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

TYRON WITTÉE NEETZOW SCHOLEM

WARIA: Geração de Procedimentos Livres de Interpretação para Processamento Robusto e Eficiente de Incidentes

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

Prof. Dr. Luciano Paschoal Gaspary
Orientador

Prof. Dr. Leandro Krug Wives
Co-orientador

Porto Alegre, dezembro de 2012

CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Scholem, Tyron Wittée Neetzow

WARIA: Geração de Procedimentos Livres de Interpretação para Processamento Robusto e Eficiente de Incidentes / Tyron Wittée Neetzow Scholem. – Porto Alegre, 2012.

51 f.: il.

Trabalho de Conclusão – Universidade Federal do Rio Grande do Sul. Engenharia de Computação, Porto Alegre, BR–RS, 2012. Orientador: Luciano Paschoal Gaspary; Co-orientador: Leandro Krug Wives.

1. Gerenciamento de incidentes, gerenciamento de eventos, geração automática de procedimentos, *workflow*. I. Gaspary, Luciano Paschoal. II. Wives, Leandro Krug.

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

“Veni, vidi, vici.”
— JULIUS CAESAR

AGRADECIMENTOS

Hoje, com este Trabalho de Conclusão de Curso, encerro a minha graduação em Engenharia de Computação. Nesse momento, gostaria de agradecer a todas as pessoas que, à sua maneira, contribuíram para o sucesso dos últimos anos. Em especial, gostaria de prestar um agradecimento para:

- Luciano Paschoal Gaspary e Leandro Krug Wives, meus orientadores, pelo apoio e atenção ao longo deste período de trabalho. Foram horas de discussões e reuniões produtivas que culminaram nesta qualificada composição.

- Meus professores, que agregaram na minha formação dentro e fora das salas de aula. Me ensinaram, sobretudo, que nunca devemos parar de aprender, de buscar o conhecimento. Os trabalhos, as provas, as conversas de corredor, hoje fazem parte do engenheiro que me torno.

- Meus pais, por estarem ao meu lado todo o tempo, provendo todas as condições necessárias para que eu pudesse me dedicar aos estudos, sempre respeitando os períodos "turbulentos" de provas e trabalhos. Pai, muito mais que lógica ou matemática, me ensinaste a ir atrás dos meus sonhos e que nunca é tarde para recomeçar – basta querer. Mãe, foi ao teu lado que vivi os melhores momentos, e foste a grande responsável pela formação do meu caráter. Obrigado por ter me incentivado e me acalmado quando precisei, sempre dispondo de um colo reconfortante quando o desespero batia.

- Meus avós paternos, Ruth e Norberto, por todo o apoio que sempre me deram, nos momentos em que mais precisei. *Oma*, obrigado pela tua dedicação em proporcionar um momento de alegria e descanso em meio à correria do dia-a-dia. *Opa*, tua desenvoltura com aparelhos eletrônicos deixa muito jovem com inveja; obrigado por me mostrar que sempre podemos aprender mais.

- Meus avós maternos, Yára e Carlos. Vô, meu companheiro de sempre, companheiro da vida; se cheguei aqui hoje, podes te sentir orgulhoso, pois foi com a tua ajuda, com as tuas caronas, e principalmente, com o teu exemplo. Vó, foste tu quem plantou a semente do conhecimento e da educação em mim. Tens uma parcela importantíssima nesta caminhada até aqui. Quando, nos meus 3 anos, me iniciaste na trajetória dos livros e do estudo, a conquista de hoje ainda estava muito distante. Etapa após etapa, sempre me incestivaste a seguir em frente: vencemos juntos o ensino fundamental, o ensino médio, o vestibular e, agora, a graduação. Obrigado pela demonstração de força e pelo exemplo que és.

- Cassiana Fülber, a quem conheci no primeiro dia desta graduação. Desde lá, nossos inúmeros trabalhos, madrugadas de estudo, foram maravilhosamente enlouquecedores. Vivenciamos tudo isso juntos e sempre foste meu forte. Em ti, era certo que eu encontraria forças para buscar o próximo desafio. Obrigado pelas palavras de carinho e por caminhar ao meu lado nesta e em outras jornadas!
- Esta instituição de ensino, Universidade Federal do Rio Grande do Sul, em especial o Instituto de Informática e a Escola de Engenharia, que possibilitaram esses anos de aprendizagem em excelentes condições.

SUMÁRIO

LISTA DE ABREVIATURAS E SIGLAS

TI	Tecnologia da Informação
IT	*Information Technology*
ITIL	*Information Technology Infrastructure Library*
CPD	Centro de Processamento de Dados
IC	Item de Configuração
CPU	*Central Processing Unit*
UT	Unidade de Tempo
SME	*Subject Matter Expert*
SLA	*Service Level Agreement*
CMDB	*Configuration Management Database*
SYMIAN	*SYMulation for Incident ANalysis*
RFC	*Request for Change*
WARIA	*Workflow Automated Refinement for Incident Analysis*
HTML	*HyperText Markup Language*

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ALGORITMOS

RESUMO

A Biblioteca de Infraestrutura de Tecnologia da Informação tornou-se, nos últimos anos, referência para o gerenciamento de infraestruturas de TI. Entre seus vários processos, destaca-se o gerenciamento de incidentes que, entre outros, é responsável pelo pronto processamento de situações diferentes do esperado ocorridas no ambiente de TI. Nesse sentido, o principal objetivo do gerenciamento de incidentes é reestabelecer o ambiente no menor tempo possível e com mínimo impacto. Um dos mecanismos sugeridos pela biblioteca é a existência de uma equipe responsável por prestar o atendimento inicial aos incidentes, executando procedimentos documentados na tentativa de solucionar a falha. O alto volume de incidentes diários demanda dos técnicos desta equipe uma rápida atuação, apesar desta atividade ser trabalhosa e basicamente manual, portanto, propensa a erros. Ao mesmo tempo, as documentações desses procedimentos seguidamente são complexas, com muitas ramificações em virtude dos diferentes cenários das falhas. Com o intuito de tornar o atendimento de incidentes mais ágil e tolerante a erros, este trabalho propõe a formalização desses procedimentos em *workflows*; além disso, é proposto o refino desses *workflows*, tornando-os livres de decisões para execução pelo técnico de suporte. Dessa forma, é gerada uma lista de ações específica para cada incidente, ratificando que as ações executadas serão adequadas a cada situação. Com isso, pretende-se reduzir as implicações causadas por falhas humanas e reduzir o tempo de atendimento, fornecendo informações claras e precisas ao técnico. Para avaliar a viabilidade e o impacto da solução, desenvolveu-se uma implementação prototípica de um sistema de gerenciamento de incidentes. Esta foi utilizada na condução de experimentos sobre uma base de incidentes de um ambiente real de TI, e os resultados obtidos foram avaliados de maneira qualitativa e quantitativa.

Palavras-chave: Gerenciamento de incidentes, gerenciamento de eventos, geração automática de procedimentos, *workflow*.

ABSTRACT

The Information Technology Infrastructure Library (ITIL) has become, over the past few years, the *de facto* standard for Information Technology (IT) infrastructure management. Among its diversified processes, incident management is responsible for the prompt handling of IT incidents, with the main objective of reducing downtime and impact of failing services. One of ITIL's best-practices mechanisms is the existence of a team responsible for providing the first support as soon as an incident is reported. This is achieved by technicians who execute documented procedures. Nevertheless, the high amount of daily incidents on typical IT environments, added to the fact that this activity is generally cumbersome and manual, leads to an error-prone situation. Documented procedures are often complex, primarily due to several branches on the flow, caused by the high variability of the situations. In order to provide a faster and more fault-tolerant support, this work proposes to a solution to formalize the procedures in workflows; besides, the refinement of these workflows is proposed, turning them into decision-free flows for execution by the support technician. Thus, the generated action list is specific for each incident, rectifying that the actions executed will be adequated to the problem. Aiming to evaluate the concept of the solution, a prototype of an incident management system was implemented and a set of experiments was conducted against a real IT environment. The results were analyzed in a quantitative and qualitaty aspect.

Keywords: incident management, event management, automated procedures generation, workflow.

1 INTRODUÇÃO

Atualmente, a Tecnologia da Informação (TI) desempenha um papel fundamental no suporte às operações de negócios, estando intrinsecamente ligada com a administração empresarial (COSTA CORDEIRO et al., 2009). No cenário corporativo, a TI não está mais limitada ao controle fabril; pelo contrário, ela está presente nos mais diversos setores, inclusive em nível executivo, auxiliando na tomada de decisões e definição de estratégias. Considerando esse cenário, a infraestrutura de TI em si representa um conjunto de ativos (*assets*) vitais para a continuidade dos negócios da empresa. O aumento contínuo na complexidade do gerenciamento desse ambiente, intensificado pela criticidade do mesmo, estabelece um novo desafio de gestão.

Nesse contexto, a Biblioteca de Infraestrutura de TI (*Information Technology Infrastructure Library*, ITIL) representa um conjunto de melhores práticas para gerenciamento de infraestruturas de Tecnologia da Informação (OFFICE OF GOVERNMENT COMMERCE, 2006). Dentre as diversas áreas abordadas, destaca-se o gerenciamento de incidentes (*Incident Management*). Segundo ITIL, um incidente consiste em um evento que não faz parte da operação normal de um serviço, podendo ou não impactar em sua disponibilidade. Basicamente, incidentes podem ser registrados de duas formas: reportados por um usuário ou identificado por um sistema de monitoração da infraestrutura (MARCU et al., 2009). Baseado nas melhores práticas, o gerenciamento de incidentes busca reestabelecer o ambiente de TI no menor tempo possível e com impacto mínimo. Considerando o alto número de ativos envolvidos nessa infraestrutura, o número de falhas, iminência de falhas ou alertas facilmente ultrapassa centenas de registros diários (BARTOLINI; DAY, 2010).

Tipicamente, cada incidente é tratado por um grupo técnico especialista na tecnologia relacionada à causa do incidente. Nesse sentido, um problema recorrente do gerenciamento de incidentes é a falta ou a descentralização de informações para o correto encaminhamento de incidentes entre os grupos de atendimento. O roteamento inadequado desses incidentes normalmente é ocasionado por falha humana, e dele decorre perda de tempo de atuação na situação (BARTOLINI; DAY, 2010). A ineficácia dos procedimentos e a sobrecarga de trabalho do técnico que faz o encaminhamento dos incidentes provocam o mau roteamento, que por sua vez atrasa o processo de tratamento de falhas. Claramente esse cenário apresenta fatores custosos de gestão, como sobrecarga do técnico, aumento do tempo de reestabelecimento do ambiente e aumento do número de incidentes.

Automatizar e padronizar o encaminhamento de incidentes tem como principal vantagem garantir que a equipe responsável por tratar a falha seja adequada à situação. Dessa forma, é possível reestabelecer o ambiente no menor tempo possível, evitando novos alertas e até a evolução da situação. A ITIL sugere diferentes níveis de atendimento para as equipes de suporte, sendo que o primeiro nível é responsável pela análise inicial do

incidente e execução de procedimentos documentados. Assim sendo, outra vantagem de automatizar o processo é diminuir a carga do primeiro nível de atendimento, possibilitando uma melhoria na qualidade do serviço.

Com o intuito de contribuir com a resolução do problema recém mencionado, esse trabalho tem como principal objetivo propor uma solução que permita avaliar cada incidente e, assim, determinar o encaminhamento correto de cada registro. Para avaliar a proposta, foi elaborado um protótipo da solução, testado sobre um banco de incidentes de uma infraestrutura real de TI. Nesse estudo, observou-se resultados tanto qualitativos quando quantitativos. Considerando que o volume de incidentes gerados por sistemas de monitoração é maior do que os incidentes reportados por usuários – uma vez que se considera que qualquer falha reportada por um usuário também foi diagnosticada pelo sistema automatizado –, neste estudo considerou-se somente incidentes reportados por ferramentas de monitoração.

O restante do trabalho está organizado da seguinte forma: no Capítulo 2, os fundamentos de gerenciamento de ambientes corporativos de TI são revisados, assim como conceitos relativos ao gerenciamento e atendimento de incidentes. Ao final do capítulo, são estudados trabalhos relacionados com o tema e sua relação com o presente trabalho. No Capítulo 3 são apresentadas a arquitetura e os componentes que integram a solução proposta. No Capítulo 4, a solução é estudada através da implementação de um protótipo. No Capítulo 5, a proposta é avaliada, e apresenta-se o protótipo do sistema de gerenciamento de incidentes desenvolvido, bem como as avaliações qualitativas e quantitativas. Por fim, no Capítulo 6, apresenta-se as considerações finais e perspectivas para trabalhos futuros.

2 FUNDAMENTOS E ESTADO DA ARTE

Este capítulo tem como objetivo revisar os principais conceitos utilizados em ambientes de TI. O leitor mais familiarizado com gerenciamento de infraestruturas de TI pode continuar sua leitura a partir da Seção 2.5, em que são analisados trabalhos relacionados.

2.1 Ambientes corporativos de TI

No cenário corporativo atual, o uso da TI como auxílio na gestão não é mais opção, mas regra entre companhias de diferentes portes (COSTA CORDEIRO et al., 2009). A complexidade das transações empresariais, aliada à facilidade oferecida pela informatização dos processos, fez com que as empresas buscassem a TI como suporte à gestão e tomadas de decisões. Nesse contexto, não apenas a infraestrutura de TI é importante, mas a sua própria gestão.

Uma infraestrutura de TI é formada por uma série de ativos. Em termos de infraestrutura física, o ambiente consiste em um Centro de Processamento de Dados (CPD) ou *datacenter*, composto tipicamente de servidores, estruturas de armazenamento de dados e infraestrutura de rede. Usufruindo dessa estrutura como um todo, encontra-se o nível de aplicação, composto por bancos de dados, aplicações de gestão e controle fabril, sistemas de monitoração, entre outros. Tanto os itens de *hardware* quanto os itens de *software* são conhecidos por itens de configuração (IC, *Configuration Item*), sendo considerados ativos fundamentais para a continuidade do negócio.

Dada a complexidade deste tipo ambiente, o grande número de ativos envolvidos e a importância destes para os propósitos da companhia, torna-se evidente a necessidade de monitorar o ambiente para que, em caso de falha, esse possa ser rapidamente reestabelecido. É neste contexto que surgem os sistemas de monitoração, tipicamente compostos por agentes e gerentes; os agentes são instalados nos dispositivos-alvo e possuem funções específicas para monitorar uma ou mais de suas propriedades. Em contrapartida, os gerentes são componentes para os quais os agentes reportam os dados coletados, centralizando as informações operacionais do ambiente.

2.2 Metodologia ITIL no suporte à gestão de TI

Considerando o aumento de complexidade de ambientes corporativos de TI, tornou-se necessária a criação de processos que orientassem sua gestão, garantindo um bom funcionamento da estrutura. A necessidade de padronização e unificação dos processos surgiu como uma iniciativa do governo do Reino Unido, na década de 80, da qual resultou a primeira versão da ITIL. A ITIL consiste em uma abordagem sistemática para a entrega de

serviços de TI de qualidade (OFFICE OF GOVERNMENT COMMERCE, 2006), sendo composta por um conjunto de melhores práticas.

Dentre os processos definidos pela ITIL, o gerenciamento de eventos (*Event Management*) e o gerenciamento de incidentes (*Incident Management*) destacam-se pela importância no que tange à estabilidade do ambiente. Segundo a terminologia apresentada na biblioteca, um evento pode ser definido como uma mudança de estado de um componente da infraestrutura (*hardware* ou *software*, que tem importância para o gerenciamento de um serviço de TI ou um item de configuração (OFFICE OF GOVERNMENT COMMERCE, 2006). Também são chamados de eventos os alertas gerados pelos serviços de TI ou sistemas de monitoração, e podem ser classificados como:

- Eventos que indicam uma operação normal - podem ser gerados para mostrar que uma ação esperada foi executada;
- Eventos que indicam um funcionamento anormal - notificam a ocorrência de uma situação adversa no ambiente e que requer ação para sua correção;
- Eventos que sinalizam uma operação incomum, mas não excepcional - notificam que algum subsistema requer atenção, mas que o cenário ainda não causa impacto.

Um incidente, por sua vez, é definido como "uma interrupção não planejada de um serviço de TI ou uma redução de qualidade de serviço" (VAN BONI, 2012). Um incidente pode ser reportado por um usuário ou identificado através de um sistema de monitoração da infraestrutura. Independente da forma de registro do incidente, o processo de gerenciamento de incidentes tem como objetivo reestabelecer o ambiente no menor tempo possível e com mínimo impacto.

2.3 Ciclo de vida de um evento e de um incidente

Comparando esses dois conceitos, é possivel perceber que eventos e incidentes estão intimamente relacionados. No ciclo de vida de um serviço, cada ativo vive alternâncias de estados de funcionamento, ou seja, diversos eventos compõem seu histórico. Eventos que indicam um funcionamento anormal e eventos que sinalizam uma operação incomum podem gerar incidentes, sendo que esses são a forma de alertar o suporte técnico para a iminência ou ocorrência de uma falha.

Eventos ocorrem a todo o momento; qualquer alteração no ambiente pode ser considerada um evento. No entanto, quando um evento torna-se repetitivo, ou quando os valores de ocorrência ultrapassam um limite aceitável (*threshold*), então esse evento tem potencial para ser considerado um incidente (MARCU et al., 2009).

Na Figura 2.1, por exemplo, é exibida a evolução do uso de CPU (*Central Processing Unit*) em um servidor com o passar do tempo. Observa-se que existe um limite aceitável de utilização de CPU, definido em 85%. Após transcorridas 5 unidades de tempo (ut), a taxa de utilização foi superior ao limite definido, retornando aos níveis aceitáveis somente na 9ut. Neste caso, diz-se que o ambiente ficou acima da capacidade por 4ut. Normalmente é definido um número de amostragens de eventos nestas circunstâncias para disparar a abertura de um incidente. Supondo um número de amostragens igual a 3, neste exemplo, seria observada a abertura de um incidente reportando alto uso de CPU para o ativo em questão.

No momento em que um evento passa a ser considerado um incidente, significa que há uma interrupção não planejada ou uma redução de qualidade em um dos serviços. Logo,

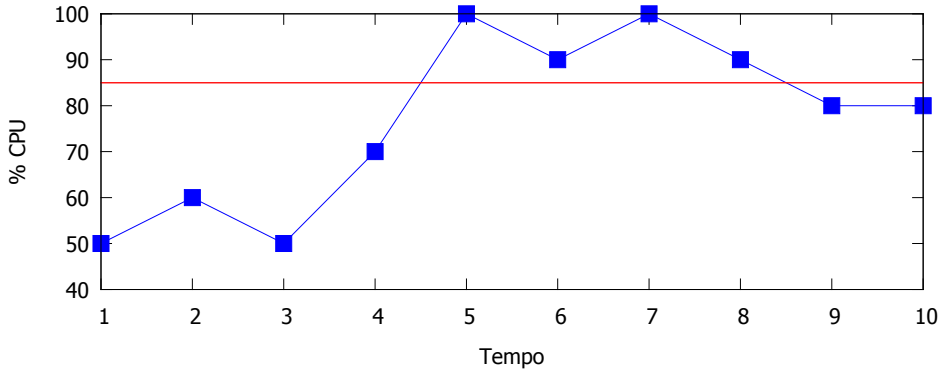

Figura 2.1: Exemplo de eventos com limite definido para uso de CPU.

há necessidade de intervenção para que o serviço seja reestabelecido. Tipicamente, o sistema de monitoração gera um registro com informações importantes sobre o incidente e encaminha para um analista técnico. Algumas das informações importantes de um incidente são data e horário de ocorrência, o ativo afetado e a situação observada. A partir dessas informações, a equipe técnica inicia o atendimento do incidente.

A metodologia ITIL sugere que o suporte técnico seja dividido em três linhas de atendimento, como apresentado na Figura 2.2. Os técnicos que compõem a primeira linha de atendimento registram e classificam os incidentes. Além disso, o time que realiza esse primeiro atendimento procura dar uma solução imediata para a situação, executando verificações e procedimentos documentados (na tentativa de reestabelecer o ambiente). Caso não consiga resolver o incidente, o encaminha para o segundo nível. No segundo nível, o incidente é analisado por um especialista no assunto (SME, *Subject Matter Expert*). Novamente, é buscada uma forma de solucionar a falha e, em caso de insucesso, um novo encaminhamento é realizado. 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, portanto, solucionar o incidente. Os fornecedores são consultados, pois, teoricamente, são os que possuem melhores condições de identificar a causa da falha, uma vez que conhecem com detalhes as tecnologias utilizadas nos produtos relacionados ao incidente. Se ainda assim não for possível determinar a solução para o incidente, pode ser invocado o processo de gerenciamento de problemas (*Problem Management*), definido também pela ITIL, ou, ainda, pode ser decidido que o incidente não tem solução.

Claramente, o processo de tratamento dos incidentes tradicionalmente é altamente dependente de pessoas. Essa característica do modelo de atendimento descrito faz com que ele seja suscetível a falhas e dependente de critérios abstratos, como a atenção e a interpretação do técnico. Por exemplo, se um técnico da primeira linha desconhece o procedimento correto a ser seguido em uma determinada situação, ou falha na sua execução, todo o processo fica comprometido. Nesse caso, o incidente é passado para o segundo nível, aumentando não apenas seu tempo de solução, mas também o seu custo. Além disso, durante esse período, o ambiente pode desestabilizar ainda mais, provocando a instanciação de novos incidentes.

Ainda, procedimentos dependentes de interpretação ou mal estruturados podem até ocasionar novas falhas, inclusive mascarando a causa raiz da situação inicialmente identificada. O modelo de atendimento como um todo é custoso, principalmente no caso de incidentes tratados pelas segunda e terceira linhas de atendimento. Logo, fica clara a necessidade de um gerenciamento de incidentes, buscando melhor qualidade de atendimento

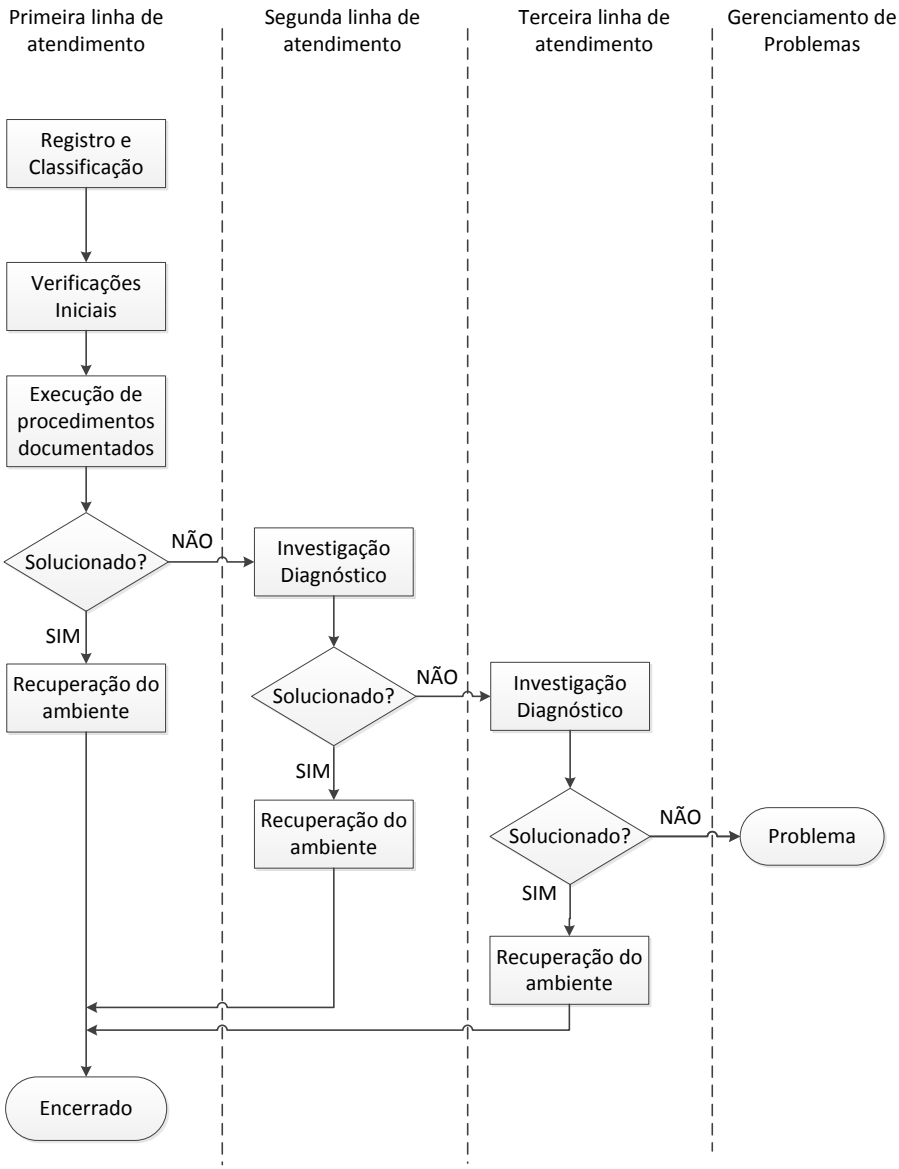

Figura 2.2: Representação das linhas de atendimento conforme metodologia ITIL.

e menor custo.

2.4 Gerenciamento de Incidentes

O objetivo principal do gerenciamento de incidentes é reestabelecer o ambiente no menor tempo possível e com mínimo impacto, garantido assim que níveis de serviço acordados (SLA, *Service Level Agreement*) sejam respeitados. Para atingir esse objetivo, uma das principais estratégias é que a primeira linha de atendimento solucione a grande maioria dos incidentes. Dessa forma, o custo e o tempo de atendimento são menores (BARTOLINI; STEFANELLI; TORTONESI, 2009). Uma segunda alternativa para garantir o bom desempenho do gerenciamento de incidentes é a sua prevenção. Para isso, é necessário entender a causa raiz dos incidentes mais ofensores e, então, atacá-la. Logo, é fundamental que o tratamento dos incidentes seja qualificado, e que a causa raiz seja combatida. Para o sucesso de ambas as estratégias, é importante documentar o maior número de procedimentos possível. No entanto, essa alternativa implica em três grandes desafios.

O primeiro desafio consiste na criação dos procedimentos. A complexidade e diversidade do ambiente influenciam diretamente na dificuldade de confecção dos roteiros de atividades. Além disso, cada técnico de suporte possui conhecimentos e experiências distintas, mas o procedimento precisa ser acessível a todos. Pode-se dizer que cada procedimento é, na realidade, um *workflow*, ou seja, uma sequência de atividades orientadas, podendo ou não conter condicionais. Nesse sentido, costuma-se dizer que cada procedimento é um documento vivo, de modo que sua manutenção e atualização podem ser necessárias, na medida em que os processos de atendimento são modificados.

O segundo desafio consiste na organização e armazenamento desses procedimentos. Considerando que todos os procedimentos devem estar documentados e sua variabilidade é inevitável, é fundamental que exista uma forma estruturada de armazenamento. Uma alternativa, por exemplo, é utilizar uma estrutura em árvore, aninhando procedimentos semelhantes, ou referentes a um mesmo ativo, ou a uma mesma categoria. De toda forma, o volume de procedimentos a serem pesquisados torna difícil a pesquisa manual, e onera o processo como um todo.

Por último, o volume diário de incidentes facilmente ultrapassa centenas, de modo que o time de suporte de primeiro nível possui uma carga altíssima de atividades. Nesse cenário, os técnicos são sobrecarregados e, consequentemente, ficam mais suscetíveis a cometer falhas. Além disso, a pesquisa constante pelo procedimento a ser executado provoca perda de tempo e reduz a produtividade do técnico.

Assim, fica claro que existe uma grande deficiência no processo de gerenciamento de incidentes, principalmente no sentido de reduzir a carga de incidentes tratados e melhorar procedimentos documentados. Observando esse cenário, é patente a oportunidade de melhorar a qualidade do processo através do tratamento automatizado de incidentes, considerando o uso de *workflows*.

2.5 Trabalhos Relacionados

No contexto de processos de TI, o gerenciamento de incidentes tem sido foco de diversos trabalhos científicos. A ITIL pode ser destacada como um dos trabalhos precursores nessa linha de estudos, buscando formalizar as melhores práticas de gerenciamento de infraestrutura de TI. Nesta seção, são destacados alguns dos importantes estudos que se

relacionam com este trabalho.

No trabalho desenvolvido por Gupta *et al.* (GUPTA; PRASAD; MOHANIA, 2008), os autores destacam as diferenças entre incidentes reportados por um usuário e gerados automaticamente através de um sistema de monitoração. No primeiro caso, a qualidade de descrição do incidente depende da interpretação e do entendimento do usuário em relação à situação apresentada, ou seja, o incidente contém apenas a percepção deste usuário. Por outro lado, um incidente gerado por um sistema de monitoração tem as informações mínimas para sua compreensão já preenchidas. Nesse sentido, incidentes do primeiro tipo podem vir a demandar muito mais esforço de interpretação e investigação do técnico de suporte. Pensando em melhorar esse processo, os autores propõem um sistema de correlação entre incidentes reportados por usuários.

Fundamentalmente, a relação entre diferentes incidentes se dá através da comparação das descrições e do IC referenciado. Inicialmente, o incidente passa por um analisador que extrai palavras-chave baseado em um dicionário de palavras típicas de um CMDB (*Configuration Management Database*). Identificados esses termos, outro componente da solução busca incidentes correlatos através de regras de similaridade. Um ponto interessante dessa busca é que o processador considera também regras de dependência entre diferentes IC, ou seja, incidentes que referenciam diferentes IC ainda assim podem ser correlatos, considerando que seus referidos IC tenham uma relação de dependência.

Ainda, Gupta *et al.* dissertam sobre as três linhas de atendimento básicas sugeridas pela ITIL, destacando que o custo do processo de gerenciamento de incidentes aumenta conforme aumenta a complexidade do atendimento. Dessa forma, é interessante para o processo como um todo concentrar o maior número possível de incidentes na primeira linha de atendimento. Para isso, é necessário que os incidentes tenham sua solução documentada de forma clara e que o técnico possa facilmente identificar qual a solução que deve ser aplicada. Os autores defendem que incidentes semelhantes normalmente têm soluções semelhantes. Dessa forma, estabelecendo a semelhança entre dois incidentes, muito provavelmente suas soluções serão parecidas, ou seja, o técnico pode utilizar um incidente como base para atender outro. Com isso, é estabelecida uma espécie de Base de Conhecimento (*Knowledge Base*).

Nesse sentido, entende-se que o trabalho promove a correlação entre incidentes na expectativa de que suas soluções sejam semelhantes. Essa estratégia assemelha-se à proposta do presente trabalho no sentido em que ambos objetivam auxiliar a primeira linha de suporte, direcionando a solução do incidente. Porém, o trabalho de Gupta *et al.* supõe que a documentação existente é bem estruturada e que, independente da aptidão do técnico, o procedimento será executado. No caso, por exemplo, de atividades que dependam de análise ou verificação, o processo está exposto a falhas. A estratégia proposta neste trabalho vai além, e propõe a construção de fluxos de ações, livres de decisões, de modo que o operador deve apenas seguir estas instruções claras e específicas. Além disso, a proposta apresentada automatiza atividades de tomadas de decisões, diminuindo ainda mais a demanda da primeira linha de atendimento.

Em uma abordagem bastante distinta, Bartolini *et al.* (BARTOLINI; STEFANELLI; TORTONESI, 2010) propõem uma solução de suporte à decisão de estratégias no gerenciamento de incidentes. Intitulada SYMIAN (*SYMulation for Incident ANalysis*), a ferramenta permite simular um contexto de suporte a incidentes considerando desde a criação do incidente até sua solução. Resumidamente, o SYMIAN permite a simulação de *what-if scenarios* através da modelagem do ambiente de suporte. São considerados, por exemplo, número de incidentes gerados, número de integrantes de cada grupo de suporte, perfil dos

operadores, entre outros. Cada grupo de suporte é tratado de forma análoga a uma fila, que recebe incidentes injetados artificialmente. Conforme o perfil do grupo (número de operadores, estratégia de solução de incidentes, tempo médio de solução, por exemplo), o SYMIAN permite simular diferentes situações. As opções são diversas, e contemplam desde simular a contratação de mais um funcionário em turno deslocado até considerar uma mudança na estratégia de atendimento. Tanto o trabalho de Bartolini *et al.* quanto a solução aqui proposta buscam tornar mais eficiente o processamento de incidentes. No contexto do SYMIAN, isto é realizado avaliando-se os benefícios que mudanças organizacionais trazem ao processo. Neste trabalho, por outro lado, a aceleração do processo é dada através da automatização de atividades de tomadas de decisão.

Além do gerenciamento de incidentes, outros processos merecem atenção em um ambiente de TI; Cordeiro *et al.* (COSTA CORDEIRO et al., 2009) abordam em seus estudos o gerenciamento de mudanças, que está intimamente relacionado com o gerenciamento de incidentes. Segundo ITIL, uma mudança consiste em qualquer alteração programada no ambiente, sendo que esse processo tipicamente é disparado pela criação de uma *RFC* (*Request for Change*, ou Requisição para Mudança). Esse documento é criado pelo cliente, e descreve em linguagem natural a solicitação, como adição de disco a um servidor ou instalação de uma nova aplicação. A execução de uma mudança pode tanto ter como objetivo solucionar um conjunto de incidentes, como também pode originar novos. Por isso, é fundamental que a especificação, o planejamento e a execução da mudança sejam processos confiáveis.

Nesse sentido, o trabalho de Cordeiro *et al.* propõe uma solução com o objetivo de formalizar e reusar o conhecimento em mudanças semelhantes ou recorrentes. Resumidamente, as ações de cada mudança são organizadas em *workflows* compostos de atividades genéricas. Assim, cada atividade da mudança tem uma descrição clara e objetiva. O processo inicia com a RFC; tipicamente, de uma forma não estruturada, esse documento apresenta o que, o por quê, quando, onde e como a mudança deve ser executada. A ferramenta *ChangeLedge* desenvolvida, dentre outras ações, transforma essa descrição em um fluxo organizado de atividades.

Esse trabalho está fortemente relacionado com o aqui proposto, no sentido de que ambos preocupam-se com a formalização e padronização da execução de uma atividade (mudanças ou tratamento de incidentes). Além disso, ambos lidam com o refino de *workflows*. Porém, o trabalho de Cordeiro *et al.* refina os *workflows* através da avaliação de dependências entre ativos por meio de consultas ao CMDB. Por outro lado, a proposta apresentada neste trabalho refere-se ao refinamento através de *pruning* (poda) das atividades a serem executadas, uma vez que atividades que envolvem tomadas de decisões são eliminadas pela automação.

3 ARQUITETURA PROPOSTA

Conforme descrito na Seção 2.2, incidentes podem ser classificados em duas categorias: reportados por usuários ou identificados através de um sistema de monitoração, uma vez que a infraestrutura de TI, seus ativos e os eventos que ocorrem são monitorados. Essa monitoração é realizada através de agentes instalados em cada um dos ativos. Esses agentes coletam periodicamente informações sobre o funcionamento do ambiente. Incidentes reportados por usuários normalmente ocorrem quando o serviço está bastante degradado, a ponto do usuário se sentir incomodado. Esses incidentes são descritos em linguagem natural, e normalmente não contém sequer as informações mínimas para a compreensão da situação.

Por outro lado, incidentes abertos automaticamente são registros bem estruturados, padronizados, e contém informações básicas, como o ativo impactado, data, horário e o sintoma observado, entre outros (GUPTA et al., 2009). Considerando que incidentes oriundos de sistemas de monitoração ocorrem com maior frequência do que os reportados por humanos, e que a rápida atuação do time de suporte pode inclusive evitar que um usuário chegue a ser impactado, neste trabalho serão considerados apenas incidentes deste tipo.

A partir do momento que um incidente é gerado, é responsabilidade da equipe de primeira linha de atendimento prestar o suporte inicial. No contexto deste trabalho, será assumido que os incidentes gerados são atendidos imediatamente, ou seja, a atuação é realizada enquanto a falha se manifesta. Neste instante, são realizadas as primeiras verificações, como análise de extensão e gravidade da situação, além de execução de rotinas que possam solucionar a falha, se aplicável. Conforme discutido no Capítulo 2, os procedimentos normalmente são descritos em linguagem natural, não são padronizados, e muitas vezes são ambíguos e descentralizados. Em adição a esses problemas, dado que o atendimento inicial é sempre realizado por esta equipe, deduz-se que a agilidade e tolerância a falhas do processo de suporte como um todo estão intimamente relacionados com a qualidade do atendimento prestado por esta equipe.

A solução proposta neste trabalho objetiva tornar mais eficiente o atendimento de incidentes, possibilitando a retomada do serviço normal com o menor impacto, no menor tempo possível. A estratégia para alcançar esse objetivo consiste em apresentar para o técnico uma lista de ações simples. Em uma infraestrutura tradicional de suporte a incidentes de TI, tipicamente um operador consulta um procedimento não estruturado para prover o atendimento inicial a um incidente. No entanto, esta estratégia é muito suscetível a falhas e à interpretação do operador. Portanto, a estratégia aqui apresentada busca padronizar, refinar e reduzir os fluxos de atendimentos (*workflows*) inicialmente complexos, em *workflows* simples, livres de decisão. Como forma de simplificar a anatomia do *workflow* a ser percorrido, algumas verificações iniciais podem ser automatizadas, eliminando condições

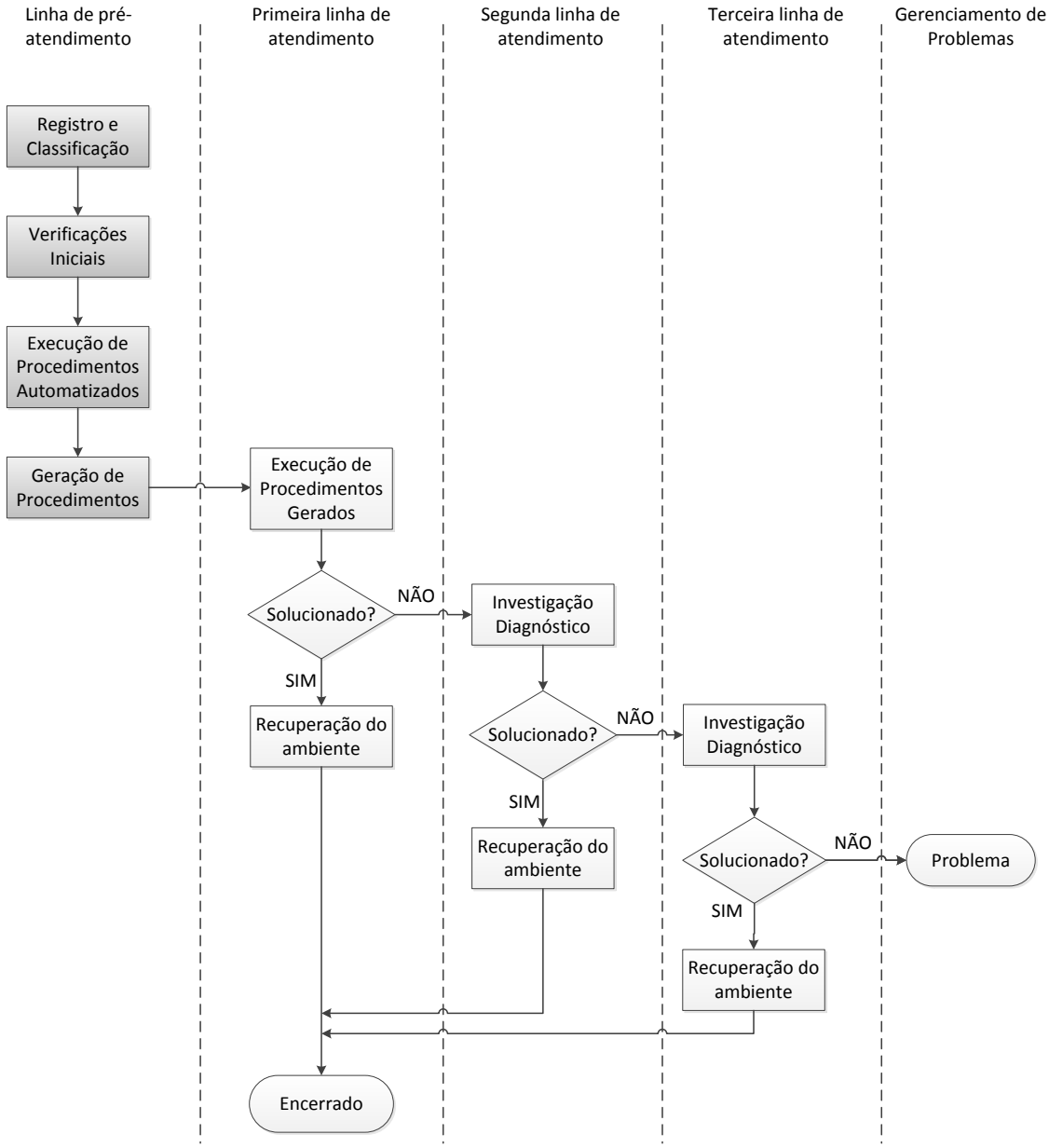

Figura 3.1: Integração da proposta de linha de pré-atendimento à metodologia ITIL.

que, de outra forma, precisariam ser analisadas pelo técnico responsável. Desta forma, no momento em que a equipe de primeiro nível receber o incidente, terá acesso não somente às suas informações básicas, mas também à lista de ações adequadas à situação identificada, sendo essas as ações necessárias para oferecer o suporte correto. A proposta não apenas tem potencial para diminuir o esforço técnico em busca do procedimento correto, como também para reduzir a probabilidade de falhas no atendimento. O benefício em termos de qualidade de processo é claro, uma vez que a sobrecarga do técnico e o tempo de atendimento são imediatamente reduzidos.

Considerando a proposta da ITIL de linhas de atendimento, conforme apresentado anteriormente na Figura 2.2, pode-se dizer que esse trabalho introduz uma nova linha no escopo de suporte, a de pré-atendimento. Essa linha automatizada de análise integra-se ao processo conforme pode ser observado na Figura 3.1.

Tipicamente, um incidente gerado por um sistema de monitoração é descrito através de

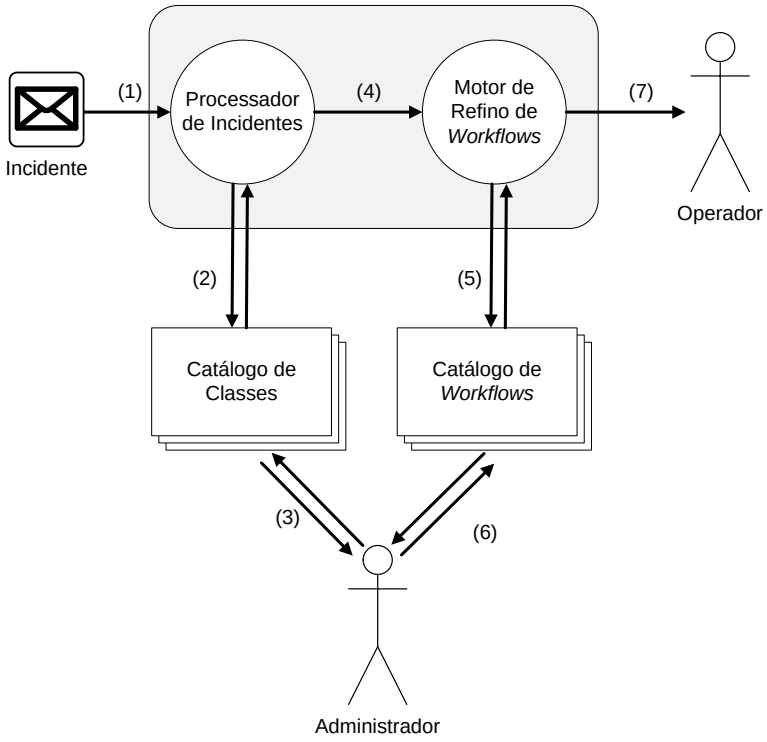

Figura 3.2: Arquitetura com os componentes da solução proposta.

algumas informações básicas, que, contextualizadas, descrevem a situação de degradação ou falha do serviço. Neste conjunto de informações, um dos atributos mais significativos é a descrição do incidente, que consiste em uma breve explicação da situação observada. Na solução apresentada neste trabalho, os incidentes são agrupados em classes com base na semelhança das suas descrições. Assim, é adicionado aos atributos de um incidente o conceito de classe.

A Figura 3.2 apresenta um panorama dos componentes que compõem a solução proposta. De maneira geral, o processo inicia quando uma degradação ou indisponibilidade de algum serviço na infraestrutura, observada na forma de um evento, resulta na instanciação de um incidente. Em um primeiro momento, o incidente é enviado (1) para um Processador de Incidentes. Após a classificação do incidente, informações sobre a classe são recuperadas (2) do Catálogo de Classes, que por sua vez é gerenciado pelo administrador do ambiente (3). O incidente, em sua forma bruta, pode então ser trabalhado, dando origem a um registro apenas com informações relevantes. O registro é encaminhado (4) ao Motor de Refino, que a partir do *workflow* (5) definido para a mesma classe (também mantido pelo administrador (6)), permite a geração da lista de ações que será retornada para análise do técnico de suporte (7). Nas próximas seções, cada subsistema será analisado separadamente.

3.1 Análise de Incidentes

O conjunto de atributos preconizados pela ITIL é extenso, e inclui dados como localização do item de configuração e *logs* para auditoria dos incidentes. No contexto deste trabalho, entretanto, será utilizado apenas um subconjunto destes atributos. Os atributos empregados variam de acordo com a classe do incidente. Mais claramente, a Figura 3.3

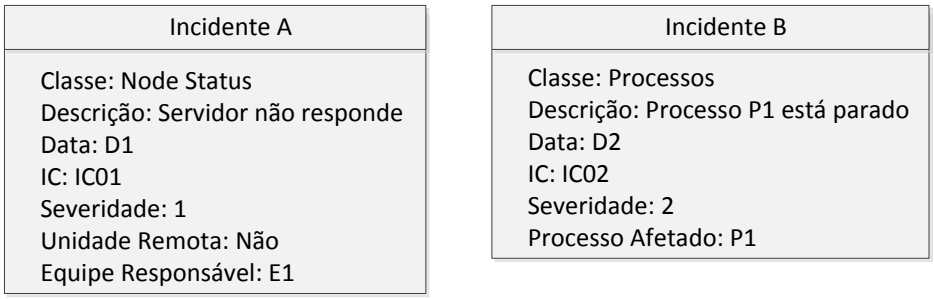

Figura 3.3: Incidentes de exemplo.

mostra dois exemplos de incidentes e os atributos considerados importantes para definição do fluxo de atendimento de cada incidente. A escolha por esses atributos se deve justamente por serem aqueles que levam a uma caracterização precisa da situação. No exemplo, o *Incidente A* ilustra uma situação em que um servidor não está respondendo, ou seja, está incomunicável. Este incidente pode ser causado por uma sensibilidade da rede, pela interrupção no fornecimento de energia, por uma falha de *hardware*, entre tantas outras possibilidades. Algumas informações são importantes para compreensão e tratamento da situação: *Classe*, *Descrição*, *Data*, *IC*. A *Severidade* do incidente está relacionada com o impacto que este tem no ambiente, e, por isso, também é um dado importante. Além disso, dependendo da situação, pode ser que o IC esteja localizado em uma planta de fábrica do cliente; por isso, observou-se a necessidade de introduzir os atributos *Unidade Remota* e *Equipe Responsável* para incidentes desta classe. O *Incidente B*, por sua vez, indica que um processo específico está parado. Esse incidente possui alguns atributos em comum com o *Incidente A*. Porém, este dispensa os atributos *Unidade Remota* e *Equipe Responsável* e necessita identificar qual o *Processo Afetado*.

Com estes exemplos, ratifica-se que cada classe possui o seu conjunto de atributos essenciais, e que essas informações são importantes não apenas para compreensão da situação, mas também para o refino do *workflow*.

3.1.1 Processador de Incidentes

Considerando o cenário recém apresentado, o primeiro componente desta solução consiste em um Processador de Incidentes, que recebe um incidente na sua forma bruta e seleciona apenas as informações relevantes. Esse processador pode ser subdividido em três subcomponentes. Primeiramente, é necessário identificar qual a classe do incidente sob análise. Como comentado, a classe é um aglutinador de incidentes distintos que reportam falhas semelhantes, e é a partir dela que são organizados os catálogos apresentados na Figura 3.2.

Com base na classe do incidente, a segunda etapa do Processador de Incidentes consiste em determinar quais atributos devem ser coletados. Esta amarração entre classes e atributos é representada (e consultada) no Catálogo de Classes. No contexto do trabalho apresentado, o conhecimento prévio destas relações é considerada uma premissa.

Após esse passo, a terceira etapa do Processador de Incidentes é iniciada, que consiste propriamente em coletar os valores dos atributos definidos como relevantes. Nesse contexto, alguns atributos são identificados diretamente do incidente, enquanto que outras informações precisam ser consultadas em fontes externas. Por exemplo, na Figura 3.3, todos os atributos necessários para a compreensão do *Incidente B* são extraídos dos próprios atributos do incidente. Por outro lado, no caso do *Incidente A*, o atributo *Equipe*

Responsável, por exemplo, é um atributo do ativo, e não do incidente. Isso significa que é preciso consultar outra base de dados (que não a de incidentes) para determinar qual a equipe responsável pelo ativo referenciado. Isso se faz necessário pois, embora essa informação não seja importante para a compreensão do incidente, é necessária para o refino do *workflow*.

Nesse sentido, os atributos selecionados podem ser de dois tipos:

- Atributos internos: é uma propriedade que depende exclusivamente do evento gerador do incidente. A informação é, normalmente, carregada através da mensagem do incidente.
- Atributos externos: caracterizam propriedades que não são explícitas na mensagem do incidente. A informação requer consulta a fontes externas.

Para os atributos internos, a única dependência é a mensagem originada pelo sistema de monitoração. A partir dela, a informação desejada pode ser extraída. Os atributos externos, por sua vez, são mais complexos e requerem o suporte de procedimentos específicos para a recuperação da informação necessária. Estes procedimentos podem ser tanto manuais, como a consulta a planilhas por parte do técnico de suporte, quanto automatizados, na forma de rotinas. A representação dos atributos é feita pelo administrador do sistema em uma fase de *set-up* ou manutenção, como será abordado em mais detalhes na Seção 4.1. Ao final, um registro estruturado é criado com os dados obtidos e encaminhado para o Motor de Refino de *Workflows*.

Resumidamente, o componente Processador de Incidentes recebe como entrada um conjunto extenso de atributos de um incidente, ainda não processados, e apresenta como resultado um registro estruturado com os atributos necessários para o refino do *workflow* correspondente à classe do incidente.

3.2 Tratamento de Incidentes

Na ocorrência de uma situação adversa no ambiente, a equipe responsável por acompanhar a monitoração é, normalmente, a primeira a saber da falha. A partir do momento em que o incidente é gerado, uma série de passos são seguidos, de forma a iniciar os procedimentos para a solução do problema. Cada um destes passos pode ser classificado como “decisão” ou “ação”.

Ações são procedimentos realizados incondicionalmente pelos técnicos de suporte. Compreendem desde execução de rotinas, como o envio de mensagens para as equipes, até verificações mais complexas, como testes de conectividade de servidores ou determinação da equipe responsável por cada ativo.

Por outro lado, são as decisões que mudam o rumo do atendimento realizado. O fluxo de ações a ser seguido pode ser diferente conforme a situação, mesmo que a classe do incidente (e, portanto, o *workflow*) seja a mesma. Um exemplo disso são incidentes em ativos críticos, que requerem uma maior atenção desde o início do atendimento, ou então incidentes registrados fora de horário comercial, que podem necessitar de um acionamento específico.

Apesar desta distinção, frequentemente passos de decisões estão vinculados a passos de ações. Nem todas as decisões são simples a ponto de serem resolvidas imediatamente pelo técnico encarregado; muitas vezes a condição só é solucionada após a execução de uma rotina de verificação ou a busca de uma informação. Retomando o exemplo da Figura 3.3, o *Incidente A* possui atributos externos (*Unidade Remota* e *Equipe Responsável*),

que exigem que uma fonte de dados externa seja consultada. Nesse caso, percebe-se que o resultado desta consulta determina qual fluxo será seguido no *workflow*: acionamento de suporte remoto ou atendimento interno.

Cada condição presente em um fluxo de atendimento implica uma escolha que o técnico deve realizar. Em cada decisão, é necessário optar por um entre dois ou mais caminhos possíveis a serem seguidos. Dessa forma, cada condição introduz uma oportunidade de erro. Portanto, este trabalho propõe um processo em que as decisões deixam de ser delegadas para os técnicos, na medida do possível, e passam a ser resolvidas de forma automatizada. Com isso, além de potencial redução de falhas no processo, eventualmente introduzidas por erro humano, também há oportunidade de diminuir o impacto que informações descentralizadas têm no tempo de atendimento a incidentes. Não obstante, também a carga de trabalho e a responsabilidade sobre o time técnico são diminuídas, possibilitando inclusive uma melhoria na qualidade de atendimento.

A seguir, o processo é ilustrado através da apresentação de um exemplo de tratamento de incidente.

3.2.1 Encaminhamento típico para um incidente

Tomando como base um registro resultante da etapa de processamento de incidentes, tal como o *Incidente A* ilustrado na Figura 3.3, a solução evolui com a recuperação do *workflow* correspondente, armazenado no Catálogo de *Workflows*. O passo subsequente consiste no percorrimento do *workflow*; na existência de condicionais, busca-se a resposta para estes no próprio registro do incidente. Percorre-se o *workflow* até o final, ficando pendentes para execução apenas as atividades que necessitam de intervenção humana. Dessa forma, o humano apenas executa as atividades indicadas, pois o *workflow* apresentado ao operador já está refinado.

Foi abordado na Seção 3.1 o processamento de um incidente sintético, da classe *Node Status*, o *Incidente A* da Figura 3.3. De forma a ilustrar também o processo de refino de *workflows*, será utilizado o *workflow* definido na Figura 3.4 como exemplo para esta classe. Para a confecção deste exemplo, supõe-se que os procedimentos a serem executados na ocorrência deste tipo de falha já foram devidamente mapeados pela equipe de suporte. Da mesma forma que ocorre no componente Processador de Incidentes, em que é suposta a existência de um Catálogo de Classes, aqui assume-se como premissa a existência de um Catálogo de *Workflows*, criado e mantido pelo administrador do ambiente.

Primeiramente, é necessário o *workflow* completo, como se fosse tratado de forma tradicional (sem qualquer automatização), representado na Figura 3.4. O roteiro de atendimento a incidentes sempre é iniciado com a etapa (1), que representa o primeiro contato do técnico de suporte com o incidente. No exemplo, a etapa (2) é uma decisão, baseada na localização[1] do ativo afetado. Caso o ativo encontre-se em uma unidade remota, o operador deve acionar o técnico da localidade para prestar o suporte (etapa (3)). Caso contrário, é o próprio operador quem tratará o incidente, partindo para a etapa (4) que consiste em verificar qual a equipe responsável pelo ativo. Seguindo essa linha, na etapa (5) o operador deve decidir qual a severidade do incidente. Em caso de severidade 1, ou seja, indisponibilidade total do serviço, a equipe responsável deve ser acionada por telefone (etapa (7)) e depois o incidente deve ser encaminhado para a equipe (etapa (8)). Caso se trate de uma severidade 2, em horário comercial (decisão na etapa (6)), basta en-

[1] Apesar de ativos normalmente ficarem instalados em um centro único (CPD), não é raro encontrar cenários em que recursos específicos (como servidores de impressão ou sistemas de arquivos) sejam instalados mais próximos do usuário final, minimizando assim a latência e o fluxo de dados na rede.

caminhar o incidente para a equipe; em caso de horário diferenciado, segue-se o mesmo fluxo de um caso de severidade 1. Para as demais severidades, independente do horário de ocorrência do incidente, o tratamento segue diretamente para a etapa (8), sem acionamentos específicos. Após o encaminhamento na etapa (8), a intervenção do operador está encerrada (etapa (9)).

O objetivo deste trabalho é que o operador possa seguir um fluxo livre de decisões, ou tomando o mínimo de decisões possíveis. Além disso, verificações também são pontos possíveis de falha, e, se passíveis de automatização, serão tratadas sem intervenção do operador. A Figura 3.5a ilustra o fluxo que é seguido considerando o *Incidente A* da Figura 3.3. Considerando os objetivos, imediatamente foi identificado que esta solução pode eliminar os passos (2), (5) e (6) com base nos atributos *Unidade Remota*, *Severidade* e *Data*. Além disso, o passo (4) também utiliza informação contida em atributo (*Equipe*). Como explicado anteriormente, na realidade esses atributos foram escolhidos como importantes para incidentes da classe *Node Status* justamente pois são eles que permitem que as decisões e verificações do *workflow* sejam automatizadas. Considerando os passos que podem ser automatizados e retomando os atributos do exemplo, a Figura 3.5b ilustra o *workflow* resultante, após o refino, que será apresentado ao operador para execução. Pode-se perceber que o número de atividades é muito menor, o que implica um menor tempo de atendimento do incidente. Além disso, as atividades do operador resumem-se em ações claras e diretas, sem necessidade de tomada de decisões, o que reduz a probabilidade de falhas no processo.

3.2.2 Motor de Refino de *Workflows*

Tomando as considerações acima apresentadas como ponto de partida, o segundo componente da solução proposta é denominado Motor de Refino de *Workflows*. Foi visto que a classificação do incidente é fundamental para possibilitar sua análise em um sistema automatizado. Após a classificação e extração de seus atributos (realizadas no componente Processador de Incidentes), é necessário iniciar uma atuação em busca da redução do *workflow* e sua transformação em um fluxo livre de decisões. As etapas desta atuação, entretanto, são particulares para cada classe de incidentes, e assume-se como premissa que cada classe possui um *workflow* definido. Buscando formalizar o raciocínio do exemplo utilizado na seção anterior, o pseudo-algoritmo 3.1 ilustra o funcionamento do Motor de Refino.

O Motor de Refino conta com duas entradas: o Registro do Incidente e o *Workflow* correspondente à classe do incidente. Dado que o objetivo de saída do Motor de Refino é uma lista de atividades a serem executadas pelo operador, o primeiro passo do algoritmo é criar uma lista vazia de atividades (linha 1). A seguir, cada atividade do *workflow* será avaliada individualmente. Lembrando que no começo desta seção foi definido que toda atividade é de decisão ou ação, o primeiro passo é avaliar se a atividade é uma ação (linha 3). Para uma atividade do tipo ação, é preciso identificar se a ação depende do conhecimento de algum atributo do incidente (linha 4); em caso positivo, busca-se o atributo no Registro do Incidente (linha 5), seu valor (linha 6) é recuperado e a atividade é atualizada com essa informação. Nesse ponto, é possível inserir a ação na lista de atividades do operador (linha 7). Esse tipo de automatização consiste em uma ação de verificação, e sua automatização evita uma possível falha humana e reduz o tempo de avaliação do incidente. Por outro lado, a atividade pode ser uma decisão (linha 10), e, nesse caso, a tomada de decisão é feita pelo Motor de Refino com base nos atributos do incidente; primeiro, identifica-se qual o atributo vinculado à decisão (linha 11), recupera-se o valor associado

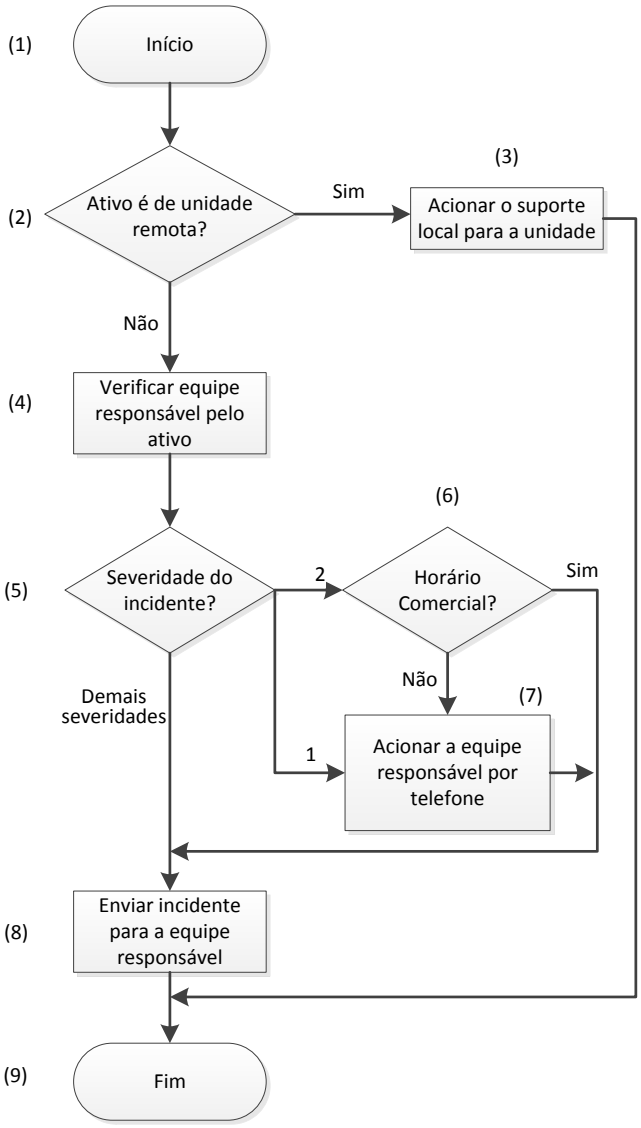

Figura 3.4: *Workflow* hipotético para suporte a incidentes da classe *Node Status*.

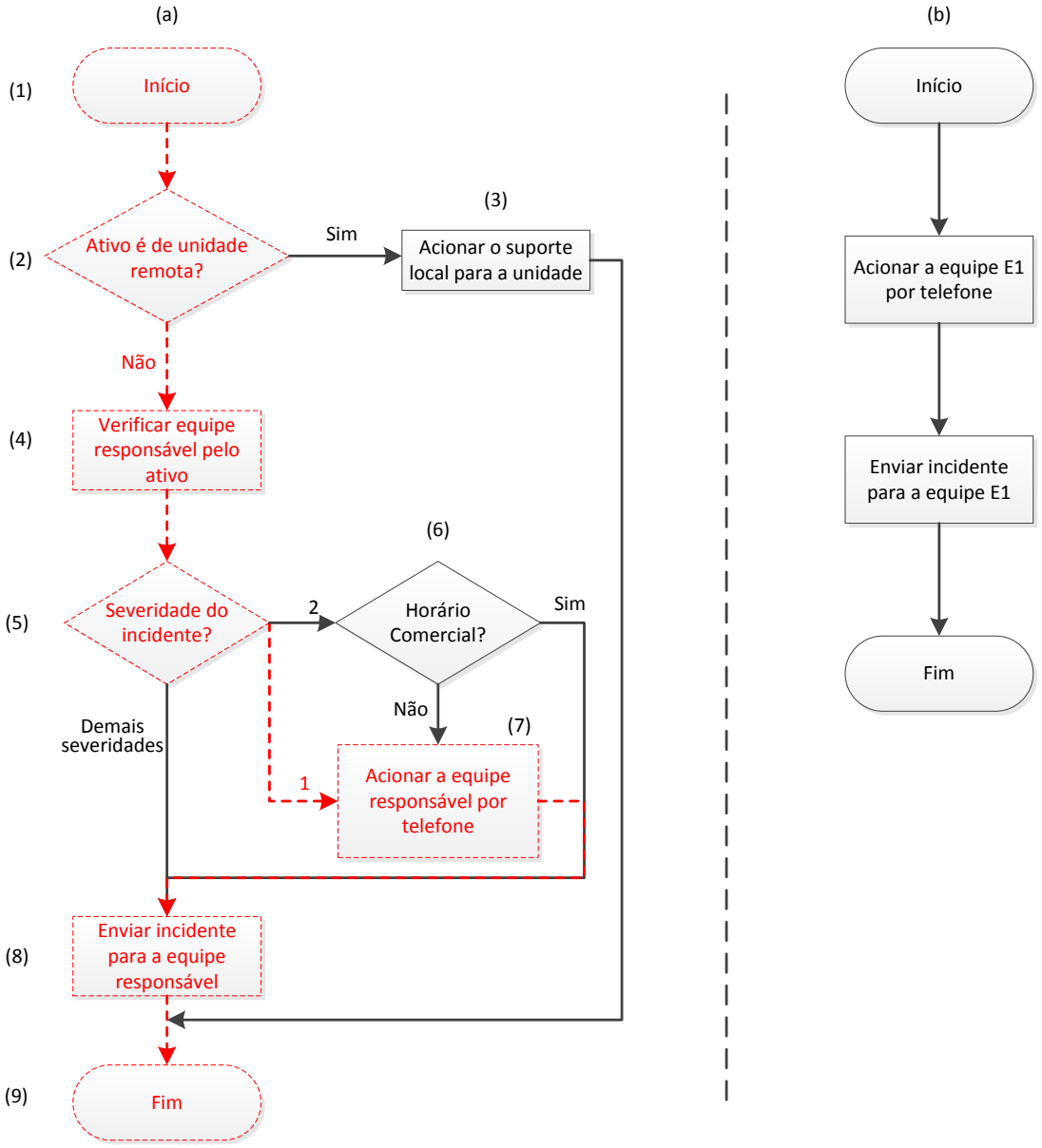

Figura 3.5: (a) *Workflow* com marcações ilustrativas do percorrimento. (b) Fluxo de ações resultante do refino do *workflow* para o incidente de exemplo.

Algoritmo 3.1: Pseudo-algoritmo do processo de refino de *workflows*.

Entrada: Registro do Incidente
Entrada: *Workflow*
Saída: Lista de Ações

```
1  criar lista vazia de ações;
2  para cada atividade ∈ Workflow faça
3      se atividade é ação então
4          se atividade depende de algum atributo então
5              buscar pelo atributo no Registro do Incidente;
6              recuperar o valor associado e preencher na atividade;
7          fim
8          inserir atividade na lista de ações;
9      fim
10     se atividade é decisão então
11         identificar o atributo atrelado a essa decisão;
12         buscar pelo atributo no Registro do Incidente;
13         testar o valor associado e determinar qual o fluxo a ser seguido;
14     fim
15 fim
16 retorna ações;
```

(linha 12), e, finalmente, decide-se, com base no atributo, qual o fluxo de atividade a ser definido (linha 13). Ao final do processamento das atividades, todas as decisões dependentes de atributos foram eliminadas, bem como as ações de verificação. O processo retorna, portanto, uma lista única e clara de ações a serem tomadas pelo operador.

Com esse componente, a solução proposta está completa. Recapitulando, o processo inicia quando um incidente contendo um conjunto extenso de atributos é recebido, então é passado pelo Processador de Incidentes, depois pelo Motor de Refino de *Workflows*, e obtém-se como resultado um fluxo livre de decisões para ser executado pelo operador. No capítulo a seguir, apresenta-se a solução denominada *Workflow Automated Refinement for Incident Analysis* (WARIA), que agrega os componentes acima descritos.

4 FERRAMENTA WARIA

Com o intuito de avaliar a viabilidade técnica da solução proposta, foi desenvolvido um protótipo de um sistema de gerenciamento de incidentes, amparado pelo processo de refino de *workflows*. Este sistema foi nomeado WARIA (*Workflow Automated Refinement for Incident Analysis*.

Conforme destacado no Capítulo 3, a solução é dividida em Processador de Incidentes e Motor de Refino de *Workflows*. Suportando estes componentes, assume-se como premissa a existência de um Catálogo de Classes e um Catálogo de *Workflows*, criados e mantidos pelo administrador do ambiente. Considerando este cenário, este capítulo é iniciado pela apresentação da construção dos catálogos, e, na sequência, mostra-se como foram implementados os demais componentes da solução.

4.1 Definição dos Catálogos

Para que o protótipo do sistema possa ser executado, é necessário criar as ferramentas de apoio, que suportam os principais componentes da proposta apresentada. Pode-se considerar que esta é a parte relativa à carga inicial da solução, pois é realizada principalmente durante o *set-up* da implantação. Apesar de ser um trabalho manual, coordenado pelo administrador do ambiente, sua execução é necessária apenas na definição de uma nova classe ou na eventual necessidade de modificar o fluxo de uma classe já existente.

Como foi visto anteriormente, tanto atributos quanto *workflows* estão intimamente relacionados com a classe do incidente. Os atributos indicam, no registro do incidente, quais os valores essenciais para o refino do *workflow* daquela classe. Posto de outra forma, pode-se dizer que a determinação de quais informações são relevantes em um incidente é feita pelo seu próprio *workflow*. Desta forma, é natural que o processo de definição de uma nova classe de incidentes inicie pela construção do *workflow* associado e, em um segundo momento, seja realizada a especificação dos atributos necessários.

4.1.1 Catálogo de *Workflows*

Baseados na proposta deste trabalho, observou-se a necessidade de representar fluxos ordenados de ações e condições de forma que pudessem ser processados sistematicamente. Verificou-se que a modelagem destes fluxos em *workflows* possibilitaria esta representação. Considerando que o cenário escolhido para testes já contava com uma série de procedimentos documentados informalmente (processo que será detalhado no Capítulo 5), a construção do Catálogo de *Workflows* consiste em transformar esses procedimentos não estruturados, descritos em linguagem natural, em fluxos padronizados.

Para realizar a especificação dos fluxos, foi utilizada a ferramenta *Intalio|Designer*,

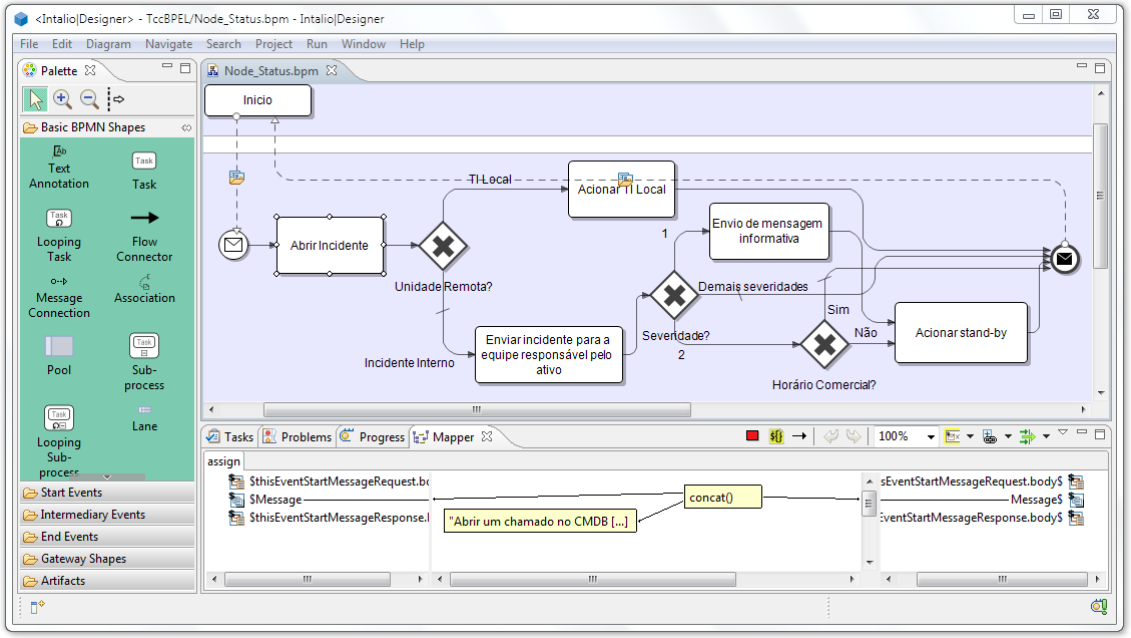

Figura 4.1: *Workflow* da classe *Node Status* em edição no Intalio.

versão 6.1.12, desenvolvida pela própria Intalio. A suíte de aplicativos fornece ferramentas para a confecção e o gerenciamento de *workflows*. Além destes recursos, funcionalidades para além da manipulação de *workflows* são oferecidas; no entanto, para esta implementação, apenas os recursos que estão relacionados com *workflows* serão utilizados.

A interface do Intalio é apresentada na Figura 4.1. A construção de *workflows* no programa é realizada de maneira bastante intuitiva, através da composição de blocos no palco principal (identificado pelo nome do arquivo, "*Node Status.bpm*" no exemplo). Estes são arrastados da biblioteca de formas (palheta, ou *palette*, à esquerda), e o fluxo de ações é definido conectando-se um bloco ao outro. Desta forma, pode-se dizer que o processo de construção é independente de uma linguagem de programação específica. Entre os blocos disponíveis, os mais utilizados são: *tasks* (que representam atividades) e *gateways* (que representam os condicionais do processo). Além destas estruturas, o Intalio também permite que ações sejam indicadas para execução em cada uma das atividades (através do painel *mapper*, na parte inferior); este recurso será explorado na Seção 4.2.2.

Novamente será utilizado como base o *Incidente A*, apresentado na Figura 3.3, para ilustrar a construção de um *workflow*. Pode-se observar na Figura 4.1 como o *workflow* da classe *Node Status* foi representado no programa. Os condicionais, relativos à localização do ativo, severidade do incidente e horário de ocorrência, foram transformados em *gateways*. De maneira semelhante, as ações foram traduzidas em *tasks*.

A execução destas etapas de construção de um *workflow* deve ser realizada para cada classe de incidentes mapeada no ambiente. O conjunto resultante destes fluxos de procedimentos é chamado de Catálogo de *Workflows*. É no momento de criação do *workflow* de uma classe que se observa quais condicionais serão necessários avaliar para efetuar o seu refino, no componente Motor de Refino de *Workflows*. Portanto, é nesta etapa que determina-se quais os atributos necessários para uma classe.

Figura 4.2: Registro integrante do Catálogo de Classes, referente à classe *Node Status*.

```
% Classe
Node Status
==
% Mensagem
Servidor não responde - Data: {{DATA}} - IC: {{STRING}} - Severidade:
    {{INT}}
==
% Atributos Internos
1,Data
2,IC
3,Severidade
==
% Atributos Externos
UnidadeRemota,IC,{
   var ws = new WebService();
   return (bool)ws.EhAtivoUnidadeRemota(IC);
}
EquipeResponsavel,IC,{
   var ws = new WebService();
   return (string)ws.EquipeResponsavel(IC);
},
StandBy,EquipeResponsavel,Data,{
   var ws = new WebService();
   return (string)ws.StandBy(EquipeResponsavel, Data);
},
HorarioComercial,EquipeResponsavel,Data,{
   var ws = new WebService();
   return (bool)ws.EhHorarioComercial(EquipeResponsavel, Data);
}
```

4.1.2 Catálogo de Classes

Uma vez que um *workflow* integra o catálogo, torna-se necessário representar os atributos apropriados para o seu refino. Isto é realizado definindo-se um Catálogo de Classes, onde cada registro que o compõe é uma estrutura de dados que armazena informações dos atributos daquela classe e a instrução de como obter os valores destes atributos. A Figura 4.2 ilustra o formato de definição de um desses registros, que é dividido em 4 seções:

- classe;
- *template* da mensagem;
- atributos internos;
- atributos externos.

A primeira seção é utilizada para identificar a classe correspondente ao registro e, portanto, contém apenas o seu nome. A segunda seção representa a mensagem que é gerada pelo sistema de monitoração, no momento do alerta do incidente. Esta mensagem normalmente contém partes fixas, que se repetem em todos os incidentes desta classe, e partes variáveis, que são específicas do incidente alarmado. Por isso, a mensagem é

Tabela 4.1: Preenchimento de um *template* para geração de um incidente.

Mensagem *template*	Servidor não responde - Data: {{DATA}} - IC: {{STRING}} - Severidade: {{INT}}
Mensagem preenchida	Servidor não responde - Data: 01/12/2012 20:56:00 - IC: SRV01 - Severidade: 2

inserida na especificação na forma de um *template*, isto é, possui marcações que indicam os trechos fixos e os variáveis – que serão substituídos no incidente real. Estes marcadores serão posteriormente traduzidos para expressões regulares, que possibilitarão o casamento entre *template* e mensagem real. A Tabela 4.1 ilustra o preenchimento de um *template* em um incidente, em que a data foi preenchida para "01/12/2012 20:56:00", o item de configuracão para "SRV01" e a severidade para "2".

As marcações estão relacionadas com os atributos internos de um incidente. Cada atributo interno é definido pela tupla "índice da posição na marcação da mensagem" e "nome do atributo". No exemplo da Figura 4.2, a terceira marcação na mensagem é referente à severidade do incidente; em virtude disso, o atributo interno é descrito como "3,Severidade". Por fim, atributos externos são definidos como em uma n-upla, em que o primeiro elemento é o nome do atributo, o último elemento contém a rotina utilizada para extrair o valor do atributo, e os elementos intermediários são os nomes dos atributos necessários para a execução da rotina. Novamente considerando o exemplo, percebe-se que o primeiro atributo externo definido, "UnidadeRemota", possui um pré-requisito: o item de configuração do incidente. No momento que este atributo externo for avaliado, o valor do IC já será conhecido, e portanto pode ser utilizado na rotina do atributo externo como chave para determinar se o ativo é ou não de unidade remota.

4.2 Núcleo do sistema

O "núcleo do sistema" é formado pelos componentes que efetivamente ficam em execução durante o processo de atendimento a incidentes, ou seja, pelo Processador de Incidentes e Motor de Refino de *Workflows*. Conforme viu-se na seção anterior, os catálogos são componentes relativamente estáveis, uma vez que só requerem alterações quando há modificações no processo. Por outro lado, os componentes que serão vistos a seguir são extremamente dinâmicos, pois o processamento de incidentes e refino de *workflows* é contínuo.

4.2.1 Processador de Incidentes

A identificação de um incidente ocorre através de um agente de monitoração, que coleta periodicamente informações diversas sobre os ativos. Nesta implementação, considera-se que, quando o agente identifica uma situação possível de risco, o incidente gerado é inserido em um banco de dados relacional. Sua estrutura fica armazenada na forma de um registro contendo uma série de informações, como apresentado na Seção 3.1.

O Processador de Incidentes foi desenvolvido para a plataforma Microsoft .NET, utilizando linguagem de programação C# e bibliotecas do *framework* .NET 4.0 para comunicação com o banco de dados e interface gráfica. Primeiramente, realiza-se uma conexão entre o protótipo e a base de dados que contém os incidentes. Quando um novo incidente é detectado, sua mensagem e classe são recuperadas. A seguir, o Catálogo de Classes é consultado para determinar quais os atributos que serão selecionados. Esses atributos po-

dem ser internos ou externos, ou seja, podem respectivamente estar contidos no registro original do incidente ou em fontes externas.

Os valores dos atributos internos estão contidos diretamente na mensagem do incidente (ou seja, na sua "descrição"). Conforme exemplificado anteriormente, no Catálogo de Classes cada classe possui os marcadores de atributos na sua mensagem. As mensagens, por sua vez, são armazenadas com os marcadores substituídos por expressões regulares, cada qual adequada para o tipo de dado que o marcador representa. Desta forma, na ocorrência de um incidente, é possível comparar sua descrição com a mensagem registrada na classe. Com base nas marcações, torna-se possível extrair da descrição os valores relativos aos atributos internos, que são armazenados na forma de uma lista.

Para o processamento dos atributos externos, a função correspondente (registrada no Catálogo de Classes) é executada, recebendo como entrada uma lista com os valores relativos aos atributos informados como pré-requisitos. Por exemplo, para determinar o atributo externo *Unidade Remota* do *Incidente A* da Figura 3.3, a função a ser executada seria uma consulta na tabela de ativos, e a entrada dessa função seria o atributo interno *Servidor*, já extraído do incidente. Como prova de conceito, o acesso a atributos externos foi implementado através de um servidor *web*, que é consultado utilizando-se *Web Services*. Os *Web Services* que possibilitam recuperar os valores dos atributos externos ilustrados na Figura 4.2 foram implementados. Estes atributos referem-se também ao *Incidente A* da Figura 3.3, e utilizados no *workflow* exibido na Figura 4.1.

Após esse tratamento, o componente Processador de Incidentes terá gerado um registro com as informações do incidente que serão utilizadas no refino do *workflow*. Será analisada, a seguir, a implementação do Motor de Refino de *Workflows*.

4.2.2 Refino de *Workflows*

O Intalio, além de possibilitar a especificação dos *workflows*, também possibilita a sua execução. Os fluxogramas definidos na ferramenta são convertidos para *Web Services*, possibilitando sua invocação por sistemas externos. Para isso, é necessário definir as entradas e saídas de cada fluxograma, ou seja, a interface que será exposta para o usuário. As entradas dos *workflows* correspondem aos atributos que integram o registro estruturado do incidente, como apresentado no final da Seção 3.1.1. Tendo em vista que o objetivo neste trabalho é percorrer o *workflow*, determinando quais atividades devem ser executadas, a saída do processo é definida como uma lista de ações para o técnico de suporte.

Durante a etapa de especificação dos fluxos, definiu-se que cada atividade representa uma instrução que será repassada para o operador. A execução do *workflow* corresponde em determinar o caminho a ser percorrido para um dado incidente. Dessa forma, busca-se reconhecer as atividades que fizerem parte deste caminho.

O Intalio permite que sejam definidas ações a serem executadas tanto para atividades quanto para condicionais durante o percorrimento do fluxo, através do painel *mapper* da Figura 4.1. Na implementação realizada, as ações definidas nos condicionais são avaliações em relação aos atributos fornecidos no registro de entrada. Já as atividades possuem ações para adicionar a instrução que representam (isto é, a ordem ao operador), descrita em linguagem natural, à lista de saida. Assim, ao final do processo, é obtido um conjunto de comandos que indicarão como o operador deve tratar a situação.

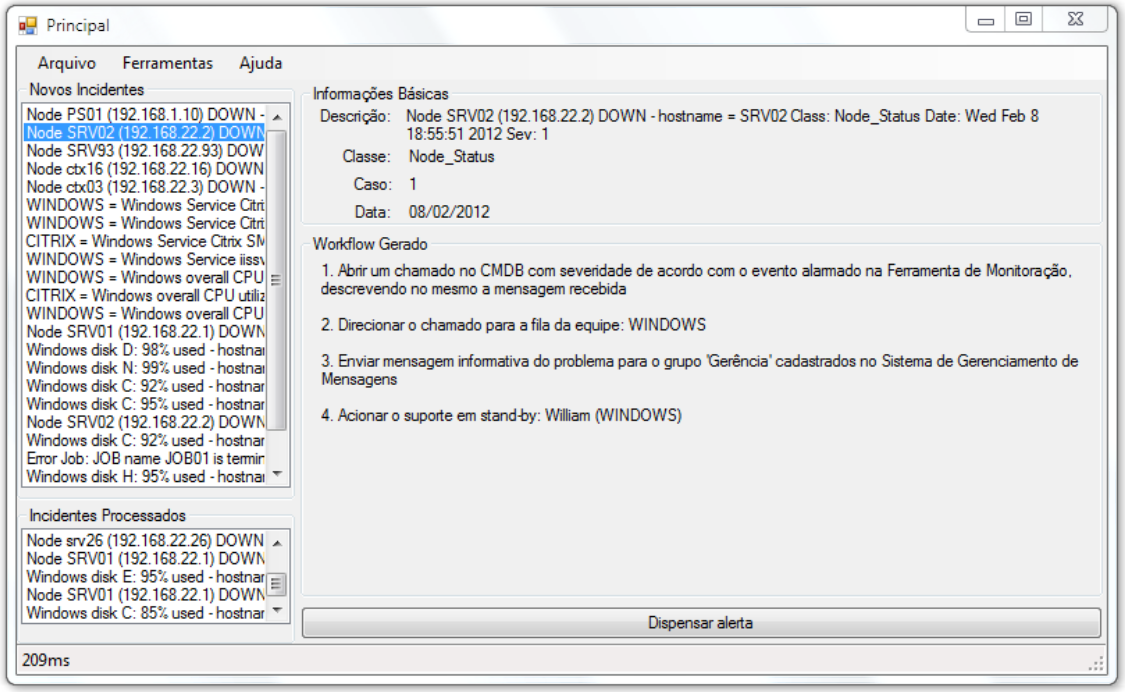

Figura 4.3: Interface do protótipo, exibindo incidente compatível com o *Incidente A*.

4.3 Interface com o operador

Nas seções anteriores, foi apresentada uma visão dos aspectos de *back-end* da solução, através dos componentes que são executados de maneira transparente para o usuário. No intuito de reproduzir uma situação real de atendimento a incidentes, foi desenvolvida uma interface simples, como prova de conceito, que visa reproduzir o tipo de interface que o operador deveria ter possibilidade de usar. O *front-end* desta solução é apresentado na Figura 4.3.

Na tela principal, existem duas listagens à esquerda: uma para incidentes não processados e uma para incidentes já processados. Ao selecionar um incidente, a parte superior da tela é preenchida com suas informações básicas. Uma consulta via *Web Services* é realizada para o servidor do Intalio, que executa o refino do *workflow* e retorna a lista de ações para o operador, exibidas logo abaixo. Por fim, um botão na parte inferior da interface representa o momento em que o técnico efetua as ações descritas nos passos; ao pressioná-lo, o incidente é "retirado" da fila de entrada, sendo registrado no histórico.

O incidente exibido na Figura 4.3 é, novamente, da classe *Node Status*. Esse incidente foi escolhido de tal forma que os valores de seus atributos são compatíveis com o *Incidente A* da Figura 3.3. Com isso, pode-se observar a saída do percorrimento do *workflow* correspondente, tal como destacado na Figura 3.5a.

O resultado do processo para o exemplo supracitado pode ser comparado com o incidente ilustrado na Figura 4.4. Neste segundo exemplo, supõe-se que o ativo afetado ("PS01") é um servidor de impressão local e, portanto, pertencente a uma Unidade Remota. Apesar da classe dos incidentes ser a mesma, esta diferença tem um grande impacto nos caminhos percorridos do *workflow*; assim, um dos principais objetivos do trabalho, que é a geração das ações de forma particular para cada incidente, é ilustrado. No próximo capítulo, será realizada uma análise dos resultados obtidos, considerando incidentes desta e de outras classes.

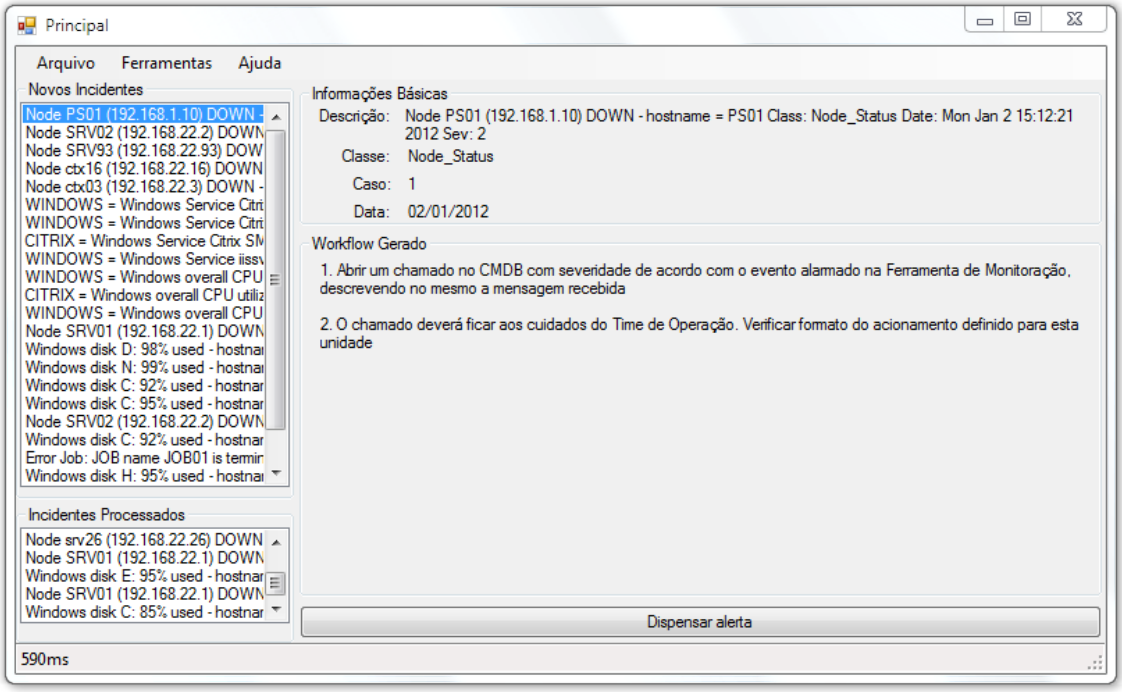

Figura 4.4: Lista de ações geradas para incidente de exemplo.

5 AVALIAÇÃO E RESULTADOS

Com o intuito de avaliar a solução proposta neste trabalho, realizou-se uma análise sobre uma base de dados real de incidentes em um ambiente corporativo de TI. Esta base é composta por cerca de 100 mil incidentes, reportados por um sistema de monitoração, coletados entre janeiro de 2008 e setembro de 2012. Neste conjunto de incidentes, observou-se a existência de cerca de 100 classes distintas. Para possibilitar uma análise mais aprofundada, os estudos foram direcionados em algumas classes, assim reduzindo o volume de dados total para análise. Foram escolhidas as classes com maior número de incidentes em relação ao total, e as classes de maior complexidade.

Como resultado desse filtro, foram amostradas cinco classes para estudo. Em relação ao número de incidentes disponíveis, fazem parte destas classes 45.927 incidentes. Ou seja, 5% das classes correspondem a aproximadamente 45% dos incidentes. As classes selecionadas, e a sua representatividade em relação ao total de incidentes, estão enumeradas na Tabela 5.1. Dessa forma, apesar da quantidade reduzida de classes, será possível analisar os resultados – e seus impactos – em relação a um número expressivo de incidentes da base. Por questões de confidencialidade, informações particulares como nome de ativos, equipes e classes foram omitidos ou modificados nos exemplos.

5.1 Análise Qualitativa

Em um primeiro momento da análise, buscou-se avaliar qualitativamente a solução proposta. A classe *Jobs* (ou rotinas) foi escolhida para este estudo de caso, pois além de ser a classe com maior número de incidentes, também apresenta um fluxo complexo de atividades de primeiro nível. O procedimento *ad-hoc* utilizado no ambiente em análise é exibido na Figura 5.1. Considerando que o operador atende diariamente a dezenas de incidentes distintos (inclusive de diversas classes), tipicamente em cada tratamento o operador consulta todo o procedimento a ser executado e precisa definir quais as atividades que devem ser executadas. Assim, todo o processo é muito dependente da interação e da

Tabela 5.1: Classes selecionadas para análise.

Classe	Total de Incidentes
Jobs	21363
Logical_Disk	7753
Node_Status	7773
Processor_Summary	1777
Services	7261

capacidade de análise do operador que, agravado pela sobrecarga de trabalho, está sujeito a erros e imprecisões.

Retomando o exemplo de estudo da classe *Jobs*, o procedimento *ad-hoc*, utilizado pelos operadores para tratamento de um incidente sem o apoio de uma ferramenta de automação, é representado na Figura 5.2. Neste procedimento, descrito em linguagem natural, todos os possíveis fluxos são considerados, tornando o procedimento complexo; da mesma forma, também o *workflow* referente a esta classe será complexo. No entanto, considerando que o operador não tem suporte de ferramentas automatizadas, ele sempre tem de avaliar todo o procedimento para determinar o atendimento correto. Por outro lado, utilizando uma ferramenta com recursos semelhantes aos disponibilizados pela ferramenta proposta nesta solução, apenas as atividades a serem executadas são apresentadas ao operador.

Para ilustrar o benefício trazido pela especificidade dos *workflows* gerados, dois incidentes distintos serão comparados, ambos pertencentes à classe *Jobs*. É importante destacar que, durante a implementação da ferramenta no ambiente utilizado para testes, surgiu a necessidade de relacionar mais de uma mensagem à mesma classe de incidentes. Por exemplo, as mensagens exibidas quando uma rotina fica em execução por um tempo maior que o normal e quando uma rotina termina a sua execução com falha são distintas. Entretanto, como ambas se relacionam com a execução de rotinas, são ditos casos diferentes da mesma classe de incidentes. Portanto, o conceito de "caso" foi implementado como uma extrapolação dos conceitos apresentados no Capítulo 4.

A Listagem 5.1 exibe os incidentes pertencentes à classe "*Jobs*" que serão analisados. Considerando uma situação sem uma solução automatizada, a instanciação de cada incidente faz com que o operador percorra manualmente os procedimentos da Figura 5.1, identificando o encaminhamento correto. Nas próximas seções, o atendimento de cada um destes incidentes será simulado, inicialmente no cenário sem a automatização, e depois considerando a implementação da solução WARIA.

Listagem 5.1: Incidentes para análise qualitativa.

(A) JOB name JOB02 is running to more than 10 minutes − hostname = SRV90 Class : Jobs Date: Sep 27 19:52:17 2011 Sev: 3
(B) JOB ROBO name JOB03 is running to more than 120 minutes − hostname = SRV91 Class: Jobs Date: Oct 10 23:28:37 2011 Sev: 1

5.1.1 Primeiro Caso

Considerando o processo não automatizado de atendimento a incidentes, a primeira ação do operador ao receber o incidente (A) da Listagem 5.1 é identificar a sua classe (*Jobs*) e abrir o arquivo de procedimentos correspondente. Em seguida, o técnico precisa decidir em qual caso a mensagem instanciada pelo sistema de monitoração ("*JOB name JOB02 is running to more than 10 minutes - hostname = SRV90 Class: Jobs Date: Sep 27 19:52:17 2011 Sev: 3*") pode ser enquadrada, dentre as presentes no arquivo de procedimentos. Após a leitura de todas as mensagens do arquivo, determina-se que o incidente corresponde ao caso 6. Com essa informação, a descrição do evento pode ser consultada, obtendo-se mais detalhes sobre o problema que ocorre; neste caso, uma rotina que está em execução há mais de 10 minutos. Neste momento, o analista passa para a leitura dos procedimentos a serem executados com o intuito de solucionar o problema.

A primeira ação do procedimento indica que o técnico deve registrar o evento no banco de dados de incidentes, com os detalhes recebidos da ferramenta de monitoração.

Arquivo de Procedimento para Jobs

Mensagem da Ferramenta de Monitoração

Caso 1 - Error Job: JOB name xxxxxx is terminate with status xxxxx - hostname = XXXXXX Class: Jobs Date: SSS MMM DD HH:MM:SS AAAA Sev: X X

Caso 2 - Cancel Job: JOB name xxxxxx is terminate with status xxxxx - hostname = XXXXXX Class: Jobs Date: SSS MMM DD HH:MM:SS AAAA Sev: X X

Caso 3: Error Job: JOB for TSM name xxxxxx is terminate with status xxxxx - hostname = XXXXXX Class: Jobs Date: SSS MMM DD HH:MM:SS AAAA Sev: X

Caso 4: Cancel Job: JOB for TSM name xxxxxx is terminate with status xxxxx - hostname = XXXXXX Class: Jobs Date: SSS MMM DD HH:MM:SS AAAA Sev: X

Caso 5 - JOBS name xxxx is status Ready - hostname = XXXXXX Class: Jobs Date: SSS MMM DD HH:MM:SS AAAA Sev: X

Caso 6- JOB name XXXXXXXX is running to more than 10 minutes.- hostname = XXXXXX Class: Jobs Date: SSS MMM DD HH:MM:SS AAAA Sev: X X

Caso 7- JOB ROBO name XXXXXXXXXXX is running to more than 120 minutes. - hostname = XXXXXX Class: Jobs Date: SSS MMM DD HH:MM:SS AAAA Sev: X X

Caso 8-TSM job %s running more than 60 minutes - hostname = XXXXXX Class: Jobs Date: SSS MMM DD HH:MM:SS AAAA Sev: X X

Caso 9- TSM job %s is in READY status for more than 60 minutes - hostname = XXXXXX Class: Jobs Date: SSS MMM DD HH:MM:SS AAAA Sev: X X

Caso 10 - PRODUCT Efetuar acionamento do Preparo: Possivel falha na conexao com a Empresa Externa - hostname = XXXXXX Class: Jobs Date: SSS MMM DD HH:MM:SS AAAA Sev: X X

Caso 11 - TSM job %s running more than 12 hours - hostname = XXXXXX Class: Jobs Date: SSS MMM DD HH:MM:SS AAAA Sev: X X

Caso 12 - SAP - Efetuar acionamento de SAP: falha no processo de restart automatico do Portal - Job Name: XXXXXX - hostname = XXXXXX Class: Jobs Date: SSS MMM DD HH:MM:SS AAAA Sev: X X

Descrição do Evento

Caso 1 e caso 2 - O Job especificado terminou com status failed, warning ou canceled.

Caso 3 e caso 4- O Job de backup do TSM executado terminou com status failed, warning ou canceled.

Caso 5 - O Job especificado está com com status de Ready a mais de 10 minutos.

Caso 6- O Job especificado está em execução por mais de 10 minutos.

Caso 7- O Job especificado está em execução por mais de 2h e necessita ser verificado.

Caso 8 - O Job de backup está em execução a mais de 1h e necessita ser verificado.

Caso 9 - O Job de backup está com status READY a mais de 1h e necessita ser verificado.

Caso 10 - Falha na conexão Empresa Externa

Caso 11- O Job especificado está em execução por mais de 12 horas.

Caso 12 - O job especificado finalizou com erro e deve ser acionado o suporte para validação do ambiente.

O que eu faço com o evento?

Ação 1: Abrir um chamado no CMDB com severidade de acordo com o evento alarmado na Ferramenta de Monitoração, descrevendo no mesmo a mensagem recebida.

Ação 2: **ATENÇÃO!** Direcionar o chamado para a fila do Suporte Produção e **COMUNICÁ-LA - das 7h00min às 18h00min** , horário comercial.

Fora do horário acima:

Caso 1 e caso 2- Verificar documentação do JOB e seguir procedimentos descritos. Caso procedimentos sejam ineficientes, acionar Stand by da Equipe Produção.

Caso 3, caso 4 e caso 5 - Verificar se o FTA - Estação no Maestro - está unlinked ou se algum serviço do TWS está parado no servidor, e caso positivo executar procedimento de link do TWS conforme documentações: Para estações clusterizadas seguir o procedimento TWS - Link de FTA Clusterizada (estação) no maestro e para estações não-clusterizadas seguir o procedimento TWS - Link de estação de trabalho no TWS. Caso esteja Linked acionar Stand by da Equipe Produção.
Encaminhar e-mail de cancelamento dos backups ao time de STORAGE.

Caso 6 - Verificar documentação do JOB e seguir procedimentos descritos. Caso procedimentos sejam ineficientes, acionar stand-by da Equipe Produção.

Caso 7 - A Operação deve, a partir do alarme, dar um KILL via Console na rotina e após SUCCESS para liberar a rotina posterior.
OBS: Se o problema não for resolvido com a ação, deve-se passar para a fila de PRODUCT.

Em caso de severidade 1, acionar o suporte responsável imediatamente e enviar mensagem informativa do problema para o grupo "Gerência" cadastrados no Sistema de Gerenciamento de Mensagens.

Ação 1: Abrir um chamado no CMDB com severidade de acordo com o evento alarmado na Ferramenta de Monitoração, descrevendo no mesmo a mensagem recebida.

Em caso de severidade 2, fora do horário comercial acionar o suporte responsável.

Caso 8 - Abrir um chamado no CMDB com severidade de acordo com o evento alarmado na Ferramenta de Monitoração, para a fila de STORAGE

Caso 9 - Abrir um chamado no CMDB com severidade de acordo com o evento alarmado na Ferramenta de Monitoração, para a fila de PRODUCT

Caso 10 - Abrir um chamado no CMDB com severidade de acordo com o evento alarmado na Ferramenta de Monitoração, para a fila de PRODUCT e acionar o analista em Stand-by da Produção.
Informação adicional: Telefone da Empresa Externa: 0800-xxx xxxx

Caso 11 - Abrir um chamado no CMDB com severidade de acordo com o evento alarmado na Ferramenta de Monitoração, para a fila de STORAGE

Caso 12 - Abrir um chamado no CMDB com severidade de acordo com o evento alarmado na Ferramenta de Monitoração, para a fila de BASIS e acionar o analista em Stand-by

Figura 5.1: Documentação para atendimento a incidentes da classe *Jobs*.

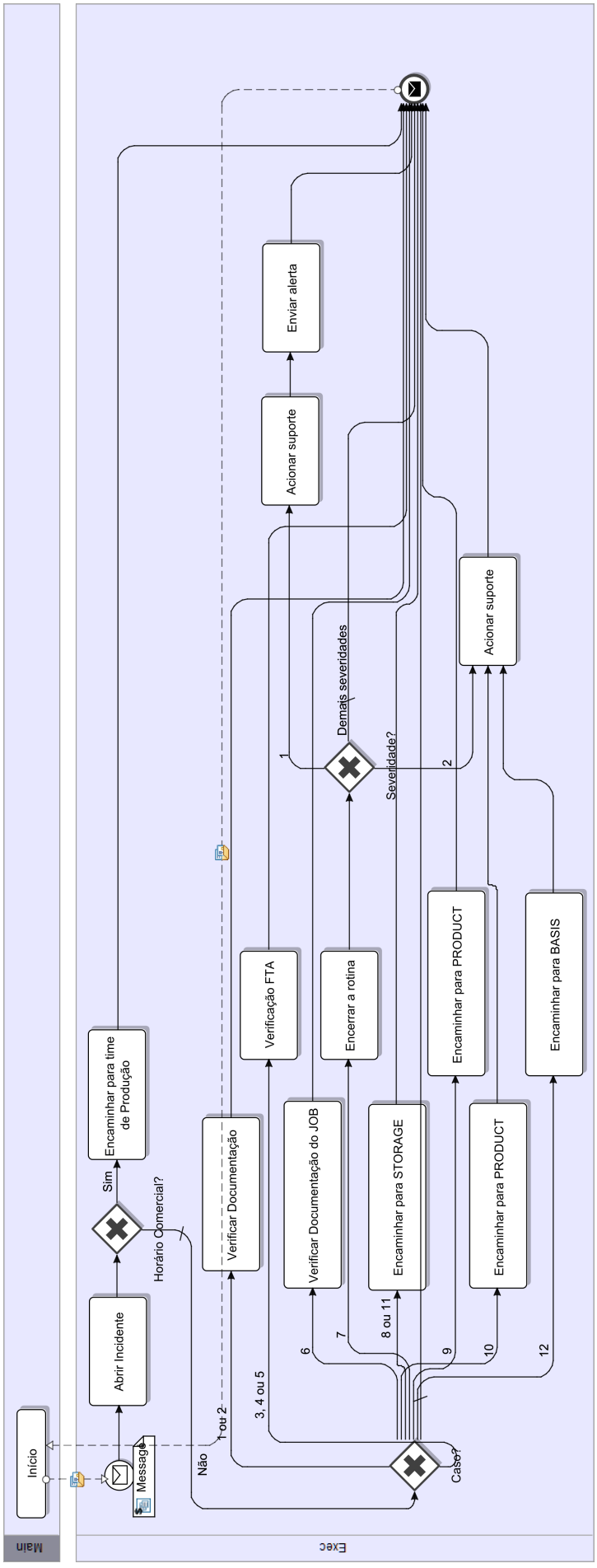

Figura 5.2: *Workflow* representado no Intalio referente à classe *Jobs*.

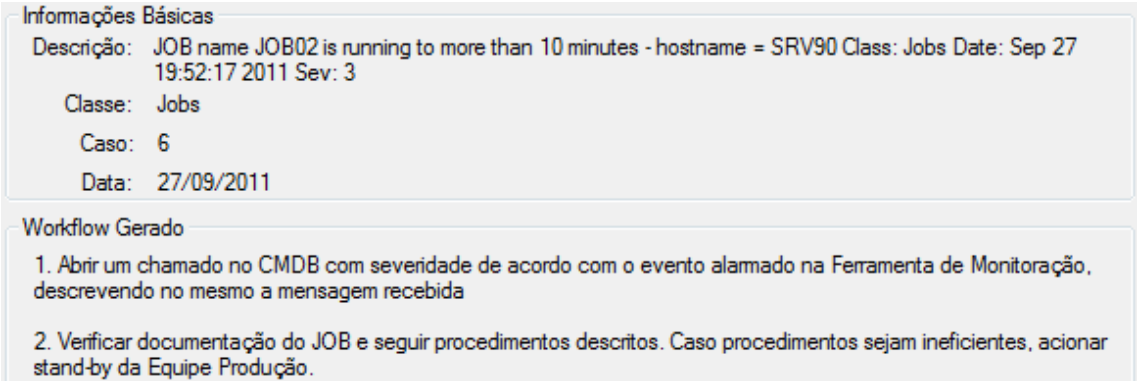

Figura 5.3: *Workflow* gerado para incidente da classe *Jobs* (caso 6).

Na sequência, a "ação 2" indica que o incidente deve receber um tratamento diferenciado baseado no horário em que o problema ocorre. Novamente, cabe ao operador decidir se o horário em que o incidente está sendo analisado enquadra-se no período atendido pela equipe. Para o incidente em análise, o horário "19:52:17" é fora do horário comercial e, portanto, a análise prossegue. A próxima ação que o técnico realiza depende do caso identificado anteriormente. Considerando o caso 6, verifica-se que o comando descrito é: "*Verificar documentação do JOB e seguir procedimentos descritos. Caso procedimentos sejam ineficientes, acionar stand-by da Equipe Produção*". Nesse caso, o operador deve consultar o catálogo de documentações das rotinas para dar continuidade ao atendimento[1].

Revisando as ações desempenhadas pelo técnico no atendimento deste incidente, pode-se resumir que os seguintes passos foram realizados:

1. Identificação da classe do incidente;

2. Reconhecimento do padrão da mensagem do incidente para identificação do caso, na documentação dos procedimentos;

3. Registro do incidente no CMDB;

4. Determinação do horário da análise, para verificação de horário comercial;

5. Verificação da documentação da rotina e tratamento.

Dentre os itens acima, apenas (3) e (5) são ações que efetivamente devem ser executadas pelo técnico. Os itens (1), (2) e (4) são verificações passíveis de automatização que, se mal avaliadas pelo técnico, implicam no mínimo perda de tempo no atendimento, podendo até resultar na execução de procedimentos incorretos.

Neste mesmo cenário, utilizou-se o protótipo da solução para realização do pré-atendimento do incidente. Devido à automatização das verificações supracitadas, o procedimento gerado e exibido para o técnico torna-se mais conciso, conforme pode ser observado na Figura 5.3. Nesta ocasião, somente as ações de registro de incidente no CMDB e verificação de documentação da rotina são exibidas para o operador.

[1] O catálogo de rotinas possui informações do funcionamento das rotinas, bem como descrição das possíveis falhas e procedimentos para sua correção. Foge do escopo deste trabalho uma análise aprofundada deste recurso, mas uma integração visando reduzir ainda mais o esforço de atendimento deste tipo de incidente é proposta como um trabalho futuro.

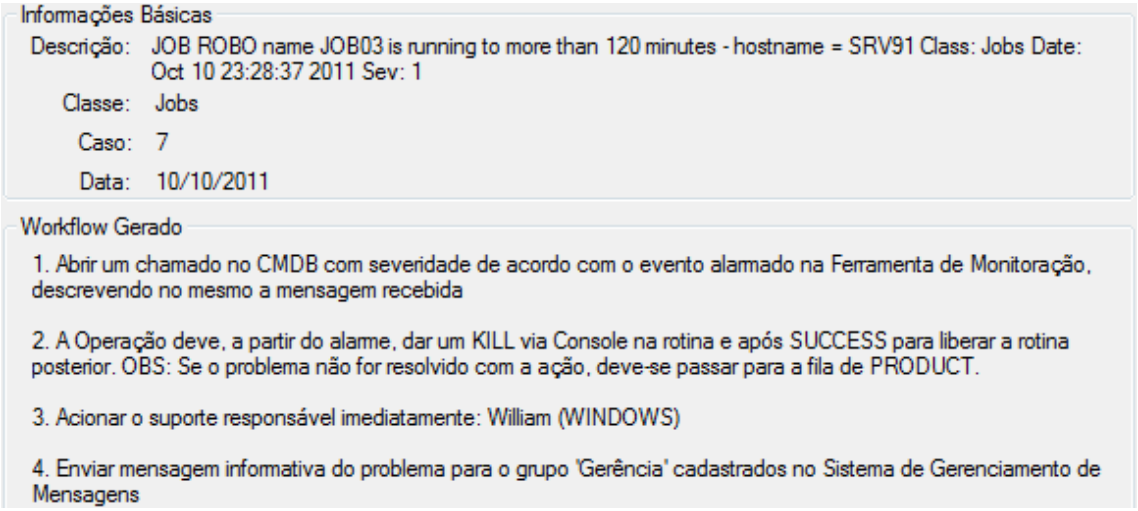

Figura 5.4: *Workflow* gerado para incidente da classe *Jobs* (caso 7).

5.1.2 Segundo Caso

O atendimento manual do caso (2) da Listagem 5.1 é semelhante ao efetuado para o caso (1), apresentado em detalhes na Seção 5.1.1 e, por isso, não será analisado no mesmo nível de detalhamento. Inicialmente, o operador identifica a sua classe (*Jobs*) e, em um segundo momento, determina qual o caso correspondente da mensagem (caso 7). Novamente, analisa se o horário do atendimento é dentro do período comercial e, considerando que é um caso negativo, verifica a próxima ação, que depende do caso determinado para o incidente. Para esta situação, um procedimento é indicado ao operador para forçar o encerramento da rotina em execução. O próximo passo, de acordo com a documentação, é verificar a severidade do evento alarmado; nesta instância, como a severidade tem o valor "1", é necessário proceder com a notificação dos responsáveis. Isto significa consultar a equipe responsável pelo ativo afetado e, posteriormente, o técnico responsável pela área.

Durante a avaliação do caso (2), tal como pôde ser observado em (1), existem etapas de verificações que, se automatizadas, minimizam as possibilidades de erros no processo. Desta forma, verificações como horário de atendimento e severidade do incidente são resolvidas automaticamente, não somente aumentando a agilidade do processo, como também a sua robustez. Assim, torna-se possível apresentar um fluxo conciso para o operador, conforme pode ser visto na Figura 5.4.

5.2 Análise Quantitativa

Com o intuito de aferir melhor os resultados obtidos, complementou-se a análise qualitativa da solução proposta com uma análise quantitativa. Neste contexto, utilizou-se a métrica de tempo de atendimento por incidente.

Para avaliar o tempo despendido no processamento dos incidentes e realizar a comparação entre o processo manual e o automatizado, foi necessário estabelecer parâmetros tangíveis para a métrica. Foram definidos dois parâmetros principais: número de condições avaliadas por fluxo e número de consultas externas. Para cada classe, foram estudados os caminhos mínimo e máximo, ou seja, os caminhos que percorrem respectivamente o menor e o maior número de condições e consultas externas. A Tabela 5.2 resume os

Tabela 5.2: Caminhos mínimo e máximo no percorrimento de *workflows*.

Classe	Caminho mínimo		Caminho máximo	
	Condições avaliadas	Consultas externas	Condições avaliadas	Consultas externas
Jobs	1	1	3	3
Logical_Disk	2	1	5	5
Node_Status	1	1	3	3
Processor_Summary	2	3	3	4
Services	2	3	4	3

valores verificados para as classes em análise.

Para ilustrar o preenchimento da tabela, a classe *Logical_Disk* será utilizada como exemplo, conforme o *workflow* definido na Figura 5.5. No caminho mínimo para esta classe, somente as verificações correspondentes à responsabilidade do ativo[2] (quando é do cliente) e à severidade (caso diferente de "1" e "2") são realizadas. Neste caso, a única consulta externa envolvida é para a determinação da responsabilidade do ativo. Já o caminho máximo desta classe ocorre quando o ativo é de responsabilidade da empresa prestadora de serviços, o incidente é do caso 3 e a severidade é 2; além da avaliação destas condições, ainda é verificado se o ativo está localizado em unidade remota e se o horário de atendimento é comercial, totalizando 5 condições. No caminho máximo desta classe, também são 5 as consultas externas realizadas: referente à responsabilidade do ativo, à sua localização, ao horário de atendimento do time responsável pelo ativo, à equipe e à pessoa que está de *stand-by*[3] no momento.

Com o intuito de analisar os resultados trazidos pelas automatizações propostas neste trabalho, fez-se necessário quantificar o tempo envolvido em cada passo que será otimizado. Com base na experiência dos operadores do ambiente em análise, foi estimado que a determinação do resultado de uma decisão consome até 10 segundos de análise, enquanto que a consulta a fontes externas requerem entre 30 e 60 segundos. Com o intuito de avaliar a solução de maneira conservadora, as estimativas de redução de tempo detalhadas a seguir foram realizadas considerando 5 segundos para tomadas de decisão e 30 segundos para consultas externas. Sendo assim, pode-se considerar que cada passo suprimido contribui para a redução do tempo total de atendimento do incidente. O tempo de redução (t_r) pode ser calculado para cada classe de incidentes, de maneira simplificada, segundo a expressão:

$$\begin{aligned} t_r = {} & 5 * numero\_condicoes\_removidas \\ & + 30 * numero\_consultas\_externas\_removidas \end{aligned} \tag{5.1}$$

A título de exemplo, no caminho máximo da classe *Logical_Disk*, reduziu-se em cinco condições avaliadas e cinco consultas externas. Com isso, o tempo reduzido no processo é de $5*5+30*5$, ou seja, 175 segundos. Aplicando a mesma fórmula aos valores determinados na Tabela 5.2, é possível obter uma projeção do tempo reduzido nos atendimentos, de acordo com a classe do incidente. O resultado desta projeção é exibido Tabela 5.3.

[2]É possível que, mesmo em um contexto de terceirização da infraestrutura de TI, alguns ativos estejam sob a gestão do cliente. Este normalmente é o caso de sistemas legados ou específicos de alguma unidade de planta de indústria.

[3]Jargão utilizado para designar o técnico responsável pelo atendimento daquela área fora do horário comercial.

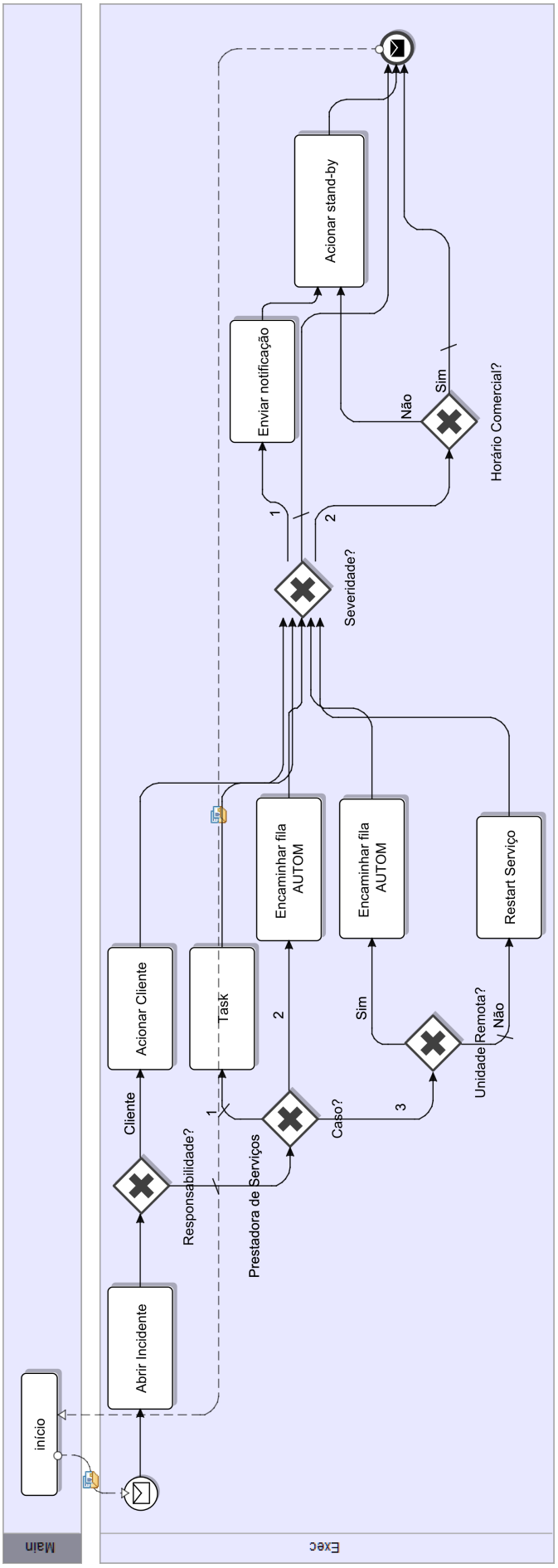

Figura 5.5: *Workflow* de atendimento a incidentes da classe *Logical_Disk*.

Tabela 5.3: Tempo reduzido no atendimento a incidentes.

Classe	Tempo reduzido (s)	
	Caminho mínimo	Caminho máximo
Jobs	35	105
Logical_Disk	40	175
Node_Status	35	105
Processor_Summary	100	135
Services	100	110

Considerando os resultados obtidos, verifica-se que a classe *Logical_Disk* é a que possui uma maior redução no caminho máximo. Este fato é muito interessante, pois conforme a Tabela 5.1 esta classe corresponde a 7.753 incidentes da base estudada. Por outro lado, comparando-se apenas os tempos de redução para cada classe, pode-se pensar que a classe *Jobs* é uma das menos beneficiadas pela solução proposta. Neste caso, é imprescindível recordar que esta classe é a mais ofensora na base de dados considerada. Portanto, mesmo que os ganhos estejam no intervalo entre 35 e 105 segundos (vide Tabela 5.3), cada um dos seus 21.363 incidentes poderia ter sido beneficiado por esta redução. Isto significa que, apenas para esta classe, entre 200 e 600 horas poderiam ter sido economizadas do tempo de técnicos durante o período em que os incidentes foram gerados.

Com o objetivo de analisar o ganho trazido pela solução proposta, as informações de números de incidentes foram cruzadas com o tempo de redução por classe. Dessa forma, a Tabela 5.4 ilustra o potencial ganho, considerando apenas a métrica de tempo, que, acredita-se, poderia ter sido obtido no período analisado.

Tabela 5.4: Potencial de redução de tempo, considerando incidentes registrados na base.

Classe	Ganho Mínimo (h)	Ganho Máximo (h)
Jobs	208	623
Logical_Disk	86	377
Node_Status	76	227
Processor_Summary	49	67
Services	202	222
Total	**621**	**1516**

Com base nos resultados da tabela, é possível inferir que há espaço para inclusão de uma ferramenta com características semelhantes àquelas proposta neste trabalho. Considerando o universo de incidentes analisados, especula-se um ganho mínimo de 600 horas de trabalho de técnicos de suporte, sendo que este número pode ser até 2,5 vezes maior, devido às diferentes situações compreendidas em cada incidente. Conforme destacado anteriormente, a abordagem seguida foi bastante conservadora no que tange aos tempos de determinação de condições e consultas, uma vez que considerou-se o tempo mínimo que o operador levaria para cada uma destas atividades. Entretanto, mesmo com esta abordagem, é possível perceber o potencial de redução de tempo de atendimento a incidentes que a solução oferece. Além disso, a eliminação das condições e consultas externas agrega robustez ao serviço prestado pelos técnicos, uma vez que tem potencial para diminuir a ocorrência de falhas humanas.

6 CONCLUSÕES E TRABALHOS FUTUROS

No início deste trabalho, foi apresentado um ambiente corporativo típico de TI e alguns dos processos que auxiliam na manutenção da sua estabilidade. Dentre os processos abordados, o gerenciamento de incidentes destaca-se pela importância em alertar que algum subsistema não está desempenhando conforme o esperado. Apesar destes alertas poderem ser reportados por usuário, uma grande parcela deste total é gerada por sistemas de monitoração, que indicam falha ou eminência de falha. Desta forma, percebe-se a necessidade de uma equipe focada em atuar na linha de frente destes problemas, agindo assim que o alerta é recebido.

A complexidade de ambientes de TI implica que, não raramente, empresas possuem dezenas ou centenas de ativos para gerenciar. Acompanhando este número, também a quantidade de incidentes pode chegar a ordem de centenas por dia. Com isso, é compreensível que a carga sobre os técnicos da primeira linha de atendimento seja alta. Com o intuito de auxiliar nas ações destes técnicos durante o atendimento a incidentes, o trabalho proposto visa fornecer um conjunto de ferramentas que possibilite reduzir o tempo de atendimento, através da automatização de tomadas de decisões. A proposta foi implementada na forma de um protótipo, que foi utilizado em experimentos sobre uma base de incidentes de um ambiente real de TI. Para isso, alguns ajustes de pré-requisitos tiveram que ser satisfeitos.

Considerando uma infraestrutura pré-existente, o primeiro passo é analisar como os procedimentos de atendimento a incidentes são organizados. Supõe-se que empresas que prestam suporte de TI para operações de negócios já tem uma documentação organizada, segundo algum critério[1]. Usualmente, estas são descritas em linguagem natural, contendo perguntas e ordens para o humano que a estiver lendo. No cenário no qual esta solução foi testada, os procedimentos relativos a cada classe de incidentes estavam organizados na forma de itens, em documentos de texto, armazenados na forma de uma página HTML (*HyperText Markup Language*). Um exemplo deste documento foi exibido na Figura 5.1.

Com base no documento de procedimentos, o primeiro desafio foi referente à tradução dos comandos para atividades de um *workflow*. Para isso, cada comando teve que ser analisado e categorizado entre condicional e tarefa. Para o tratamento de condicionais, embora algumas automatizações tenham sido realizadas (como consultas a fontes externas), outras ainda são passíveis de melhorias. Condicionais que dependem de interação com alguma interface gráfica, na qual um humano determina visualmente o *status* de um processo, foram mantidos como atividades ao operador no escopo deste trabalho. Além

[1] Apesar da suposição, não há prejuízos em realizar o primeiro mapeamento destes procedimentos junto com a implementação de uma solução tal como a proposta neste trabalho. Nestes casos, inclusive é interessante fazê-lo já estudando formas de automatizar os processos, dispensando as etapas de tradução dos procedimentos para representações digitais.

disso, a solução foi baseada no refino de *workflows* para geração de procedimentos mínimos; entretanto, a responsabilidade de executá-los ainda é do operador. Por exemplo, o reinício de um serviço de Sistema Operacional é, atualmente, uma tarefa repassada para o operador; com os recursos de integração oferecidos pelos sistemas, é possível que um mecanismo computacional controle esta solicitação. Desta forma, observa-se que há espaços para aprimoramentos da solução apresentada. Em última instância, uma abordagem audaciosa seria o estudo da eliminação da intervenção humana no processo, mostrando que esta é uma área muito interessante de ser explorada.

Um segundo desafio relacionado ao mapeamento dos procedimentos manuais foi lidar com as ambiguidades e imprecisões que a linguagem escrita causa. Dado que os procedimentos escritos em linguagem natural não são estruturados de forma a garantir a unicidade da interpretação, duas pessoas lendo o mesmo procedimento poderiam ter interpretações – e, consequentemente, ações – diferentes. No procedimento exibido na Figura 5.1, por exemplo, é possível perceber que as ações relacionadas com a severidade do incidente são descritas logo após o caso 7. Na interpretação utilizada neste trabalho, as ações de severidade foram aplicadas somente para o caso 7. Entretanto, é possível que em outra interpretação, estas ações pudessem ser consideradas para execução para todos os casos anteriores.

Durante a fase de implementação, foi necessário estabelecer uma forma de avaliação dos atributos. O protótipo foi implementado utilizando o conceito de avaliação ansiosa (*eager evaluation*), em que os valores dos atributos de uma classe são avaliados assim que são descobertos. Desta forma, ainda na etapa de processamento de incidentes, todos os valores são recuperados, e quando o processo atinge a etapa de refino de *workflows*, estes já são conhecidos. Como consequência, é possível que nem todos os atributos sejam utilizados, devido aos diferentes caminhos dos fluxos. Para resolver este problema, seria possível explorar, em trabalhos futuros, uma alternância entre etapas de processamento e refino, de modo que os atributos pudessem ser avaliados à medida que fossem necessários. Este problema já pode ser constatado na Tabela 5.2, que mostra classes com números de consultas externas distintos para caminhos diferentes.

Embora entenda-se que há uma série de possibilidades de melhorias e extensão da solução apresentada, a concepção atual, por si só, já traz uma série de benefícios, como destacou-se no Capítulo 5. A redução de tempo de solução dos incidentes e a potencialidade de melhoria na robustez do processo de gerenciamento de incidentes representam um avanço no processo de gestão. Além disso, a solução proposta foi desenhada para ser integrada a processos existentes. Por isso, é natural esperar que forneça e utilize recursos que possibilitem a extensão de suas capacidades. Uma utilização deste preceito pode ser observada no contexto dos atributos externos, que possibilitam que o administrador defina serviços externos (exemplificados na forma de *Web Services*) que sejam usados para fornecer informações ao sistema. De maneira análoga, o *software* Intalio, utilizado para definição dos *workflows* nesta implementação, também permite chamadas a *Web Services* para agregar valor ao processo – apesar deste recurso não ter sido explorado nos exemplos deste trabalho. Dessa forma, é possível alcançar extensibilidade tanto na camada de processamento de incidentes, quanto na de refino de *workflows*. Assim, sugere-se como trabalhos futuros a aplicação destes recursos para, além de automatizar consultas de atributos, também aumentar o escopo de ações executadas automaticamente.

REFERÊNCIAS

BARTOLINI, C.; DAY, P. Where have all the tickets gone? In: CNSM. **Anais...** IEEE, 2010. p.41–47.

BARTOLINI, C.; STEFANELLI, C.; TORTONESI, M. Business-impact analysis and simulation of critical incidents in IT service management. In: INTEGRATED NETWORK MANAGEMENT, 2009. IM '09. IFIP/IEEE INTERNATIONAL SYMPOSIUM ON. **Anais...** [S.l.: s.n.], 2009. p.9 –16.

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.

COSTA CORDEIRO, W. L. da et al. ChangeLedge: change design and planning in networked systems based on reuse of knowledge and automation. **Computer Networks**, [S.l.], v.53, n.16, p.2782 – 2799, 2009.

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.; PRASAD, K. H.; MOHANIA, M. Automating ITSM Incident Management Process. **Autonomic Computing, International Conference on**, Los Alamitos, CA, USA, v.0, p.141–150, 2008.

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).

VAN BONI, J. **ITIL**: guia de referência, edição 2011. [S.l.]: CAMPUS - RJ, 2012.