atitude fez e vai continuar fazendo a diferença.

- Aos meus colegas, em especial a Leonardo Faganello, Matheus Proença, Henrique Klein, Jônatas Rech, Bruno Guedes e Thiago Santini, por compartilharem comigo

o dia-a-dia desta Graduação. Juntos vivemos momentos inesquecíveis e dividimos sentimentos únicos da experiência de tornar-se um engenheiro!

- Ao meu grande amigo Vitor Mariath, por todas as conversas e ensinamentos ao longo deste último ano. Quando o trabalho parecia acabar com minhas forças, sempre me levavas para tomar um café, acompanhado de uma boa conversa de revigorar os ânimos!
- À minha família, por entender minhas ausências, respeitar meu silêncio e sempre contribuir no que fosse possível para amenizar as dificuldades enfrentadas.
- À esta instituição, Universidade Federal do Rio Grande do Sul, e em especial ao Instituto de Informática e à Escola de Engenharia, por oportunizarem esses anos de estudo sob excelentes condições.

SUMÁRIO

LISTA DE ABREVIATURAS E SIGLAS

TI	Tecnologia da Informação
IT	Information Technology
ITIL	Information Technology Infrastructure Library
ITSM	Information Technology Service Management
CI	Configuration Item
SME	Subject Matter Expert
KPI	Key Performance Indicators
SLA	Service Level Agreement
ITS	Incident Ticket System
CMDB	Configuration Management Database
KB	Knowledge Base
CBR	Case-Based Reasoning
BDIM	Business-Driven IT Management
SYMIAN	SYMulation for Incident ANalysis
IP	Internet Protocol
RAM	Random-access Memory
DB	Database
OS	Operating System
CPU	Central Processing Unit
OLEDb	Object Linking and Embedding Database
TAD	Tipo Abstrato de Dados
MSAGL	Microsoft Automatic Graph Layout
ERP	Enterprise Resource Planning
PI	Process Integration
BI	Business Inteligence

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ALGORITMOS

RESUMO

Recentemente, o Gerenciamento de Serviços de Tecnologia da Informação (ITSM, *Information Technology Service Management*) tem sido foco de diversas pesquisas na comunidade acadêmica e na indústria. Em um contexto corporativo de TI, o gerenciamento de incidentes destaca-se como peça fundamental na busca pela excelência de ITSM. Segundo a Biblioteca de Infraestrutura de Tecnologia da Informação (ITIL, *Information Technology Infrastructure Library*), um incidente é caracterizado como um evento que diverge da operação normal de um serviço. O principal objetivo do gerenciamento de incidentes é restabelecer a operação normal do ambiente no menor tempo possível, com impacto mínimo. Esse trabalho propõe a análise de conjuntos de incidentes periodicamente, entendendo como estes se correlacionam, com o objetivo de identificar as melhores formas de atuar no ambiente para manter a qualidade do serviço.

Palavras-chave: Gerenciamento de incidentes, gerenciamento de serviços de tecnologia da informação, correlação de incidentes, reincidência.

ABSTRACT

Recently, Information Technology Service Management (ITSM) has been the focus of various studies, both in academic community and in industry. In a corporate IT context, incident management stands out as a key role in pursuing excellence in ITSM. According to Information Technology Infrastructure Library (ITIL), an incident is defined as an event which diverges from a service normal operation. The main objective of incident management is to reestablish the normal operation of the environment as quickly as possible and with minimum impact. This work proposes the analyses of a set of incidents periodically, understanding how they correlate, intending to identify the best ways of acting in the environment in order to maintain quality of service.

Keywords: Incident management, information technology service management, incident correlation, recurrence.

1 INTRODUÇÃO

Considerando um ambiente corporativo de Tecnologia da Informação (TI), o gerenciamento de incidentes é uma das principais atividades relacionadas à manutenção da saúde do ambiente como um todo. Segundo a Biblioteca de Infraestrutura de Tecnologia da Informação (ITIL, *Information Technology Infrastructure Library*), um incidente consiste em qualquer evento que não faz parte da operação normal de um serviço e que pode impactar sua qualidade ou sua disponibilidade. O gerenciamento de incidentes tem como principal objetivo restaurar o funcionamento normal do ambiente, no menor tempo possível e com mínimo impacto. Além disso, igualmente importante é a prevenção desses incidentes, mitigando o risco de sua ocorrência. Para alcançar tais objetivos, é necessária a atuação de um time de suporte técnico (responsável por restabelecer o ambiente) e de um gerente de incidentes (que coordena o processo). Dependendo da complexidade do problema observado, um incidente pode ser solucionado com a atuação de apenas um técnico ou de uma equipe.

Considerando o alto número de dispositivos propensos a falhas, o volume diário de incidentes pode facilmente ultrapassar centenas de registros. Neste cenário, não raro diversos incidentes repetem-se ou estão relacionados a uma mesma causa raiz (KANG et al., 2010). Com as soluções disponíveis atualmente, a detecção dessas correlações é difícil e normalmente é de responsabilidade do gerente de incidentes, onerando o processo como um todo. Quando o humano falha nessa correlação, possivelmente a causa raiz dos incidentes não será identificada, resultando em novos incidentes tratados individualmente e mais esforços empregados desnecessariamente.

Realizar uma correlação de incidentes de forma automatizada tem três grandes benefícios para o gerenciamento de incidentes. O primeiro é que o agrupamento de incidentes correlacionados diminui o volume de registros a serem analisados, reduzindo a carga de trabalho do time de suporte. O segundo é que a chance de erro na correlação entre eventos é menor, pois tipicamente o gerente de incidentes fica sobrecarregado com esta tarefa e nem sempre consegue analisar todo o volume de registros. O terceiro é que a qualidade da solução dos incidentes é melhorada, uma vez que os técnicos têm menos incidentes para tratar individualmente e, assim, conseguem focar melhor em suas atividades e identificar mais facilmente a causa raiz dos incidentes.

Nesse contexto, diversos trabalhos foram documentados na literatura recente com o objetivo principal de melhorar o processo de gerenciamento de incidentes como um todo. No entanto, os trabalhos analisados são limitados, como será melhor detalhado no Capítulo 3, principalmente no que tange os tipos de incidentes abordados e as premissas assumidas em termos de dados conhecidos.

Para suprir as limitações das soluções disponíveis e proporcionar melhorias no processo de gerenciamento de incidentes, elaboramos uma solução conceitual para identificar

incidentes correlacionados. O foco desse trabalho é a análise de incidentes identificados através de sistemas de monitoração de *hardware* e de aplicação. Não são contemplados incidentes reportados por usuários, em linguagem natural, visto que estes normalmente são minoria e que seu objetivo tipicamente é alteração de algum serviço, e não o reporte de uma degradação de ambiente. Além disso, nos casos em que o usuário percebe a degradação do ambiente, espera-se que a monitoração também identifique a situação. Logo, a prevenção da ocorrência desse tipo de incidente e seu correto tratamento evitam também a ocorrência de incidentes de insatisfação do usuário. Como prova de conceito, implementamos um protótipo de sistema de correlação de incidentes, e o testamos sobre uma base de dados sintética.

No Capítulo 2, revisamos fundamentos de gerenciamento de incidentes e de prestação de serviços de TI. No Capítulo 3, comentamos trabalhos relacionados ao tema e como a nossa proposta pode contribuir para o avanço dos estudos na área. No Capítulo 4, apresentamos uma visão geral da implementação da solução e quais as etapas que a compõe. No Capítulo 5, mostramos os resultados da aplicação da solução proposta sobre uma base de dados de incidentes. Finalmente, no Capítulo 6, concluímos os estudos e apresentamos propostas de continuidade como atividades futuras.

2 FUNDAMENTOS

2.1 Ambiente corporativo de TI

Considerando um contexto corporativo de TI, o Gerenciamento de Serviços (ITSM, *Information Technology Service Management*) consiste em um conjunto de diversos processos estreitamente relacionados e integrados. Dentre esses, podemos destacar o gerenciamento de mudanças, o gerenciamento de problemas e o gerenciamento de incidentes. Esses processos compartilham fundamentalmente o fato de lidarem diretamente com a manutenção da saúde do ambiente, seja evitando a ocorrência de falhas, corrigindo-as ou mesmo identificando sua causa raiz.

Tipicamente o ambiente de TI é composto basicamente por servidores (onde as aplicações são executadas), *storages* (conjuntos de discos onde os dados são armazenados) e *switches* (que fazem a comunicação entre esses *hardwares*). Ainda, temos bibliotecas de fitas e servidores dedicados exclusivamente ao *backup* dos dados. Além disso, na camada de aplicação existem diferentes bancos de dados, sistemas de monitoração, aplicações do cliente, entre outros. Esses componentes de *hardware* e *software*, citando um jargão da área de TI, são conhecidos como itens de configuração ou CIs (*Configuration Item*). Considerando os diversos componente citados, conclui-se que a complexidade do cenário de TI é proporcional ao volume e à diversidade das aplicações consideradas (ANEROUSIS; DIAO; HECHING, 2011).

A gestão do ambiente de TI representa um desafio à parte, levando muitas corporações de médio e grande porte a contratarem prestadoras de serviço, terceirizando esta atividade. Neste caso, podem ser considerados tanto requisitos de implementação como de manutenção do ambiente, dependendo da necessidade do cliente.

Evidentemente, o cliente espera que esse ambiente seja controlado e que, uma vez identificada uma falha, essa seja imediatamente tratada, minimizando o impacto ao negócio. Logo, além da existência de uma Central de Serviço para atendimento ao usuário final, faz-se necessária a implantação de um serviço de monitoração. Essa pode ser aplicada tanto em nível de *hardware*, quanto nas aplicações, e é tão robusta quanto maior for a necessidade do cliente de garantir estabilidade. Uma vez que o sistema de monitoração identifica uma falha (ou uma propensão à falha), um alerta deve ser gerado para que uma equipe de suporte seja acionada (GUPTA et al., 2009).

2.2 Metodologia ITIL

Percebendo a complexidade deste cenário, a comunidade de TI trabalhou no desenvolvimento da ITIL, um conjunto de melhores práticas para ITSM. Atualmente, a metodologia ITIL é reconhecida mundialmente como um guia para gerenciamento de serviços,

sendo reconhecida, inclusive, pelo ISO 9000.

Segundo a terminologia ITIL (Office of Government Commerce, 2006), um incidente pode ser definido como qualquer evento que não faça parte da operação normal de um serviço e que causa, ou pode causar, uma interrupção ou redução da qualidade desse serviço. Incidentes podem ser categorizados em incidentes de aplicação (como indisponibilidade de um serviço ou alta utilização de disco); de *hardware* (como servidor indisponível ou problemas de configuração); ou requisições de serviços (como requisições de informações ou troca de senha). Assim, tipicamente, incidentes de aplicação ou de *hardware* são identificados através da monitoração implantada no ambiente de TI, enquanto que as requisições de serviços partem do cliente ou do time de suporte.

Dependendo do modelo de gestão de TI, o incidente, quando reportado, pode ter mais ou menos informações. No entanto, alguns dados são essenciais para o entendimento da situação, como qual componente apresenta degradação, quando esta situação foi detectada, qual time solucionou o incidente, entre outros. Além disso, a metodologia ITIL define que incidentes devem ser classificados em severidades, de acordo com o impacto que têm no negócio do cliente.

Normalmente são consideradas três severidades de incidente. Falhas com severidade 3 representam um alerta, indicando que o time técnico deve verificar se a situação reportada é normal ou não, e qual ação pode ser tomada para evitar que se agrave. Severidade 2 consiste em uma falha que reflete uma degradação do ambiente para o usuário final. Quando há indisponibilidade da aplicação, o incidente é considerado como severidade 1.

Após a identificação do incidente, este deve ser tratado pelo time de suporte. ITIL preconiza que existam três linhas de atendimento, conforme apresentado na Figura 2.1. A primeira linha de atendimento tem como papel registrar e classificar os incidentes. Esse time procura dar uma solução imediata para a situação identificada, executando procedimentos documentados; caso não consiga resolver o incidente, este será encaminhado para o segundo nível. No segundo nível, o incidente é analisado por um especialista no assunto (SME, *Subject Matter Expert*), conforme classificado pelo primeiro nível. O time de segundo nível é responsável por restaurar, o mais rápido possível, o ambiente onde foi identificada a situação. A terceira linha de suporte é composta por fornecedores de *hardware* e *software*, que são acionados quando o time de segunda linha não é capaz de identificar a causa e a solução do incidente, necessitando de informações mais precisas sobre o produto.

Do ponto de vista da gerência, ITIL defende que para avaliar o desempenho do processo de gerenciamento de incidentes devem ser utilizados Principais Identificadores de Desempenho (KPI, *Key Performance Indicators*):

- número total de incidentes;
- tempo médio de resolução do incidente;
- porcentagem de incidentes que não ultrapassam o tempo de solução ou de atendimento definido no contrato de prestação de serviço, respeitando o Acordo de Nível de Serviço (SLA, *Service Level Agreement*);
- custo médio por incidente;
- porcentagem de incidentes solucionados na primeira linha de atendimento;
- taxa de incidentes processados por número de funcionários no time de suporte;

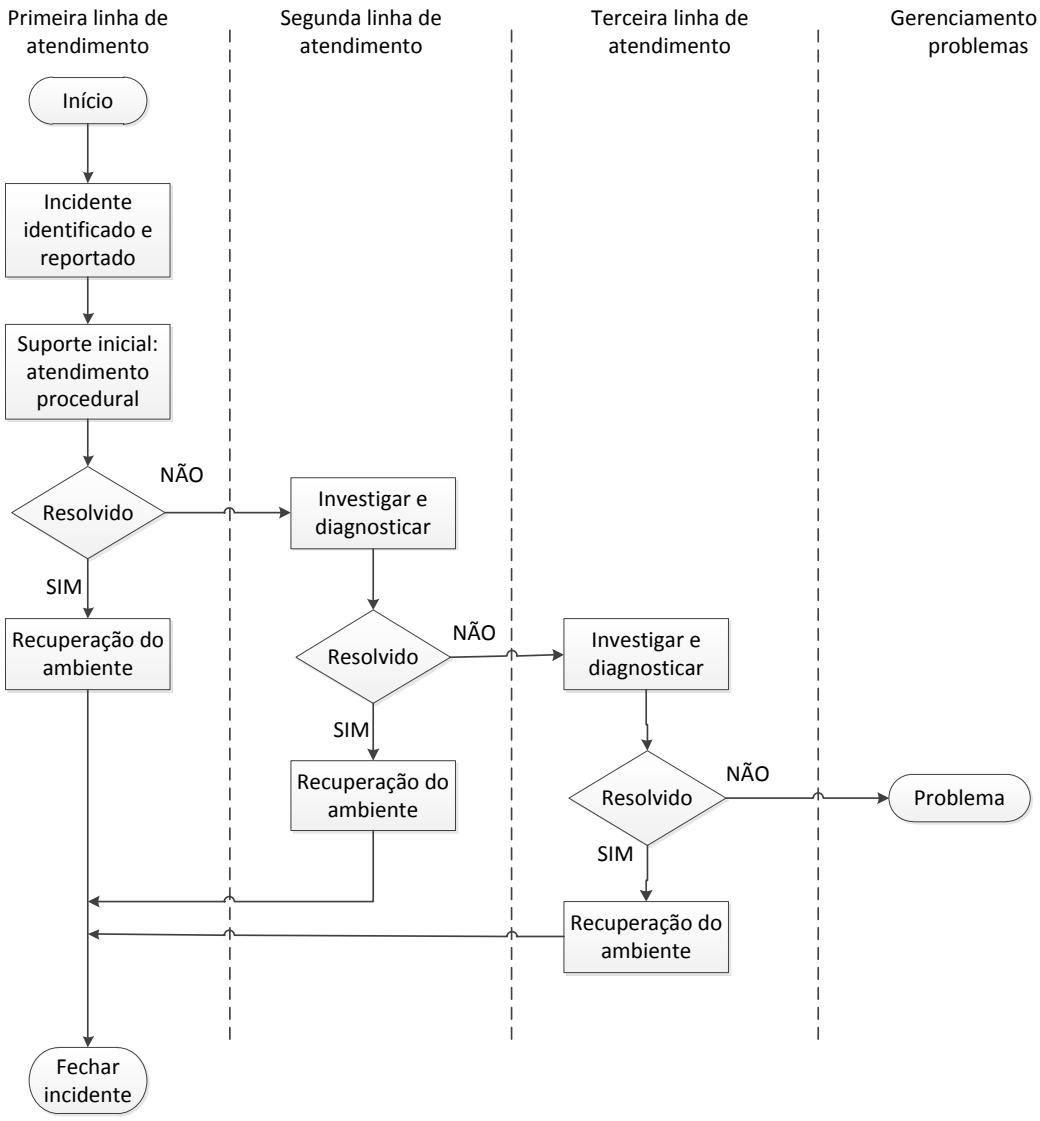

Figura 2.1: Fluxograma de roteamento de incidentes entre as diversas linhas de atendimento.

- número e porcentagem de incidentes atendidos remotamente (sem necessidade de intervenção local).

2.3 Gerenciamento de incidentes

O gerenciamento de incidentes pode ser entendido como um conjunto de etapas a serem seguidas, desde a identificação da situação até sua solução. Dessa forma, faz-se necessária uma hierarquia composta por gerente de incidentes e time de suporte. O time de suporte tem como papel registrar e classificar os incidentes, encaminhá-los para os times de suporte específicos (quando necessário), prover uma solução e fechar o incidente. Na realidade, o time de suporte como um todo é composto pelas três linhas de suporte comentadas previamente. Por outro lado, é clara a necessidade de um gerente de incidentes, responsável por coordenar todo este processo, incluindo algumas tarefas como: controlar a eficiência e eficácia do gerenciamento de incidentes; produzir relatórios para

níveis superiores de gerência; supervisionar o time de suporte; analisar o cumprimento do processo e a qualidade do atendimento; e garantir manutenção dos sistemas utilizados. A Figura 2.2 apresenta um modelo típico de gerenciamento de incidentes, explicado na sequência.

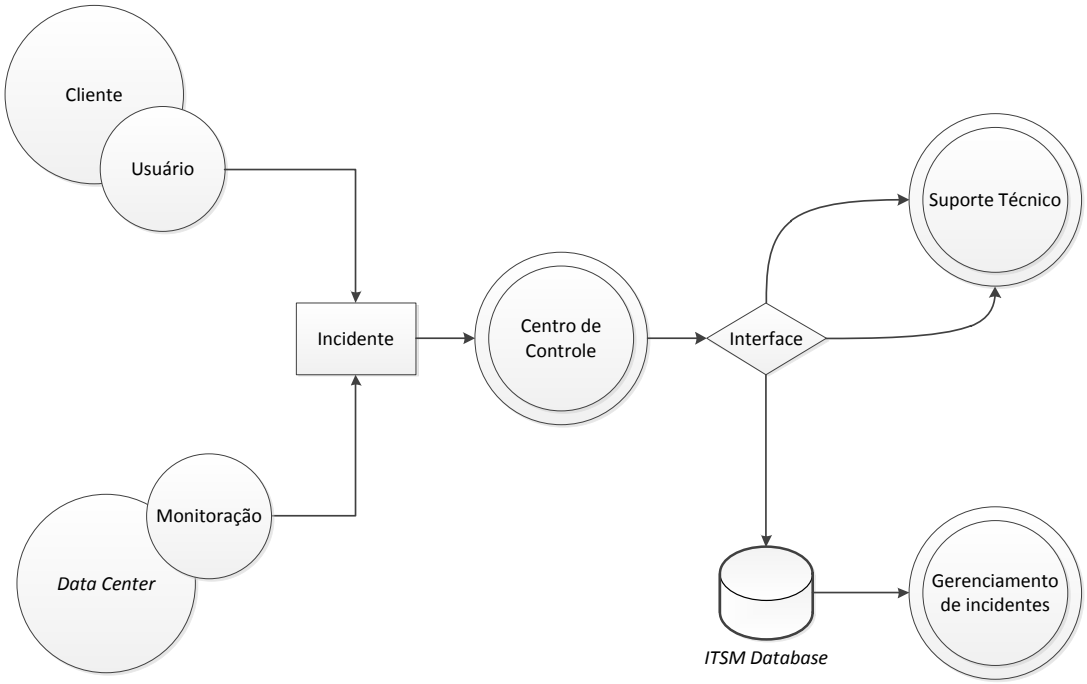

Figura 2.2: Modelo tradicional de gerenciamento de incidentes.

Existem duas formas básicas de identificação de um incidente: incidente reportado por um usuário do serviço de TI (cliente) ou incidentes reportados por um sistema de monitoração, que monitora os ativos de um *Data Center*. Os incidentes abertos pelo cliente tipicamente são descritos com linguagem natural, sem nenhuma padronização. Muitas vezes, o usuário que reporta o incidente sequer consegue identificar adequadamente qual é situação ou onde ela se manifesta. Cabe à primeira linha de atendimento interpretar os dados fornecidos. Por outro lado, incidentes provenientes de um sistema de monitoração são gerados por um agente instalado nos servidores, que coleta periodicamente informações sobre o ambiente; assim, usualmente seguem uma estrutura padronizada de descrição.

Ambas as categorias de incidentes são atendidas por um Centro de Controle, composto pela primeira linha de atendimento. O técnico irá reportar o incidente, classificá-lo e tentar resolvê-lo. Para reportar, classificar, resolver e mesmo fechar um incidente, utiliza-se, via de regra, um sistema, sendo as informações de cada incidente registradas em um banco de dados. Caso seja necessário encaminhar o incidente para o Suporte Técnico (segunda linha de atendimento), este visualizará o incidente pela mesma interface, dando continuidade ao processo.

Durante todo esse processo, o gerente de incidentes deve acompanhar o time de suporte, verificando se os incidentes estão sendo atendidos dentro dos prazos estabelecidos e seguindo o processo definido pela companhia. Periodicamente, o gerente de incidentes gera relatórios a partir da base de dados de incidentes, para analisar o desempenho do processo. Através dos relatórios, o gerente de incidentes extrai os KPIs conforme

recomendado pelo ITIL. Além disso, é feita uma análise de reincidências, que, potencialmente, indicam um problema maior no ambiente.

Considerando que os principais objetivos do gerenciamento de incidentes são o restabelecimento do ambiente em caso de indisponibilidade ou degradação, e a prevenção dessas situações, dois tipos de correlação de eventos são analisados: relações de causa e efeito e reincidências. A primeira alternativa objetiva melhorar o desempenho do processo de recuperação do ambiente. A segunda busca analisar quais as situações e ativos mais ofensores e quais ações podem ser tomadas para evitar a recorrência desses incidentes.

Embora as duas propostas resultem em melhorias claras no processo de gerenciamento de incidentes, nem sempre é possível colocá-las em prática. O time de suporte técnico é sobrecarregado por um alto volume diário de incidentes, e quiçá observa relação entre diferentes registros ou reincidências. O gerente de incidentes, por sua vez, não tem como prerrogativa para sua função compreender tecnicamente o ambiente, e assim não tem o treinamento necessário para identificar diversas correlações entre diferentes incidentes reportados. De uma forma geral, o trabalho do gerente de incidentes acaba se tornando um processo bastante manual e oneroso, mas o resultado desse empenho nem sempre é claro(GUPTA; HIMA PRASAD; MOHANIA, 2008).

Dessa forma, uma falha no ambiente que resulta em diversos incidentes possivelmente não será observada como um problema único, mas sim como eventos isolados. Para exemplificar essa situação, considere o seguinte cenário: a indisponibilidade de um servidor pertencente a um *cluster* operando no modo *ativo-ativo* pode prejudicar o desempenho de uma aplicação que executa em outro nodo do *cluster* (pois este segundo nodo fica sobrecarregado ao atender as demandas do primeiro). Como os eventos são de categorias diferentes (servidor indisponível *versus* aplicação com baixo desempenho) e servidores diferentes, possivelmente o relacionamento entre essas falhas passará despercebido. Considerando um outro exemplo, analisando uma situação de um servidor com diversas instâncias de banco de dados, gerando incidentes de indisponibilidade, pode ser considerada uma reincidência, pois o mesmo servidor apresenta o mesmo tipo de falha (indisponibilidade de banco de dados). No entanto, se uma das instâncias falha periodicamente e corresponde à maioria dos incidentes, enquanto que as demais falharam apenas uma vez, apenas o primeiro caso deveria ser considerado reincidência.

Dessa forma, fica claro que existem problemas no gerenciamento de incidentes. Assim, propomos uma solução inteligente, configurável e automatizada de análise de correlação de incidentes, com o objetivo de reduzir os custos do processo e melhorar a acurácia das análises. O projeto agrega valor ao negócio e qualidade à prestação de serviços, pois uma vez identificadas correlações entre incidentes e reincidências, as ações tomadas como solução real do problema podem ser trabalhadas de fato em sua origem, com uma compreensão clara das falhas.

2.4 Gerenciamento de mudanças e gerenciamento de problemas

Além do gerenciamento de incidentes, a metodologia ITIL propõe dois outros processos fortemente relacionados: o gerenciamento de mudanças e o gerenciamento de problemas.

Toda vez que alguma alteração for executada no ambiente, essa deve ser registrada através do processo de mudança. O principal objetivo desse processo é garantir que o ambiente de TI e o negócio estejam e se mantenham alinhados de forma eficiente e com o mínimo possível de interrupções e riscos associados. O escopo do processo cobre mu-

danças em ativos e CIs do ambiente, podendo ser tanto uma mudança corretiva quanto uma implementação de uma requisição de aditivo. Nesse contexto, o gerenciamento de mudanças está intimamente relacionado com o gerenciamento de incidentes em dois sentidos. O primeiro deles é que uma mudança pode introduzir novos incidentes no ambiente, como consequência da alteração realizada. Por outro lado, uma mudança pode ter como objetivo solucionar a causa raiz de um ou mais incidentes e, assim, diminuir o volume diário de registros (CASTAGNA LUNARDI et al., 2009).

O gerenciamento de problemas, por sua vez, tem como principais objetivos minimizar o impacto de incidentes e problemas identificados no ambiente e previnir recorrência desses eventos. Para isso, fundamentalmente este processo busca identificar a causa raiz dos incidentes. Segundo a metodologia ITIL, um problema é a causa não identificada de um ou mais incidentes, e um erro conhecido é um problema que foi coretamente identificado e tratado. Nesse sentido, podemos inclusive afirmar que a proposta deste Trabalho de Graduação promove melhorias também no gerenciamento de problemas, uma vez que promove uma visão integrada de incidentes provavelmente correlatos.

3 TRABALHOS RELACIONADOS

A área de ITSM tem recebido grande atenção da comunidade científica e da indústria no que tange gerenciamento de incidentes. Neste capítulo fazemos uma síntese de alguns dos principais estudos que se relacionam com o nosso trabalho. Marcu *et al*. (MARCU et al., 2009) consideram duas categorias de incidentes: incidentes gerados por sistemas de monitoração do ambiente (*resource tickets*) e incidentes reportados pelo usuário (*service tickets*). No primeiro caso, o sistema de controle de incidentes (ITS, *Incident Ticket System*) armazena *tickets*, que são registros estruturados que descrevem a situação observada por um agente de monitoração, que periodicamente coleta dados do ambiente. No segundo caso, o usuário final reporta a sua percepção do problema que atinge a infraestrutura de TI. Quando um usuário reporta um incidente relatando uma situação de degradação ou indisponibilidade do ambiente, pressupõe-se que também o sistema de monitoração tenha identificado a mesma situação. Logo, embora estruturalmente muito diferentes, as duas categorias de incidentes têm alto potencial de correlação. A partir dessa constatação, os autores do trabalho destacam correlações entre eventos de diferentes categorias (*i.e.*, incidentes gerados por humanos e por sistemas de monitoração), que não são imediatamente identificadas através de uma análise manual.

O principal objetivo do referido trabalho é reduzir o volume de incidentes redundantes e, assim, também o custo do gerenciamento de incidentes e de atendimento. Marcu *et al*. propõem que a correlação entre os incidentes ocorra no momento em que um usuário reporta um mal comportamento do serviço, oportunidade em que o mecanismo proposto procura incidentes correlatos gerados pelo sistema de monitoração. Com isso, a expectativa dos autores é que incidentes relacionados sejam agrupados e tratados conjuntamente. Para tal, os autores propõem três etapas de relacionamento entre os incidentes: correlação baseada em categorias, que filtra os incidentes por meio de regras de similaridade; correlação baseada em itens de configuração críticos para o funcionamento do serviço; e, finalmente, correlação temporal. Nos três casos, as correlações são estabelecidas considerando incidentes reportados pelo usuário e incidentes gerados pelo sistema de monitoração. Inicialmente, incidentes que são relacionados a uma mesma categoria de serviços são agrupados através de regras de similaridade e ordenados temporalmente. Paralelamente, identificam-se os CIs que se relacionam com o serviço que apresenta problema ou com a categoria desse serviço. Caso essas duas etapas tenham sucesso, os incidentes inicialmente selecionados são refinados, restando apenas os que incluem em sua descrição os CIs identificados como críticos para o serviço. Caso contrário, utilizam-se CIs da árvore de dependência de *Business Service CIs*, para que esses sejam usados no refinamento dos incidentes inicialmente selecionados.

No trabalho apresentado por Gupta *et al*. (GUPTA; PRASAD; MOHANIA, 2008), os autores propõem um mecanismo de automatização do processo de correlação de in-

cidentes. Analogamente ao trabalho de Marcu *et al*., Gupta também identifica as duas categorias de incidentes, mas intensifica seu estudo apenas em incidentes reportados por usuários. A estratégia assemelha-se a do trabalho anteriormente apresentado, baseando a correlação de incidentes no relacionamento e na dependência entre diferentes CIs cadastrados no CMDB (*Configuration Management Data Base*), que contém dados de relacionamentos entre os ativos. Para implementar esse mecanismo de correlação, os autores propõem um algoritmo, explicado a seguir. A primeira etapa do algoritmo consiste em identificar palavras-chave na descrição dos incidentes feitos pelo usuário. Para tal, cada incidente é editado em busca de palavras típicas de um CMDB (como os próprios CIs, endereços IP e tipos de CI) usando um dicionário; em seguida, as palavras são normalizadas conforme o padrão de nomenclatura utilizado no CMDB em questão. A partir daí, o algoritmo executa em busca de incidentes correlatos, *i.e.*, através de regras de similaridade que consideram diversos atributos de um incidente, incluindo o CI identificado.

A título de exemplo, considere o cenário apresentado nas Figuras 3.1 e 3.2. No exemplo da Figura 3.1, os incidentes 1 e 2 estão claramente relacionados, pois possuem em sua descrição o mesmo ativo. Por outro lado, na Figura 3.2 os incidentes 3 e 4 também são correlatos, embora a percepção dessa relação dependa de conhecimentos prévios de dependência entre CIs. No caso, os ativos 2 e 3 estão relacionados ao ativo 4 (sendo essa relação conhecida graças ao CMDB) e, portanto também os incidentes em questão potencialmente são correlatos.

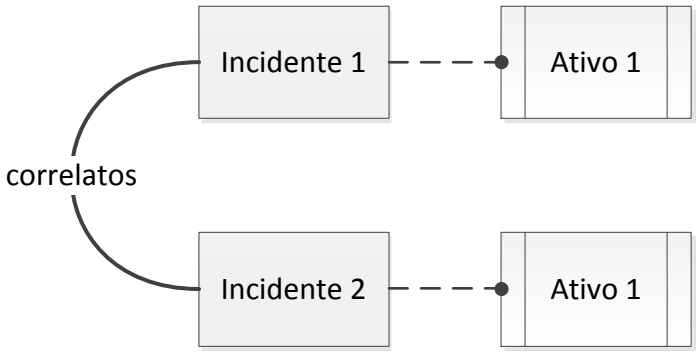

Figura 3.1: Incidentes diretamente correlacionados; ambos contém em sua descrição o mesmo ativo.

Gupta destaca um modelo de organização da infraestrutura de atendimento ao usuário baseada em níveis de atendimento. Nesse modelo, inicialmente todo incidente é atendido por um operador da Central de Serviço, que tentará identificar e classificar o problema reportado pelo usuário. Assim, embora o incidente originado no sistema de ITSM consista fundamentalmente na percepção do usuário que reportou a falha, o operador pode procurar entender a situação mais amplamente, questionando o usuário. Após essa interpretação inicial, conforme a categoria do incidente, o próprio operador tenta solucionar o problema, através de procedimentos. Caso não tenha sucesso na normalização da situação, o incidente é repassado ao especialista na tecnologia, conhecido como SME. O SME atende o incidente ou delega a atividade a algum integrante de seu time, treinado para atividades de maior complexidade. Se ainda assim não for possível normalizar o ambiente, um terceiro nível de atendimento é acionado, sendo esse time composto por recursos altamente qualificados e especializados na tecnologia que apresenta os sinto-

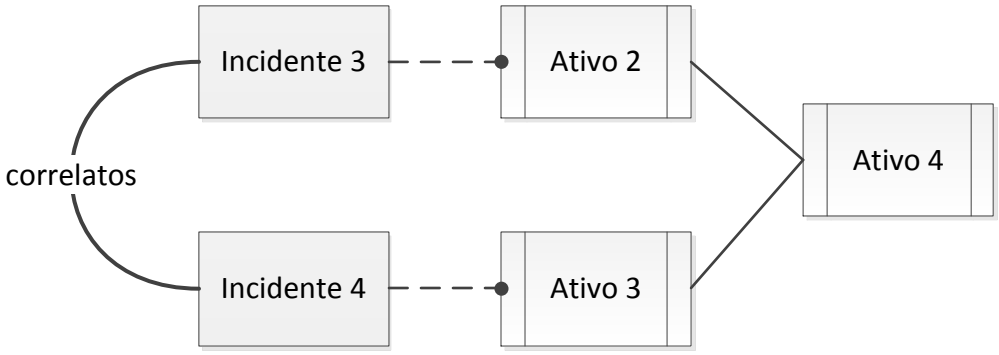

Figura 3.2: Incidentes indiretamente correlacionados; cada incidente refere-se a um ativo, mas ambos estão relacionados com o Ativo 4.

mas. Dessa forma, percebe-se que quanto mais alto o nível de atendimento, mais custoso torna-se o processo de gerenciamento de incidentes. Nesse contexto, os autores propõem que um maior número de incidentes seja solucionado pelo primeiro nível de atendimento, reduzindo os custos e diminuindo a carga de trabalho sobre os SMEs.

Os autores partem da premissa de que incidentes semelhantes possuem também resoluções semelhantes. Assim, pode-se utilizar incidentes já solucionados como modelo para atendimento de novos incidentes. Aumentando o número de incidentes documentados e utilizando o algoritmo de busca proposto sobre os dados do ITS, os operadores aumentam o conjunto de incidentes sobre o qual possuem domínio e conhecem a solução. Assim, constitui-se uma espécie de base de conhecimento (KB, *Knowledge Base*), aumentando o escopo das atividades solucionadas pelo primeiro nível de atendimento e, portanto, diminuindo a carga de trabalho dos SMEs e o custo do processo como um todo. É para lidar com essa redução de esforço e custo que o mecanismo de correlação de incidentes é aplicado.

Ambos trabalhos destacam fortemente os incidentes abertos pelo usuário final, no momento em que um problema ou uma certa degradação da qualidade de serviço são evidentes. Nesse quesito, nosso trabalho difere no sentido em que abordamos fundamentalmente incidentes reportados pelo sistema de monitoração, com o objetivo de solucioná-los e normalizar o ambiente antes mesmo que o usuário seja impactado. Ainda, Gupta *et al*. e Marcu *et al*. têm como premissa a existência de um CMDB estruturado e completo, o que é uma exigência muito forte. Tal se deve ao fato de que a implementação e a manutenção de um CMBD é custosa e onerosa, fazendo com que grande parte das empresas não disponha dessa base. Nosso trabalho, por sua vez, dispensa a existência de um CMDB, sendo necessário apenas que cada incidente indique qual o ativo a que se refere. Desse modo, não se faz possível comparar CIs cuja relação não é conhecida, mas aumenta-se o possível escopo de análise.

Raman *et al*. (RAMAN; CROSS, 1996) também apresentam um trabalho interessante utilizando o conceito de Raciocínio Baseado em Casos (CBR, *Case-based Reasoning*). CBR é uma metodologia utilizada para solucionar um problema baseando-se no conhecimento adquirido na solução prévia de problemas semelhantes. O trabalho apresentado propõe o uso da técnica de CBR para melhorar o desempenho da primeira linha de atendimento, buscando na base de incidentes registros semelhantes e qual a solução dada em cada caso.

Basicamente, o sistema de CBR implementado é dividido em cinco etapas: representação e armazenamento do caso (semelhante à etapa de identificação e registro mencionada no ITIL); casamento de padrões do incidente com casos previamente solucionados; adaptação da solução recuperada; validação; e atualização do sistema com as informações da solução aplicada. O ponto mais interessante desta proposta, que destaca-se em relação aos outros trabalhos revisados, é a etapa de validação (*feedback*). Os autores implementam uma interface que permite que o técnico revise a solução do algoritmo, validando ou não a proposta. Dessa forma, quando o algoritmo acerta uma solução, seu peso é incrementado, e sua posição no *ranking* sobe. A cada nova consulta, o algoritmo prioriza soluções bem qualificadas no *ranking*, ou seja, que foram confirmadas diversas vezes como sendo as soluções ideais para uma determinada classificação de incidentes. O trabalho evidencia a importância de padronização e formatação dos dados avaliados, previamente à sua manipulação, e da interação com o usuário, permitindo que o sistema adapte-se à necessidade e à dinâmica do ambiente avaliado. Esses são aspectos que de certa forma influenciaram a concepção do projeto apresentado nesse Trabalho de Graduação.

Finalmente, discutimos os trabalhos de Claudio Bartolini ((BARTOLINI; STEFANELLI; TORTONESI, 2010) e (BARTOLINI; SALLE; TRASTOUR, 2006)), que apresentam um sistema de suporte à tomada de decisão, fundamentada em um modelo de gerenciamento conhecido como BDIM (*Business-driven IT Management*). A ferramenta, intitulada SYMIAN (*SYMulation for Incident ANalysis*), auxilia na tomada de decisões considerando cenários *what-if*, analisando e otimizando o processo de gerenciamento de incidentes. O protótipo permite a construção de modelos de ambientes reais de suporte de TI, avaliação de desempenho e indicação de melhorias resultantes de alterações estruturais e comportamentais. O modelo consiste basicamente em injetar diversos incidentes em um ambiente simulado de TI, sendo os parâmetros dos mesmos determinados de acordo com regras de construção. A seguir os incidentes são despachados para filas de atendimento de um grupo de suporte. O tempo de solução de um incidente considera tanto as suas características quanto as regras de comportamento definidas para o time de atendimento. Por exemplo, um time pode ter uma política que prega o atendimento de incidentes críticos com alta prioridade, dando menos importância para incidentes mais simples; assim, no caso de abertura de um incidente de alta criticidade, mesmo que um recurso estivesse atendendo um incidente simples, ele seria interrompido e passaria a atender o novo incidente. Por outro lado, a política do time poderia ser alocar incidentes para os recursos do time de acordo com a complexidade do incidente e a qualificação do técnico; nesse caso, poderíamos ter tanto recursos subutilizados quanto sobrecarregados.

Resumidamente, o SYMIAN permite que seu usuário simule cenários *what-if* através de um modelo de uma organização de TI. Para tal, o ambiente de TI é tratado como um conjunto de filas abertas. Cada grupo de suporte é tratado como uma fila multiservidores, com um determinado período de operação e um número limitado de recursos. Nesse contexto, incidentes são injetados no sistema através de um gerador de incidentes. Assim, a simulação do ambiente de TI é dada pela relação entre os grupos de suporte e a entidade geradora de incidentes.

Considerando os incidentes criados e as políticas dos grupos de suporte, a simulação mostra indicadores, considerados pelos autores como críticos para avaliação da qualidade do processo de gerenciamento de incidentes. A maior contribuição desse trabalho consiste na apresentação dos dados da simulação de forma que fique claro para o gerenciamento de incidentes qual estratégia pode melhorar o desempenho e a qualidade do serviço. Esse trabalho não tem uma relação tão direta com o trabalho apresentado. Todavia, este é de

notável contribuição para a comunidade científica no que tange gerenciamento de incidentes, sendo uma referência fundamental para este Trabalho de Graduação.

4 SOLUÇÃO DE CORRELAÇÃO

Apesar dos trabalhos apresentados anteriormente evidentemente contribuírem para os estudos relacionados ao gerenciamento de incidentes, não fez parte do escopo desses trabalhos correlações mais complexas, como relações de causa e efeito. Dessa forma, as deficiências comentadas do processo de gerenciamento de incidentes ainda não são supridas. Para preencher essa lacuna, propomos uma solução de correlação de eventos, detalhada a seguir.

Conforme explicamos no Capítulo 2 (mais especificamente na Figura 2.2), um sistema típico de gerenciamento de incidentes em um ambiente de TI considera incidentes reportados por usuários e incidentes reportados por sistemas de monitoração. Esses incidentes são tratados por um Centro de Controle ou pelo time de Suporte Técnico, sob supervisão do Gerente de Incidentes. Os registros dos incidentes são armazenados em uma base de dados (referenciada como *ITSM Database*), sendo que cada registro contém uma série de atributos que descrevem a situação observada e seu tratamento. Retomando esse contexto, a Figura 4.1 apresenta nossa proposta, que integra-se ao processo de gerenciamento de incidentes.

Considerando este cenário, propomos um sistema de correlações que recebe como entrada (1) os incidentes produzidos automaticamente por sistemas de monitoração. Após o processamento desses registros pelo nosso sistema de correlações, os dados processados e relacionados são apresentados ao gerente de incidentes (2) e ao time de suporte técnico (3). Entendemos que o processo de correlação de incidentes pode ser segmentado em três componentes fundamentais: normalização, correlação e visualização, conforme podemos observar na Figura 4.1. Primeiramente, é necessário normalizar a massa de incidentes analisada, selecionando os atributos importantes. Depois de normalizados, os registros de incidentes passam por um algoritmo de correlação, buscando identificar relações não-óbvias no ambiente. Finalmente, os resultados dessa análise serão apresentados em uma interface para o gerente de incidentes e para o time de suporte. A seguir, descrevemos conceitualmente cada uma dessas etapas.

Inicialmente, destacamos algumas premissas consideradas na elaboração dessa solução. A primeira delas considera que incidentes reportados por um sistema de monitoração seguem uma estrutura pré-determinada. Em segundo lugar, consideramos exclusivamente correlações sobre um mesmo ativo. Dessa forma, inferir correlações entre dois ativos distintos não faz parte do escopo desse trabalho. Ainda, consideramos que, no ambiente em que a solução é implementada, um incidente é encerrado tão logo a sua solução tenha sido alcançada. Assim, desconsideramos a possibilidade de que o operador do sistema de controle de incidentes aguarde um período para fechar um grupo de incidentes, a menos que suas soluções de fato ocorram conjuntamente. Entendemos que, dependendo do ambiente em que a solução venha a ser empregada, essa é uma premissa que pode ser

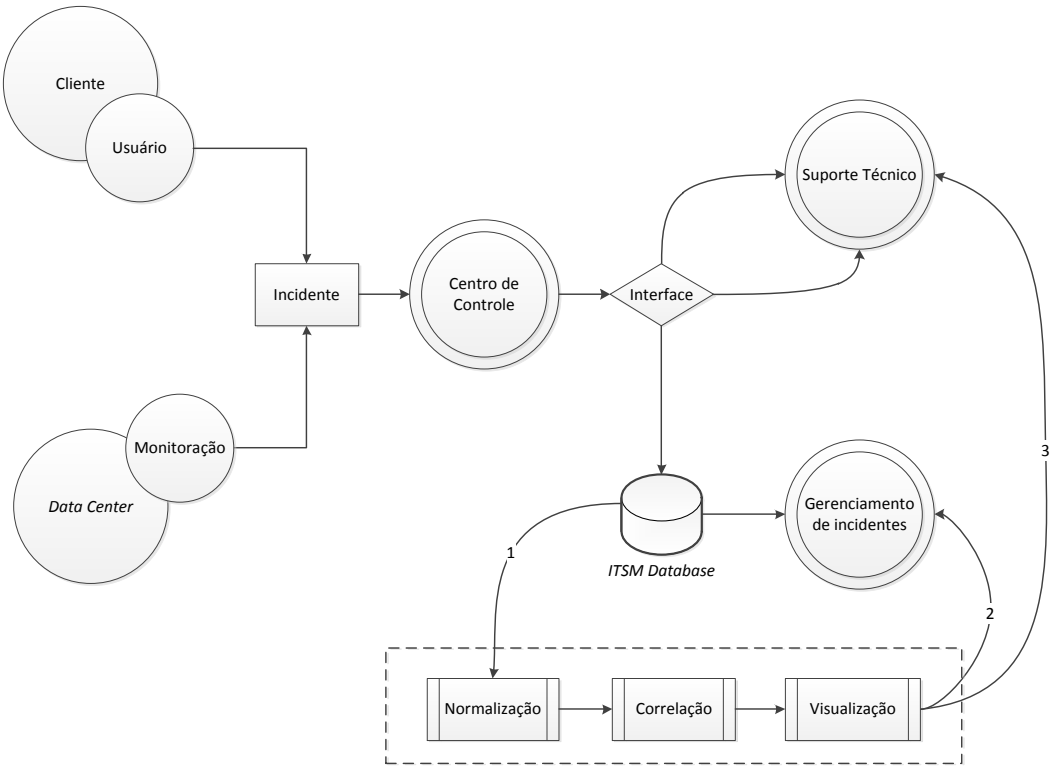

Figura 4.1: Visão geral do processo de gerenciamento de incidentes e da solução proposta.

entendida como forte. Por outro lado, em uma corporação que tenha um processo mais sistematizado, essa premissa é plenamente factível. Nessa primeira iteração para lidar com correlações de eventos essa premissa é mantida, ficando o seu relaxamento como trabalho futuro.

4.1 Normalização dos dados

Tipicamente, sistemas de gerenciamento de incidentes consideram um número alto de atributos, sendo que muitos deles não são utilizados para o entendimento da situação reportada, sendo necessários apenas para processos de auditoria. Portanto, consideramos que um conjunto reduzido de informações é suficiente para a identificação da causa raiz de uma situação de degradação ou indisponibilidade. Dessa forma, a etapa de normalização do nosso algoritmo consiste em ler os incidentes reportados no sistema de monitoração, selecionando apenas os dados de interesse. Como exemplo, a Tabela 4.1 mostra campos tipicamente utilizados em sistemas de gerenciamento de incidentes, seu significado e se esse campo é mantido em nossa normalização.

Tabela 4.1: Campos típicos de um registro de incidente em um sistema de gerenciamento de incidentes.

Campo no Sistema	**Descrição**	**Essencial para Correlações**
Incident	Identificador único do incidente	Sim

Tabela 4.1 (continuação)

Campo no Sistema	**Descrição**	**Essencial para Correlações**
Classification	Classificação do incidente de acordo com grandes áreas (por exemplo *hardware*, *software*, administração de usuário, entre outros)	Não
Reported Date	Data e hora em que o incidente foi aberto	Sim
Affected Date	Data e hora em que o incidente é percebido pela monitoração (tipicamente idêntico à data reportada, mas pode diferir em alguns minutos)	Não
Created By	Indivíduo ou sistema que criou o incidente	Não
Created by Group	Grupo ao qual pertence o indivíduo que criou o incidente (não se aplica em caso de incidentes gerados por sistemas de monitoração)	Não
Affected Person	Indivíduo ou área afetada pelo incidente (depende de conhecimento de uma matriz de responsabilidade, relacionando cada ativo com um responsável)	Não
Summary	Resumo do incidente	Não
Details	Descrição completa do incidente	Sim
Reported Priority	Prioridade com que o incidente foi reportado (equivalente à severidade referenciada pelo ITIL)	Sim
Internal Priority	Prioridade com a qual o incidente foi tratado (pode ser igual ou inferior a prioridade reportada)	Não
Status	Estado do incidente (aguardando atendimento, em atendimento ou encerrado)	Sim
Asset	Ativo no qual a situação de degradação ou indisponibilidade foi identificada pelo agente de monitoração	Sim
Location	Localização do ativo identificado	Não
Asset Site	Sítio de localização do ativo (tipicamente utilizado para localização dentro de um *data center*)	Não
Vendor	Empresa responsável pelo ativo reportado	Não
Assigned to	Nome do técnico que está atendendo o incidente	Não

Tabela 4.1 (continuação)

Campo no Sistema	**Descrição**	**Essencial para Correlações**
Owner Group	Time de suporte ao qual pertence o técnico que está atendendo o incidente	Não
Symptom	Sintoma do incidente, ou seja, uma descrição em linguagem natural feita pelo técnico no momento de encerramento do incidente	Não
Cause	No momento do encerramento do incidente o técnico descreve qual a causa do incidente, no seu entendimento	Não
Resolution	Solução dada ao incidente, descrita pelo técnico no momento do encerramento	Não
Resolution Code	Código referente ao modo de solução do incidente (por exemplo, *reset* de senha, limpeza de *log*, extensão de *filesystem*, entre outros)	Não
Work log	Campo de escrita livre utilizado pelos técnicos para descrever em mais detalhes alguma atividade	Não
Audit log	*Logs* de auditoria; registram transferências do incidente entre diferentes times de suporte ou entre técnicos	Não
Target Start	Data e hora limites para o início do atendimento, respeitando métricas de contrato (SLA)	Não
Target Resolution	Data e hora limites para solução do incidente, respeitando métricas de contrato (SLA)	Não
Actual Start	Data e hora em que o técnico inicia o atendimento (tipicamente muito semelhante à data de abertura do incidente, devido ao atendimento de primeiro nível)	Não
Actual Resolution	Data e hora de solução (encerramento) do incidente	Sim

Assim, decidimos tratar apenas informações essenciais para a inferência de correlação, simplificando a complexidade do *ticket*. Os campos de *logs* (*Work log* e *Auditlog*) podem ser descartados, uma vez que são utilizados fundalmentalmente em casos de auditoria. Também são desconsiderados campos referentes ao fechamento dos incidentes (*Symptom*, *Cause* e *Resolution*), dado que a etapa de correlação proposta inicia antes do fechamento do *ticket*. Informações sobre quais as áreas afetadas e seus responsáveis dependem do conhecimento de uma matriz de responsabilidade e, quiçá, de um CMDB; portanto, muitas vezes são campos vazios ou apresentam informações desatualizadas, sendo

desconsideradas em nossa análise. Referente à data e horários do incidente, consideramos apenas a data de abertura e de solução do incidente. O campo *Details*, por sua vez, é um campo fundamental em nossa análise, pois é ele que contém a descrição estruturada resultante do sistema de monitoração.

Sabendo que os incidentes gerados por um sistema de monitoração seguem uma estrutura padronizada em sua descrição, é possível trabalhar sobre esse campo para extrair informações úteis. Por exemplo, o campo *Classification*, indicado pelo ITIL como sendo necessário, está fortemente relacionado com a descrição da situação. A classe do incidente, em resumo, indica genericamente qual o estado do ativo, ou qual o tipo de situação observada, sendo esse dado fundamental para inferência de correlações. Assim, entendemos que a classe do incidente pode ser determinada analisando sua descrição através de regras. Em nossa solução, realizamos um casamento de padrões entre regras pré-definidas e a descrição de cada incidente para determinar sua classe.

Assim, a normalização dos incidentes consiste em escrever um novo registro para cada incidente (contendo um número reduzido de atributos) e determinar sua classe através da descrição. Após a normalização dos dados, esses podem ser tratados pela solução de correlação implementada. Nesse momento, o volume de dados a serem analisados consiste em um conjunto reduzido, já que diversos atributos são desconsiderados. Em síntese, os atributos considerados de cada incidente são:

- *Incident*: identificador único;
- *Details*: descrição estruturada do incidente;
- *Classification*: classe do incidente, determinada através da análise de sua descrição;
- *Reported Date*: data em que o incidente foi identificado pelo sistema de monitoração;
- *Reported Priority*: prioridade do incidente, conforme reportado pelo sistema de monitoração;
- *Asset*: código do ativo sobre o qual o incidente foi identificado;
- *Actual Resolution*: data de resolução (fechamento) do incidente.

4.2 Algoritmo de correlação

Após a normalização, os incidentes não mais são tratados como originalmente, mas sim como registros reduzidos de informações. A próxima etapa da solução consiste em avaliar esses dados em busca de correlações entre eles. O objetivo dessa correlação é tratar grupos de incidentes correlatos conjuntamente, diminuindo a carga de trabalho do time de suporte e melhorando a qualidade do serviço de gerenciamento de incidentes. Dessa forma, acredita-se que a solução dada para os incidentes tem potencial para ser mais precisa, atacando sua verdadeira causa raiz. Nesse contexto, consideramos dois tipos de correlações: reincidências e relações de causa e efeito.

A reincidência é identificada através da recorrência de incidentes da mesma classe sobre um mesmo ativo em um determinado período de tempo. A análise de reincidências permite que sejam identificadas as situações mais recorrentes no ambiente. Entendendo que esses incidentes correlatos possam ter a mesma causa, a compreensão e o fechamento

de um incidente podem levar à solução e quiçá prevenção de todo o conjunto de incidentes de uma vez só.

O segundo tipo de correlação que abordamos nesse trabalho são as relações de causa e efeito. Para isso, buscamos observar temporalmente encadeamentos de incidentes em um determinado período. Quando um grupo de incidentes referentes a um mesmo ativo é aberto e atendido em janelas temporais que se sobrepõem, entendemos que existe uma grande probabilidade de que esses incidentes estejam correlacionados, mesmo que suas classes sejam diferentes, ou mesmo que o time de suporte seja outro. Analogamente ao caso anterior, entendemos que esses incidentes podem ser tratados conjuntamente, reduzindo o esforço do time de suporte. Inclusive, a análise de incidentes de diversas classes conjuntamente pode enriquecer a análise da situação e contribuir para a identificação da causa raiz do conjunto de incidentes. Sendo a causa identificada com mais precisão, também o plano de ação para restabelecimento do ambiente e a prevenção de novos incidentes têm potencial para serem feitos com mais precisão e qualidade.

Nessa etapa da solução, os dados disponíveis correspondem aos incidentes já normalizados. Considerando esses registros, cada incidente tem uma classe (campo *Classification*). Na nossa solução de correlação, propomos que cada classe de incidente seja tratada como o estado de um ativo. Conforme incidentes vão sendo abertos sobre um mesmo ativo, consideramos que esse percorre diferentes estados. Assim, a sequência de incidentes de diferentes classes provoca uma transição entre estados. Ainda, caso um incidente da mesma classe seja aberto, consideramos uma transição do estado para ele próprio, ou seja, um *loop*.

Dessa forma, os incidentes gerados sobre um ativo permitem a geração de um multidígrafo, que representa as diversas situações enfrentadas conjuntamente. Mais especificamente, um multidígrafo é um grafo direcionado que permite ocorrência de *loops* em um nodo. Para facilitar a leitura, deste ponto em diante utilizaresmo os termos multidígrafos e grafos de forma indistinta.

Como objetivamos inferir correlações entre esses eventos, entendemos que cada ativo deve ter não apenas um grafo, mas sim um conjunto de grafos, sendo que cada uma delas representa um momento em que grupos de incidentes foram observados. Basicamente, entendemos que um grupo de incidentes de um mesmo ativo, quando temporalmente próximos, podem estar relacionados. Uma alternativa simples seria agrupar incidentes cuja data de abertura estivesse contida em uma janela de tempo. No entanto, entendemos essa solução como simplória, e propomos um modelo mais complexo de agrupamento, considerando interpolações de janelas temporais. No Algoritmo 4.1 apresentamos um pseudo-código do nossa solução, explicado detalhadamente a seguir; o algoritmo apresentado representa a análise feita sobre um conjunto de incidentes de um único ativo.

Conforme explicado anteriormente, cada ativo pode ter um ou mais grupos de incidentes, ou seja, um conjunto de grupos de incidentes (linha 2). A cada novo incidente aberto para o ativo em questão, o primeiro passo é verificar se existe um grupo de incidentes aberto (linha 4). Em caso positivo, adicionamos o novo incidente no grupo e incrementamos o contador de incidentes abertos nesse grupo (linhas 5 e 6). Além disso, recuperamos a informação do estado atual do ativo (variável *estadoAtual*, linha 7) e definimos o novo estado como sendo a classe do referido incidente (variável *novoEstado*, linha 8). Nesse ponto, precisamos verificar qual a transição de estados que deve ser feita. Para isso, primeiramente verificamos se a classe do incidente já pertence ao grupo (linha 9). Em caso positivo, basta adicionarmos a transição entre o estado atual e o novo estado no grupo (linha 10) e redefinir o estado atual (linha 11). Caso a classe ainda não faça parte do grupo

```
1  para cada ativo faça
2      novo ConjuntoGrupos conjuntoGrupos;
3      para cada incidenteAberto faça
4          se grupo.GetStatus() = aberto então
5              grupo.AdicionaIncidente(incidenteAberto);
6              grupo.incidentesAbertos++;
7              estadoAtual = grupo.getEstado();
8              novoEstado = incidenteAberto.Classe;
9              se existe(incidenteAberto.Classe, grupo) então
10                 grupo.AddTransicao(estadoAtual, novoEstado);
11                 grupo.setEstado(novoEstado);
12             senão
13                 grupo.AddEstado(novoEstado);
14                 grupo.AddTransicao(estadoAtual, novoEstado);
15                 grupo.setEstado(novoEstado);
16             fim
17         senão
18             new Grupo grupo;
19             grupo.SetStatus(aberto);
20             grupo.AddEstado(incidenteAberto.Classe);
21             grupo.SetEstado(incidenteAberto.Classe);
22             conjuntoGrupos.AddGrupo(grupo);
23         fim
24     fim
25     para cada incidenteFechado faça
26         grupo.incidentesAbertos−−;
27         se grupo.incidentesAbertos = 0 então
28             grupo.SetStatus(fechado);
29         senão
30             break;
31         fim
32     fim
33 fim
```

Algoritmo 4.1: Algoritmo principal da solução de correlação de incidentes.

(linha 12), significa que essa é a primeira ocorrência de um incidente dessa classe nesse grupo. Logo, precisamos adicionar um novo estado (linha 13) e então definir a transição entre os estados (linha 14). Por outro lado, caso não haja nenhum grupo aberto (linha 17), é preciso criar um novo grupo, com *status* correspondente a *aberto*, e cujo estado corresponde à classe do incidente aberto (linhas 18 a 21). Finalmente, o novo grupo deve ser inserido no conjunto de grupos do ativo (linha 22). Assim, conclui-se a parcela do algortimo referente ao tratamento de incidentes abertos.

Por outro lado, para cada incidente fechado, devemos decrementar o número de incidentes abertos no grupo (linha 26). Caso o número de incidentes abertos seja igual a zero, então esse grupo é definido como fechado (linha 28). Caso contrário, o algoritmo prossegue, e o grupo permanece aberto.

Genericamente, cada novo incidente aberto para o mesmo ativo antes do fechamento de todos os demais incidentes é incluído em um grupo. Quando todos os incidentes de um grupo estão fechados, também o grupo é fechado e é dentro desse grupo que devemos observar relações de causa e efeito. Dessa forma, entendemos que os incidentes pertencentes a um grupo compartilharam janelas temporais, o que indica que se manisfestaram conjuntamente e, assim, entendemos que provavelmente são eventos correlatos.

Uma vez explicado o algoritmo de construção dos grupos de incidentes e dos conjuntos de grupos, exemplificamos a execução do algoritmo. Para facilitar a visualização e compreensão dos exemplos, consideramos apenas a criação e fechamento de um grupo de incidentes, e não de um conjunto de grupos. A Figura 4.2 e a Figura 4.3 exemplificam respectivamente casos de reincidência e relações de causa e efeito, e são explicadas detalhadamente a seguir.

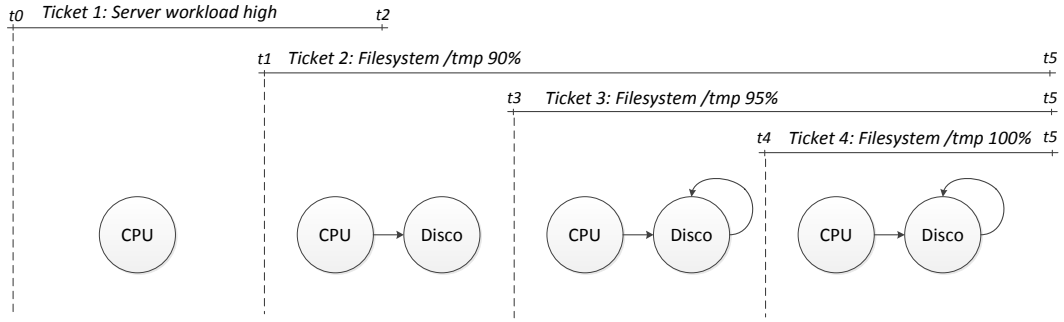

Figura 4.2: Simulação de um conjunto de incidentes, representados através de multidígrafos. A correlação entre os incidentes demonstra reincidências.

Na Figura 4.2, um novo grupo é aberto no momento de criação do incidente *Ticket 1* no momento *t0*. Conforme o resultado da etapa de normalização, a classe desse incidente é *CPU*, e sua descrição é *Server workload high*, indicando que o servidor está com alta carga, implicando em alto consumo da capacidade de processamento. A seguir, o incidente *Ticket 2* é aberto sobre o mesmo ativo no instante *t1*, dessa vez relativo ao uso de um sistema de arquivos (classe *Disco*), conforme vemos por sua descrição. Analisando globalmente a situação, podemos inferir que a alta utilização de CPU deve-se à execução de uma ou mais rotinas onerosas, que, inclusive, devem estar escrevendo em arquivos armazenados no sistema de arquivos */tmp*, provocando a abertura desses dois primeiros incidentes. Observe que no instante *t2* o *Ticket 1* é fechado, mas isso não altera o nosso grafo; esse incidente é apenas marcado como fechado no grupo. Ainda, no instante *t3* um

segundo incidente (*Ticket 3*) relativo a utilização do sistema de arquivos */tmp* é aberto. Como esse incidente pertence à mesma classe do incidente *Ticket 2*, apenas cria-se um laço sobre o nodo *Disco*. No instante *t4*, o terceiro incidente relativo a mesma situação é reportado, sem provocar alterações estruturais no grafo. Finalmente, no instante *t5* os três incidentes ainda abertos (todos relativos à utilização do sistema de arquivos */tmp*) são fechados. Nesse instante, também o grupo é fechado, dado que não resta nenhum incidente aberto nesse intervalo temporal.

Neste exemplo podemos observar dois pontos interessantes do algoritmo de correlação. O primeiro deles é que a data de fechamento do incidente não altera a estrutura do grafo, apenas decrementa o contador de incidentes abertos no grupo. O segundo ponto é que, após a instanciação de um estado (correspondente à classe do incidente) ou de uma transição entre estado, essa não mais precisa ser repetida na estrutura do grafo. Embora seja importante saber quantos e quais os incidentes relativos a cada estado, não é necessário modificar o grafo.

Analisando qualitativamente o grafo, observamos duas características de registros tratados separadamente: incidentes provavelmente correlatos (incidente da classe *CPU* e incidentes da classe *Disco*), e, principalmente, a ocorrência de três incidentes da mesma classe (inclusive com a mesma descrição). Assim, o grafo permite visualizar a evolução do quadro do servidor, proporcionando um entendimento da situação como um todo, além de indicar implicitamente uma forma de prevenção dos demais incidentes.

Embora esse seja um caso típico de ocorrência de incidentes em um ambiente de TI, a Figura 4.3, apresentada e explicada a seguir, demonstra uma situação igualmente típica, cuja correlação não é tão evidente.

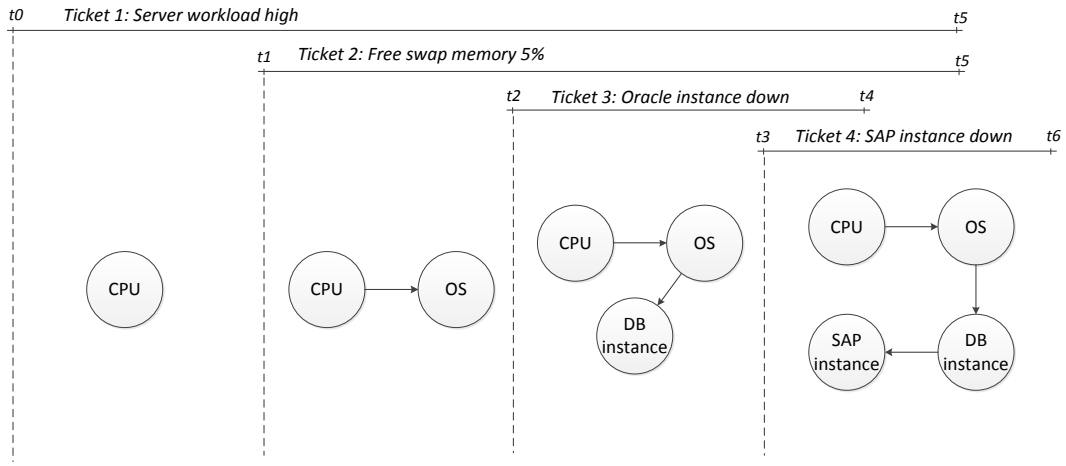

Figura 4.3: Simulação de um conjunto de incidentes, representados através de multidígrafos. A correlação entre os incidentes demonstra relações de causa e efeito.

Assim como no exemplo anterior, o grupo de incidentes deste ativo é iniciado com a abertura do incidente *Ticket 1* no momento *t0* relativo ao alto uso de processamento do servidor. Tipicamente, esse incidente, por se tratar de uma situação relativa à infraestrutura (classe *CPU*), seria tratado pelo time de suporte responsável pelo servidor, seja este um time de infraestrutura local ou de sistema operacional. A seguir, o incidente *Ticket 2* reporta alta utilização da memória de *swap* do servidor, indicando que as rotinas em andamento estão consumindo muito esta área. Em casos como esse, pode ocorrer degradação tanto da aplicação quanto do sistema operacional, dado que a demanda das rotinas por

este recurso estão altas. Tipicamente, a área de *swap* é uma área em disco utilizada para armazenar páginas de memória que estejam inativas. Essa é uma área de baixo desempenho em relação à memória RAM, sendo utilizada apenas em casos em que a RAM está 100% utilizada. Portanto, quando ocorre um alerta de pouca área de *swap* livre, entendemos que também a memória RAM está esgotada. Logo, o servidor dispõe de poucos recursos para a escrita, sendo que essa operação pode implicar em escrita em disco, troca de páginas da memória, entre outros, prejudicando o desempenho das rotinas correntes. Embora esse incidente seja de uma classe diferente do primeiro (classes *CPU* e *OS*), este incidente também seria tipicamente tratado pelo mesmo time do *Ticket 1*. Muitas vezes, o próprio sistema operacional, para previnir uma ruptura de seu serviço, para os serviços de aplicação, com o objetivo de liberar a área de *swap* e continuar executando suas próprias rotinas. Como consequência, no momento *t2*, o incidente *Ticket 3* é aberto e indica que uma instância de banco de dados Oracle não está respondendo. Como a própria classe do incidente indica (*DB instance*), essa situação é tratada pelo time de banco de dados. Ainda, no instante *t3* o incidente *Ticket 4* é recebido e reporta que a própria instância do SAP[1] não está respondendo. Embora esse tipo de incidente seja comumente tratado isoladamente pelo time de SAP, claramente neste caso está relacionado com a indisponibilidade do banco de dados.

O grafo apresentado permite a observação que estes incidentes são correlatos. No entanto, dado que tipicamente estes são tratados por times de suporte distintos, a correlação entre eles muitas vezes não é identificada e, por mais que sua causa seja a mesma, os incidentes são tratados e solucionados individualmente. Os tempos de fechamento dos incidentes ratificam essa observação; observe que o *Ticket 3* é o primeiro a ser fechado (instante *t4*). Uma interpretação plausível para esse caso seria que a parada do banco de dados e da aplicação SAP (instantes *t2* e *t3*) tenha cancelado algumas das rotinas onerosas, liberando a área de *swap* e possibilitando o restabelecimento do banco de dados no instante *t4*. No instante *t5* o time de infraestrutura observa que o uso do *swap* e dos processadores foi normalizado, e encerra os incidentes *Ticket 1* e *Ticket 2*. Finalmente, no instante *t6* também a aplicação consegue ser reiniciada, uma vez que a situação do servidor está normalizada e o banco de dados está disponível.

Se este grupo de incidentes fosse tratado conjuntamente, os três times estariam envolvidos e a causa raiz do problema poderia ser tratada verdadeiramente, dado que a evolução da ssituação observada no servidor seria compreendida completamente. Além disso, imediatamente após o momento em que o banco de dados foi restabelecido os demais incidentes poderiam ter sido fechados.

Dessa forma, percebemos que a correlação dos incidentes em grupos temporalmente associados claramente traz benefícios à correlação de eventos, que, por sua vez, permite um gerenciamento de incidentes mais rico.

4.3 Visualização

Após a correlação entre os eventos, entendemos que o suporte visual ao usuário é fundamental. Conforme explicado anteriormente, inferimos as correlações sobre grupos de incidentes, e consideramos esses grupos como multidígrafos. Os exemplos apresentados nas Figura 4.2 e Figura 4.3 apresentam uma prévia representação dos grafos, sendo os nodos equivalente às classes dos incidentes e as transições representando o encadeamento

[1]Referente a aplicações da empresa alemã SAP (*Systeme, Anwendungen und Produkte in der Datenverarbeitung*), criadora de *softwares* de gestão de empresas.

entre os eventos. Ainda, entendemos que a visualização, além de apresentar os grafos, deve permitir acesso aos incidentes de cada classe. Além disso, numerando as transições entre os estados temos uma visão mais completa da ordem de ocorrência dos eventos, enriquecendo ainda mais o cenário. Com isso, redesenhamos as figuras Figura 4.2 e Figura 4.3 considerando os multidígrafos com as características de visualização desejadas, conforme apresentado, respectivamente, nas Figuras 4.4 e 4.5.

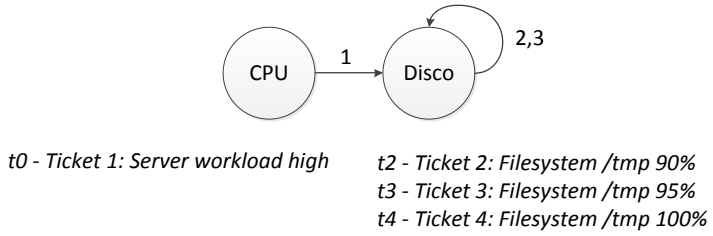

Figura 4.4: Proposta de visualização do grafo final da Figura 4.2.

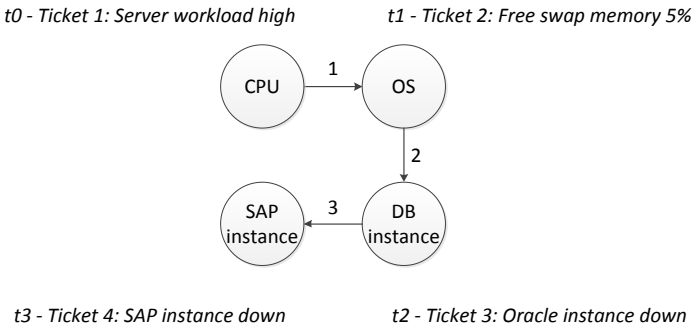

Figura 4.5: Proposta de visualização do grafo de estados final da Figura 4.3.

Dessa forma, percebemos como a visualização agrega valor e contribui para o entendimento da situação do ambiente no momento dos incidentes. Por um lado, os estados descrevem genericamente a situação observada no ambiente; por outro, a listagem dos incidentes permite uma leitura de qual a situação específica ocorreu. Além disso, as transições numeradas entre os estados permitem que o usuário entenda temporalmente os eventos.

Após a visualização detalhada do cenário, a solução está completa. Com essa proposta, tanto o gerente de incidentes quanto o time de suporte tem uma visão dos eventos correlatos, separados em classes (estados) e ordenados temporalmente.

5 IMPLEMENTAÇÃO E AVALIAÇÃO

5.1 Implementação do protótipo

A proposta apresentada nesse trabalho foi implementada na forma de um protótipo, o qual denominamos SMARTIC. Por tratar-se de uma implementação prototípica, o sistema foca em demonstrar a viabilidade técnica da solução de correlação. Conforme explicado no Capítulo 4, a solução de correlação elaborada é dividida em três etapas: normalização dos incidentes, correlação e visualização. Nessa Seção explicamos como cada uma dessas etapas foi implementada.

5.1.1 Normalização dos incidentes

Essa primeira etapa da solução consiste não apenas na normalização dos dados, mas também no acesso ao banco de dados original (que armazena os incidentes na sua forma bruta), na reestruturação dos registros de incidentes e seu armazenamento em uma nova base (normalizada). A base de incidentes utilizada para a avaliação desse protótipo consiste em um banco DB2 versão V9. Utilizando uma conexão do tipo *OleDb* acessamos o banco de dados do sistema de monitoração real, lendo todos os incidentes abertos. Aplicando uma seleção sobre essa base, importamos os campos previamente definidos como importantes para a correlação para um banco de dados local, no caso, o Microsoft SQL Server 2008, versão 10.50.1600.

Após esse passo, temos em nosso banco de dados um conjunto de incidentes com número reduzido de atributos. No entanto, a descrição do incidente (campo *Details*) ainda não foi tratada. Assim, a próxima etapa consiste em analisar e extrair desse campo a classe do incidente. Visto que a descrição do incidente é gerada por um sistema de monitoração, esse campo possui uma estrutura típica, sendo analisado através de expressões regulares. Como resultado temos a descrição da situação, sua classe e prioridade; essas informações são armazenadas como atributos de cada incidente. Cada incidente, portanto, é composto pelos seguintes atributos:

- *Incident*: identificador único;
- *Details*: descrição estruturada do incidente;
- *Classification*: classe do incidente, determinada através da análise de sua descrição;
- *Reported Date*: data em que o incidente foi identificado pelo sistema de monitoração;
- *Reported Priority*: prioridade do incidente, conforme reportado pelo sistema de monitoração;

- *Asset*: código do ativo sobre o qual o incidente foi identificado;
- *Actual Resolution*: data de resolução (fechamento) do incidente.

Depois de normalizados, os incidentes são inseridos em uma lista de incidentes. Cada ativo possui a sua lista, e os incidentes são ordenados temporalmente. Como a construção dos grupos de incidentes na próxima etapa da solução depende da data de fechamento de cada incidente, entendemos ser necessária uma marca de fechamento. Considerando que a etapa de correlação utiliza a classe de um incidente como base para a construção dos grupos, criamos uma classe auxiliar que representa um estado normal do ambiente. Mais precisamente, cada vez que um incidente é fechado, inserimos na lista um incidente sintético cuja classe é *Normal*. Como esses incidente atuam como uma marca de encerramento de um outro incidente, sua data de abertura é considerada a data de fechamento do referido incidente; por outro lado, não é necessário determinar uma data de encerramento. Assim, após a normalização dos eventos temos para cada ativo uma lista ordenada temporalmente de incidentes, sendo essa lista composta tanto por incidentes reais quanto por sintéticos (classe *Normal*), conforme pode ser observado na Figura 5.1.

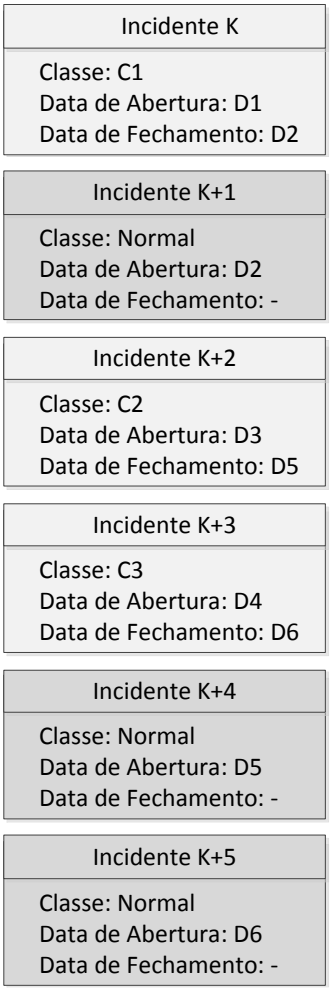

Figura 5.1: Organização da lista de incidentes, ordenados pela *Data de Abertura*.

Analisando o exemplo da Figura 5.1, o primeiro incidente aberto é o incidente *k*, da classe *C1*; sua data de abertura é *D1* e a de fechamento *D2*. Observe que o próximo

incidente é da classe *Normal*, e sua data de abertura corresponde à data de fechamento do incidente *k*. Nos instantes *D3* e *D4*, os incidentes *k+2* e *k+3* são abertos. Quando fechados, respectivamente nos instantes *D5* e *D6*, provocam a criação dos incidentes *k+4* e *k+5*, da classe *Normal*.

5.1.2 Algoritmo de correlação

Para cada ativo, a partir da listagem de incidentes ordenada temporalmente, com incidentes reais e sintéticos intercalados, criamos grupos de incidentes. Em um primeiro momento, nenhum grupo está aberto; portanto, o primeiro incidente ocasiona a abertura do primeiro grupo e instancia um contador de incidentes abertos nesse grupo (valor unitário). Para cada incidente aberto na sequência, analisamos se sua classe é *Normal* (indicando o encerramento de um incidente aberto) ou não (no caso de um novo incidente real). No segundo caso, o incidente aberto é incluido no grupo, e o contador de incidentes abertos é incrementado. Cada vez que um novo incidente do tipo *Normal* é identificado, o contador é decrementado. Quando esse contador chega a zero, entendemos que todos os incidentes abertos na janela temporal do grupo foram fechados e, portanto, também o grupo deve ser encerrado.

Considerando que podemos tratar incidentes de diversos ativos simultaneamente e que cada grupo refere-se exclusivamente a um ativo, na implementação utilizamos o conceito de dicionários para possibilitar esse tratamento. O dicionário consiste em um vetor indexado de dados, sendo que a chave utilizada em nosso algoritmo é o código do ativo do incidente, e o conteúdo de cada elemento do vetor é composto por todos os grupos de incidentes criados para este ativo. Para representar esse conjunto de grupos de incidentes, definimos um Tipo Abstrato de Dados (TAD), que possui uma lista em que cada elemento consiste em um grupo de incidentes. Esse TAD também expõe as funcionalidades de criação de um novo grupo, inserção de um elemento (incidente) no grupo aberto e encerramento de um grupo.

Figura 5.2: Organização dos eventos: dicionário, conjunto de grupos, grupos de incidentes e incidentes.

A título de ilustração e analisando a Figura 5.2, consideramos um ambiente com *p* ativos, sendo cada um desses ativos a chave para um dicionário (vetor indexado de elementos). Cada elemento deste dicionário corresponde ao conjunto de grupos de incidentes do respectivo ativo. No exemplo apresentado, selecionamos o conjunto de grupos do *i*-ésimo ativo. Este ativo contém *q* grupos de incidentes. Ainda, analisando o *j*-ésimo grupo de incidentes, vemos que este é composto por *r* incidentes. Assim, temos o entendimento completo da organização dos eventos no sistema.

Dessa forma, para acessar o incidente *k*, por exemplo, primeiramente acessamos o valor contido na posição de índice *i* do dicionário, que corresponde ao conjunto de grupos do ativo *i*. A seguir, selecionamos a *j*-ésima entrada do conjunto de grupos, que corresponde ao grupo *j*. Finalmente, dentro deste grupo, selecionamos a *k*-ésima posição, que corresponde ao incidente de interesse.

Ao longo da implementação desse protótipo, observamos casos em que incidentes temporalmente muito próximos eram alocados em grupos distintos de incidentes. Esse fato é natural, e indica apenas que um incidente foi aberto e fechado antes mesmo que a situação como um todo e os demais incidentes fossem abertos. Sob essa perspectiva, entendemos que esses incidentes, embora em diferentes grupos, também têm alto potencial de correlação. Para tratar essa questão, definimos uma variável de tolerância, que compara a data de fechamento do último incidente de um grupo com a data de abertura do primeiro incidente do grupo seguinte. Caso a diferença temporal seja menor do que a tolerância estabelecida, fazemos uma junção entre os grupos, para que os incidentes sejam tratados conjuntamente. Na Sessão 5.2 apresentamos o impacto desse ajuste na construção dos grupos de correlação.

5.1.3 Visualização

Dado que entendemos o nosso algoritmo como multidígrafos, a visualização dos resultados dos grupos de incidentes se dá através de grafos. Utilizamos a biblioteca de visualização Microsoft Automatic Graph Layout (MSAGL). A seguir apresentamos algumas características de aspectos visuais do protótipo implementado.

Na Figura 5.3 apresentamos a tela principal do protótipo do sistema de correlação de incidentes. Considerando este cenário, a primeira ação do usuário é selecionar qual o ativo a ser analisado utilizando o *dropdown* no canto superior esquerdo da tela (código 11239 no exemplo). A seguir, o conjunto de grupos de incidentes é apresentado logo abaixo na forma de uma lista, já demonstrando a quantidade de incidentes presentes naquele grupo. Após selecionar um dos grupos, este pode ser visualizado na parte central da janela, ou seja, o usuário tem a visão do estado final do grafo, em que todos os incidentes estão fechados. Além disso, na parte inferior central da tela podemos visualizar a lista de incidentes. Ainda, o usuário pode filtrar a lista incidentes selecionando apenas uma das classes. Além disso, no canto inferior esquerdo da tela apresentamos a contagem dos grupos daquele ativo e a variável de tolerância temporal entre os grupos. Na Figura 5.3, esses valores correspondem a 52 grupos e 15 minutos, respectivamente.

No contexto desse trabalho, trabalhamos sobre uma base de dados estática. No entanto, para emular como seria o sistema considerando a abertura de incidentes em tempo real, simulamos a construção dos grupos de incidentes através do uso das setas direcionais apresentadas logo abaixo da tela principal. Para ilustrar esse benefício, redesenhamos passo-a-passo o grafo da Figura 5.3 na Figura 5.4.

Conforme podemos observar na Figura 5.4, inicialmente temos apenas um incidente aberto, sendo sua classe *R3 Alerts* (Figura 5.4(a)). A seguir, dois outros incidentes dessa classe são abertos (Figura 5.4(b) e Figura 5.4(c)), criando um *loop* sobre essa classe. Observe que no instante Figura 5.21(c) temos três incidentes abertos e, portanto, duas transições de estado (que indicam a sequência de ocorrência dos eventos). Na Figura 5.4(d), um novo incidente é aberto, sendo esse da classe *Log Entries*, criando, portanto, uma transição entre diferentes estados. NasFigura 5.4(e) e Figura 5.4(f) dois novos incidentes da classe *Log Entries* são abertos, criando um *loop* sobre esse estado.

Com os recursos apresentados, conseguimos tanto visualizar o grupo de incidentes

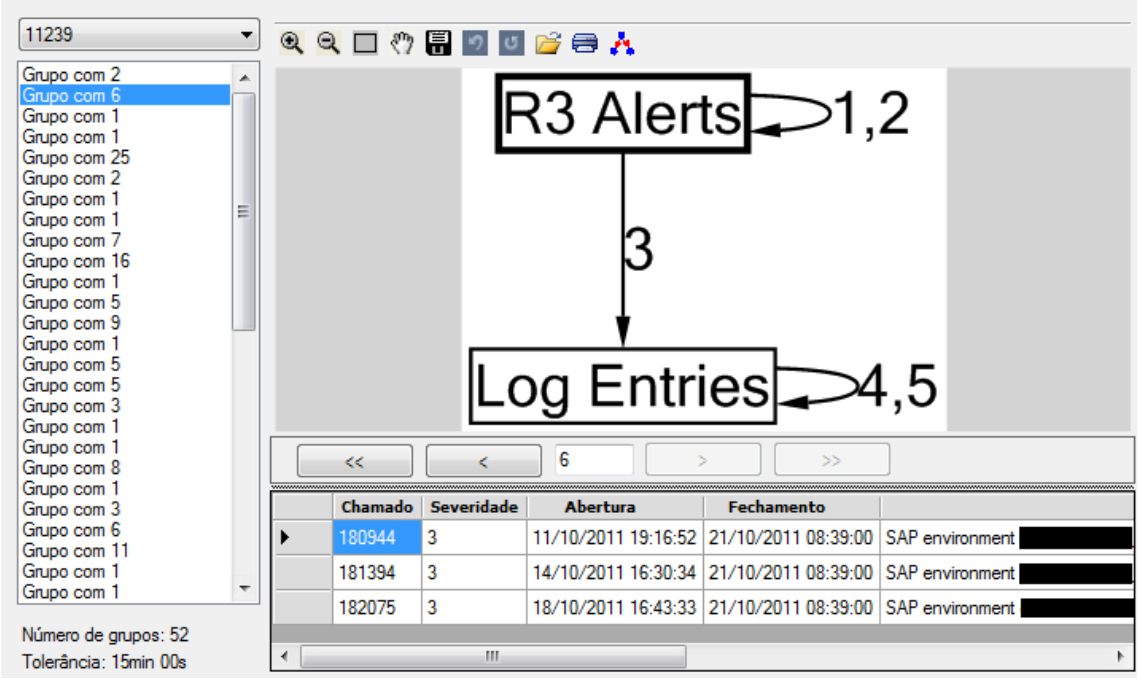

Figura 5.3: Visão geral da tela do sistema de correlações de incidentes.

correlatos quanto cada instante de sua criação. Além disso, as transições entre os nodos do grafo são numeradas, de modo que podemos observar a sequência de eventos. Acreditamos que, com funcionalidades como as recém apresentadas, a equipe de suporte passa a ter uma nova condição de avaliação e de discernimento acerca dos incidentes e suas relações. A seguir, na Seção 5.2, apresentamos alguns exemplos preliminares de que tipo de situações podem vir a ser identificadas (que seriam impossíveis ou muito difíceis de serem obtidas em um sistema tradicional de gerenciamento de incidentes).

5.2 Resultados

Como prova de conceito da solução SMARTIC, desenvolvemos um estudo sobre uma base de dados real, porém estática, composta por incidentes coletados em um ambiente corporativo real de TI. Essa base de dados é composta por cerca de 130 mil incidentes, compreendidos entre janeiro de 2008 e junho de 2012. Para reduzir o volume de dados analisados, em nosso estudo de caso analisamos cerca de 60 mil incidentes, compreendidos entre setembro de 2011 e junho de 2012. Considerando essa massa de dados, focamos nossa apresentação de resultados em três ativos, que somam aproximadamente 6% dos incidentes analisados. Informações relacionadas aos sitemas e à própria natureza dos incidentes são suprimidas neste trabalho para presevar a confidencialidade dos dados e o anonimato do cliente.

O primeiro ativo escolhido corresponde ao principal servidor de *Enterprise Resource Planning* (*ERP*) do cliente estudado. Esse ativo é fundamental para o funcionamento do cliente, uma vez que o ERP é reponsável por integrar dados e processos da corporação. O ambiente é composto por uma aplicação SAP que executa sobre um banco de dados Oracle. A arquitetura do sistema é composta por um servidor *Central Instance* (ativo selecionado para estudo), onde fica o banco de dados, e diversos *Application Servers*, que processam rotinas acessando remotamente a *Central Instance*. Considerando a complexi-

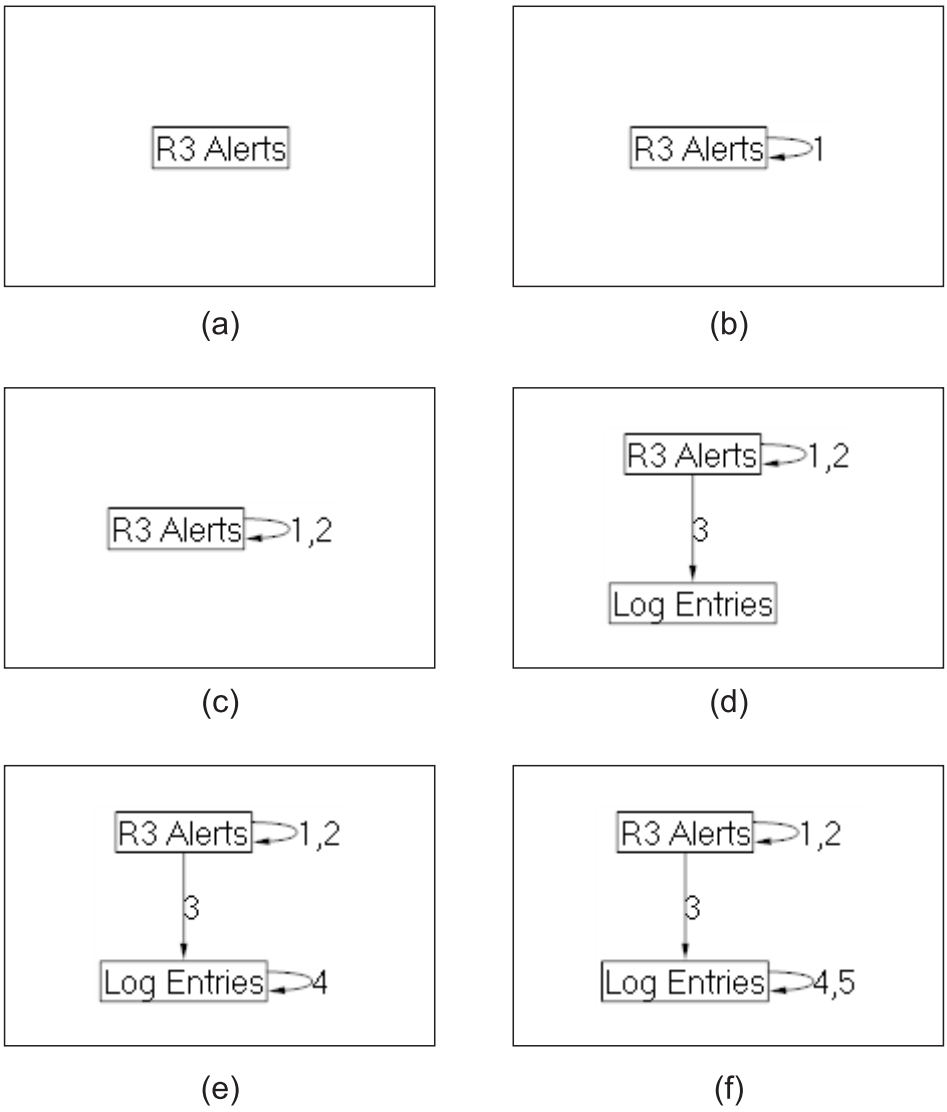

Figura 5.4: Exemplo de simulação da criação passo-a-passo de um grafo.

dade e a importância desse ambiente, entendemos que o ativo selecionado é crítico e que qualquer indisponibilidade ou degradação do serviço deve ser tratada imediatamente e, se possível, evitada.

O segundo ativo selecionado, que também tem papel importante na organização, é o ambiente SAP PI (*Process Integration*). A principal funcionalidade desse ambiente é permitir a comunicação segura e estável de componentes de software internos e externos do cliente. Assim, esse sistema está fortemente relacionado com os demais, podendo impactar gravemente o desempenho de outros ambientes. Dessa forma, entendemos que esse também é um caso crítico e sua análise merece destaque.

O terceiro ativo comporta a solução de *Business Inteligence* (BI) da corporação, tendo alta interação com outros ambientes e uma carga diária alta de processamento. A principal funcionalidade desse ambiente é processar dados do negócio e, dessa forma, auxiliar na tomada de decisões estratégicas.

Com esses três ativos, temos um conjunto diversificado de características, e esperamos encontrar grupos de incidentes correlatos bastante distintos. Dado que o volume de incidentes desses três ativos representa aproximadamente 6% da massa estudado, enten-

demos que os ativos selecionados são grandes ofensores no ambiente de TI, consequência de sua importância e do volume de dados e transações executadas. A seguir, vamos analisar alguns grupos de incidentes dos referidos ativos, observando e analisando a correlação entre eventos agrupados.

5.2.1 Ativo 1: Ambiente *Enterprise Resource Planning*

Dentre o conjunto analisado de incidentes, 3,5% referem-se a esse ambiente. Esses incidentes são divididos em 104 grupos, sendo a média de incidentes por grupo próxima a 18. A Figura 5.5 ilustra um histograma do número de incidentes em cada grupo, desconsiderando grupos de apenas um incidente.

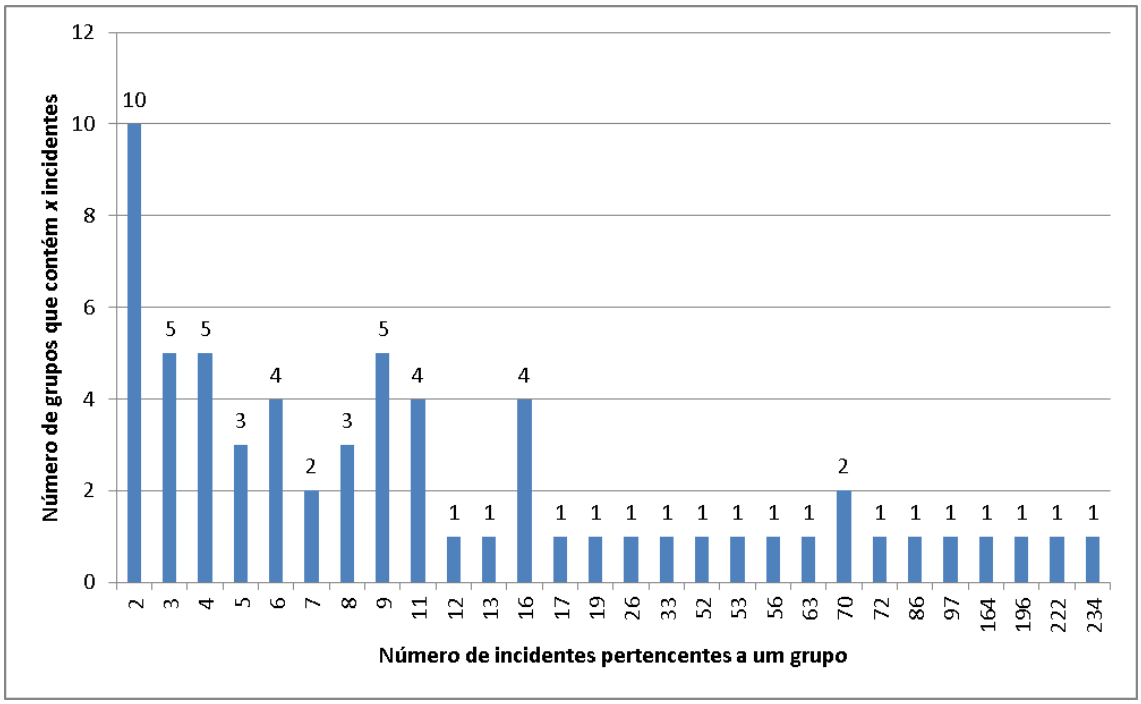

Figura 5.5: Histograma do número de incidentes em cada grupo (ambiente ERP), desconsiderando grupos formados por apenas um incidente.

Analisando o histograma da Figura 5.5, percebemos que este ativo mais de 70% dos incidentes em grupos grandes (com mais de 50 incidentes). Por outro lado, essas massa de incidentes está divida em apenas 13 dos 104 grupos. A grande maioria dos grupos possui entre 2 e 33 incidentes. Considerando o alto número de grupos com número elevado de incidentes e também a diversidade de classes, entendemos que, tipicamente, nesse ativo observaremos relações de causa e efeito.

A seguir apresentamos dois exemplos de sequências de eventos que geraram grafos diferentes, porém muito semelhantes. A Figura 5.6 representa o grafo do grupo 4, composto de 5 nodos e 7 transições (portanto, 8 eventos). A Figura 5.8, por sua vez, apresenta o grupo 9, com o mesmo número de incidentes e transições, mas apenas 4 estados. Observando os dois casos, vemos que entre os grafos a classe que difere é *Oracle Statistics*, sendo as demais iguais. Para facilitar o entendimento dos casos apresentados, a Figura 5.7 contém a lista dos incidentes apresentados na Figura 5.6. Analogamente, a Figura 5.9 contém a lista dos incidentes e apresentados na Figura 5.8.

O grupo 4, apresentado na Figura 5.6 inicia o grafo com um incidente da classe *Jobs*, que indica a falha de alguma rotina específica. O segundo incidente indica a iminência

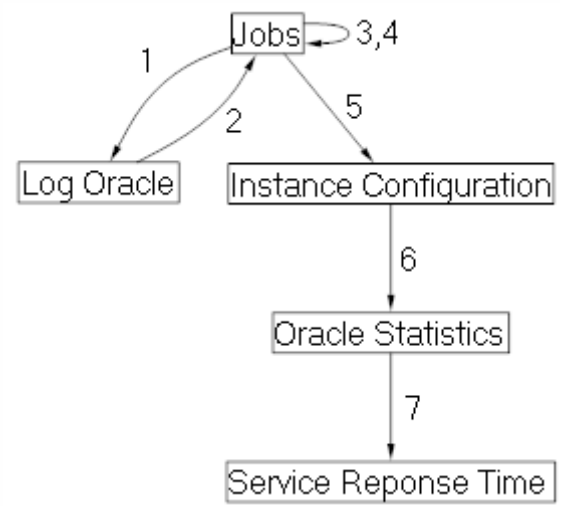

Figura 5.6: Grupo 4 de incidentes do sistema ERP, contendo 5 estados.

Descrição
Error Job: JOB in TWS name [redacted] is terminate with status Abend
DBA Ambiente iminente de parada acionar DBA - Tipo TOO_MANY_EXTENTS no [redacted], ambiente [redacted], tablespace [redacted],
Error Job: JOB in TWS name [redacted] is terminate with status Abend
Error Job: JOB for TSM in TWS name [redacted] is terminate with status Abend
Error Job: JOB in TWS name 1 is terminate with status Abend
SAP The percent of dialog process in running is 100 in the instance [redacted]
Oracle deadlocks is 82 in database [redacted]
SAP Avg response time of dialog work process is 8382 - Instance: [redacted] System Name: [redacted]

Figura 5.7: Lista de incidentes ordenados temporalmente pertencentes ao grupo 4, apresentado na Figura 5.6.

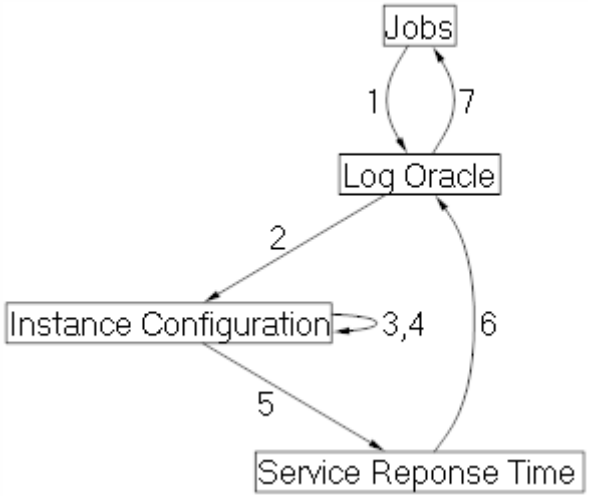

Figura 5.8: Grupo 9 de incidentes do sistema ERP, contendo 4 estados.

de parada do ambiente de banco de dados, provocando uma transição para o estado *Log Oracle*. Os 3 incidentes seguintes são da classe *Jobs*, indicando o término de mais rotinas com falhas (conforme vemos pelas transições *2*, *3* e *4* na Figura 5.6). A seguir, um incidente da classe *Instance Configuration* indica que um número excessivo de processos do tipo *dialog* do SAP estão executando (transição *5* na Figura 5.6). Em seguida, a moni-

Descrição
Error Job: JOB in TWS name [redacted] is terminate with status Abend
DBA Ambiente iminente de parada acionar DBA - Alerta CRITICO, verificar log dbck.log [redacted], ambiente [redacted]
SAP The percent of dialog process in running is 98 in the instance [redacted]
SAP The percent of dialog process in running is 98 in the instance [redacted]
SAP Instance [redacted] down in [redacted]
SAP Avg response time of dialog work process is 11683 - Instance: [redacted] System Name: [redacted]
DBA Oracle tablespace [redacted] total free space is 96.17%
Error Job: JOB in TWS name [redacted] is terminate with status Abend

Figura 5.9: Lista de incidentes ordenados temporalmente pertencentes ao grupo 9, apresentado na Figura 5.8.

toração de Oracle abre um incidente indicando alto número de *deadlocks* em uma tabela do banco, certamente ocasionada pelo alto número de processos SAP tentando acessá-la (transição *6* na Figura 5.6). Finalmente, o último incidente do grupo é aberto, indicando a degradação do serviço através do alto tempo de resposta dos processos do SAP (transição *7* na Figura 5.6). Analisando esse cenário, entendemos que o ambiente sofreu uma alta demanda, de modo que muitos processos de SAP estavam executando e fazendo *locks* no banco de dados Oracle. A demanda foi tanta que provocou um número excessivo de *deadlocks* no banco, aumentando o tempo de resposta de cada processo SAP e, inclusive, cancelando uma série de rotinas.

A possível causa para a ocorrência desses eventos poderiam ser, por exemplo, um programa executando com falha e fazendo diversas requisições do tipo *dialog* (processos de interações com telas), provocando *deadlocks* no banco de dados, cancelando rotinas e aumentando drasticamente o tempo de resposta do sistema. Nesse caso, a solução seria identificar qual o programa com falha e acionar o responsável para que sua execução seja cancelada. Nesse contexto, o grafo apresentado ajuda o time de suporte a entender o impacto no ambiente e os recursos envolvidos. Ainda, a correlação entre os incidentes agrupados é altamente provável, e consideramos que tratar esses incidentes conjuntamente traria benefícios ao processo.

Analisando o grupo 9, apresentado na Figura 5.8, vemos que os dois primeiros incidentes abertos são semelhantes aos do grupo 4, mas a sequência de eventos é distinta. Após a segunda transição, os três incidentes seguintes são da classe *Instance Configuration* (transições *2*, *3* e *4* na Figura 5.8), porém têm descrições diferentes: os dois primeiros indicam alto número de processos do tipo *dialog*, enquanto que o terceiro indica que um dos *Application Servers* não está respondendo. O próximo incidente indica que o tempo de resposta dos processos *dialog* está alto (classe *Service Response Time*). Os próximos dois incidentes são, respectivamente, das classes *Log Oracle* e *Jobs*, indicando pouco espaço livre em uma *tablespace* do Oracle (transição *6* na Figura 5.8) e mais uma rotina cancelada (transição *7* na Figura 5.8).

Comparando os dois casos, percebemos que as duas situações descritas são bastante semelhantes. A principal diferença é a ocorrência de um incidente que indica um dos *application servers* não está respondendo. Considerando o contexto apresentado, podemos inferir que esse incidente também se enquadra no cenário descrito na Figura 5.6. Embora as classes e mesmo as descrições dos incidentes sejam diferentes, o quadro observado potencialmente retrata situações semelhantes. Dessa forma, entendemos que, além da correlação entre incidentes dentro de um mesmo grupo, nosso sistema permite observar o comportamento típico de um ativo, através da comparação de diferentes grupos entre

si. No caso, observamos esses dois grafos que, embora distintos, são, de certa forma, logicamente equivalentes.

Além desses exemplos, o ambiente ERP também apresenta casos claros de reincidência, como podemos ver na Figura 5.10, que ilustra o grupo 23 de incidentes. Também nesse exemplo, a Figura 5.11 mostra a lista dos referidos incidentes.

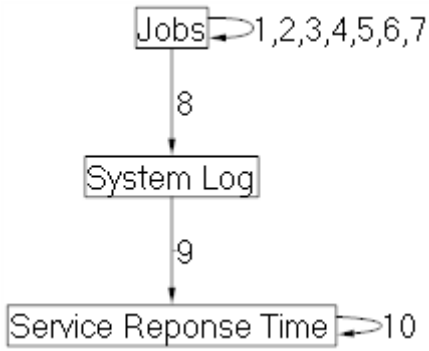

Figura 5.10: Grupo 23, indicando reincidência sobre a classe *Jobs*.

Descrição
PRODUCT = Error Job: JOB in TWS name [redacted] is terminate with status Abend - Job Command: [redacted]
PRODUCT = Error Job: JOB in TWS name [redacted] is terminate with status Abend - Job Command: [redacted]
PRODUCT = Error Job: JOB in TWS name [redacted] is terminate with status Abend - Job Command: [redacted]
PRODUCT = Error Job: JOB in TWS name 6 is terminate with status Abend - Job Command: [redacted]
PRODUCT = Error Job: JOB in TWS name [redacted] is terminate with status Abend - Job Command: [redacted]
PRODUCT = Error Job: JOB in TWS name 4 is terminate with status Abend - Job Command: [redacted]
PRODUCT = Error Job: JOB in TWS name 23 is terminate with status Abend - Job Command: [redacted]
PRODUCT = Error Job: JOB in TWS name [redacted] is terminate with status Abend - Job Command: [redacted]
BASIS = SAP System log Warning messages [redacted] - Instance: [redacted] - System Name: [redacted]
BASIS = SAP Avg response time of dialog work process is 15542 - Instance: [redacted] System Name: [redacted]
BASIS = SAP Avg response time of dialog work process is 9017 - Instance: [redacted] System Name: [redacted]

Figura 5.11: Lista de incidentes ordenados temporalmente pertencentes ao grupo 23, apresentado na Figura 5.10.

Na Figura 5.10 a reincidência ocorre no estado *Jobs*, indicando que várias rotinas falharam. Após esses incidentes, outros três são abertos em classes distintas. O incidente da classe *System Log* indica que um *warning* foi criado no ambiente; a classe *Service Response Time* registra alto tempo de espera de processos SAP. Uma possível causa para essa situação seria a dependência entre as rotinas, ocasionando uma sequência de falhas.

Embora esse exemplo também apresente traços de relação de causa e efeito entre os incidentes, a reincidência é uma característica mais marcante. Analisando o conjunto de grupos de incidentes desse ativo (excluindo grafos com apenas um estado), concluímos que cerca de 70% dos grupos evidencia fortemente relações de causa e efeito, 7% reincidências e o restante refere-se a grupos mistos, em que os dois tipos de correlação são observados. Com isso, percebemos uma característica do ativo, em que as reincidências tipicamente estão envolvidas em grupos maiores de correlação, e não isoladas. Além disso, percebemos que as classes *Jobs*, *Service Response Time*, *System Log* e *OS Performance* constumam ser as mais ofensoras (contando respectivamente com 522, 309, 273 e 228 eventos) e também as mais reincidentes.

Destacamos ainda alguns grupos em particular, que são compostos por um número muito alto de incidentes (acima de 50). Esses grandes grupos em alguns casos levam dias

até terem todos os incidentes fechados. Casos como esses, embora raros, poderiam ser beneficiados pelo nosso sistema. Uma análise mais detalhada desses casos extremos fica como trabalho futuro.

Considerando os exemplos apresentados, entendemos que este ambiente poderia ter grandes benefícios com a implementação do nosso sistema. Não apenas o esforço do time técnico e do gerente de incidentes seriam reduzidos, ao tratar incidentes conjuntamente, mas também os planos de ação para restabelecimento do ambiente poderiam ser melhor elaborados. Ainda, observamos que alguns padrões de situações são repetitivos, como apresentado nos exemplos da Figura 5.6 e da Figura 5.8. Nesses casos, com o auxílio do nosso sistema, o time poderia identificar a recorrência de não apenas um incidente, mas sim do grupo como um todo, inclusive baseando a solução da situação no caso anteriormente tratado.

5.2.2 Ativo 2: Ambiente *Process Integration*

Esse ambiente contém 221 incidentes, divididos entre 52 grupos. A Figura 5.12 mostra um histograma do número de incidentes em cada grupo, desconsiderando grupos de apenas um incidente.

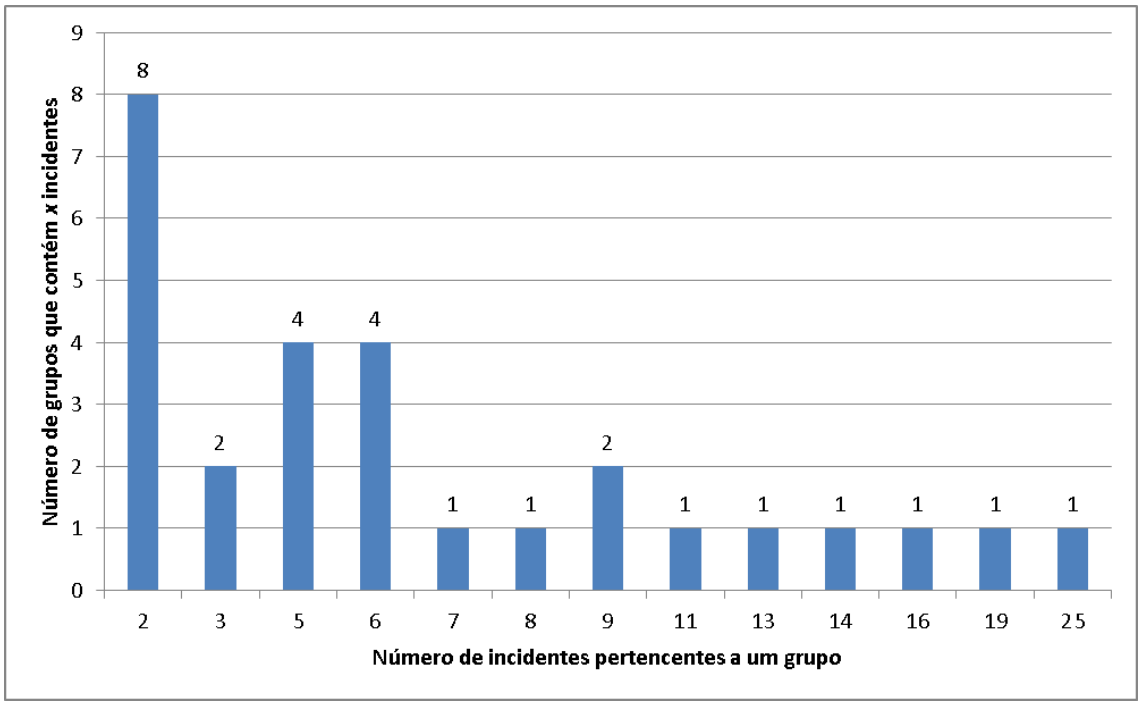

Figura 5.12: Histograma do número de incidentes em cada grupo (ambiente PI), desconsiderando grupos formados por apenas um incidente.

Diferentemente do ativo ERP, observamos que o PI possui grupos consideravelmente menores de incidentes, sendo seu maior grupo composto por 25 incidentes. Ainda, a maioria dos grupos tem entre 2 e 9 incidentes. Além disso, a maioria dos grupos de incidentes correlatos evidencia reincidência, mas não relações de causa e efeito. A Figura 5.13 ilustra um caso de reincidência sobre a classe *R3 Alerts*; a Figura 5.14 mostra a lista dos referidos incidentes.

A classe *R3 Alerts* é uma classe típica de aplicações SAP, referente a erros da aplicação e, principalmente, a *status* de filas de processos de usuário. Nesse ativo a classe mais

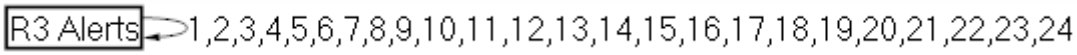

Figura 5.13: Grupo 5 de incidentes do ambiente PI, indicando forte reincidência sobre a classe *R3 Alerts*.

Descrição
SAP environment [redacted] - Inbound: Error in application - 54 IDocs for this monitoring object found but threshold value is 50
SAP environment [redacted] - Blocked queues: Client 001 - Blocked inbound queue: Client 001 Q name [redacted] status SYSFAIL dest [redacted]
SAP environment [redacted] - Blocked queues: Client 001 - Blocked inbound queue: Client 001 Q name [redacted] status SYSFAIL dest [redacted]
SAP environment [redacted] - Blocked queues: Client 001 - Blocked inbound queue: Client 001 Q name [redacted] status SYSFAIL dest [redacted]
SAP environment [redacted] - Blocked queues: Client 001 - Blocked inbound queue: Client 001 Q name [redacted] status SYSFAIL dest [redacted]
SAP environment [redacted] - Blocked queues: Client 001 - Blocked inbound queue: Client 001 Q name [redacted] status SYSFAIL dest [redacted]
SAP environment [redacted] - Blocked queues: Client 001 - Blocked inbound queue: Client 001 Q name [redacted] status SYSFAIL dest [redacted]
SAP environment [redacted] - Blocked queues: Client 001 - Blocked inbound queue: Client 001 Q name [redacted] status SYSFAIL dest [redacted]
SAP environment [redacted] - Blocked queues: Client 001 - Blocked inbound queue: Client 001 Q name [redacted] status SYSFAIL dest [redacted]
SAP environment [redacted] - Blocked queues: Client 001 - Blocked inbound queue: Client 001 Q name [redacted] status SYSFAIL dest [redacted]
SAP environment [redacted] - Blocked queues: Client 001 - Blocked inbound queue: Client 001 Q name [redacted] status SYSFAIL dest [redacted]
SAP environment [redacted] - Blocked queues: Client 001 - Blocked inbound queue: Client 001 Q name [redacted] status SYSFAIL dest [redacted]
SAP environment [redacted] - Blocked queues: Client 001 - Blocked inbound queue: Client 001 Q name [redacted] status SYSFAIL dest [redacted]
SAP environment [redacted] - Blocked queues: Client 001 - Blocked inbound queue: Client 001 Q name [redacted] status SYSFAIL dest [redacted]
SAP environment [redacted] - Blocked queues: Client 001 - Blocked inbound queue: Client 001 Q name [redacted] status SYSFAIL dest [redacted]
SAP environment [redacted] - Blocked queues: Client 001 - Blocked inbound queue: Client 001 Q name [redacted] status SYSFAIL dest [redacted]
SAP environment [redacted] - Blocked queues: Client 001 - Blocked inbound queue: Client 001 Q name [redacted] status SYSFAIL dest [redacted]
SAP environment [redacted] - Blocked queues: Client 001 - Blocked inbound queue: Client 001 Q name [redacted] status SYSFAIL dest [redacted]
SAP environment [redacted] - Blocked queues: Client 001 - Blocked inbound queue: Client 001 Q name [redacted] status SYSFAIL dest [redacted]
SAP environment [redacted] - Blocked queues: Client 001 - Blocked inbound queue: Client 001 Q name [redacted] status SYSFAIL dest [redacted]
SAP environment [redacted] - Blocked queues: Client 001 - Blocked inbound queue: Client 001 Q name [redacted] status SYSFAIL dest [redacted]
SAP environment [redacted] - Blocked queues: Client 001 - Blocked inbound queue: Client 001 Q name [redacted] status SYSFAIL dest [redacted]
SAP environment [redacted] - Blocked queues: Client 001 - Blocked inbound queue: Client 001 Q name [redacted] status SYSFAIL dest [redacted]
SAP environment [redacted] - Blocked queues: Client 001 - Blocked inbound queue: Client 001 Q name [redacted] status SYSFAIL dest [redacted]
SAP environment [redacted] - Inbound: Error in application - 25 IDocs for this monitoring object found but threshold value is 20
SAP environment [redacted] - Calls w/Execution Errors - SYSFAIL - tRFC: React to alerts

Figura 5.14: Lista de incidentes ordenados temporalmente pertencentes ao grupo 5, apresentado na Figura 5.13.

ofensora é a *R3 Alerts*, que é composta por 172 eventos. No caso, os alertas do grupo 5 indicam diversas filas de entrada bloqueadas em um mesmo SAP *client*. O período do grupo dura 5 dias, indicando que a situação demorou a ter sua causa raiz identificada e tratada e, portanto, cada vez mais incidentes foram abertos. Caso o time de suporte tivesse uma ferramenta com as características do SMARTIC, poderia observar conjuntamente todas filas bloqueadas e, possivelmente, analisar a situação de forma mais concisa.

5.2.3 Ativo 3: Ambiente *Business Inteligence*

Esse ambiente contém cerca de 1100 incidentes (cerca de 2% do volume de incidentes analisado), separados em 175 grupos. A Figura 5.15 mostra um histograma do número de incidentes em cada grupo, desconsiderando grupos de apenas um incidente.

Como podemos observar, esse ativo tem um alto volume de incidentes, mas seus grupos não são tão grandes como os do ativo do ERP. Apenas 4 grupos ultrapassam 50 incidentes, sendo a maioria dos grupos concentrados entre 2 e 10 incidentes. Observamos também a distribuição dos incidentes entre as classes, sendo as mais ofensoras *Oracle Statistics*, *Jobs* e *Oracle Log*, respectivamente com 372, 211 e 190 incidentes. Semelhante ao ativo do ambiente ERP, o BI aprensenta grupos tipicamente de relações de causa e efeito. A correlação entre as classes é evidenciada, por exemplo, através de ciclos formados entre os estados, conforme podemos ver na Figura 5.16; a lista de incidentes desse grupo é apresentada na Figura 5.17.

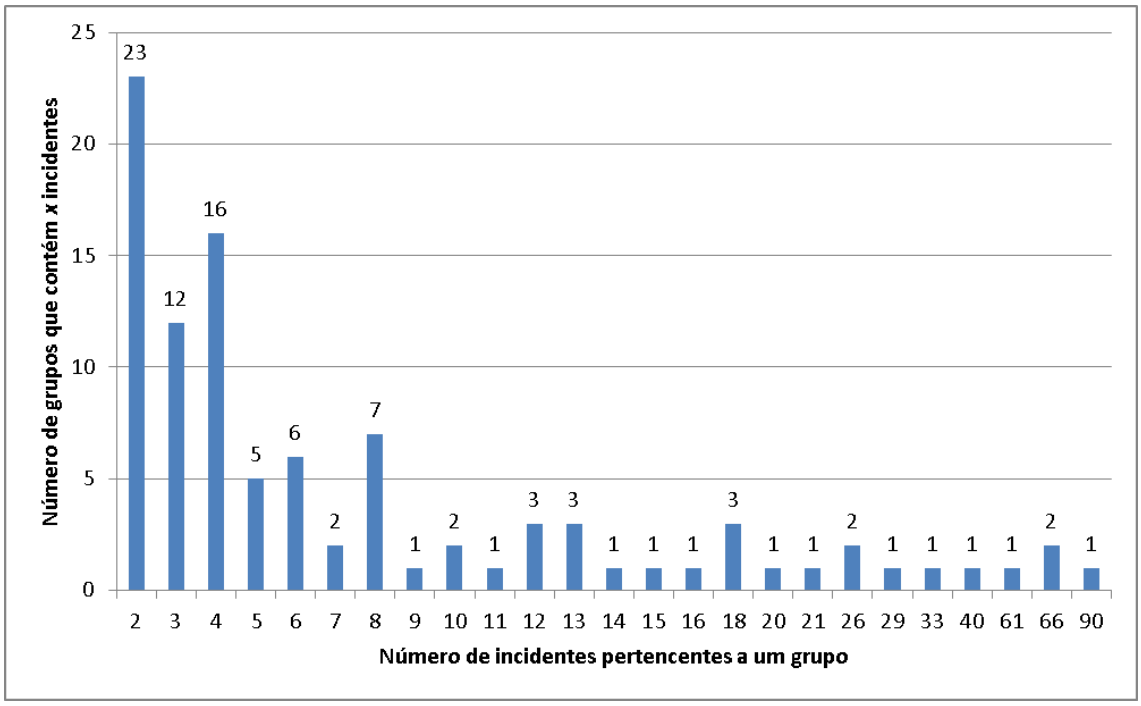

Figura 5.15: Histograma do número de incidentes em cada grupo (ambiente BI), desconsiderando grupos formados por apenas um incidente.

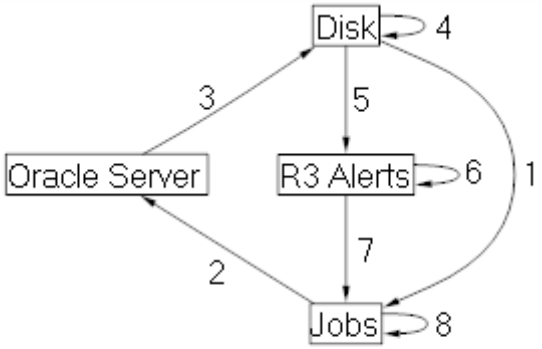

Figura 5.16: Grupo 142 de incidentes do ambiente BI. Destaca-se o laço formado pelas transições *1*, *2* e *3*.

No grupo 142 de incidentes, destacamos o ciclo criado entre as classes *Disk*, *Jobs* e *Oracle Server*, conforme observamos pelas transições *1*, *2* e *3*. Os quatro primeiros incidentes desse grupo (que formam o ciclo) tem alto potencial de correlação. O primeiro deles indica que o *filesystem* disponínel para *archive* do banco de dados está com pouca área disponível. No segundo incidente, uma rotina de *arqchive* falha (classe *Jobs*), certamente por não haver área suficiente. A seguir, um incidente da classe *Oracle Server* é aberto, indicando que o banco de dados tem pouco espaço para *archive*. O quarto incidente fecha o cilco mais uma vez alarmando alta utilização do *filesystem* de *archive*. Analisando essa sequência, vemos que os incidentes são praticamente redundantes e in-

Descrição
STORAGE = Archive filesystem /oracle/[redacted]/oraarch is 90% used
PRODUCT = Error Job: JOB in TWS name [redacted] is terminate with status Abend - Job Command: [redacted]
ORACLE = Oracle space for the archives is 91.77% in database [redacted]
STORAGE = Archive filesystem /oracle/[redacted]/oraarch is 89% used
STORAGE = Archive filesystem /oracle/[redacted]/oraarch is 71% used
BASIS = SAP Blocked outbound queue: Client [redacted] name [redacted] status EXECUTED dest [redacted] Blocked queues: Client [redacted] in [redacted]
BASIS = SAP environment [redacted] - Blocked queues: Client [redacted] - Blocked outbound queue: Client [redacted] name [redacted] status EXECUTED dest [redacted]
PRODUCT = Error Job: JOB in TWS name [redacted] is terminate with status Abend - Job Command: [redacted]
PRODUCT = Error Job: JOB in TWS name [redacted] is terminate with status Abend - Job Command: [redacted]

Figura 5.17: Lista de incidentes ordenados temporalmente pertencentes ao grupo 142, apresentado na Figura 5.16.

discutivelmente correlatos. Nesse exemplo, temos uma relação clara de causa e efeito. Porém, em uma linha de atendimento de um sistema tradicional de gerenciamento de incidentes, cada um desses alertas poderia ser tratado por times de suporte diferente, por exemplo, classe *Disk* tratada pelo time de infraestrutura, classe *Jobs* tratada pelo time de produção e classe *Oracle Server* tratada pelo time de banco de dados. O suporte de uma ferramenta como o SMARTIC permite que esses eventos sejam tratados conjuntamente.

A Figura 5.18 e a Figura 5.19 ratificam a presença de ciclos entre os estados. Ainda, observamos que nesses dois casos temos incidentes divididos entre as classes *R3 Alerts* e *Jobs*. Análogos a esse, observamos diversos casos, e entendemos que esses grafos indicam um padrão de incidência no ambiente BI.

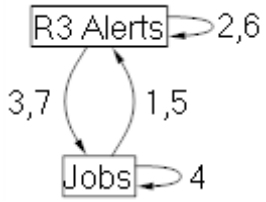

Figura 5.18: Grupo 172 de incidentes do ambiente BI, indicando a formação de ciclos entre as classes *R3 Alerts* e *Jobs*.

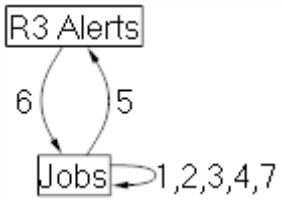

Figura 5.19: Grupo 173 de incidentes do ambiente BI, indicando a formação de ciclos entre as classes *R3 Alerts* e *Jobs*.

Conforme, explicamos anteriormente, a classe *R3 Alerts* tipicamente refere-se a *status* de filas de processos. Já a classe *Jobs* indica falhas em rotinas agendas pela equipe de produção. Considerando que as filas de processos estejam, por exemplo, bloqueadas, é natural imaginar que diversas rotinas cancelem, uma vez que a fila não suporta novas entradas. Assim, a correlação entre essas classes é altamente provável, e o ambiente BI exemplifica diversos casos em que essa situação se repete. Novamente, esse é um caso claro em que observamos o potencial de correlação entre diferentes grupos de incidentes.

5.2.4 Tolerância entre grupos de incidentes

Conforme explicado na Seção 5.1.2, observamos a existência de grupos de incidentes temporalmente próximos que são tratados separadamente. Assim, definimos uma variável de tolerência, que é comparada com a distância temporal entre os grupos. Nesse contexto, aplicamos essa variável sobre o ambiente ERP, recalculando os grupos.

Utilizando uma tolerância de 15 minutos, o número de grupos do ambiente ERP reduz de 104 para 99, indicando que cinco grupos foram aglutinados. Conforme aumentamos a tolerância, aumentamos também a flexibilidade da nossa análise. Com tolerância de 30 minutos, por exemplo, o total de grupos de incidentes é 91. A determinação de qual a tolerância ideal é um processo delicado e implica em uma análise detalhada do perfil de abertura de incidentes do ambiente. A título de exemplo, apresentamos os grupos aglutinados com tolerância de 15 minutos.

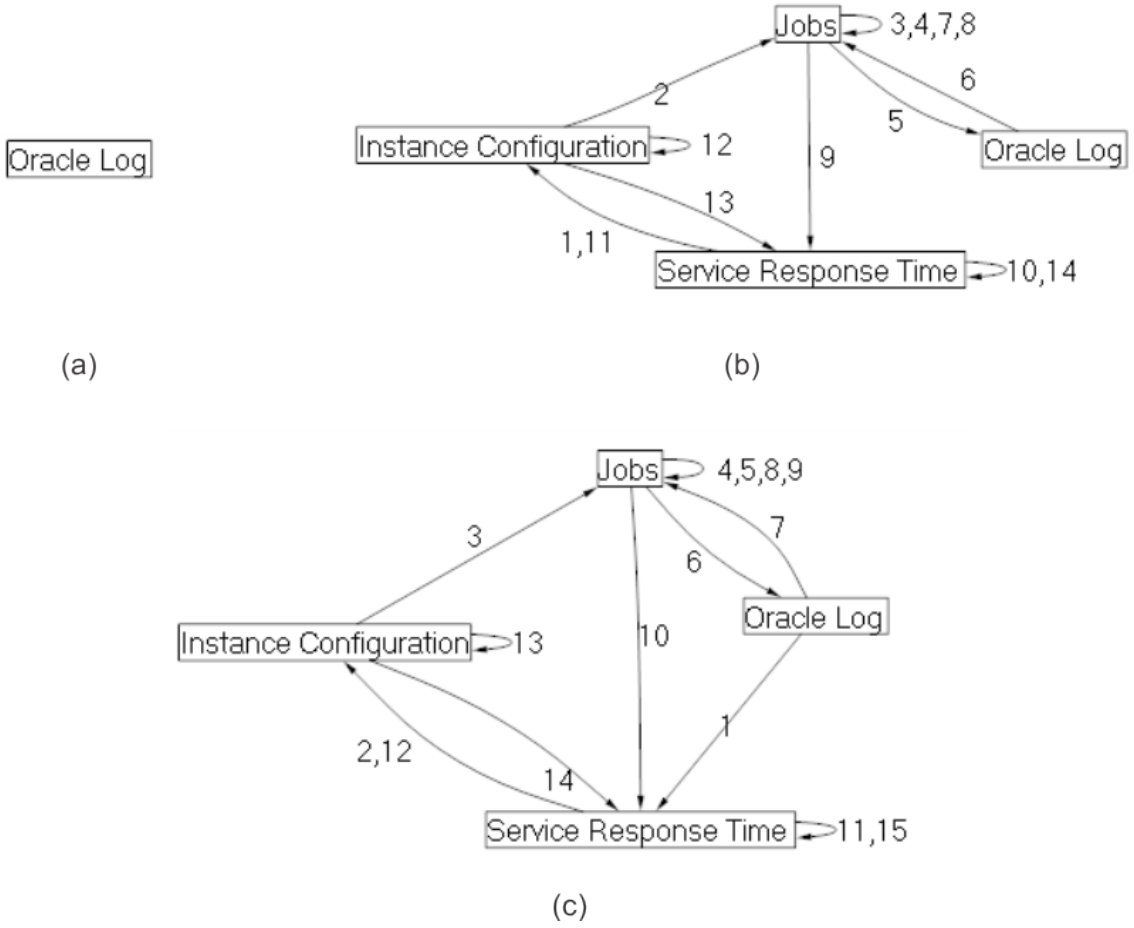

Figura 5.20: Com o uso de tolerância de 15 minutos, os grupos (a) e (b) aglutinaram-se, formando o grupo (c).

Na Figura 5.20, um grupo contendo um incidente (Figura 5.21(a)) é aglutinado a outro grupo de 15 incidentes (Figura 5.21(b)), resultando no grupo apresentado na Figura 5.21(c). Observamos que nesse caso, pouco da estrutura do grafo muda; apenas é adicionada uma transição entre os estados *Oracle Log* e *Service Response Time*, correspondentes ao arco *1* da Figura 5.21(c). Como esse incidente é classificado em um estado que já se manifestava no grupo e a proximidade temporal de sua ocorrência com os demais é pequena, entendemos que a proposta de aglutinar os grupos faz sentido, pois esse

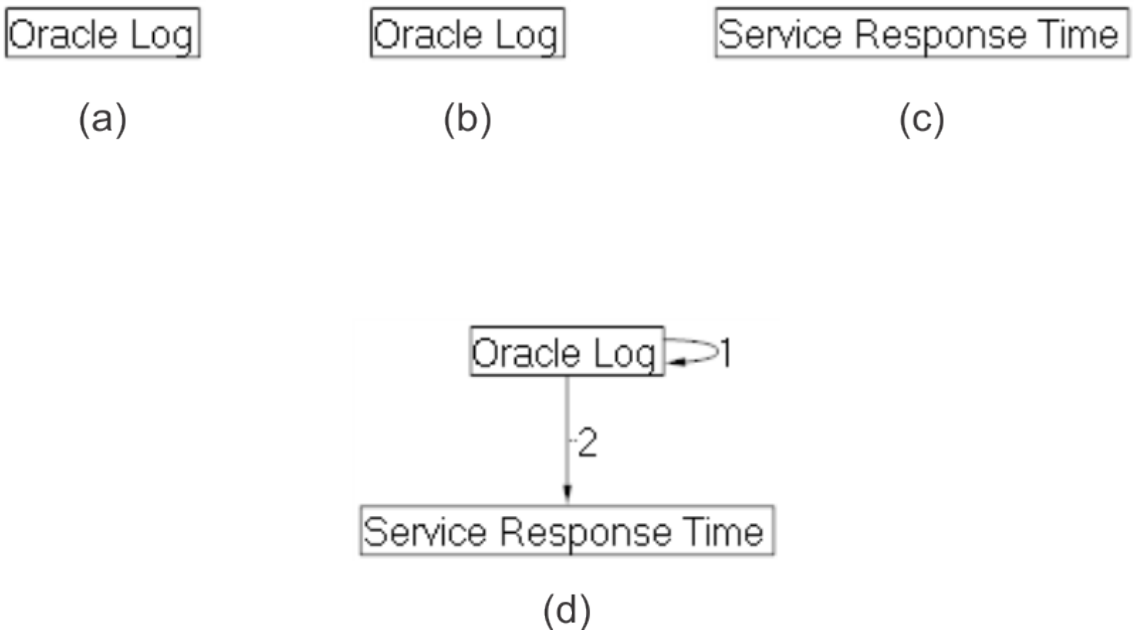

Figura 5.21: Com o uso de tolerância de 15 minutos, os grupos (a), (b) e (c) aglutinaram-se, formando o grupo (d).

incidente tem alto potencial de correlação com os demais.

Ainda, na Figura 5.21, analisamos três grupos de incidentes individuais que são aglutinados. Esse caso é interessante, pois os três incidentes, antes tratados isoladamente, agora são considerados correlatos e apresentam indícios de relações de causa e efeito. Assim, nesse caso a implementação da tolerância entre os grupos resulta em benefícios claros ao time de suporte.

6 CONCLUSÕES E TRABALHOS FUTUROS

No início desta monografia, destacamos o conceito de incidente considerando um ambiente corporativo de TI, destacando as diferenças entre incidentes reportados por um usuário e incidentes identificados através de um sistema de monitoração. No primeiro caso, os incidentes são descritos em linguagem natural e retratam a visão do próprio usuário da situação enfrentada; consequentemente, frequentemente os *tickets* são reportados com informações faltantes e pouco expressivas. Por outro lado, incidentes reportados por um sistema de monitoração seguem uma estrutura padronizada de descrição e o preenchimento dos campos tipicamente é confiável e mais completo.

Dada a complexidade de um ambiente de TI, o volume diário de incidentes facilmente ultrapassa centenas. Considerando que as atividades do time de suporte (que busca restabelecer o ambiente) e do gerente de incidentes consistem tipicamente em atividades manuais, o processo como um todo acaba sendo onerado. Inclusive, a solução e identificação da causa real dos incidentes pode ser prejudicada pela alta demanda.

Conforme ratificamos através da análise de trabalhos relacionados, os principais desafios do gerenciamento de incidentes consistem na diminuição do volume de incidentes a serem tratados e na qualidade de atendimento dos *tickets*. Os trabalhos analisados contribuem fortemente para a melhoria do processo de gerenciamento de incidentes, sendo as soluções propostas pelos autores fundamentalmente baseadas na correlação entre incidentes. Todavia, não foi escopo dos trabalhos analisados realizar correlações mais complexas, como relações de causa e efeito.

Nossa solução de correlação de incidentes busca agrupar eventos de um mesmo ativo que ocorrem dentro de uma mesma janela temporal. Ainda, previamente à inferência das correlações, normalizamos os incidentes (reduzindo o conjunto de atributos avaliados) e determinamos uma classe, baseada em sua descrição. Com esse registro normalizado de incidente, estabelecemos um algoritmo de agrupamento considerando fundamentalmente a sequência temporal dos eventos e suas classes. Para avaliar nossa proposta, implementamos a ferramenta SMARTIC e aplicamos sobre uma base de dados resultante da extração parcial de uma base real de um ambiente corporativo de TI.

Sobre essa massa de dados, avaliamos os grupos de incidentes de três ativos, considerados críticos para a continuidade do negócio do cliente. Analisando os grupos de incidentes gerados, observamos uma redução drástica no volume de dados a serem analisados. No primeiro ativo analisado (ambiente ERP), por exemplo, 1995 incidentes resultaram em 104 grupos. Considerando que a proposta da solução é que os grupos de incidentes seja tratados conjuntamente, teríamos uma redução de quase 95% de esforço de tratamento dos *tickets* propriamente ditos. Dessa forma, o tempo do time de suporte poderia ser melhor empregado em uma análise mais detalhada da situação e, inclusive, no trabalho pró-ativo buscando mitigar o risco de sua recorrência. Ainda no primeiro ativo,

observamos claramente relações de causa e efeito, destacadas pelas classes dos incidentes e as transições entre elas. No segundo ativo analisado (ambiente PI), reincidências são frequentemente observadas. Também o terceiro ativo (ambiente BI) apresenta grupos de incidentes caracterizados pelas relações de causa e efeito. Nos três casos, a análise desses incidentes individualmente provoca um esforço grande de tratamento dos registros, que é minimizado quando tratamos a situação como um todo.

Embora o foco do trabalho seja correlacionar incidentes de modo a diminuir o esforço do time de suporte e melhorar a qualidade do atendimento, os resultados analisados trazem ainda mais benefícios. A análise de grupos de incidentes de diferentes ativos permite, por exemplo, uma comparação entre os mesmos. No caso, observamos que os ativos analisados tem comportamentos típicos bastante distintos; o primeiro e o terceiro ativos possuem grupos grandes de incidentes (inclusive com centenas de registros), enquanto que o segundo ativo forma grupos menores. Dessa forma, entendemos que alguns ativos costumam apresentar situações mais complexas, compostas de um número de incidentes cuja correlação muito dificilmente seria identificada caso a análise dos eventos não fosse suportada por um sistema com características como as oferecidas pelo SMARTIC. No entanto, esses grandes grupos de incidentes correlatos são justamente as situações que desejamos preferencialmente identificar, pois eles indicam situações críticas no ambiente, em que a solução e identificação da causa raiz foram complexas e demoradas.

Outra diferença entre os servidores refere-se às classes mais ofensoras. No ativo do ambiente ERP, predominam incidentes das classes *Jobs*, *Service Response Time*, *System Log* e *OS Performance*; no ambiente PI predominam incidentes da classe *R3 Alerts*; já no ambiente BI predominam as classes *Oracle Statistics*, *Jobs* e *Oracle Log*. Dadas as características dos incidentes das classes *Jobs* e *R3 Alerts*, é natural entendermos suas reincidências. O grande número de eventos nas demais classes ofensoras, por sua vez, ilustra a particularidade de cada ambiente avaliado. Nesse sentido, entendemos que o gerente de incidentes deve orientar o time de suporte a atentar especialmente para ocorrência de eventos de algumas classes de incidentes em cada ambiente. Dessa forma, pode-se diminuir a ocorrência de reincidência sobre uma mesma classe.

Por outro lado, uma característica muito interessante do ativo do ambiente de BI refere-se à semelhança entre diferentes grupos de incidentes, conforme apresentado na Figura 5.18 e na Figura 5.19. Nesse contexto, avaliamos não apenas a estrutura do grafo, mas também o conjunto de incidentes correlatos e a situação que eles representam. Assim, não necessariamente grafos semelhantes são estruturalmente parecidos, mas sim indicam uma situação semelhante. Em nossa análise, percebemos alguns casos como esse, principalmente no ativo do ambiente BI. Dessa forma, como trabalho futuro propomos a busca por grafos e subgrafos semelhantes sobre os grupos de um mesmo ativo, e quiçá sobre conjuntos de grupos de diferentes ativos.

Ainda, conforme comentamos no Capítulo 4, assumimos um conjunto de premissas como verdadeiras. Dentre elas, consideramos que um incidente é fechado imediatamente após a sua solução, sendo o relaxamento dessa premissa também proposto para um próximo trabalho. Relativo a essa situação, em nosso protótipo implementamos uma variável de tolerância entre grupos de incidentes, aglutinando-os em caso de proximidade termporal. Nossa expectativa era de que, talvez, grupos formados por um único incidente (ou seja, sem correlação com outro evento) fossem aglutinados, indicando a correlação entre eles. No entanto, percebemos que apenas o ativo do ambiente ERP teve um número razoável de aglutinações. Porém, embora poucos grupos tenham sido tratados diferentemente, esses realmente têm alto potencial de correlação entre os eventos, conforme observamos

pela sequência de incidentes. Assim, concluímos que o uso de tolerância entre os grupos continua sendo uma estratégia válida.

Considerando as análises realizadas, acreditamos que uma análise de um conjunto maior de ativos e incidentes pode proporcionar uma visão do ambiente de TI como um todo, indicando semelhanças, diferenças e perfis típicos de cada ambiente. De toda forma, acreditamos que a implementação do sistema SMARTIC pode contribuir para o tratamento de incidentes de forma mais inteligente, considerando não registros individuais, mas o contexto em que cada registro está inserido. Logo, a causa raiz dos incidentes pode ser melhor identificada e, consequentemente, as ações para restabelecimento do ambiente e mitigação do risco de degradação são determinadas de forma mais precisa. Ainda, fica clara a diminuição do volume de *tickets* a serem tratados pelo gerenciamente de incidente, uma vez que tratam-se grupos de incidentes, e não registros individuais. Dessa forma, o esforço manual do time é reduzido, e os recursos podem focar-se na qualidade do atendimento.

REFERÊNCIAS

ANEROUSIS, N.; DIAO, Y.; HECHING, A. The cost of service quality in IT Outsourcing. In: INTEGRATED NETWORK MANAGEMENT (IM), 2011 IFIP/IEEE INTERNATIONAL SYMPOSIUM ON. **Anais...** [S.l.: s.n.], 2011. p.773 –784.

BARTOLINI, C.; SALLE, M.; TRASTOUR, D. IT service management driven by business objectives An application to incident management. In: NETWORK OPERATIONS AND MANAGEMENT SYMPOSIUM, 2006. NOMS 2006. 10TH IEEE/IFIP. **Anais...** [S.l.: s.n.], 2006. p.45 –55.

BARTOLINI, C.; STEFANELLI, C.; TORTONESI, M. SYMIAN: analysis and performance improvement of the it incident management process. **Network and Service Management, IEEE Transactions on**, [S.l.], v.7, n.3, p.132 –144, september 2010.

CASTAGNA LUNARDI, R. et al. ChangeAdvisor: a solution to support alignment of it change design with business objectives/constraints. In: BARTOLINI, C.; GASPARY, L. (Ed.). **Integrated Management of Systems, Services, Processes and People in IT**. [S.l.]: Springer Berlin / Heidelberg, 2009. p.138–151. (Lecture Notes in Computer Science, v.5841). 10.1007/978-3-642-04989-7_11.

GUPTA, R. et al. Multi-dimensional Knowledge Integration for Efficient Incident Management in a Services Cloud. In: SERVICES COMPUTING, 2009. SCC '09. IEEE INTERNATIONAL CONFERENCE ON. **Anais...** [S.l.: s.n.], 2009. p.57 –64.

GUPTA, R.; HIMA PRASAD, K.; MOHANIA, M. Information integration techniques to automate incident management. In: NETWORK OPERATIONS AND MANAGEMENT SYMPOSIUM, 2008. NOMS 2008. IEEE. **Anais...** [S.l.: s.n.], 2008. p.979 –982.

GUPTA, R.; PRASAD, K.; MOHANIA, M. Automating ITSM Incident Management Process. In: AUTONOMIC COMPUTING, 2008. ICAC '08. INTERNATIONAL CONFERENCE ON. **Anais...** [S.l.: s.n.], 2008. p.141 –150.

KANG, Y.-B. et al. A knowledge-rich similarity measure for improving IT incident resolution process. In: ACM SYMPOSIUM ON APPLIED COMPUTING, 2010., New York, NY, USA. **Proceedings...** ACM, 2010. p.1781–1788. (SAC '10).

MARCU, P. et al. Towards an optimized model of incident ticket correlation. In: INTEGRATED NETWORK MANAGEMENT, 2009. IM '09. IFIP/IEEE INTERNATIONAL SYMPOSIUM ON. **Anais...** [S.l.: s.n.], 2009. p.569 –576.

Office of Government Commerce. Best practice for Service Support. In: **Anais...** [S.l.: s.n.], 2006. (IT Infrastructure Library).

RAMAN, P.; CROSS, J. H. A Self-improving Helpdesk Service System Using Case-Based Reasoning Techniques. In: COMPUTERS IN INDUSTRY. **Anais...** [S.l.: s.n.], 1996. p.113–125.