

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
INSTITUTO DE INFORMÁTICA  
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

ROBEN CASTAGNA LUNARDI

**CHANGEADVISOR: Uma Solução para  
Alinhamento de Planos de Mudanças em  
Infraestruturas de TI a Objetivos/Restrições  
de Negócios**

Dissertação apresentada como requisito parcial  
para a obtenção do grau de Mestre em Ciência  
da Computação

Prof. Dr. Luciano Paschoal Gasparry  
Orientador

Porto Alegre, Setembro de 2010.

## CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Lunardi, Roben Castagna

Normas para Apresentação de Dissertações do Instituto de Informática e do PPGC / Roben Castagna Lunardi – Porto Alegre: Programa de Pós-Graduação em Computação, 2010.

95 f.:il.

Dissertação (mestrado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação. Porto Alegre, BR – RS, 2010. Orientador: Luciano Paschoal Gasparry.

1.ABNT. 2. Processadores de Texto 3.Formatção eletrônica de documentos. I. Gasparry, Luciano Paschoal. II. Título.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Carlos Alexandre Netto

Vice-Reitor: Prof. Rui Vicente Oppermann

Pró-Reitor de Pós-Graduação: Prof. Aldo Bolten Lucion

Diretor do Instituto de Informática: Prof. Flávio Rech Wagner

Coordenador do PPGC: Prof. Álvaro Freitas Moreira

Bibliotecária-Chefe do Instituto de Informática: Beatriz Regina Bastos Haro

*“...And be a simple kind of man  
Oh be something you love and understand...”*  
(Lynyrd Skynyrd)

-  
*“...We can change the world  
We gotta change the way we think about it...”*  
(Chickenfoot)

## AGRADECIMENTO

Primeiramente agradeço aos meus pais e irmãos pelo exemplo, educação e ensinamentos. Com eles tive a oportunidade de entender o verdadeiro conceito de família e união. Sem o incentivo, apoio e carinho deles não poderia chegar onde estou. Sem esquecer meus avós que, com sua experiência, mostraram-me como viver dignamente.

Após estes, mas não menos importantes, agradeço ao apoio de todos os integrantes e colegas do Grupo de Redes da UFRGS, em especial: ao Professor Luciano Paschoal Gaspary por toda a orientação e ensinamentos ao longo do mestrado (fazer um “ogro” aprender a prestar um pouco mais de atenção em detalhes não foi nem um pouco fácil, garanto!); ao Juliano Araujo Wickboldt, colega de longas jornadas noturnas (e até mesmo dominicais) no laboratório (e viva ao *cup noodles*), parceiro de diversas discussões sobre a pesquisa ou não (futebol, carros, cerveja, dentre outros), e companheiro em longas viagens (em todos os sentidos); ao Weverton Luis da Costa Cordeiro (vulgo Mocorongo), que perdeu horas discutindo formas de escrever os artigos e de descrever algoritmos (a culpa do “clusterizer” estar aqui hoje é dele!); ao Fabrício Girardi Andreis, “brother” que, sempre na “correria”, esteve presente implementando uma última funcionalidade (sempre existe mais uma última funcionalidade) antes do *deadline* de algum evento ou discutindo, nos 45 do segundo tempo, algum caso em que o algoritmo poderia não gerar resultados bons; ao Guilherme Sperb Machado, que diretamente da “(Res)Tinga” foi o primeiro a me ensinar as gírias porto-alegrenses (“deu os doce”, “O meu, tá ligado?!”, ahh... e, é claro, o seu “*handshake* infinito”), além de discutir assuntos relacionados a pesquisa; ao Alan Diego dos Santos, o “pirulito”, que sempre surgia com idéias malucas e histórias inacreditáveis (o apelido tem significado...); ao professor Lisandro Zambenedetti Granville, o “mestre dos magos”, que após os seus nobres conselhos (com ou sem sentido) sumia sem deixar rastro; aos “guris” Bruno Lopes Dalmazo, Ricardo Luis dos Santos e Luis Bianchin (Nautilus), que mesmo com pouca “experiência” puderam contribuir para o trabalho e manter o bom humor no laboratório; aos gerentes de projeto Cristiano Bonato Both e Abraham Lincoln Rabelo que sempre acharam um jeito de fazer com que a bolsa entrasse na conta (eu sei que isso não foi nada fácil...).

Agradeço, ainda, o Professor Raul Ceretta Nunes, meu orientador durante a graduação na UFSM, de fundamental importância para eu ter continuado na área acadêmica. E Koiti Ozaki e Nelson Jorge Schuch, meus orientadores no CRS/INPE, pelos meus primeiros passos no mundo científico.

Com carinho agradeço a todos os amigos que me apoiaram na minha trajetória, e em especial à Aline Ibaldo Gonçalves pelo carinho, compreensão, companheirismo e, é claro, por agüentar noites de “roncos” fortes.

## SUMÁRIO

<b>LISTA DE ABREVIATURAS E SIGLAS.....</b>	<b>7</b>
<b>LISTA DE FIGURAS .....</b>	<b>8</b>
<b>LISTA DE TABELAS .....</b>	<b>9</b>
<b>RESUMO .....</b>	<b>10</b>
<b>ABSTRACT.....</b>	<b>11</b>
<b>1 INTRODUÇÃO .....</b>	<b>12</b>
<b>2 REVISÃO BIBLIOGRÁFICA .....</b>	<b>14</b>
<b>2.1 Gerenciamento de Mudanças .....</b>	<b>16</b>
<b>2.2 Alocação de humanos .....</b>	<b>17</b>
<b>2.3 Síntese comparativa.....</b>	<b>18</b>
<b>3 SOLUÇÃO CONCEITUAL .....</b>	<b>20</b>
<b>3.1 <i>Change Refiner</i>.....</b>	<b>21</b>
<b>3.2 <i>Change Aligner</i>.....</b>	<b>25</b>
3.2.1     Objetivos / restrições de negócio e métricas de custos / capacidades ....	25
3.2.2     Algoritmo de alinhamento a objetivos/restrições de negócio.....	26
<b>3.3 Resumo do Capítulo .....</b>	<b>28</b>
<b>4 ESTRATÉGIAS PARA O ALINHAMENTO DE PLANOS DE MUDANÇA   A OBJETIVOS/RESTRICÇÕES DE NEGÓCIO .....</b>	<b>30</b>
<b>4.1 Alocação do melhor para o pior humano disponível.....</b>	<b>31</b>
<b>4.2 Alocação do melhor para o pior humano disponível com ordenação das     atividades.....</b>	<b>32</b>
<b>4.3 Alocação do melhor para o pior humano disponível com ordenação de     atividades por <i>clusters</i> .....</b>	<b>34</b>
<b>4.4 Alocação de humanos baseada nas preferências do Gerente de TI .....</b>	<b>35</b>
<b>4.5 Alocação de humanos baseada nas preferências do Gerente de TI com     ordenação de atividades .....</b>	<b>36</b>
<b>4.6 Alocação de humanos baseada nas preferências do Gerente de TI com     ordenação de atividade por <i>Clusters</i> .....</b>	<b>36</b>
<b>4.7 Resumo do Capítulo .....</b>	<b>37</b>

<b>5</b>	<b>O SISTEMA CHANGEADVISOR .....</b>	<b>38</b>
<b>5.1</b>	<b>Interface gráfica do CHANGEADVISOR.....</b>	<b>39</b>
<b>5.2</b>	<b>Resumo do Capítulo .....</b>	<b>43</b>
<b>6</b>	<b>AVALIAÇÃO EXPERIMENTAL.....</b>	<b>44</b>
<b>6.1</b>	<b>Primeiro estudo de caso .....</b>	<b>44</b>
<b>6.2</b>	<b>Segundo estudo de caso .....</b>	<b>47</b>
<b>6.3</b>	<b>Resumo do Capítulo .....</b>	<b>50</b>
<b>7</b>	<b>CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS.....</b>	<b>51</b>
<b>7.1</b>	<b>Resultados obtidos .....</b>	<b>51</b>
<b>7.2</b>	<b>Questões em aberto e possíveis investigações futuras .....</b>	<b>52</b>
	<b>REFERÊNCIAS.....</b>	<b>54</b>
	<b>ANEXO A ARTIGO PUBLICADO NO SBRC 2009.....</b>	<b>57</b>
	<b>ANEXO B ARTIGO PUBLICADO NO DSOM 2009.....</b>	<b>72</b>
	<b>ANEXO C ARTIGO PUBLICADO NO NOMS 2010.....</b>	<b>87</b>

## LISTA DE ABREVIATURAS E SIGLAS

AMN	<i>Activity Modeling Notation</i>
BPEL	<i>Business Process Execution Language</i>
CAB	<i>Change Advisory Board</i>
CHAMPS	<i>Change Management with Planning and Scheduling</i>
CI	<i>Configuration Item</i>
CIM	<i>Common Information Model</i>
CMS	<i>Configuration Management System</i>
DBMS	<i>Database Management System</i>
DML	<i>Definitive Media Library</i>
DMTF	<i>Distributed Management Task Force</i>
DNS	<i>Domain Name System</i>
ISO	<i>International Organization for Standardization</i>
ITIL	<i>Information Technology Infrastructure Library</i>
ITSM	<i>Information Technology Service Management</i>
OASIS	<i>Advancing Open Standards for the Information Society</i>
OGC	<i>Office of Government Commerce</i>
P & S	<i>Planner &amp; Scheduler</i>
QoS	<i>Quality of Service</i>
RFC	<i>Requests for Change</i>
SLA	<i>Service Level Agreement</i>
StAX	<i>Streaming API for XML</i>
TI	Tecnologia da Informação
TGB	<i>Task Graph Builder</i>
WfMC	<i>Workflow Management Coalition</i>
XML	<i>Extended Markup Language</i>

## LISTA DE FIGURAS

<i>Figura 2.1. Processos do ITIL classificados por escopo de atuação.....</i>	<i>14</i>
<i>Figura 3.1. Arquitetura conceitual da solução proposta.....</i>	<i>20</i>
<i>Figura 3.2. Modelo usado para a representação da infraestrutura de TI e da DML....</i>	<i>22</i>
<i>Figura 3.3. Instâncias de software (SoftwareElement) presentes na DML e de Papéis (Roles) presentes na infraestrutura de TI.....</i>	<i>23</i>
<i>Figura 3.4. Exemplo de RFC e de plano preliminar de mudança para migração de servidor de e-mail e respectivo front-end Web.....</i>	<i>23</i>
<i>Figura 3.5. Algoritmo para geração de planos de mudança alinhados a objetivos/restrições de negócio.....</i>	<i>26</i>
<i>Figura 3.6. Exemplo de plano de mudança detalhado para RFC de migração de servidor de e-mail.....</i>	<i>28</i>
<i>Figura 4.1. Exemplo genérico de plano de mudança refinado C.....</i>	<i>31</i>
<i>Figura 4.2. Algoritmo de criação de clusters.....</i>	<i>34</i>
<i>Figura 5.1 Interface gráfica do ChangeAdvisor.....</i>	<i>39</i>
<i>Figura 5.2 Seleção de RFC na interface do ChangeAdvisor.....</i>	<i>39</i>
<i>Figura 5.3 Seleção de objetivo/restrição de negócio no ChangeAdvisor.....</i>	<i>40</i>
<i>Figura 5.4 Definição do valor máximo em minutos e em dólares no ChangeAdvisor ..</i>	<i>40</i>
<i>Figura 5.5 Escolha da estratégia de alocação de humanos a ser adotada no ChangeAdvisor.....</i>	<i>41</i>
<i>Figura 5.6 Escolha do número de humanos por papel para conduzir as atividades no ChangeAdvisor.....</i>	<i>41</i>
<i>Figura 5.7 Escolha do formato de saída e nível de detalhamento no ChangeAdvisor ..</i>	<i>41</i>
<i>Figura 5.8 Exemplo de plano de mudança alinhado gerado pelo ChangeAdvisor .....</i>	<i>42</i>
<i>Figura 6.1. Plano de mudança alternativo para RFC de migração de servidor de e-mail para o estudo de caso 1.....</i>	<i>45</i>
<i>Figura 6.2. Plano de mudança do Cenário 4 alinhado considerando a Estratégia 3 ...</i>	<i>49</i>
<i>Figura 6.3. Plano de mudança do Cenário 2 alinhado considerando a Estratégia 5 ...</i>	<i>49</i>



## LISTA DE TABELAS

<i>Tabela 3.1. Descrição da notação AMN .....</i>	<i>24</i>
<i>Tabela 4.1. Estratégias para o alinhamento de planos de mudança, classificadas de acordo com as heurísticas empregadas .....</i>	<i>31</i>
<i>Tabela 4.2. Exemplo genérico de papéis de humanos disponíveis em uma organização com os respectivos custos por hora.....</i>	<i>32</i>
<i>Tabela 6.1. Papéis disponíveis para alocação de humanos para os estudos de caso ...</i>	<i>44</i>
<i>Tabela 6.2. RFC e objetivos/restrições de negócio empregado no estudo de caso 1 ....</i>	<i>45</i>
<i>Tabela 6.3. Número de atividades, tempo estimado de execução e custos de alocação para os planos detalhados gerados no estudo de caso 1 .....</i>	<i>46</i>
<i>Tabela 6.4. Tempo despendido na geração automática dos planos detalhados e alinhados no estudo de caso 1.....</i>	<i>46</i>
<i>Tabela 6.5. Características dos planos detalhados avaliados no estudo de caso 2 .....</i>	<i>47</i>
<i>Tabela 6.6. Tempo total de execução e custo de alocação de humanos estimados para os planos detalhados avaliados no estudo de caso 2 .....</i>	<i>47</i>

## RESUMO

O planejamento de mudanças é um elemento-chave para a gerência e a operação de infraestruturas de redes e serviços. Ele compreende desde a especificação, em alto nível, de requisições e planos preliminares de mudança (por um operador da mudança) até a geração, possivelmente automatizada, de planos detalhados que, se executados, concretizarão as mudanças solicitadas (por exemplo, alteração da configuração de dispositivos de rede e implantação de novos serviços).

Por limitação das soluções de refinamento existentes, um problema fundamental é que, embora corretos, tais planos de mudança podem não necessariamente estar alinhados com os propósitos definidos em nível de negócio (ex: minimização do tempo de indisponibilidade de um dado serviço). Outro problema encontrado no planejamento de mudanças de TI é a ausência de modelos e algoritmos para realizar a especificação adequada da associação de humanos a tarefas. A correta especificação pode impactar em resultados mais otimizados (menor custo de execução) e com maior qualidade (conformidade com as habilidades/experiências, requeridas pelas atividades, e adquiridas pelos recursos humanos).

Para abordar estes problemas, nesta dissertação propõe-se uma solução para alinhar o planejamento de mudanças em infraestruturas de TI a objetivos/restrições de negócio, utilizando estratégias inovadoras de alocação de humanos a atividades. A solução proposta é avaliada experimentalmente por meio da implementação prototípica de um sistema de suporte à decisão denominado CHANGEADVISOR, o qual auxilia os gerentes de TI a entender o "compromisso" entre planos de mudança alternativos e a efetuar escolhas mais adequadas aos propósitos de negócio.

**Palavras-Chave:** Gerenciamento de Serviços de TI, Gerenciamento de TI Orientado a Negócios, Gerenciamento de Mudanças em TI.

# **CHANGEADVISOR: A Solution to Support Alignment of IT Change Design with Business Objectives/Constraints**

## **ABSTRACT**

Change planning represents a key element for the operation and management of Information Technology infrastructures and services. Its scope ranges from the high level design of change requests and change plans (by an operator) to the generation, possibly automated, of detailed plans that, if executed, will perform the requested changes (*e.g.*, modification of network device settings and deployment of new services).

A fundamental problem - considering the limitations of existing refinement solutions - is that, although correct, these refined change plans can not necessarily be aligned with the purposes defined in the business level (*eg*, minimizing the downtime of a particular service). Another problem encountered in the IT change planning is the absence of models and algorithms to achieve the correct association of human to tasks. These correct association can impact in results more optimized (lower cost of deployment) and with higher quality (according to the skills/experience required for the activities and acquired by human resources).

To overcome this problem, in this thesis we propose a solution for the alignment of change plans with business objectives/constraints using novel strategies for the human assignments to activities. The proposed solution is experimentally evaluated through a prototypical implementation of a decision support system called CHANGEADVISOR, which helps IT managers to understand the trade-offs between alternative change designs and to choose change plans more aligned to the business purposes.

**Keywords:** IT Service Management, Business-Driven IT Management, IT Change Management.

# 1 INTRODUÇÃO

O tamanho e a complexidade crescentes das infraestruturas de rede e serviços – também chamadas genericamente de infraestruturas de Tecnologia da Informação (TI) ao longo desta dissertação – demandam, por parte das organizações, a adoção de boas práticas e processos para assegurar o seu funcionamento correto e eficiente. Como exemplo de tamanho das infraestruturas de TI citadas, considere empresas de médio e grande porte, as quais possuem centenas de computadores (por exemplo, terminais de acesso à *internet*, estações de trabalho e servidores), dezenas de ativos de rede (por exemplo, *hubs*, *switches* e roteadores) e dezenas de *software*. Por sua vez, a complexidade elevada se deve ao fato do grande volume de informação a ser analisado inerente ao gerenciamento de todos esses dispositivos de *hardware* e *software*, e suas relações de dependência (ex: bibliotecas necessárias para instalação de um serviço de *e-mail* ou categoria de cabeamento adequada a ser utilizada em um roteador).

Com o objetivo de auxiliar as organizações nessa tarefa de gerenciar e otimizar recursos de infraestruturas de redes e serviços, a Biblioteca de Infraestrutura de Tecnologia da Informação (*Information Technology Infrastructure Library*, ITIL) (ITIL, 2010), uma das referências mais importantes neste contexto, apresenta um conjunto de boas práticas e processos para a implantação e a manutenção de infraestruturas de TI. Ciente dos altos custos que a condução de mudanças em infraestruturas de TI representa para as organizações, a biblioteca ITIL propõe o processo de Gerenciamento de Mudanças (*Change Management*), o qual auxilia no tratamento imediato e eficiente de mudanças que venham a ser necessárias na infraestrutura de *hardware* e *software* das organizações.

Considerado um dos elementos chave do referido processo, o *planejamento de mudanças* (ITIL: *Service Transition*, 2007) compreende desde a especificação da mudança por um solicitante, em alto nível de abstração (ex: ampliação da capacidade de canais de comunicação, migração de serviços, implantação de novas aplicações), até a geração, possivelmente automatizada, de planos com alto grau de detalhamento (isto é, planos que compreendem atividades de baixo nível como, por exemplo, manipulação de rotas em roteadores e modificação de políticas de *firewall*). Esses planos, se executados, concretizarão as mudanças especificadas pelo solicitante na infraestrutura gerenciada.

No contexto de planejamento de mudanças, diversas pesquisas foram conduzidas, no passado recente com o objetivo de automatizar a geração de planos de mudanças (CORDEIRO *et al.*, 2008a) (CORDEIRO *et al.*, 2008b) (KELLER *et al.*, 2004). Porém, um problema fundamental até então não considerado acerca da geração dos planos de mudança é que, a partir de uma mesma especificação em alto nível, podem ser obtidos diferentes planos detalhados. Embora corretos, esses planos não necessariamente estão

alinhados com os propósitos definidos em nível de negócio. Os propósitos podem ser de natureza técnica, tais como a imposição de *restrições* quanto à indisponibilidade de serviços providos pela infraestrutura gerenciada, ou podem definir *objetivos* para otimizar a alocação de recursos durante a execução da mudança (por exemplo, minimizar o número de operadores envolvidos na manutenção de um serviço de *e-mail*). Sem levar em conta tais considerações, os planos de mudança gerados podem levar a resultados que violam políticas pré-definidas no nível de negócio (por exemplo, paradas para manutenção no serviço de *e-mail* não devem exceder 10 horas mensais).

Destacam-se três motivos que justificam a importância de alinhar planos de mudança em infraestruturas de TI com propósitos de negócio. Primeiro, existe a possibilidade de otimização no uso dos recursos disponíveis, que normalmente são escassos e/ou caros. Segundo, tal alinhamento tem potencial para contribuir com a redução de custos como, por exemplo, o tempo despendido para completar uma mudança. Terceiro – e mais importante – a possibilidade de alinhar o planejamento de mudanças a propósitos distintos permite ao gerente de TI entender os “compromissos” resultantes ao optar por um plano de mudança dentre as opções possíveis.

Diante do exposto, nesta dissertação propõe-se uma solução para alinhar o planejamento de mudanças em infraestruturas de TI com propósitos definidos em nível de negócio. Em contraste com as propostas já estabelecidas na literatura, o foco desta dissertação concentra-se na geração automatizada (ou com participação humana reduzida na geração) de planos de mudança guiada pelos objetivos/restrições definidos durante a especificação da mesma. Para tal, são propostos seis algoritmos para o planejamento da alocação de recursos humanos a atividades de mudança. A solução proposta é avaliada experimentalmente por meio do CHANGEADVISOR, uma implementação prototípica de um sistema de suporte à decisão que fornece apoio para o entendimento dos “compromissos” entre planos de mudança alternativos. Durante a avaliação – realizada utilizando cenários baseados em casos reais – foram consideradas a correção e a completude dos planos gerados, bem como o alinhamento dos mesmos aos propósitos de negócio definidos na fase de especificação das mudanças.

O restante desta dissertação está organizado como segue. O Capítulo 2 discute alguns dos principais trabalhos publicados na área de Gerenciamento de Mudanças em infraestruturas de TI. O Capítulo 3 apresenta a solução conceitual para a geração de planos de mudanças alinhados a propósitos de negócio. O Capítulo 4 discute seis diferentes algoritmos de alocação de recursos humanos a atividades de mudança, bem como as suas diferenças e peculiaridades. O capítulo 5 detalha o protótipo CHANGEADVISOR desenvolvido no contexto desta dissertação, enquanto o Capítulo 6 descreve a avaliação experimental conduzida utilizando esse sistema. Por fim, o Capítulo 7 conclui a dissertação com as considerações finais e perspectivas de trabalhos futuros.

## 2 REVISÃO BIBLIOGRÁFICA

A grande área de gerenciamento de infraestruturas de TI tem recebido grande atenção da comunidade científica nos últimos anos. Conforme mencionado na introdução, a biblioteca ITIL (ITIL, 2010) é um dos maiores pilares da área. Cinco livros (ITIL: Service Strategy, 2007) (ITIL: Service Design, 2007) (ITIL: Service Transition, 2007) (ITIL: Service Operation, 2007) (ITIL: Continual Service Improvement, 2007) cobrem todo o ciclo de vida de serviços de TI, propondo diversos processos para gerenciá-los.

Alguns dos diversos processos sugeridos pela biblioteca ITIL (principalmente os sugeridos nos livros *Service Transition* e *Service Operation*) estão destacados na Figura 2.1. Nesta figura, os processos estão separados pelo escopo de atuação, conforme descrito no padrão ISO/IEC 20000 (DUGMORE e JENNY, 2006). Por exemplo, os processos *Gerenciamento de Configuração* (*Configuration Management*) e *Gerenciamento de Mudanças* (*Change Management*) estão classificados como processos de controle (*Control Processes*). Por sua vez, o processo de *Gerenciamento de Releases e Implantação* (*Releases and Deployment Management*) é classificado como *Processo de Release* (*Release Processes*).

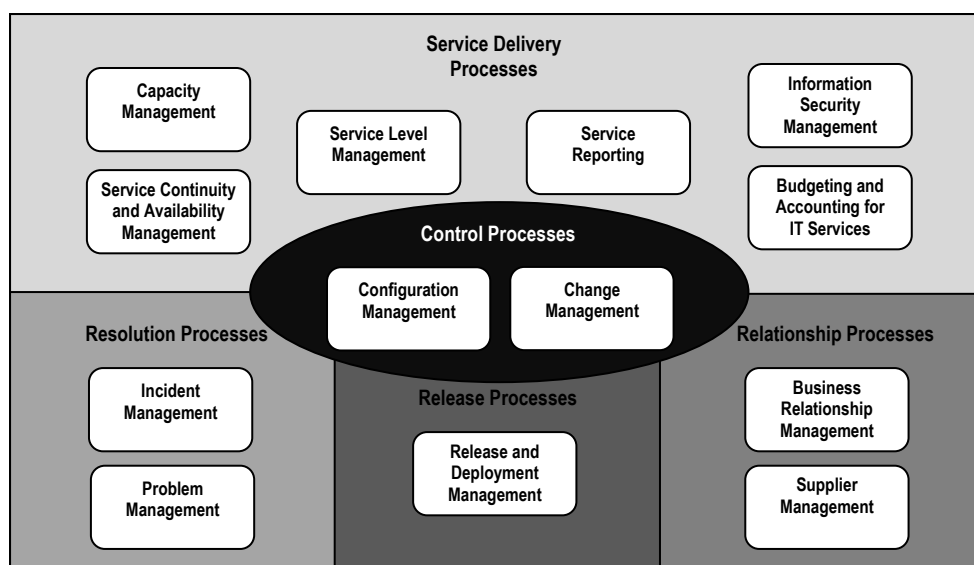


Figura 2.1. Processos do ITIL classificados por escopo de atuação

Abaixo segue uma breve descrição dos principais processos apresentados na Figura 2.1, como definidos pelo ITIL:

- **Gerenciamento de Portfólio de Serviços** (*Service Portfolio Management*): tem como propósito gerenciar os investimentos da organização, controlar os recursos dos projetos que estão em andamento e identificar novas oportunidades de investimentos (ITIL: Service Strategy, 2007) (ITIL: Service Design, 2007);

- **Gerenciamento de Mudanças** (*Change Management*): responsável por estabelecer como conduzir mudanças em infraestruturas de TI, com o objetivo de otimizar os recursos envolvidos, gerar o menor impacto nos serviços providos e

assegurar que, ao fim das mudanças, a infraestrutura esteja em um estado estável (ITIL: Service Transition, 2007);

- **Gerenciamento de Incidentes** (*Incident Management*): tem como principal objetivo fazer com que, na ocorrência de incidentes, os serviços retornem ao funcionamento normal o mais rápido possível, buscando o menor impacto possível aos negócios da organização (ITIL: Service Operation, 2007) (ITIL: Continual Service Improvement, 2007);

- **Gerenciamento de Níveis de Serviço** (*Service Level Management*): tem a finalidade de garantir que todos os serviços em operação estejam no estado “executando” (*running*) e os seus respectivos desempenhos e medições estejam corretos e dentro dos limites definidos por Acordos de Nível de Serviço (*Service Level Agreements - SLAs*) – os quais são estabelecidos nos contratos de prestação de serviço entre organizações produtoras e consumidoras (ITIL: Service Design, 2007) (ITIL: Continual Service Improvement, 2007);

- **Gerenciamento de Releases e Implantação** (*Release and Deployment Management*): tem como principais objetivos a definição e a verificação junto aos clientes da adequação dos planos de releases e de implantação (*release and deployment plans*), além de garantir a integridade e compatibilidade dos pacotes de *release* (ITIL: Service Transition, 2007) (ITIL: Service Operation, 2007);

- **Gerenciamento de Configuração** (*Configuration Management*): tem como propósito identificar, controlar, medir, gravar, reportar, verificar e auditar os itens de configuração (como por exemplo, os ativos da organização), além de garantir funcionamento e comportamento adequados dos serviços em operação (ITIL: Service Transition, 2007) (ITIL: Service Operation, 2007);

- **Gerenciamento de Segurança da Informação** (*Information Security Management*): responsável por alinhar os aspectos de segurança da infraestrutura de TI com as políticas de segurança da organização, e garantir que as políticas de segurança da informação adotadas são utilizadas em todos os processos da organização (ITIL: Service Design, 2007) (ITIL: Service Operation, 2007).

Entretanto, nestes livros é apresentando apenas **o que** deve ser feito pelos gerentes de TI, não explorando **como** ser feito. Por esta razão, nos últimos anos a comunidade científica tem concentrado esforços para entender como resolver problemas nos processos de gerenciamento de TI. Diversos aspectos vêm sendo explorados, tais como modelos (RODOSEK, 2003), automação (BROWN e KELLER, 2006) e alinhamento a propósitos de negócio (MOURA *et al.*, 2008). Analisando-se especificamente a subárea de Gerenciamento de Mudanças, observa-se atividade semelhante. Por exemplo, passos importantes têm sido dados em tópicos como agendamento (REBOUÇAS *et al.*, 2007), (TRASTOUR *et al.*, 2007), *rollback* (MACHADO *et al.*, 2008), (MACHADO *et al.*, 2009), e planejamento de mudanças (KELLER *et al.*, 2004), (CORDEIRO *et al.*, 2008a), (CORDEIRO *et al.*, 2008b), (CORDEIRO *et al.*, 2009). A seguir são discutidos os principais trabalhos na área de (i) Gerenciamento de Mudanças, (ii) alocação de recursos humanos a atividades, e, por fim, (iii) são apresentadas as principais diferenças dos trabalhos existentes com o que está sendo proposto nesta dissertação.

## 2.1 Gerenciamento de Mudanças

Como já mencionado, o processo de Gerenciamento de Mudanças é responsável por conduzir desde a especificação da mudança – em documentos chamados de requisição de mudança – até a implantação da mesma. Neste contexto, um dos passos mais importantes, e que vêm recebendo muita atenção da comunidade científica, é o *planejamento de mudanças*. Esse passo, como o próprio nome sugere, é responsável por planejar – tanto em alto nível quanto em baixo nível – o que deve ser realizado nos itens da infraestrutura de TI para que uma requisição de mudança seja satisfeita.

Como trabalho seminal da área, Keller *et al.* (2004) propuseram CHAMPS, um sistema para automação da geração de planos de mudança. A principal contribuição desse trabalho foi a elaboração de um algoritmo que permite produzir, de forma automática, planos de mudança executáveis. O projeto do sistema CHAMPS seguiu uma abordagem que busca utilizar ferramentas e técnicas existentes e bem difundidas como, por exemplo, *workflows* e gerenciamento de projetos, aplicadas no contexto de Gerenciamento de Mudanças. Em termos de arquitetura, o referido sistema é constituído por um construtor de seqüência de atividades a serem executadas na infraestrutura gerenciada, denominado *Task Graph Builder* (TGB), e um gerador e escalonador de planos de mudança, denominado *Planner & Scheduler* (P & S). O TGB determina as necessidades temporais e as restrições de cada tarefa para completar uma requisição de mudança (*Request for Change, RFC*). O P & S utiliza o TGB para construir o plano de mudança. Ainda nesse trabalho, busca-se explorar alto grau de paralelismo na execução das tarefas para melhorar o desempenho da execução das requisições de mudança solicitadas. Embora seja capaz de aperfeiçoar o agendamento de tarefas considerando os recursos de *hardware* disponíveis para executá-las, CHAMPS foi um primeiro esforço para o *planejamento de mudanças* e, por este motivo, diversos aspectos não foram considerados. Dentre os principais aspectos não considerados, podem ser citados a associação de recursos humanos a tarefas e a possibilidade de especificar objetivos/restrições em requisições de mudanças.

Em um trabalho recente, Cordeiro *et al.* (2008a) propuseram uma solução para formalizar, preservar e reusar o conhecimento adquirido durante mudanças recorrentes em infraestruturas de TI. A solução, baseada em *modelos de mudança (templates de mudança)*, permite que passos rotineiros (por exemplo, alterações e/ou implantações de serviços) sejam especificados em alto nível e recorrentemente instanciados (por exemplo, a partir da indicação dos serviços que serão manipulados). Além disso, é apresentado e detalhado o CHANGELEDGE, um sistema de gerenciamento de mudanças para gerar, de forma automatizada, planos de mudança. Tais planos são refinados à luz de informações sobre a infraestrutura de TI, pacotes de software e suas respectivas dependências, e do esboço de planos preliminares desenvolvidos por operadores e/ou gerados a partir de *templates de mudança*. Embora a solução compreenda um algoritmo para a geração de planos detalhados a partir da especificação em alto nível, está fora do escopo desse algoritmo, como uma premissa simplificadoria, alinhar os planos gerados a propósitos de negócio.

Em um trabalho subsequente (CORDEIRO *et al.*, 2008b), o algoritmo de geração de planos de mudança foi aprimorado, de modo a levar em consideração as restrições técnicas impostas pelo ambiente gerenciado (por exemplo, disponibilidade de recursos como espaço em disco e memória, e permissões para a execução de ações em determinados dispositivos). Em contraste com pesquisas anteriores, esta solução



concentra-se em prever e considerar o impacto que atividades já computadas para compor um plano irão causar na infraestrutura de TI, a fim de calcular a viabilidade de executá-lo. Para poder realizar este cálculo, apresenta-se o conceito de *snapshots* da infraestrutura de TI: representações de estados intermediários que a infraestrutura de TI pode alcançar com a execução de diferentes caminhos do plano de mudança. Como resultado, os planos de mudança refinados gerados pela solução tendem a estar menos propensos a falhar prematuramente, reduzindo assim a ocorrência de incidentes relacionados com as execuções de atividades mal dimensionadas nos planos de mudança.

Outro conjunto de trabalhos, apesar de não tratar do problema de planejamento de mudanças, merece breve discussão por, de alguma forma, explorar alinhamento com aspectos de nível de negócio. Por exemplo, Keller (2005) introduziu o conceito de contratos eletrônicos. No trabalho em questão são propostos quatro tipos de contratos: Requisições de Mudança (RFC), *Deployment Descriptors*, Políticas e Boas Práticas (*Policies and Best Practices*) e Acordos de Nível de Serviço (*Service Level Agreements*, SLAs). Esses quatro documentos apresentados por Keller (2005), que possuem diferentes propósitos, formatos e níveis de detalhamento, são utilizados para a especificação de objetivos/restrições e têm fundamental importância para o processo de execução de mudanças.

Na mesma linha de investigação, Rebouças *et al.* (2007) propuseram uma abordagem para a priorização e o escalonamento de mudanças visando à minimização de custos (por exemplo, custos operacionais e perdas financeiras devido à violação de SLAs). O principal problema abordado é o agendamento/escalonamento ótimo de mudanças em janelas (*change windows*). Não trata de problemas relativos ao planejamento de mudanças; apenas prioriza a execução dos planos de mudança aprovados, ou seja, não busca otimizações na geração dos planos de mudança em si, mas na ordem de execução dos mesmos.

## 2.2 Alocação de humanos

Os desafios que envolvem a atribuição de humanos a atividades de TI vêm sendo investigados há muito tempo nas mais diversas áreas da computação. Isso se deve ao alto grau de complexidade e diversidade dos fatores que devem ser analisados para cada uma das áreas em que os humanos são utilizados (variando conforme os objetivos de cada ambiente analisado). Nesta seção são apresentados os principais trabalhos recentes que procuram mensurar e otimizar os custos que estão envolvidos com a utilização de humanos.

Em uma pesquisa recente, Zhou (2008) propôs um método para alocar humanos em projetos de software. O método consiste em analisar aspectos das tarefas a serem realizadas, detalhes da arquitetura do software a ser desenvolvido, habilidades/preferências dos humanos e relações interpessoais dos recursos humanos envolvidos. Para isso foram consideradas a satisfação e a adequação do problema por parte dos recursos humanos e também os atributos das tarefas para as quais estes estão mais qualificados. Além disso, também foi considerada a relação entre as tarefas e as relações interpessoais. Apesar do trabalho em questão levar em consideração esses diversos aspectos, pouco é discutido sobre como influenciam na execução das tarefas, ou seja, pouco é argumentado sobre os impactos que uma escolha gera para o projeto como um todo. Ainda, não é apresentada a métrica utilizada para analisar as habilidades de cada humano (somente são apresentados os capazes e os incapazes de realizarem as

tarefas). Além disso, o trabalho foca nos aspectos do desenvolvimento de *software*, não considerando diversos problemas relativos às infraestruturas de TI, como por exemplo, a dependência entre os recursos e os serviços.

Na mesma linha, Zhengyuan e Lihua (2008) propuseram um modelo matemático para conduzir a alocação de humanos baseado em múltiplos critérios. Para isso, vislumbraram um algoritmo particular de otimização de enxames com objetivos múltiplos (*multi-objective particle swarm optimization*). Apesar de o trabalho criar um modelo eficiente (capaz de gerar resultados otimizados em pouco tempo), os parâmetros utilizados são apenas dois: “custo” e “proficiência” dos recursos humanos. Portanto, aspectos relativos às infraestruturas de TI e os possíveis impactos das alocações (por exemplo, tempo e qualidade) não são analisados.

Alguns aspectos diferentes foram investigados pela pesquisa realizada por Bennour *et al.* (2005), como por exemplo as habilidades específicas de cada recurso humano. No trabalho em questão, Bennour *et al.* propõem a associação de humanos a atividades de *workflows* por meio de uma solução baseada em busca binária. A solução proposta é comparada, sob diferentes pontos de vista (custos temporais e financeiros, por exemplo), com alguns algoritmos provindos da área de inteligência artificial, tais como *simulated annealing* e busca *Taboo*. O trabalho em questão produz resultados muito interessantes (quantitativos e qualitativos), porém não considera aspectos fundamentais para o gerenciamento de TI, como, por exemplo, o impacto da alocação para os serviços em andamento, a indisponibilidade dos recursos envolvidos e a adequação da alocação (se os humanos alocados possuem capacidade de executar as tarefas a que foram submetidos).

Utilizando como estudo de caso o processo de *Gerenciamento de Incidentes*, Diao e Heching (2009) propuseram uma abordagem para o gerenciamento de pessoas e carga de trabalho (*workload*). O objetivo do trabalho é aumentar a produtividade dos serviços fornecidos pela organização. Para tal, os recursos humanos são alocados conforme as habilidades técnicas (*skills*), e busca-se uma distribuição uniforme de tarefas entre os participantes de cada grupo de trabalho (carga de trabalho homogênea entre os grupos). Caso existam mudanças nos recursos gerenciados ou nos objetivos (humanos podem mudar de grupos conforme os serviços requisitados), emprega-se um sistema de controle de *feedback* para identificar e ajustar as métricas dinamicamente. Apesar desta proposta abordar gerenciamento de carga de trabalho, pouco é discutido sobre outros aspectos que influenciam a escolha e a utilização de humanos como, por exemplo, os custos de cada alocação.

### **2.3 Síntese comparativa**

Em síntese, embora o planejamento de mudanças tenha sido foco de investigações recentes, o alinhamento das mesmas com os propósitos definidos em nível de negócio não foi abordado. Isso significa que, não raro, o planejamento e, por conseguinte, a execução de mudanças é conduzida em direção oposta, ou de forma pouco alinhada, com os objetivos/restrições de negócio demandados pelo solicitante. Conseqüentemente, o negócio das organizações acaba sendo prejudicado por planos de mudança ineficientes e não otimizados.

Por sua vez, os trabalhos sobre alocação de humanos são muito específicos, ou seja, os aspectos analisados variam conforme o ambiente em que a alocação é realizada. Deste modo, os trabalhos apresentados buscam otimizações dos recursos humanos

disponíveis, analisando os custos da alocação (como por exemplo, custo financeiro para a alocação dos humanos e tempo para a realização das tarefas).

Para que os planos de mudança estejam de acordo com os objetivos/restrições de negócio é necessário explorar a alocação adequada dos recursos disponíveis na infraestrutura. Com este objetivo, os capítulos seguintes apresentam uma solução conceitual para o alinhamento de planos de mudança a objetivos/restrições de negócios; diferentes algoritmos para gerar este alinhamento; além de apresentar o sistema desenvolvido para suportá-la e, por fim, os resultados obtidos com a utilização do mesmo.

### 3 SOLUÇÃO CONCEITUAL

Visando oferecer suporte ao planejamento de mudanças alinhado a propósitos definidos em nível de negócio, foram introduzidos novos componentes na solução conceitual para o planejamento e a implantação de mudanças, proposta em trabalhos anteriores do nosso grupo de pesquisa (CORDEIRO *et al.*, 2008a), (CORDEIRO *et al.*, 2008b), (CORDEIRO *et al.*, 2009). Os novos componentes – destacados na área em cinza – materializam o mecanismo para o planejamento guiado por propósitos de negócio e se encaixam adequadamente, sem modificação significativa, na solução conceitual proposta anteriormente. A Figura 3.1 apresenta uma visão geral da solução conceitual proposta neste trabalho, destacando os seus principais componentes, atores envolvidos e interações entre esses elementos.

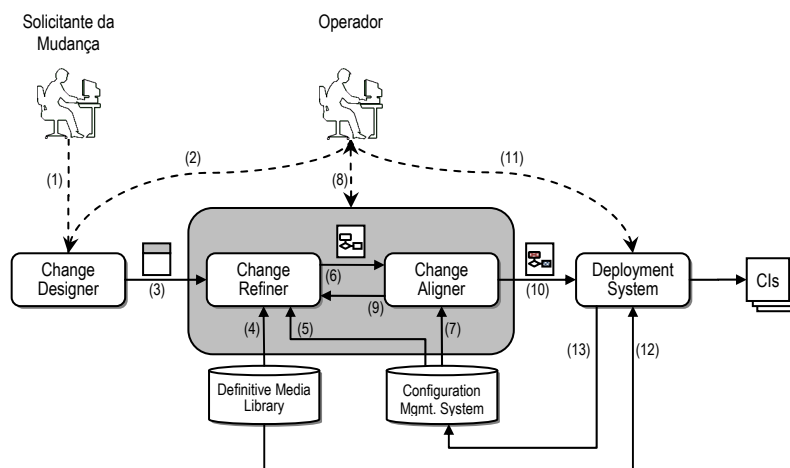


Figura 3.1. Arquitetura conceitual da solução proposta

O solicitante da mudança inicia um processo de mudança interagindo com o componente *Change Designer* (fluxo 1 na Figura 3.1) para elaborar uma requisição de mudança (*Request for Change, RFC*). A RFC descreve, em linhas gerais, quais modificações devem ser acomodadas na infraestrutura gerenciada, os itens de configuração (*Configuration Items, CIs*) primariamente afetados (por exemplo, *firewall, switches, serviços, aplicações, etc.*), e os propósitos de negócio a serem alcançados. A RFC, no entanto, não especifica os detalhes de como a mesma deve ser materializada. Esses devem ser esboçados em um passo subsequente, por um operador, por meio da especificação de um “plano preliminar” junto ao componente *Change Designer* (fluxo 2). O plano preliminar consiste de um *workflow* de atividades (ou ações) que descrevem, em alto nível de abstração, como a mudança requisitada deve ser materializada na infraestrutura de TI gerenciada.

A geração do plano detalhado de mudança a partir da especificação preliminar esboçada pelo operador é realizada, sem intervenção humana, pelo componente *Change*

*Refiner* (3). Esse processamento é efetuado utilizando um algoritmo de refinamento de atividades, com base em informações sobre (pacotes de) *software*, disponíveis na *Definitive Media Library* (DML) (4), e em informações sobre a infraestrutura de TI alvo da mudança (5), presentes no *Configuration Management System* (CMS).

Após o refinamento da mudança, o plano resultante é consumido pelo componente *Change Aligner* (6), que possui um papel central no alinhamento do plano aos objetivos/restrições de negócio expressos na RFC. Para guiar este alinhamento, o componente *Change Aligner* obtém no CMS (7) informações sobre custos de alocação de operadores e suas habilidades (por exemplo, um operador especialista em serviço de *e-mail* cuja hora de trabalho custa 40 unidades monetárias), bem como capacidade dos dispositivos a serem manipulados (por exemplo, poder computacional, no caso de estações de trabalho e servidores).

Posteriormente (8), os planos gerados podem ser modificados pelo operador (de modo a refletir precisamente suas necessidades), e avaliados quanto aos “compromissos” existentes entre as várias alternativas. Entre os critérios que podem servir de parâmetros para essa análise estão o tempo necessário, os custos financeiros associados e os recursos humanos que devem ser alocados para a execução dos planos. Com base nessa análise, o operador poderá selecionar o plano mais adequado para ser executado sobre a infraestrutura. É importante ressaltar que o processo de refinamento e alinhamento da mudança é reiterado (9), durante um período de tempo definido *a priori* pelo operador, visando à geração de diferentes planos detalhados que atendam aos mesmos propósitos de negócio definidos na RFC. O resultado do processo de alinhamento do *Change Aligner* é um plano detalhado com uma escolha de alocação de papéis de humanos para executarem as atividades manuais.

Como próximo passo do processo de mudança, o operador invoca a execução do plano selecionado (10) interagindo com o componente *Deployment System* (11). Para realizar algumas das atividades descritas no plano, tal componente pode consumir pacotes de *software* disponíveis na *Definitive Media Library* (12). Após a execução, o *Deployment System* deverá atualizar o CMS com as modificações realizadas em cada CI da infraestrutura gerenciada (13).

Tendo apresentado uma visão geral da solução proposta, as próximas seções têm por objetivo descrever (i) a geração de planos detalhados a partir da especificação preliminar de mudanças e (ii) o processo de alinhamento de tais planos a propósitos definidos em nível de negócio.

### **3.1 *Change Refiner***

O componente *Change Refiner* desempenha um papel importante no processo de geração de um plano de mudança detalhado a partir do plano preliminar fornecido pelo solicitante da mudança. Para materializar esse componente, propõe-se um algoritmo, inspirado em um trabalho anterior (CORDEIRO *et al.*, 2009) do nosso grupo de pesquisa, que oferece suporte à geração de diferentes alternativas de planos para o processo de alinhamento. O novo algoritmo, bem como seu relacionamento com as outras entidades da solução conceitual, é brevemente descrito a seguir.

Conforme apresentado na solução conceitual, a geração de planos de mudanças detalhados requer informações sobre as entidades que compõem a infraestrutura e seus relacionamentos, e os pacotes de *software* disponíveis para a implantação de mudanças. Com o objetivo de materializar o armazenamento dessas informações no CMS e na

DML, respectivamente, é empregado nesta dissertação um subconjunto do *Common Information Model* (CIM) (DMTF, 2010). A Figura 3.2 apresenta uma visão parcial do modelo, o qual permite representar qualquer CI (item de configuração) que compõe a infraestrutura gerenciada. Relacionamentos tais como associações, composições e agregações, a maioria omitidas na figura por questões de legibilidade, mapeiam as dependências entre os elementos que compõem a infraestrutura.

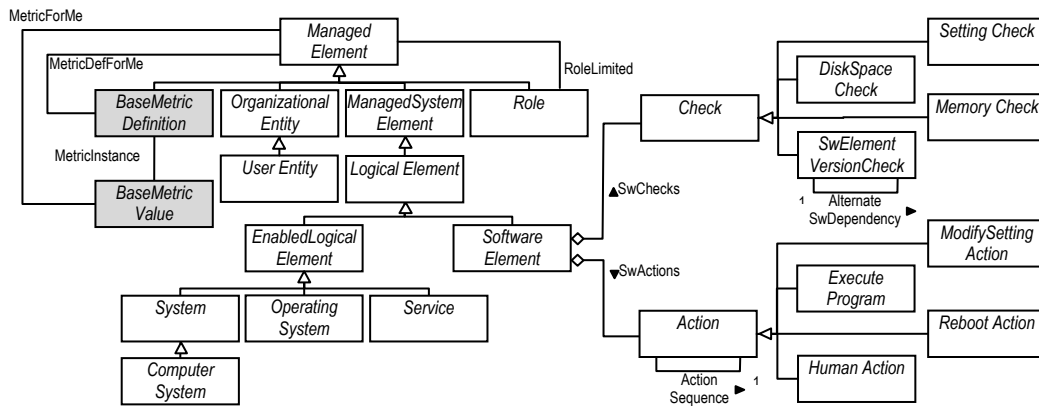


Figura 3.2. Modelo usado para a representação da infraestrutura de TI e da DML

O modelo apresentado incorpora também classes *Check* e *Action*, que mapeiam informações necessárias para o cômputo de dependências e refinamento das atividades que compõem o plano preliminar. Uma instância da classe *Check* define uma condição a ser satisfeita ou uma característica requerida pelo *software* associado para que o mesmo possa evoluir para um novo estado (ex: *deployable*, *installable*, *executable* ou *running*).

Para exemplificar as classes descritas acima, na Figura 3.3 são apresentadas instâncias de *Checks* associadas ao *software Squirrelmail*. As duas primeiras expressam, respectivamente, que os *softwares Exim4* e *Apache2* devem ser instalados antes da instalação do *Squirrelmail*. As duas últimas, por sua vez, expressam que o *Squirrelmail* requer 1 MB de memória e 3 MB de espaço em disco para sua instalação e execução. Cada instância da classe *Action*, por sua vez, representa uma atividade exigida para mudar o estado do *software* associado (ex: de *installable* para *executable*, isto é, um processo de instalação). Utilizando novamente como exemplo o *Squirrelmail*, são ilustradas na Figura 3.3 duas atividades associadas ao processo de instalação do mesmo: *Install Squirrelmail*, responsável pela cópia dos arquivos binários do *Squirrelmail*; e *Configure Squirrelmail*, a qual gera as configurações necessárias para o funcionamento do serviço provido pelo mesmo.

Outro requisito importante para a geração automatizada de planos detalhados é a definição, de forma não ambígua e interpretável pelo mecanismo de refinamento, dos objetivos das atividades presentes no plano preliminar. Para tal, é utilizada uma notação, proposta em um trabalho proposto anteriormente pelo grupo de pesquisa (CORDEIRO *et al.*, 2009), denominada *Activity Modeling Notation* (AMN), representada na Tabela 3.1. Como exemplo, considere novamente a atividade *Install Squirrelmail*. Sua representação, de acordo com a AMN, é *install SoftwareElement Squirrelmail at ComputerSystem www.example.com*. Neste exemplo, *Squirrelmail* e *www.example.com* são referências a objetos presentes na DML e no CMS, respectivamente.

O algoritmo para o refinamento de mudanças, também denominado de *change\_refinement*, recebe como entrada a especificação de mudança a ser refinada, além de dados sobre o atual estado da infraestrutura gerenciada (oriundos do CMS) e

sobre pacotes disponíveis para o processo de mudança (provenientes da DML). Para cada uma das atividades que compõe o plano preliminar, *change\_refinement* computa o conjunto de atividades que satisfazem as dependências para a execução da mesma.

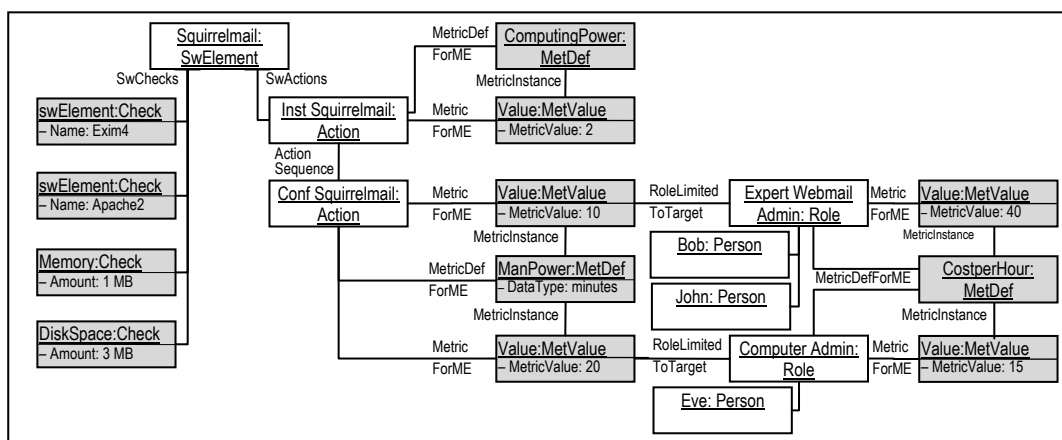


Figura 3.3. Instâncias de software (*SoftwareElement*) presentes na DML e de Papéis (*Roles*) presentes na infraestrutura de TI

Para ilustrar o processo de refinamento, considere a RFC *Migrate Mail System*, apresentada na Figura 3.4 (fundo em cinza na figura). Ela consiste na migração de um serviço de *e-mail* do servidor *X* para o *Y*, e tem como objetivo minimizar o tempo de execução estimado para, no máximo, 150 minutos (de modo a não comprometer as operações da organização). O plano preliminar associado à especificação (fundo em branco) é composto por seis atividades em alto nível, que compreendem desde a instalação do sistema operacional até a migração de dados e configurações do serviço de *e-mail* localizado em *X* para *Y* (ambos servidores presentes na rede gerenciada).

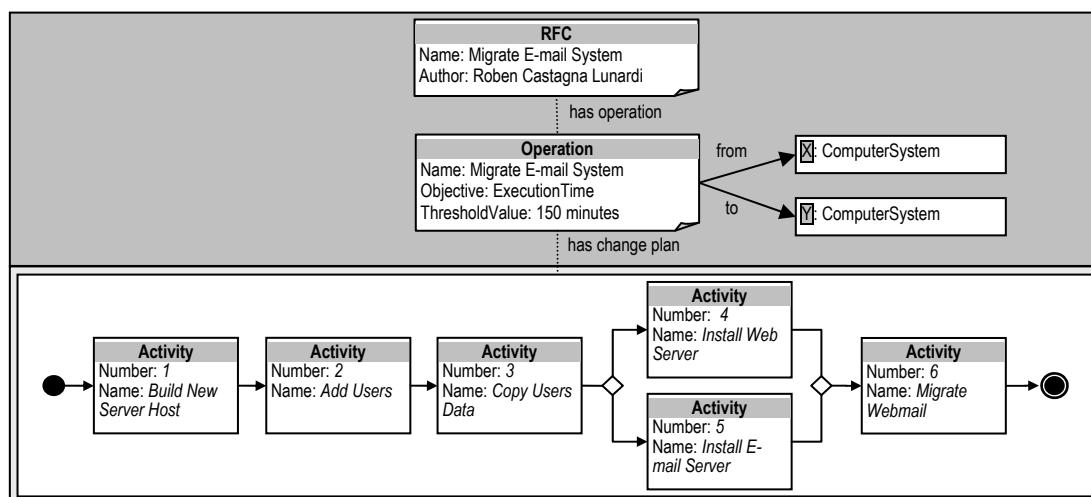


Figura 3.4. Exemplo de RFC e de plano preliminar de mudança para migração de servidor de *e-mail* e respectivo *front-end Web*

Focando na atividade *Install E-mail Server* (ilustrada no plano preliminar), *change\_refinement* identifica – a partir das pré-condições (*Checks*) apresentadas na DML (Figura 3.3) – que a execução dessa atividade depende da instalação prévia dos softwares *Apache2* (servidor Web) e *Exim4* (servidor de *e-mail*). Logo, duas novas atividades serão geradas para satisfazer as dependências de *Install Squirrelmail: Install Apache2* e *Install Squirrelmail: Install Exim4*. Esse processo continua recursivamente, até que todas as

dependências para execução das atividades do plano sejam satisfeitas. Ao final desse processo, quando um plano detalhado é obtido (por exemplo, o plano ilustrado na Figura 3.6, página 28), *change refinement* armazena o “estado atual” (isto é, as opções de refinamento percorridas para se chegar a tal plano). Esse passo é importante para que, após o alinhamento (discutido em detalhes na Seção 3.2), seja possível gerar novos planos detalhados utilizando refinamentos alternativos. Desta forma, diferentes planos de mudança detalhados podem ser gerados para a mesma especificação inicial.

Tabela 3.1. Descrição da notação AMN

Atividades Abstratas	Especificação Abstrata
Iniciar um serviço	start Service ⟨Service⟩ at ComputerSystem ⟨Host⟩
Parar um serviço	stop Service ⟨Service⟩ at ComputerSystem ⟨Host⟩
Recarregar as configurações de um serviço	reload Service ⟨Service⟩ at ComputerSystem ⟨Host⟩
Reiniciar um serviço	restart Service ⟨Service⟩ at ComputerSystem ⟨Host⟩
Instalar um <i>software</i>	install SoftwareElement ⟨Software⟩ at ComputerSystem ⟨Host⟩
Desinstalar um <i>software</i>	uninstall SoftwareElement ⟨Software⟩ from ComputerSystem ⟨Host⟩
Efetuar backup das configurações de um <i>software</i>	backup Setting from SoftwareElement ⟨Software⟩ at ComputerSystem ⟨Host⟩ to DataFile ⟨File⟩ at ComputerSystem ⟨Backup Server⟩
Restaurar as configurações de um <i>software</i>	restore Setting for SoftwareElement ⟨Software⟩ at ComputerSystem ⟨Host⟩ from DataFile ⟨File⟩ at ComputerSystem ⟨Backup Server⟩
Copiar configurações específicas de um <i>software</i>	copy Setting ⟨Setting⟩ from SoftwareElement ⟨Software⟩ at ComputerSystem ⟨Host⟩ to SoftwareElement ⟨Software⟩ at ComputerSystem ⟨Host⟩
Copiar configurações de um <i>software</i>	copy Setting from SoftwareElement ⟨Software⟩ at ComputerSystem ⟨Host⟩ to SoftwareElement ⟨Software⟩ at ComputerSystem ⟨Host⟩
Adicionar novo usuário	add UserEntity ⟨User⟩ at ComputerSystem ⟨Host⟩
Remover um usuário	del UserEntity ⟨User⟩ from ComputerSystem ⟨Host⟩
Modificar os atributos de um usuário	modify UserEntity ⟨User⟩ at ComputerSystem ⟨Host⟩ set ⟨Attribute⟩ to value ⟨New Value⟩
Adicionar nova entrada no DNS	add HostName ⟨Host⟩ at ComputerSystem ⟨DNS Host⟩
Apagar uma entrada existente do DNS	delete HostName ⟨Host⟩ at ComputerSystem ⟨DNS Host⟩
Modificar uma entrada existente do DNS	modify HostName ⟨Host⟩ at ComputerSystem ⟨DNS Host⟩ set ⟨Attribute⟩ to ⟨New Value⟩
Criar novo arquivo	create DataFile ⟨File⟩ in Directory ⟨Dir⟩ at ComputerSystem ⟨Host⟩
Apagar um arquivo existente	delete DataFile ⟨File⟩ in Directory ⟨Dir⟩ from ComputerSystem ⟨Host⟩
Renomear um arquivo	rename DataFile ⟨File⟩ to ⟨New File⟩ in Directory ⟨Dir⟩ at ComputerSystem ⟨Host⟩
Copiar arquivo para uma pasta	copy DataFile ⟨File⟩ in Directory ⟨Target Directory⟩ at ComputerSystem ⟨Host⟩ to Directory ⟨Target Directory⟩ at ComputerSystem ⟨Host⟩
Mover arquivo para uma pasta	move DataFile ⟨File⟩ in Directory ⟨Target Directory⟩ at ComputerSystem ⟨Host⟩ to Directory ⟨Target Directory⟩ at ComputerSystem ⟨Host⟩
Configurar um <i>software</i> /dispositivo genérico	configure ManagedSystemElement ⟨Element⟩ of type ⟨Element Type⟩ at ComputerSystem ⟨ComputerSystem⟩ using Setting ⟨Setting⟩
Modificar os atributos de um <i>software</i> /dispositivo genérico	modify ManagedSystemElement ⟨Element⟩ of type ⟨Element Type⟩ at ComputerSystem ⟨ComputerSystem⟩ set ⟨Attribute⟩ to ⟨New Value⟩
Copiar <i>schema</i> do Banco de Dados para um arquivo	dump schema from DatabaseSystem ⟨Database⟩ at ComputerSystem ⟨Host⟩ to DataFile ⟨File⟩ in Directory ⟨Dir⟩ at ComputerSystem ⟨Host⟩
Carregar um <i>schema</i> para um Banco de Dados	load schema from DataFile ⟨File⟩ in Directory ⟨Dir⟩ at ComputerSystem ⟨Host⟩ to DatabaseSystem ⟨Database⟩ at ComputerSystem ⟨Host⟩



### 3.2 *Change Aligner*

A materialização do componente *Change Aligner*, elemento chave para o alinhamento dos planos detalhados aos objetivos de negócio, está fundamentada em dois blocos construtores principais: (i) a formalização de propósitos de negócio e os modelos para representar métricas de custos/capacidades e (ii) o algoritmo para o alinhamento de mudanças em função dos propósitos especificados e custos/capacidades associados aos elementos da infraestrutura. Cada um desses blocos construtores é detalhado a seguir.

#### 3.2.1 **Objetivos / restrições de negócio e métricas de custos / capacidades**

Para que os planos de mudança produzidos pelo componente *Change Refiner* possam ser alinhados a objetivos/restrições de negócio, o(s) objetivo(s) e/ou restrição(ões) a ser(em) satisfeito(s) precisa(m) ser especificado(s) na RFC. Duas categorias de objetivos/restrições são contempladas nesta dissertação. São elas: o tempo necessário (*ExecutionTime*) e o custo financeiro associado ao uso de recursos humanos (*ExecutionCost*) para completar uma mudança. Sua escolha deveu-se por terem sido apontadas como essenciais por um grupo de operadores de rede consultados, e por serem encontradas em trabalhos relacionados apresentados anteriormente. Acompanha a definição de cada objetivo e/ou restrição um limiar (*ThresholdValue*), representando o tempo máximo admitido e/ou o custo máximo tolerado, calculado em função dos humanos alocados, para a execução da mudança. Como pode ser observado na Figura 3.4, foi especificado na RFC *Migrate E-mail System* que a mudança, ao ser planejada, deve considerar um tempo máximo de execução de 150 minutos.

Outras informações indispensáveis ao processo de alinhamento são as *métricas de custo* e de *capacidade*. Enquanto as primeiras são usadas para caracterizar o esforço demandado para a execução de atividades ligadas a gerência e a operação de infraestruturas de redes e serviços, a última permite expressar uma expectativa de rendimento de computadores para executá-las. As métricas empregadas neste trabalho, enumeradas a seguir, são adotadas e consolidadas em áreas como Sistemas Distribuídos (SHEPHERD, 1977), Engenharia de Software (PILLAI, SUKUMARAN NAIR, 1997) e Economia (BANYAHIA, 1996). Do ponto de vista de modelagem, são especificadas, e seus valores instanciados, no modelo usado para a representação da infraestrutura de TI e da DML (ilustrado na Figura 3.2) por meio das classes *BaseMetricDefinition* e *BaseMetricValue*, respectivamente.

As métricas de custo são subdivididas em *ManPower* (MP), *ComputingPower* (CP) e *CostperHour* (CH). MP é usada para quantificar o esforço humano necessário para executar uma atividade (no nosso caso, representa o tempo médio em minutos gasto por um humano de um determinado papel para realizar determinada tarefa). CP, por sua vez, denota o esforço computacional requerido para executar uma atividade automatizada, *i.e.*, com nenhuma (ou desprezível) intervenção humana. CH representa o custo financeiro por hora dos integrantes de cada papel de humanos. Instâncias de valores para essas métricas são utilizadas para “anotar” custos em atividades associadas a elementos de *software* registrados na DML. Na Figura 3.3, pode-se observar, a título de ilustração, que a atividade *Install Squirrelmail* possui custo de execução associado, em CP, de 2 (duas) unidades. Por outro lado, a atividade de configuração do referido serviço, *Configure Squirrelmail* necessita de esforço (MP), em média, de 10 (dez) minutos de um administrador especializado em *Webmail* (custando 40 unidades

monetárias por hora para a organização) e, em média de 20 (vinte) minutos de um administrador sem especialização em *Webmail* (custo de 15 unidades monetárias).

A métrica de capacidade *ComputingPower/Hour* (CP/H) denota a carga de trabalho sustentada por computadores para executar atividades. Instâncias de valores da métrica CP/H são vinculadas a cada um dos computadores presente na infraestrutura de TI, refletindo o seu desempenho esperado.

É importante destacar que a investigação de formas de valoração de custos e capacidades, apesar de fundamentais para a solução proposta, não foram alvo da pesquisa. Para esta finalidade assumimos o emprego de algumas das propostas publicadas e consolidadas na literatura, muitas das quais foram recentemente compiladas por Jorgensen e Shepperd (2007). No caso dos custos CP e MP, utilizamos dados históricos, ou seja, analisamos execuções de planos de mudanças passados e extraímos os valores médios de tempo gasto com cada atividade por cada computador/papel. Tomando as mesmas como base, são calculados os custos para as demais atividades a serem planejadas. A estimativa das capacidades dos recursos computacionais (CP/H) é realizada, também, considerando históricos de execução das mesmas em um determinado ambiente, durante um determinado período de tempo.

### 3.2.2 Algoritmo de alinhamento a objetivos/restrições de negócio

O mecanismo de refinamento, os modelos e as métricas apresentados anteriormente formam a base sobre a qual o algoritmo para o alinhamento de planos de mudança a propósitos de negócio está constituído. O algoritmo, ilustrado na Figura 3.5, é denotado por *change\_alignment* ( $C, DML, IT, H$ ), sendo  $C$  o plano de mudança conforme gerado pelo *change\_refinement*,  $DML$  o repositório que contém informações sobre pacotes de *software* disponíveis,  $IT$  o repositório de informações sobre a infraestrutura gerenciada, e  $H$  o conjunto de operadores humanos disponíveis para a condução de mudanças.

```

01  change_alignmnet (C, DML, IT, H)
02  C' ← empty change plan
03  D ← set of activities from C
04  recursive_alignment (C, C', D, DML, IT, H)
05
06  recursive_alignment (C, C', D, DML, IT, H)
07  if D is empty and cost of deploying C' comply with objective threshold, given DML, IT, then
08  return C' as solution
09  extract activity ai from D
10  if ai is a human activity then
11  for all h in H do
12  for all ai in C do
13  allocable ← true
14  if ai is parallel to aj in C and h is allocated to aj then
15  allocable ← false
16  if allocable is true then
17  associate human h to ai
18  add activity ai to change plan C'
19  if cost of deploying C' comply with objective threshold, given DML, IT, then
20  recursive_alignment (C, C', D, DML, IT, H)
21  else
22  add activity ai to change plan C'
23  if cost of deploying C' comply with objective threshold, given DML, IT, then
24  recursive_alignment (C, C', D, DML, IT, H)
25  return failure

```

Figura 3.5. Algoritmo para geração de planos de mudança alinhados a objetivos/restrições de negócio

Como primeiro passo do processo de alinhamento, é criado um plano de mudança “vazio”, o qual conterà o resultado deste processo (linha 02 na Figura 3.5). Em seguida, as atividades presentes em  $C$  são copiadas para o conjunto  $D$  (linha 03), e o alinhamento continua a partir da invocação da função *recursive\_alignment* (04).

O processo de alinhamento, conforme descrito em *recursive\_alignment*, inicia-se pela verificação se  $D$  (o conjunto de atividades ainda não processadas) é vazio ou não, e se o plano  $C'$  recebido satisfaz aos propósitos de negócio especificados na RFC (07), verificando o respectivo valor de custo máximo (denominado *threshold* na descrição do algoritmo). Caso positivo, o plano  $C'$  é retornado como solução. Caso contrário, os seguintes passos são executados. Inicialmente, uma atividade  $a_i$  – as estratégias para ordenar a escolha de atividades são descritas no Capítulo 4 – é extraída do conjunto  $D$ , para processamento (09). Em seguida, é verificado se  $a_i$  requer (ou não) apoio de um operador humano para a sua execução (10). Caso  $a_i$  demande intervenção humana, o algoritmo tenta alocar o melhor operador  $h$  (disponível em  $H$ , o conjunto de operadores humanos) que não esteja alocado a nenhuma outra atividade paralela a  $a_i$  (linhas 11-18). As estratégias de escolha do melhor recurso humano para cada atividade são descritas no Capítulo 4.

É importante mencionar que, para o cômputo do alinhamento, a alocação dos operadores é conduzida de forma não-nominal. Em outras palavras, é especificado que um humano com papel  $x$  deverá executar um conjunto de atividades. A alocação efetiva (ou nominal) de operadores às atividades é postergada para a etapa de agendamento (*scheduling*) da mudança. Essa etapa está fora do escopo da nossa solução proposta, sendo abordada por diversos trabalhos relacionados (KELLER *et al.*, 2004), (REBOUÇAS *et al.*, 2007), (TRASTOUR *et al.*, 2007).

Uma vez alocado um humano para  $a_i$ , o algoritmo verifica qual o custo de executar o plano  $C'$  sobre a infraestrutura gerenciada, considerando tal alocação (linha 19). Se o custo não violar o valor de limiar (*ThresholdValue*) definido em função do objetivo especificado na RFC, o processo de alinhamento continua recursivamente (20), de modo a processar as outras atividades restantes em  $D$ . Vale ressaltar que a semântica do *custo* (e por conseguinte, do *limiar*) está atrelada ao objetivo/restrrição associado à RFC. Por exemplo, caso seja especificada a redução do tempo de execução (*ExecutionTime*), *limiar* e *custo* são avaliados em função de *tempo*.

Alternativamente, a atividade  $a_i$  extraída do conjunto  $D$  pode não necessitar de intervenção humana (21). Nesse caso, a mesma é adicionada diretamente ao plano que será retornado como solução (22). O processamento segue recursivamente (24), caso o custo de execução de  $C'$  esteja alinhado com o limiar e objetivo definidos (23).

Caso nenhum dos humanos disponíveis possa ser alocado para  $a_i$ , ou o custo para executar  $C'$  (já com a adição de  $a_i$ ) supere o valor de limiar definido, o processo de alinhamento falhará (linha 25). Conseqüentemente, as alocações de operadores realizadas em recursões anteriores serão re-feitas, de modo a explorar novas alternativas para o alinhamento. No pior caso, *recursive\_alignment* retornará o *feedback* de que não é possível alinhar o plano detalhado  $C$  ao propósito de negócio indicado na RFC.

Para ilustrar o processo de alinhamento, considere o plano apresentado na Figura 3.6, gerado a partir da RFC *Migrate E-mail System* (introduzida na Figura 3.4). Os diferentes papéis de humanos alocados para a execução do plano são representados utilizando áreas hachuradas em torno das atividades, sendo as atividades com sombreamento cinza as que requerem intervenção humana. Nesse exemplo, há apenas um humano para cada papel definido em  $H$ , sendo *Expert Webserver Admin* (administrador especialista em *web servers*), *Junior Webserver Admin* (administrador de *web servers* com pouca experiência) e *Computer Admin* (administrador de computadores sem especialização) os papéis utilizados. O plano apresentado obedece ao

objetivo *ExecutionTime*, de acordo com o especificado na RFC. Note que as atividades relacionadas à instalação do *Exim4* (serviço de *e-mail*) requerem maior intervenção humana; por este motivo, neste caso possuem alocado o melhor humano disponível (no caso, pertencente ao papel *Expert Webserver Admin*). De forma análoga, a alocação de humanos é efetuada para as demais atividades do plano, levando em conta os paralelismos existentes, bem como os custos associados.

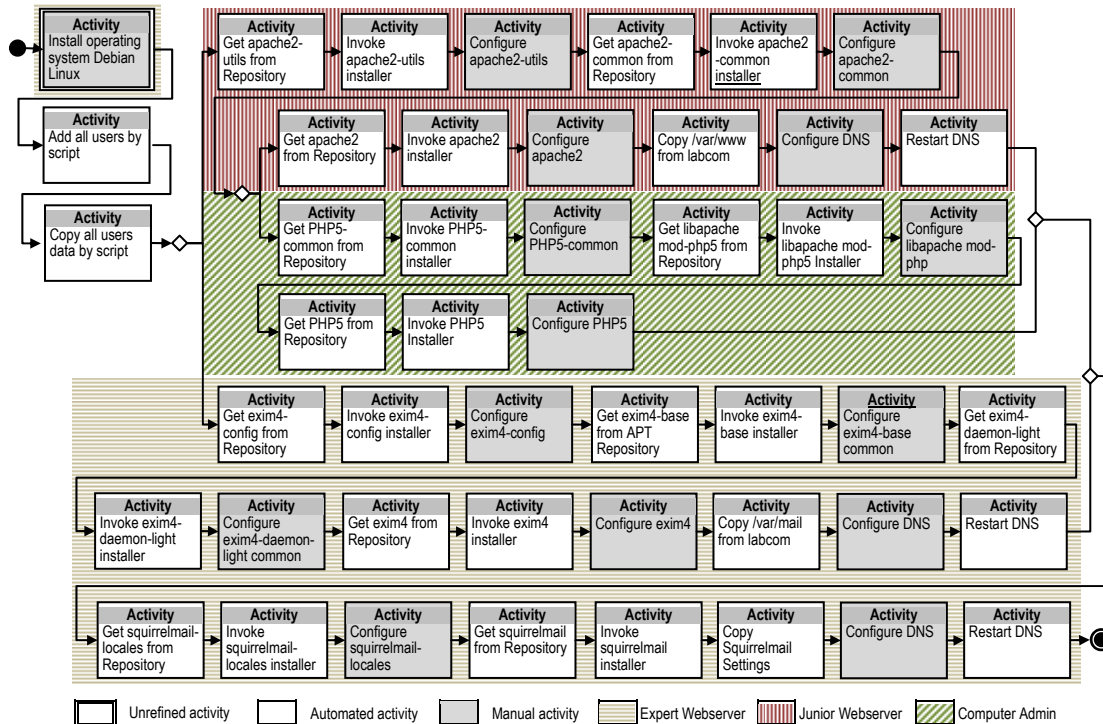


Figura 3.6. Exemplo de plano de mudança detalhado para RFC de migração de servidor de *e-mail*

### 3.3 Resumo do Capítulo

A solução proposta para o planejamento de mudanças alinhado a propósitos de negócio pode ser dividida em três partes fundamentais: (i) entradas (requisição de mudança, informações sobre o estado atual da infraestrutura gerenciada, repositório de *softwares* e plano preliminar); (ii) processamento (*Change Refiner* e *Change Aligner*), e (iii) saídas (planos alinhados aos objetivos de negócio e relatórios de custos). Cada uma destas partes é sintetizada a seguir.

**Entradas:** Os documentos de requisição de mudança (RFC – *Request for Change*) são gerados pelos solicitantes das mudanças e contém informações sobre qual é o resultado desejado/objetivo de cada mudança (por exemplo, migração de um servidor de *e-mails*). Por outro lado, as informações sobre a infraestrutura de TI gerenciada (CMS) e a respeito dos pacotes de *software* disponíveis (DML) são mantidos e alimentados pelo sistema de Gerenciamento de Mudanças (sempre que há uma mudança as informações devem ser atualizadas e mantidas consistentes). Em contrapartida, o plano preliminar de mudança é gerado pelo operador do sistema (responsável por conduzir o processo de planejamento de mudança), sendo composto por atividades em alto nível do que deve ser efetuada para alcançar as metas definidas na requisição de mudanças.

**Processamento:** O componente *Change Refiner* é responsável pela geração – com alto nível de automação – dos planos de mudança detalhados. Os planos são gerados levando em consideração as dependências de *hardware* e *software* disponíveis na infraestrutura de TI e nos pacotes de *software* utilizados. Por sua vez, o componente *Change Aligner* é responsável pelo alinhamento dos planos de mudança detalhados aos objetivos/restrições de negócio. Para isso, calcula-se os custos de cada plano, utilizando diferentes alocações de humanos, e entrega-se para o operador um plano com custos respeitando os objetivos/restrições estabelecidos na requisição da mudança.

**Saídas:** Os planos de mudança detalhados e alinhados aos objetivos/restrições de negócio são entregues ao operador, juntamente com os respectivos relatórios de custos e requisitos para executar cada plano. Vale ressaltar que as alocações propostas não são nominais, ou seja, apenas são estabelecidas quais categorias/papeis de humanos devem ser utilizados, deixando para um processo posterior (agendamento da mudança) a tarefa de especificar os humanos que irão executar cada atividade.

A seguir, no Capítulo 4, são apresentadas as diferentes estratégias para o alinhamento dos planos de mudança aos objetivos/restrições de negócio. Essas estratégias são utilizadas pelo algoritmo de alinhamento apresentado anteriormente na Figura 3.5 de forma a atender as necessidades e preferências dos operadores da mudança.

## 4 ESTRATÉGIAS PARA O ALINHAMENTO DE PLANOS DE MUDANÇA A OBJETIVOS/RESTRICÇÕES DE NEGÓCIO

A solução conceitual, os modelos e o algoritmo apresentados no capítulo anterior formam a base para a construção das estratégias de alinhamento de planos de mudança a objetivos/restricções de negócio. Essas estratégias têm por objetivo definir em qual ordem as atividades e os humanos serão analisados e usados pelo algoritmo de alinhamento. Para este fim, neste capítulo serão apresentados os detalhes de cada estratégia proposta.

Buscar o plano de mudança detalhado “ótimo” e que esteja alinhado aos objetivos/restricções de negócio especificados na RFC é considerado como um problema *NP-hard* (não polinomial difícil). Isso se deve ao fato de ser necessário avaliar e combinar diversos aspectos, como o estado inicial da infraestrutura de TI (a qual contém as informações armazenadas no CMS), o conjunto de pacotes de *software* (advindos da DML), o número de operadores de TI disponíveis (juntamente com os respectivos papéis), o documento de requisição de mudança (acompanhado do plano preliminar de mudança) e os objetivos/restricções de negócio a serem satisfeitos pela RFC em questão. Com o objetivo de reduzir a complexidade deste problema, foram propostas seis estratégias inovadoras, as quais utilizam dois tipos de heurísticas: (i) ordenamento de atividades e (ii) ordenamento dos recursos humanos a serem utilizados. A combinação desses dois tipos de heurísticas irá resultar em cada uma das estratégias. A Tabela 4.1 sumariza as seis estratégias citadas.

De maneira mais clara, as estratégias descritas na Tabela 4.1 apóiam, por meio de heurísticas, esses processos de ordenação (i) em que os humanos serão considerados (discutido anteriormente de maneira genérica na linha 09 da Figura 3.5) e (ii) em que as atividades do plano serão processadas visando à associação de humanos a elas (apresentado na linha 11 da Figura 3.5). Nas próximas seções, cada estratégia é descrita em detalhes e são analisadas as potencialidades e limitações para cada caso onde forem aplicadas.

Tabela 4.1. Estratégias para o alinhamento de planos de mudança, classificadas de acordo com as heurísticas empregadas

		Ordenamento das Atividades		
		Sem Ordenamento	Por Atividade	Por <i>Cluster</i> de Atividades
Alocação dos Recursos Humanos	Alocação do Melhor para o Pior Recurso Humano Disponível	<b>Estratégia 1</b> – Percorre o <i>workflow</i> de atividades, e tenta alocar o melhor recurso humano disponível para cada atividade	<b>Estratégia 2</b> – Ordena as atividades pelo custo (em ordem decrescente), e percorre a lista resultante, tentando alocar o melhor recurso humano disponível para cada atividade	<b>Estratégia 3</b> – Agrupa as atividades em <i>Clusters</i> considerando a topologia do <i>workflow</i> , ordena os clusters pelo custo, e percorre as atividades da lista resultante, tentando alocar o melhor recurso humano disponível para cada atividade
	Alocação Baseada nas Preferências do Gerente de TI	<b>Estratégia 4</b> – Percorre o <i>workflow</i> de atividades, e tenta alocar os recursos humanos para cada atividade, de acordo com as preferências do Gerente de TI	<b>Estratégia 5</b> – Ordena as atividades pelo custo (em ordem decrescente), e percorre a lista resultante, tentando alocar os recursos humanos para cada atividade, de acordo com as preferências do Gerente de TI	<b>Estratégia 6</b> – Agrupa as atividades em <i>Clusters</i> considerando a topologia do <i>workflow</i> , ordena os clusters pelo custo, e percorre as atividades da lista resultante, tentando alocar os recursos humanos para cada atividade, de acordo com as preferências do Gerente de TI

#### 4.1 Alocação do melhor para o pior humano disponível

Esta estratégia (*Estratégia 1* na Tabela 4.1) representa uma solução simplista e gulosa para o alinhamento dos planos de mudança a objetivos/restrições de negócio. Em síntese, essa estratégia processa as atividades na ordem original do *workflow* que compõe o plano de mudança, e tenta alocar o melhor humano disponível (considerando a capacidade produtiva) para executar a tarefa.

Para ilustrar, considere a função  $custo(a, p)$ , a qual retorna o custo (financeiro ou de tempo) para um humano do papel  $p$  para executar a atividade  $a$ . Considere, também, o plano de mudança  $C$  ilustrado na Figura 4.1, e os humanos apresentados na Tabela 4.2. Assumindo que (i) o objetivo de negócio de  $C$  é reduzir o custo financeiro gasto com os humanos, (ii)  $c_1 < c_2 < c_3$ , (iii) toda atividade pode ser executada por um operador de qualquer papel, (iv) todas atividades demoram 1 hora para ser executada, independentemente do papel do humano alocado, e (v) para todas as alocações de um humano  $h_j$  (do papel  $p_i$ ) para executar a atividade  $a_i$ ,  $custo(a_i, p_j) = 1 \text{ hour} \times c_j$ , o resultado da alocação – de acordo com esta estratégia – seria  $(a_1, h_1)$ ,  $(a_2, h_2)$ ,  $(a_3, h_1)$ ,  $(a_4, h_2)$ ,  $(a_5, h_3)$ ,  $(a_6, h_3)$ ,  $(a_7, h_1)$  e  $(a_8, h_4)$ .

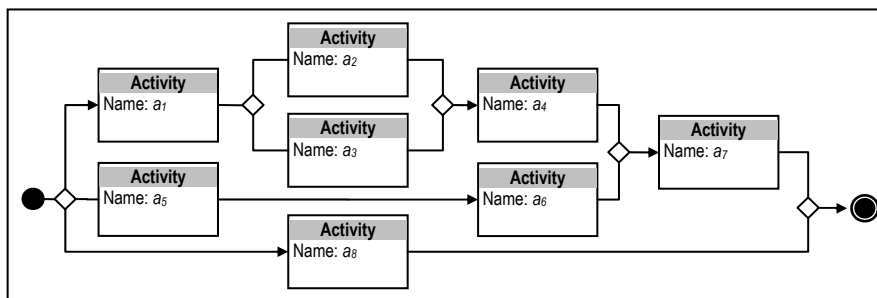


Figura 4.1. Exemplo genérico de plano de mudança refinado  $C$

Note que a partir do exemplo acima, a maioria das alocações foram realizadas utilizando recursos humanos pertencentes ao papel  $p_1$ . Tal se deve ao fato de que o custo (financeiro) por hora de um humano do papel  $p_1$  é mais barato para a organização. Porém, é possível notar que o humano alocado para  $a_5$  não pertence ao papel  $p_1$ . Isso acontece devido a todos os humanos deste papel estarem sendo utilizados em atividades paralelas a  $a_5$ . Neste caso, o humano do segundo papel mais “barato” disponível ( $p_2$ ) foi escolhido ( $h_3$ ).

Tabela 4.2. Exemplo genérico de papéis de humanos disponíveis em uma organização com os respectivos custos por hora

Papel	Operador Humano	Custo por hora (C/H)
$p_1$	$h_1$	$c_1$
$p_1$	$h_2$	$c_1$
$p_2$	$h_3$	$c_2$
$p_3$	$h_4$	$c_3$
$p_3$	$h_5$	$c_3$

O principal aspecto positivo desta estratégia é que o melhor recurso humano disponível é sempre alocado para executar atividade(s). Esta característica tem o potencial de aumentar a eficiência da execução de cada atividade e é altamente efetiva em cenários em que existe grande disponibilidade de humanos pertencentes ao “melhor” papel (que possui menor custo, por exemplo).

Entretanto, existem situações para as quais esta estratégia gulosa pode ser inconveniente. Por exemplo, supondo que  $p_1$  (papel com menor custo financeiro) tem apenas um operador disponível, o qual é alocado para a atividade  $a_8$  (assumindo, é claro, que  $a_8$  é a primeira atividade a ser analisada no plano). Neste cenário, as atividades  $a_1$ ,  $a_2$ ,  $a_4$ , e  $a_7$  (seqüência de atividades potencialmente mais demandante de esforço e paralelas a  $a_8$ ) teriam uma alocação com um operador de custo maior, podendo aumentar o custo total gasto com humanos para implantar o plano de mudança  $C$ . Outro aspecto negativo existente é que existe a possibilidade de gerar “sobre” alocações (quando um humano mais caro – considerado mais capaz de executar tarefas – é alocado para uma atividade simples) e “sub” alocações (quando operadores mais baratos – considerados menos capazes – são alocados para atividades mais complexas). Nas próximas seções serão apresentadas estratégias que buscam resolver esses aspectos negativos.

## 4.2 Alocação do melhor para o pior humano disponível com ordenação das atividades

A abordagem gulosa utilizada na estratégia apresentada anteriormente, apesar de correta, pode falhar (gerar resultados não otimizados) quando utilizada em cenários nos quais o processamento das atividades em sua ordem original (como aparecem no plano de mudança) leva a alocações de humanos com priorização de atividades simples (atividades com baixa complexidade sendo executadas por humanos altamente qualificados). Para resolver esse problema, a estratégia anterior é expandida para processar as atividades considerando os respectivos custos de implantação. Nesta



estratégia (*Estratégia 2* da Tabela 4.1), as atividades são ordenadas (de forma descendente) de acordo com os respectivos *custos médios* para executar cada atividade (detalhado a seguir). Logo após – similar ao que acontece na estratégia anterior – o algoritmo de alinhamento processa as atividades na ordem estabelecida, tentando alocar o humano do melhor papel disponível para cada atividade.

Primeiramente, para entender o *custo médio* para implantar uma atividade  $a$ , é necessário lembrar que as atividades podem ser executadas por um ou mais opções de papéis. No exemplo da Figura 3.3, a atividade *Install Squirrelmail* pode ser executada tanto por um Administrador especializado em *Webmail (Expert Webmail Admin)*, quanto por um Administrador sem especialização (*Computer Admin*). O *custo médio*, formalizado na Equação 1, corresponde à soma dos custos (tempo ou financeiro, dependendo do objetivo escolhido) de todos os papéis que podem executar a atividade  $a$  – valor dado pela fórmula *custo* para cada papel – dividido pelo número de opções de papéis.

Considerando o exemplo da Figura 3.3, o *custo médio* da atividade *Install Squirrelmail* corresponde a aproximadamente \$ 5,83  $((10*40/60+20*15/60)/2)$  ou 15 minutos  $((10+20)/2)$ . Na Equação,  $n$  é o número de opções de papéis, onde  $p_{ka}$  representa o  $k$ -th papel de operador que pode ser alocado para executar  $a$ .

$$custo_{medio}(a) = \frac{1}{n} \times \sum_{k=1}^n custo(a, p_{ka}) \quad (1)$$

Para ilustrar esta estratégia, considere novamente o plano detalhado da Figura 4.1 e os recursos humanos da Tabela 4.2. Agora assuma que (i) o objetivo de negócio do plano detalhado  $C$  é minimizar o custo financeiro total de implantação da mudança, (ii)  $c_1 < c_2 < c_3$ , (iii) todas as atividades podem ser executadas por todos os humanos de qualquer papel, e (iv)  $m_4 < m_1 < m_7 < m_6 < m_2 < m_3 < m_5 < m_8$  (onde  $m_k = custo_{medio}(a_k)$ ). Tendo como bases estas premissas, a alocação resultante é  $(a_8, h_1)$ ,  $(a_5, h_2)$ ,  $(a_3, h_3)$ ,  $(a_2, h_4)$ ,  $(a_6, h_2)$ ,  $(a_7, h_2)$ ,  $(a_1, h_3)$  e  $(a_4, h_3)$ .

Note que neste exemplo a atividade  $a_8$  possui maior prioridade no processo de alocação (por possuir *custo médio* maior que as demais atividades). Em razão disto, foi alocado a ela o humano pertencente ao melhor papel disponível na organização. Em contrapartida,  $a_4$  foi classificada com a menor prioridade, resultando na alocação do humano pertencente ao segundo melhor papel disponível,  $p_2$  (por todos os operadores do papel  $p_1$  já estarem alocados a atividades paralelas a  $a_4$ ).

A maior vantagem desta estratégia é que, na maioria dos casos, os humanos alocados tendem a atingir a capacidade (e também ter a experiência necessária) requerida pela atividade a qual foram designados a realizar (resultando na resolução de diversos casos de “sub” e “sobre” alocações). Ainda, esta estratégia é mais adequada quando o plano de mudança possui baixo grau de paralelismo, já que as tarefas paralelas destes planos têm alta probabilidade de possuírem os melhores humanos disponíveis alocados. Por outro lado, esta estratégia pode levar a “sub” alocações em cenário em que todas (ou um grande número de) atividades possuem baixo custo e alto grau de paralelismo (não conseguindo gerar alocações ótimas por não conseguir balancear corretamente as alocações dos humanos).

### 4.3 Alocação do melhor para o pior humano disponível com ordenação de atividades por *clusters*

Nesta estratégia (*Estratégia 3*), é adotada uma abordagem diferente na maneira como as atividades do plano de mudança são processadas. Ao invés de processar as atividades separadamente, estas são tratadas considerando os *clusters* de atividades (fluxos de um *workflow*) e manipuladas pelo algoritmo de alinhamento considerando a posição dentro desses *clusters*. Para ilustrar como a “clusterização” (criação de *clusters*) de atividades é realizada, considere o algoritmo apresentado na Figura 4.2.

O algoritmo, denominado *clusterizer*, recebe como entrada o plano de mudança em alinhamento ( $C$ ) e o conjunto de atividades armazenado em ( $D$ ). Como primeiro passo, um conjunto vazio de *clusters* (*Cluster*) é criado (linha 2 na Figura 4.2). Em seguida, um *cluster* unitário é criado para cada atividade em  $D$  (linhas 3-4) e copiado para dentro de *Cluster* (linha 5). Considerando o plano de mudança da Figura 4.1, o conjunto resultante de *clusters* é  $Cluster = \{\{a_1\}, \{a_2\}, \{a_3\}, \{a_4\}, \{a_5\}, \{a_6\}, \{a_7\}, \{a_8\}\}$ .

```

01  clusterizer (C, D) {
02      Cluster ← empty set of clusters
03      for each activity  $a_i \in D$  do
04          create cluster  $c_i$  containing activity  $a_i$ 
05          add cluster  $c_i$  to Cluster
06      Temp ← Cluster
07      while Temp ≠ ∅ do
08          P ← ∅
09          for each cluster  $c_i \in Temp$  do
10               $P_i \leftarrow$  activities from clusters that are parallel to  $c_i$  in C
11              if  $P_i \neq \emptyset$  then
12                  add  $P_i$  to P
13          Temp ← ∅
14          for each pair  $(P_i, P_j) \in P \mid i < j$  do
15               $P_{ij} \leftarrow P_i \cap P_j$ 
16              if  $P_{ij} \neq \emptyset$  then
17                  for each cluster  $c_k \in Cluster$  do
18                      if  $c_k \cap P_{ij} \neq \emptyset$  and  $c_k \not\subseteq P_{ij}$  and  $P_{ij} \not\subseteq c_k$  then
19                          remove cluster  $c_k$  from Cluster
20                      if no cluster  $c_k$  was removed from Cluster then
21                          add cluster  $P_{ij}$  to Temp
22          add all clusters  $c_i \in Temp$  to Cluster
23      end-while
24      sort clusters in Cluster considering cost, in descending order
25      Result ← ∅
26      for each cluster  $c_i \in Cluster$  do
27          for each activity  $a_i \in c_i \mid a_i \notin E$ 
28              add  $a_i$  to Result
29      return Result
30  }
```

Figura 4.2. Algoritmo de criação de *clusters*

No passo seguinte deste processo é criado um conjunto temporário (*Temp*), o qual vai conter todos os *clusters* criados até então (linha 6). Em seguida, enquanto houver *clusters* a serem processados em *Temp* (7), as seguintes ações são realizadas. Primeiro, um conjunto temporário  $P$  é criado (8). Posteriormente, para cada *cluster*  $c_i$  existente em *Temp* (9), as atividades contidas nos *clusters* que são paralelas a  $c_i$  (considerando a topologia do plano de mudança  $C$ ) são adicionadas a  $P_i$  (10). Logo após, se o conjunto  $P_i$  não for vazio, este é adicionado a  $P$  (11-12). Considerando o exemplo da Figura 4.1, os seguintes conjuntos de atividades  $P_i$  são criados:  $P_1 = \{a_5, a_6, a_8\}$ ;  $P_2 = \{a_3, a_5, a_6, a_8\}$ ;  $P_3 = \{a_2, a_5, a_6, a_8\}$ ;  $P_4 = \{a_5, a_6, a_8\}$ ;  $P_5 = \{a_1, a_2, a_3, a_4, a_8\}$ ;  $P_6 = \{a_1, a_2, a_3, a_4, a_8\}$ ;  $P_7 = \{a_1, a_2, a_3, a_4\}$ ; e  $P_8 = \{a_7\}$ .

O processo continua definindo o conjunto *Temp* como vazio (13). Logo após, para todo par de conjuntos  $(P_i, P_j) \in P$  (14), um *cluster* em potencial  $P_{ij}$  é criado como resultado da intersecção entre  $P_i$  e  $P_j$  (15). Se esta intersecção não for um *cluster* vazio (16), o algoritmo verifica se  $P_{ij}$  possui elementos em comum com algum outro *cluster*

pertencente ao conjunto *Cluster* (17-18). Se existir um *cluster*  $c_i$  em *Cluster* que possua elemento em comum com  $P_{ij}$ , e  $P_{ij}$  não for um subconjunto  $c_i$ , nem  $c_i$  for conjunto de  $P_{ij}$  (portanto, somente poucas serão comum a ambos *clusters*), o *cluster*  $c_i$  é removido de *Cluster* (19). No final, se nenhum  $c_i$  foi removido,  $P_{ij}$  é confirmado como um novo *cluster* criado, e adicionado a *Temp* para processamento futuro (em uma iteração subsequente do laço *while*) (21). Ainda, os novos *clusters* criados são adicionados a *Cluster* (22). No exemplo da Figura 4.1, os seguintes novos *clusters* são criados (após duas iterações):  $c_1 = \{a_5, a_6\}$ ; e  $c_2 = \{a_1, a_2, a_3, a_4\}$ . Todo este processamento (até então) é realizado para identificar os ramos de atividades do *workflow* do plano de mudança.

Após os *clusters* serem criados e adicionados, estes são ordenados (em ordem decrescente) considerando a soma do *custo médio* de todas as atividades que os compõem. (24). Para ilustrar, considere  $Cluster = \{\{a_1\}, \{a_2\}, \{a_3\}, \{a_4\}, \{a_5\}, \{a_6\}, \{a_7\}, \{a_8\}, \{a_5, a_6\}, \{a_1, a_2, a_3, a_4\}\}$ . O *custo médio* de cada *cluster* é  $\{\{m_1\}, \{m_2\}, \{m_3\}, \{m_4\}, \{m_5\}, \{m_6\}, \{m_7\}, \{m_8\}, \{m_5 + m_6\}, \{m_1 + m_2 + m_3 + m_4\}\}$  (onde  $m_k = \text{custo médio}(a_k)$ ).

Como passo final (26-28), cada *cluster* é processado em ordem, e as respectivas atividades são adicionadas em *Result* (caso ainda não tenham sido adicionadas). Em seguida, *Result* é retornado como o conjunto de atividades ordenadas para ser utilizado pelo algoritmo de alinhamento (29). Por exemplo, assumindo que  $m_1 < m_2 < m_3 < m_4 < m_5 < m_6 < m_7 < m_8 < (m_1 + m_2 + m_3 + m_4) < (m_5 + m_6)$ . Neste caso, o conjunto ordenado é  $Result = \{a_5, a_6, a_1, a_2, a_3, a_4, a_8, a_7\}$ .

O principal aspecto positivo desta estratégia é que é considerada a execução dos fluxos de atividades do plano de mudança, ou seja, considera-se a topologia na qual as atividades estão inseridas, ao invés de analisá-las separadamente. Note que  $a_5, a_6$  e  $a_1, a_2, a_3, a_4$  formam dois fluxos de execução distintos no plano de mudanças apresentado na Figura 4.1. Por este motivo, são ordenados de acordo com o seu *custo médio*, e não analisando apenas as atividades individualmente. Essa abordagem potencialmente reduz os custos dos ramos de atividades, o que normalmente diminui o custo total do plano de mudança. Em contrapartida, essa estratégia pode não resultar adequada para alocação em planos de mudança em que exista baixo grau de paralelismo (em contraste com a *Estratégia 2*, a qual possui maior eficiência nesse tipo de cenário).

#### 4.4 Alocação de humanos baseada nas preferências do Gerente de TI

Embora as estratégias introduzidas e apresentadas até então reduzam os custos associados às mudanças de forma eficiente, a *expertise* (conhecimento adquirido) dos gerentes de TI não é considerada para esboçar alocações. Baseado em sua própria experiência, por exemplo, o Gerente de TI pode desejar que um administrador *Web* configure um servidor *Apache*, ao invés de um operador de redes. Em outro cenário, pode desejar a alocação de um operador não especialista para conduzir configurações “menores” em um servidor de banco de dados, ao invés de utilizar um analista de banco de dados experiente. Para possibilitar a utilização dessa *expertise* dos gerentes, a *Estratégia 1* foi estendida para priorizar, em cada atividade, a alocação de humanos pertencentes a papéis indicados pelo Gerente de TI (operador do sistema que conduz a geração da mudança).

Essa *expertise* é representada como uma métrica (instanciada com a classe *BaseMetricDefinition*, Figura 3.2) para indicar a **preferência** dos Gerentes de TI para a alocação de cada papel (instâncias da classe *Role*, Figura 3.2). Essa preferência de

alocação pode assumir valores (representado pela classe *BaseMetricValue*) entre 0 (zero) – papel de humanos menos preferencial até 255 (duzentos e cinquenta e cinco) – papel com maior preferência. Portanto, o humano a ser utilizado pela estratégia será o que possuir o maior valor dessa métrica.

Para ilustrar como essa estratégia opera, considere novamente o plano de mudança da Figura 4.1 e os recursos humanos apresentados na Tabela 4.2. Sob as mesmas premissas adotadas na *Estratégia 1*, a alocação resultante é  $(a_1, h_1)$ ,  $(a_2, h_2)$ ,  $(a_3, h_1)$ ,  $(a_4, h_2)$ ,  $(a_5, h_3)$ ,  $(a_6, h_3)$ ,  $(a_7, h_1)$  e  $(a_8, h_4)$ . Agora considere que o gerente de TI priorizou a alocação dos humanos do papel  $p_3$  para as atividades  $a_2$  e  $a_3$ . Sob este novo cenário, temos a alocação resultante  $(a_1, h_1)$ ,  $(a_2, h_4)$ ,  $(a_3, h_5)$ ,  $(a_4, h_1)$ ,  $(a_5, h_2)$ ,  $(a_6, h_2)$ ,  $(a_7, h_1)$  e  $(a_8, h_3)$ .

O principal aspecto positivo dessa estratégia (*Estratégia 4*) reside no fato de ser disponibilizada ao gerente de TI a possibilidade de elaborar alocações que melhor se adéquem ao propósito de negócio da organização, usando o seu conhecimento adquirido com o uso de recursos humanos em mudanças passadas. Entretanto, é importante ressaltar que a alocação de humanos baseada nas prioridades estabelecidas pelo gerente de TI pode levar a planos que não estão em conformidade com os objetivos/restrições de negócio (já que o custo do plano não é o foco principal da alocação). Outro aspecto negativo é a possibilidade de ocorrer *inversão de prioridade*, ou seja, quando as preferências do gerente de TI não são alcançadas (devido à exaustão dos recursos humanos qualificados) para atividades que possuam alto impacto – em termos financeiros ou de tempo necessário para execução. Esse problema, como discutido no caso da *Estratégia 1*, ocorre principalmente por se considerar a ordem das atividades como são encontradas no plano (sem ordenação prévia).

#### **4.5 Alocação de humanos baseada nas preferências do Gerente de TI com ordenação de atividades**

Esta estratégia (*Estratégia 5*) incorpora as heurísticas propostas na *Estratégia 2* (ordenamento das atividades) e *Estratégia 4* (preferências dos Gerentes de TI), o que permite que o gerente de TI possa influenciar na escolha dos papéis de humanos a serem utilizados nas atividades que possuem maior impacto (no nosso caso, atividades com alto *custo médio*) no plano de mudança. Esse é um aspecto de grande importância, especialmente em mudanças que possuem muitas atividades de alto impacto e com cenários onde existe baixa disponibilidade de recursos humanos altamente qualificados.

Contraopondo os aspectos positivos mencionados, essa estratégia pode se tornar ineficaz para mudanças com grande número de atividades com baixo *custo médio*, o que pode levar a alocações inadequadas por desperdiçar recursos importantes com atividades simples.

#### **4.6 Alocação de humanos baseada nas preferências do Gerente de TI com ordenação de atividade por *Clusters***

Esta estratégia (*Estratégia 6*) estende a *Estratégia 3*, incorporando a possibilidade dos gerentes de TI influenciarem o processo de alocação de humanos. Note que a ordem das atividades através do método de “clusterização” não é alterada, já que o processo de alocação de humanos é feito posteriormente à escolha de atividades.

Em contraste com a *Estratégia 5*, as atividades com menor custo, porém em um fluxo de execução, possuem maior probabilidade de receberem uma alocação mais adequada, conforme a indicação do Gerente de TI. Entretanto, nota-se que as preferências por atividades de maior impacto (em isolado) podem não ser consideradas em planos contendo baixo grau de paralelismo.

Outro ponto negativo desta estratégia é a maior probabilidade de acontecer uma inversão de prioridades. Por exemplo, considere que a atividade  $a_5$  represente o maior impacto no plano de mudança da Figura 4.1, e a atividade  $a_6$  represente o menor impacto. Com essas premissas, o fluxo composto por essas atividades possuirá o maior *custo médio* do plano, e a atividade  $a_6$  acabará recebendo maior prioridade de alocação do que as demais atividades, mesmo que possuindo o menor *custo* entre as atividades.

## 4.7 Resumo do Capítulo

Em síntese, o Capítulo 4 apresentou seis estratégias para a alocação de humanos a atividades, propostas no contexto desta dissertação. De modo geral, observou-se que os custos estimados e qualidade das alocações de humanos para a execução dos planos podem ser diferentes para cada estratégia utilizada.

Intuitivamente, a se confirmar nos estudos de caso do capítulo 6 (Avaliação Experimental), espera-se que, na maioria dos casos, a *Estratégia 3* resulte em planos mais otimizados, principalmente em cenários com alto grau de paralelismo (muitas atividades em paralelo), pois leva em consideração o custo de todas as atividades em cada ramo paralelo. Por sua vez, espera-se que a *Estratégia 2* gere resultados mais otimizados quando houver baixo grau de paralelismo (poucas atividades em paralelo) com atividades de alta complexidade (alto custo de execução), por priorizar as atividades pelos custos. Ainda, para a maioria dos casos, a *Estratégia 5* deve gerar os planos com alocações mais adequadas considerando as habilidades/capacidade produtiva dos humanos, por priorizar a alocação de humanos para as atividades mais complexas (e normalmente mais relevantes). Por outro lado, a *Estratégia 1* deve gerar os piores resultados em termos de custos de execução dos planos de mudança, pois tende a efetuar alocações sem critérios de ordenação das atividades. Essas afirmações, de certa forma, serão confirmadas nos estudos de caso do Capítulo 6 (Avaliação Experimental).

A seguir, no Capítulo 5, é apresentado o protótipo desenvolvido para o alinhamento dos planos de mudança aos objetivos/restrições de negócio. Esse protótipo, chamado de CHANGEADVISOR, implementa os modelos e algoritmos apresentados nos capítulos anteriores, bem como possibilita a escolha das estratégias, apresentadas no presente Capítulo, a serem utilizadas durante o processo alinhamento do plano de mudança.

## 5 O SISTEMA CHANGEADVISOR

A solução para o planejamento de mudanças em infraestruturas de TI, apresentada nos capítulos anteriores, é materializada pelo protótipo de um sistema de suporte a decisão denominado CHANGEADVISOR. Esse sistema está acomodado no contexto do CHANGELEDGE (CORDEIRO *et al.*, 2008a), um *framework* para a gerência de mudanças com foco no reuso de conhecimento e automação. Neste capítulo serão apresentados alguns dos aspectos principais do sistema implementado, as interfaces do mesmo com os componentes conceituais do *framework* CHANGELEDGE, e a interface gráfica implementada para facilitar a utilização e a interação com o usuário.

O sistema CHANGEADVISOR, implementado utilizando a linguagem de programação *Java*, materializa as funcionalidades dos componentes *ChangeRefiner* e *ChangeAligner* (destacados na área cinza da Figura 3.1). As especificações de mudança consumidas pelo sistema são codificadas utilizando a linguagem XML, e processadas por meio da *Streaming API for XML* (StAX) (StAX, 2010). A partir das especificações recebidas, o sistema gera, como saída, planos de mudança detalhados (*workflows* executáveis) codificados em XML e respeitando o padrão *Business Process Execution Language* (BPEL) (OASIS, 2010). É importante mencionar que a escolha por BPEL se deve à grande popularidade do padrão e a sua aderência para coordenar atividades distribuídas em infraestruturas de redes de computadores.

A manipulação de dados sobre a infraestrutura gerenciada, além da obtenção de informações sobre os pacotes de *software* disponíveis para o processo de mudança, é feita utilizando o *framework* para mapeamento objeto/relacional Hibernate (Hibernate, 2010). A persistência dos dados manipulados, por sua vez, é realizada pelo sistema de gerência de banco de dados MySQL (MySQL, 2010). Para desacoplar as funcionalidades do sistema CHANGEADVISOR dos aspectos relativos à persistência e recuperação de dados, foi adotado o padrão de projeto *Data Access Object* (DAO).

A comunicação entre os componentes conceituais do *framework* CHANGELEDGE e o sistema CHANGEADVISOR é feita por meio de interfaces e padrões bem definidos e entendidos por ambas as partes. Entre as interfaces existentes, merecem destaque as que comunicam o sistema com os componentes *Change Designer* e *Deployment System*. No primeiro caso, o *Change Designer* gera documentos especificados em XML, adotando como padrão o mesmo XML *Schema* que o CHANGEADVISOR. No segundo caso, os documentos BPEL gerados pelo CHANGEADVISOR são consumidos pelo orquestrador ActiveBPEL (2010), o qual implementa as funcionalidades do *Deployment System*. O leitor interessado em informações adicionais sobre a execução dos *workflows* e a interface necessária nos CIs (itens de configuração) para sua manipulação remota poderá encontrá-las em trabalho anterior publicado recentemente pelo nosso grupo de pesquisa (MACHADO *et al.*, 2008).

## 5.1 Interface gráfica do CHANGEADVISOR

Com o objetivo de facilitar a condução do processo de alinhamento de planos de mudança, foi implementada uma interface gráfica amigável utilizando a biblioteca Java/Swing GUI Builder (2010). A Figura 5.1 ilustra a interface apresentada ao operador do sistema CHANGEADVISOR para conduzir processos de alinhamento. É importante destacar que as RFCs (requisições de mudança) e os respectivos planos de mudança refinados são funcionalidades oferecidas pelo sistema CHANGELEDGE (CORDEIRO *et al.*, 2008a), (CORDEIRO *et al.*, 2008b), (CORDEIRO *et al.*, 2009) e, portanto, são aqui omitidas.

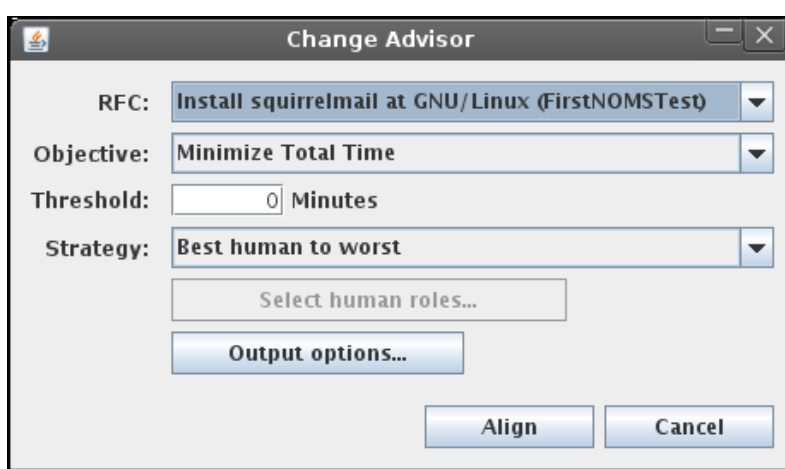


Figura 5.1 Interface gráfica do CHANGEADVISOR

A primeira opção disponível para interação é a seleção da requisição de mudança (RFC) a ser alinhada. Como exemplo, na Figura 5.2 o gerente de TI (responsável por conduzir o alinhamento da mudança) pode selecionar uma de quatro RFCs.

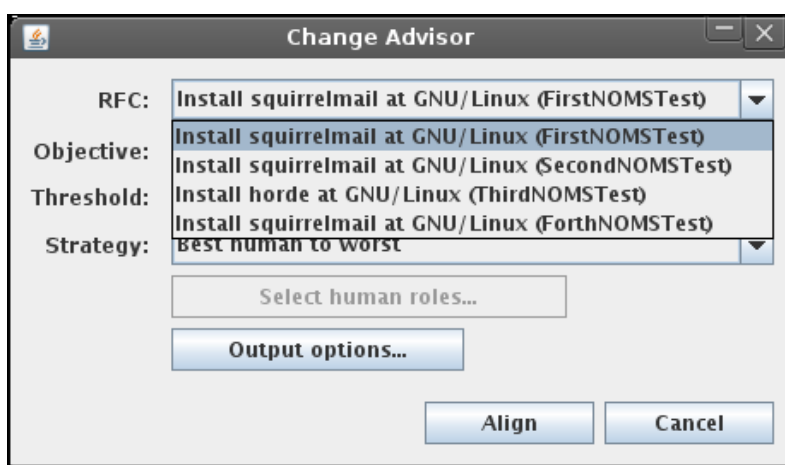


Figura 5.2 Seleção de RFC na interface do CHANGEADVISOR

O próximo item disponível para interação é a seleção de qual objetivo/restrrição de negócio (*Objective*) deverá ser seguido no processo de alinhamento. Como demonstrado na Figura 5.3, existem dois objetivos disponíveis: (i) diminuir o tempo total para executar a mudança (*Minimize Total Time*); ou (ii) reduzir o custo financeiro

despendido com os recursos humanos (*Minimize Financial Cost*). Após selecionar o objetivo de negócio, é possível definir qual o valor máximo (*Threshold*) tolerável para este objetivo. Por exemplo, caso seja selecionado o objetivo de reduzir o tempo total de execução do plano de mudança (*Minimize Total Time*), o valor máximo será representado em minutos (*minutes*), como demonstrado na Figura 5.4 (a). Por outro lado, caso o objetivo selecionado for o de reduzir o custo financeiro despendido com humanos (*Minimize Financial Cost*), o valor máximo será representado em dólares (*Dollars*), como demonstrado na Figura 5.4 (b).

Logo em seguida, o gerente de TI pode escolher uma estratégia - dentre as seis disponíveis - para efetuar o alinhamento do plano de mudança, como demonstrado na Figura 5.5.

Após a escolha da estratégia a ser utilizada, é possível definir também o número de humanos de cada papel (*Select human roles...*) que podem ser utilizados para conduzir as atividades manuais do plano de mudança. Os humanos disponíveis por papéis são apresentados em uma nova janela na qual o gerente pode inserir ou retirar participantes, como demonstrado na Figura 5.6.

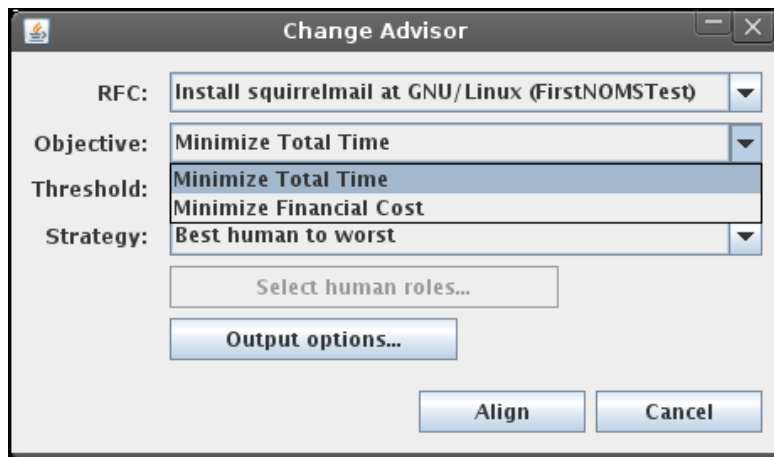


Figura 5.3 Seleção de objetivo/restrrição de negócio no CHANGEADVISOR

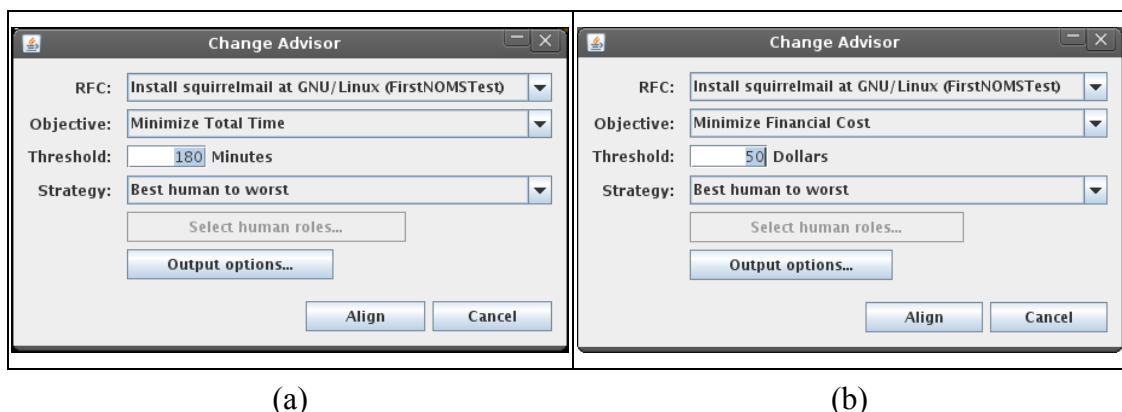


Figura 5.4 Definição do valor máximo em minutos e em dólares no CHANGEADVISOR



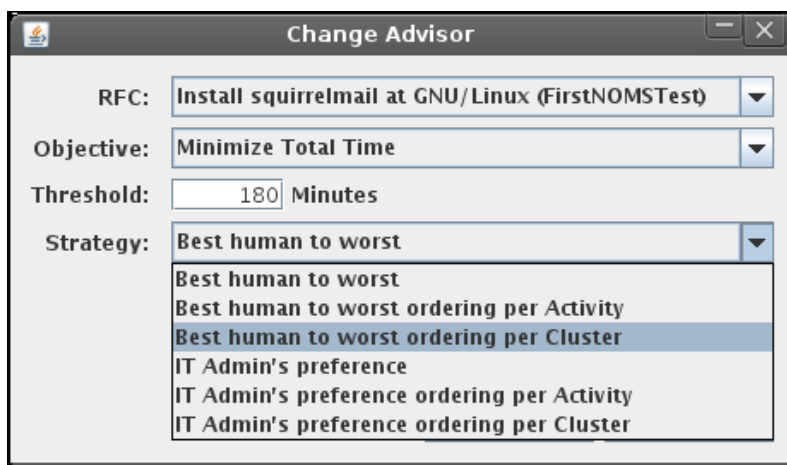


Figura 5.5 Escolha da estratégia de alocação de humanos a ser adotada no CHANGEADVISOR

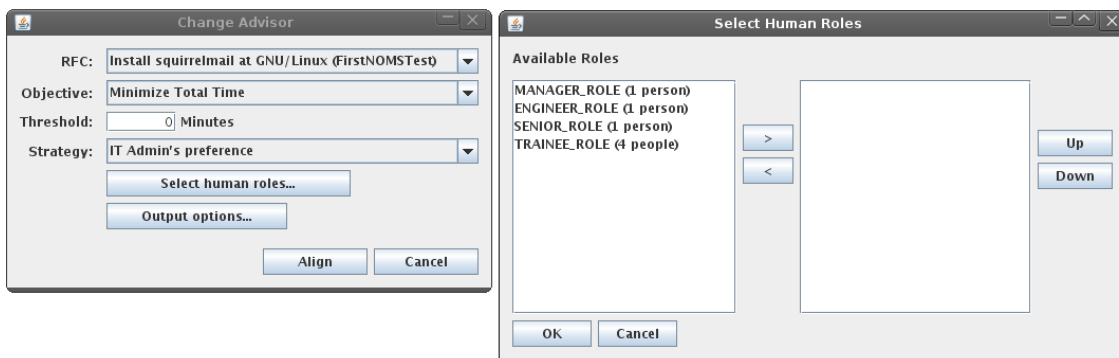


Figura 5.6 Escolha do número de humanos por papel para conduzir as atividades no CHANGEADVISOR

Como última opção disponível, existe a possibilidade de se selecionar um de dois formatos de saída para planos de mudança gerados (Figura 5.7). A primeira é uma saída gráfica no formato de imagem “png” e a outra, saída compatível com o *software* de edição Dia (DIA, 2010). Além disso, é possível escolher entre uma versão resumida, apenas mostrando atividades em mais alto nível (*Colapsed*), ou uma versão completa (Figura 5.8) das atividades e suas respectivas dependências (*Expanded*).

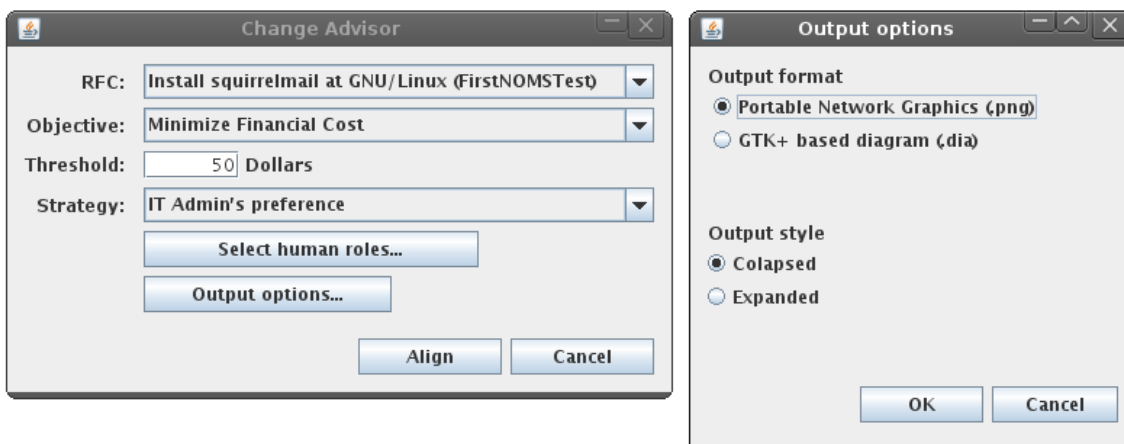


Figura 5.7 Escolha do formato de saída e nível de detalhamento no CHANGEADVISOR

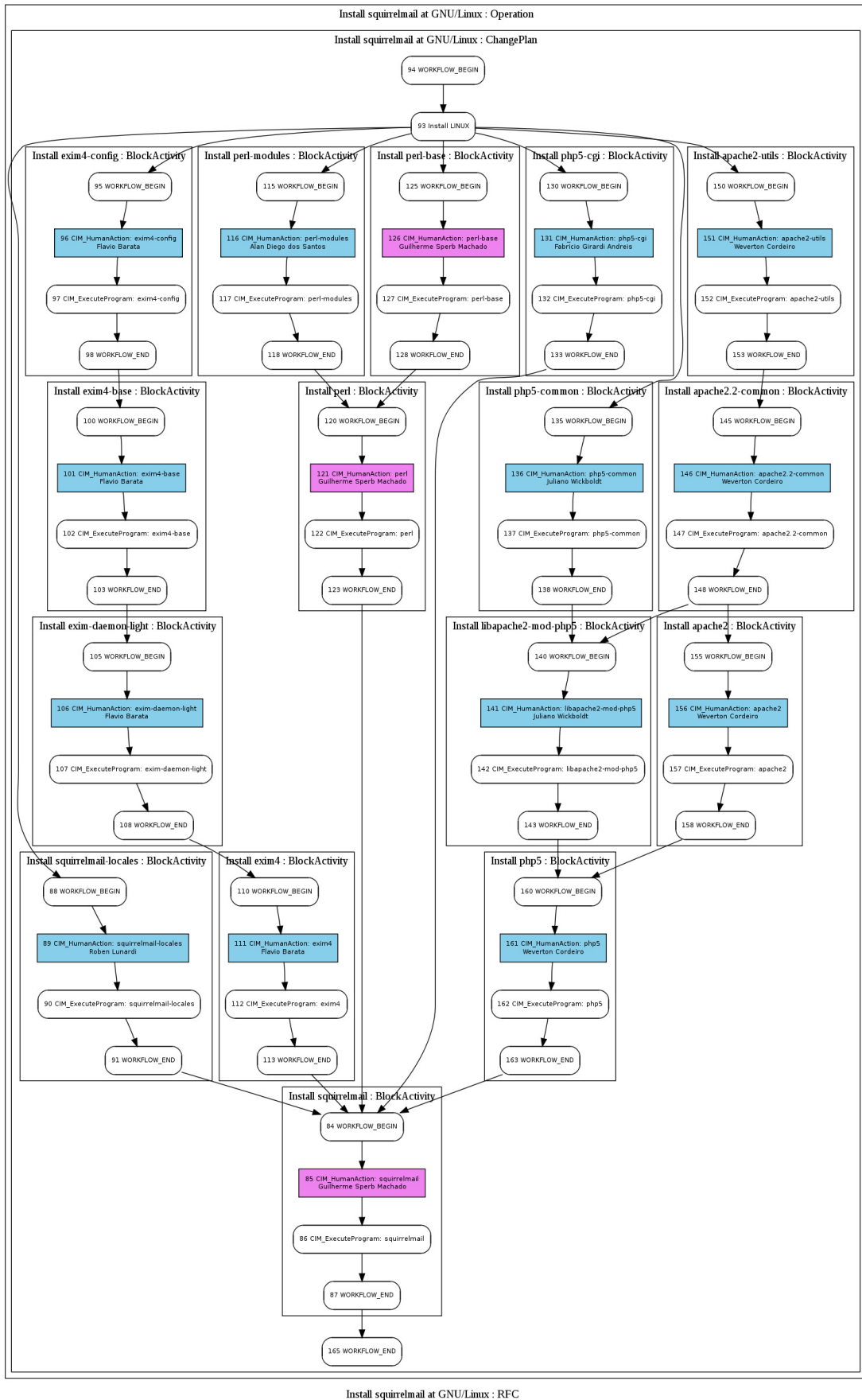


Figura 5.8 Exemplo de plano de mudança alinhado gerado pelo CHANGEADVISOR

## 5.2 Resumo do Capítulo

Em síntese, o Capítulo 5 apresentou os principais detalhes de implementação do protótipo para a geração de planos de mudança alinhados a objetivos/restrições de negócio, chamado CHANGEADVISOR. Os principais módulos da solução foram implementados utilizando a linguagem JAVA, a persistência dos dados é consolidada utilizando MySQL e a manipulação dos dados é realizada via Hibernate.

Além disso, foi implementada uma interface gráfica amigável, utilizando Java/Swing, para facilitar a condução do processo de alinhamento de planos de mudança. Essa interface permite escolher os objetivos/restrições de negócios e seus respectivos valores máximos, as estratégias de alocação de humanos a serem utilizadas, os humanos envolvidos no processo e a forma de visualização dos planos gerados, bem como os respectivos relatórios comparativos.

No Capítulo 6, o protótipo desenvolvido é utilizado para conduzir a avaliação experimental. Nessa avaliação serão analisados dois estudos de caso, para demonstrar a eficiência e eficácia da solução e do protótipo desenvolvido.

## 6 AVALIAÇÃO EXPERIMENTAL

Para avaliar a viabilidade técnica do planejamento de mudanças em TI alinhado a propósitos de negócio, foram conduzidos dois estudos de caso utilizando o sistema CHANGEADVISOR. No primeiro estudo de caso comparou-se a alocação de humanos sem e com a nossa solução proposta, com o objetivo de demonstrar a importância do alinhamento dos planos de mudança aos objetivos/restrições de negócio. No segundo estudo de caso foram analisados diferentes planos de mudança. Cada plano foi gerado utilizando as seis estratégias de alocação de humanos, propostas anteriormente no Capítulo 4. O objetivo deste segundo estudo de caso é comparar resultado da utilização das estratégias propostas em diferentes cenários (*workflows* com topologias e custos diferentes). Como resultado geral de ambos os estudos de caso, foi observado o alinhamento dos planos de mudança detalhados aos propósitos de negócio (caracterizando uma análise mais qualitativa da solução proposta), bem como apresentados indicadores de desempenho e adequação dos humanos às atividades (análise quantitativa). A seguir serão apresentados os dois estudos de caso, bem como os resultados obtidos.

### 6.1 Primeiro estudo de caso

A infraestrutura de TI empregada na avaliação é composta por 2 servidores, *labcom* e *noc*, sendo que apenas o servidor *labcom* tem um sistema operacional instalado, o *Debian GNU/Linux*. Há também três papéis de operadores disponíveis para conduzir a execução das mudanças, cujos custos financeiros relativos a sua alocação são descritos na Tabela 6.1. De acordo com a Tabela 6.1, considerando o custo de \$40 para um administrador *Expert Webserver Admin*, se esse humano for alocado por meia hora em uma atividade, tal alocação implicará na adição de \$20 para o custo financeiro de execução da respectiva mudança.

Tabela 6.1. Papéis disponíveis para alocação de humanos para os estudos de caso

Papel	Número de operadores disponíveis	Custo de Alocação (\$ / hora)
<i>Expert Webserver Admin</i>	1	40
<i>Junior Webserver Admin</i>	1	25
<i>Computer Admin</i>	3	12

Por fim, a Tabela 6.2 descreve os cenários empregados na avaliação. Nos cenários apresentados existe a mesma necessidade: migrar um serviço de e-mail do servidor *noc* para o servidor *labcom*. Essa necessidade pode ser representada pela requisição de mudança (RFC) e no respectivo plano preliminar de mudança apresentados

anteriormente na Figura 3.4. Além disso, na Tabela 6.2 são apresentados os objetivos/restrição de negócio para cada cenário. No primeiro cenário, o objetivo é que a execução do plano de mudança não ultrapasse 150 (cento e cinquenta) minutos. Por sua vez, no segundo cenário, existe uma restrição financeira a ser despendida com humanos de 30 dólares. Finalmente, no terceiro cenário, não existe nenhum objetivo/restrição de negócio a mudança solicitada.

Tabela 6.2. RFC e objetivos/restrições de negócio empregado no estudo de caso 1

Cenário	Finalidade	Objetivo/Restrição Associada	Valor de Limiar
1	Migrar Serviço de e-mail do Servidor <i>labcom</i> para <i>noc</i>	Diminuir Tempo Total de Implantação	150
2	Migrar Serviço de e-mail do Servidor <i>labcom</i> para <i>noc</i>	Diminuir Custo com Alocação de Operadores	30
3	Migrar Serviço de e-mail do Servidor <i>labcom</i> para <i>noc</i>	Nenhum	-

Uma visão parcial dos planos gerados para a RFC de migração do serviço de e-mail, nos cenários 1 e 2, é apresentada nas Figuras 3.6 (página 28) e 6.1, respectivamente. Para facilitar a visualização, devido ao tamanho dos planos, o refinamento das atividades *Install Debian GNU/Linux* (Figura 3.6) e *Install Windows 2003 Server* (Figura 6.1) é suprimido em ambas as figuras. Ademais, estruturas de decisão são omitidas para facilitar a sua legibilidade. A ligação entre as atividades presentes nos planos ilustrados reflete a dependência entre os pacotes de *software* instalados, conforme identificado na DML. Por exemplo, para a instalação do *apache2* no *Debian GNU/Linux*, é necessário que as bibliotecas básicas do mesmo, providas pelo *apache2-common*, já estejam instaladas.

A partir das figuras apresentadas, observam-se as diferentes alocações de papéis para a execução das atividades *manuais*. Por exemplo, no cenário 1, a execução do plano de mudança requer três operadores com papéis distintos (um *Expert Webserver Admin*, um *Junior Webserver Admin* e um *Computer Admin*) para que o objetivo especificado seja atingido. O plano do cenário 2, por sua vez, requer apenas um operador (*Expert Webserver Admin*), de modo a satisfazer o objetivo especificado.

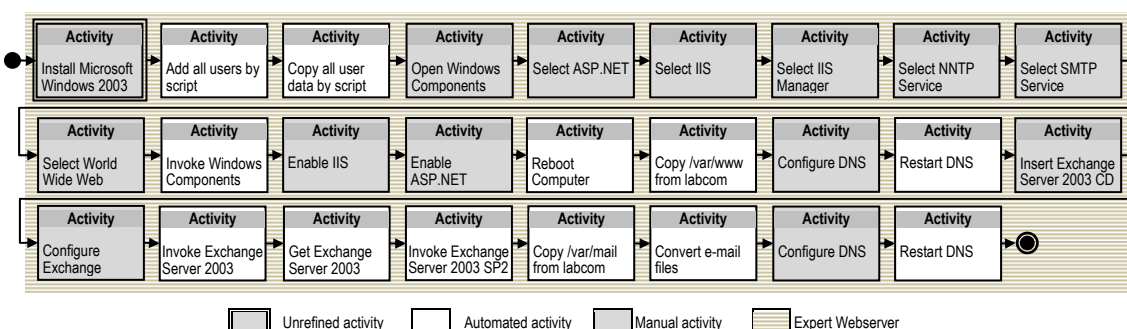


Figura 6.1. Plano de mudança alternativo para RFC de migração de servidor de e-mail para o estudo de caso 1

A Tabela 6.3 apresenta os custos associados à execução dos planos detalhados gerados para cada um dos cenários avaliados. Nessa tabela é possível verificar, comparativamente, que os planos detalhados são capazes de prover melhores resultados face aos objetivos/restrições especificados. Por exemplo, pôde ser concretizada por dois

planos distintos: um que possui tempo de execução baixo (118 minutos), embora apresentando alto custo de implantação (\$34,21), e o outro que possui custo razoável (\$7,75), apesar de demandar mais tempo para sua execução (156 minutos).

A Tabela 6.3 apresenta também, para efeito comparativo, plano alternativo não alinhado aos objetivos/restrições. Embora corretos, planos como esse tendem a apresentar custos maiores em comparação às alternativas que seguem algum alinhamento. Por exemplo, o plano gerado para o cenário 3 tem custo com alocação de humanos estimado em \$45,98, valor 493% maior que o plano alinhado ao objetivo de minimização de tal custo. Tal pode ser explicado, por exemplo, pelos diferentes CIs envolvidos na mudança, em virtude dos diferentes refinamentos feitos para um mesmo plano preliminar. Esses, por sua vez, podem eventualmente implicar em atividades com diferentes tempos de duração, bem como em diferentes alocações de humanos (de papéis distintos). Note que os valores de custos obtidos, se analisados individualmente, podem ser desprezíveis. No entanto, ao considerar a execução de diversas mudanças no longo prazo, alinhamentos têm potencial para contribuir para a redução de custos de gerência e operação da empresa.

Tabela 6.3. Número de atividades, tempo estimado de execução e custos de alocação para os planos detalhados gerados no estudo de caso 1

Cenário	Atividades	Tempo Estimado de Execução (min)	Custo com Alocação de Humanos (\$)
1	69	118,75	34,21
2	38	156,12	7,75
3	69	157,08	45,98

O desempenho do sistema CHANGEADVISOR para gerar os planos executáveis caracterizados acima é apresentado na Tabela 6.4. Executado em um computador equipado com um processador Intel® Core™ 2 Duo 2.33 GHz, 4.096 KB de memória *cache* e 2 GB de memória RAM, o sistema mostrou-se bastante eficiente, exigindo poucos segundos para gerar os planos. Também foi calculado um intervalo de confiança de 95% sobre os tempos medidos, considerando 10 repetições do processo de geração de planos de mudança. Conforme apresentado na Tabela 6.4, é esperado que o tempo de geração de planos varie minimamente para cada cenário. Esses resultados permitem observar que a solução, além de gerar planos alinhados a propósitos de negócio, tem potencial para reduzir, sensivelmente, o esforço e o tempo exigidos para tal.

Tabela 6.4. Tempo despendido na geração automática dos planos detalhados e alinhados no estudo de caso 1

Cenário	Tempo Total de Geração dos Planos Detalhados (ms)	Intervalo de Confiança do Processo de Geração de Planos Detalhados	
		Limite Inferior (ms)	Limite Superior (ms)
1	17.475	17.247	17.703
2	14.185	14.030	14.340
3	16.761	16.680	16.842

## 6.2 Segundo estudo de caso

No segundo estudo de caso foram realizados refinamentos/alinhamentos para a mesma RFC e (considerando) a mesma infraestrutura de TI utilizados na seção anterior. Basicamente, analisou-se possibilidades de alinhamento de quatro planos de mudança detalhados gerados para a referida RFC. A ênfase aqui foi analisar a qualidade dos planos gerados e os resultados obtidos com o emprego das seis estratégias de alinhamento propostas no Capítulo 4.

Os diferentes planos detalhados, gerados para avaliar as diferentes estratégias, materializam a RFC de migração de um serviço de *Webmail* de um servidor para outro (*labcom* para *noc*). Desta vez, contudo, foram utilizados dois objetivos/restrições de negócio para o mesmo processo de alinhamento (ao mesmo tempo): o tempo total para executar o plano gerado deveria ser de no máximo 150 minutos e o custo monetário gasto com recursos humano deveria ser inferior a \$30 (trinta unidades monetárias). As informações relativas aos planos detalhados, que são relevantes ao escopo desta avaliação, são descritos na Tabela 6.5.

Tabela 6.5. Características dos planos detalhados avaliados no estudo de caso 2

Cenário	Atividades	Programas utilizados
1	69	OS Debian Linux, Apache2, PHP5, Exim4, Squirrelmail
2	60	OS Debian Linux, Apache2, PHP5, Postfix, Squirrelmail
3	69	OS Debian Linux, Apache2, PHP5, Horde
4	69	OS Debian Linux, Apache2, PHP5, Sendmail, Squirrelmail

A Tabela 6.6 apresenta o custo associado à execução de cada um dos diferentes planos apresentados na Tabela 6.5, considerando cada estratégia apresentada no Capítulo 4. São ressaltados com fundo preto (e letras brancas) os planos com melhor alocação de humanos para executar as atividades do plano de cada cenário. Essa avaliação é realizada considerando habilidades/esforço requerido pelas atividades e habilidades/experiência dos humanos disponíveis (materializadas pelas preferências dos gerentes de TI). Além disso, foram sublinhados os planos com melhores custos para cada plano de mudança, considerando os propósitos de negócio definidos na RFC.

Tabela 6.6. Tempo total de execução e custo de alocação de humanos estimados para os planos detalhados avaliados no estudo de caso 2

Estratégia	Cenário 1		Cenário 2		Cenário 3		Cenário 4	
	Minutos	\$	Minutos	\$	Minutos	\$	Minutos	\$
1	107,74	22,19	86,49	15,25	120,69	21,20	83,49	15,50
2	107,74	22,19	83,49	15,45	<u>116,57</u>	<u>21,00</u>	83,49	15,50
3	<u>100,37</u>	<u>22,10</u>	<u>78,24</u>	<u>15,00</u>	120,69	21,20	<u>70,99</u>	<u>15,60</u>
4	128	23,30	109	17,60	126,58	21,50	100,5	17,55
5	118,34	21,00	92,10	15,50	131,16	18,00	93,34	13,55
6	117,9	21,20	91,72	15,60	133,07	19,40	92,97	17,50

Como demonstrado na Tabela 6.6, o melhor custo, em relação ao tempo total de execução estimado, é obtido com o plano detalhado no *Cenário 4* utilizando a *Estratégia 3* (70,99 minutos). Por sua vez, o menor (melhor) custo monetário despendido com humanos é obtido pelo *Cenário 2*, também utilizando a *Estratégia 3* (\$15,00). Note, entretanto, que os planos alinhados que possuem alocações mais adequadas em relação às preferências dos gerentes de TI (ressaltados com fundo preto) não são os mesmos que possuem os melhores custos de execução (sublinhados). Por exemplo, estima-se que o plano detalhado do *Cenário 2*, alinhado com a *Estratégia 5*, necessita de aproximadamente 14 minutos a mais para ser executado que o plano gerado pelo *Cenário 2* utilizando a *Estratégia 3*, e precisa de 22 minutos a mais que o plano do *Cenário 4* alinhado com a *Estratégia 3* (plano que possui melhor custo em relação ao tempo de execução estimado dentre todos os planos alinhados). Observe que esta comparação é válida, já que estamos comparando as diferentes estratégias utilizando diversos planos alinhados que materializam a mesma requisição de mudança (RFC).

Em geral, a estratégia que gerou os planos alinhados com melhores custos, na maioria dos cenários, é a *Estratégia 3*. Isto acontece, pois ao reduzir o custo do ramo mais custoso tem maior potencial de diminuir o custo de todo o plano de mudança. É importante mencionar, ainda, que nem sempre o melhor custo é observado nas *Estratégias 4, 5, e 6*. Isso ocorre porque as alocações de humanos preferenciais dos gerentes de TI podem não necessariamente coincidir com custos baixos. Por outro lado, essas estratégias tendem a gerar planos de mudança com melhor qualidade, ou seja, alocações mais adequadas em relação às habilidades dos humanos disponíveis e às necessidades de cada atividade do *workflow*.

Em relação à qualidade das alocações de humanos às atividades, a *Estratégia 5* gerou planos alinhados mais harmoniosos (de acordo com as preferências dos gerentes de TI) para a maioria dos cenários, por analisar primeiro as atividades mais custosas, e conseqüentemente, as que possuem maior impacto na implantação do plano de mudança. Ainda sobre qualidade, a *Estratégia 4* foi prejudicada pela *inversão de prioridade (priority inversion)* (SHA, 1990), ou seja, as atividades são processadas na ordem original em que aparecem no plano de mudança e, por conseguinte, há grande probabilidade de um humano muito importante acabar sendo assinalado para executar uma tarefa que não possui grande prioridade ou, então, um humano com baixa qualificação ser alocado para atividades de grande relevância. Por outro lado, a *Estratégia 6* considera os ramos de atividades mais custosos. Ou seja, ao efetuar alocação dos humanos priorizará conjuntos de atividades e será eficiente quando diversas atividades pequenas de um ramo forem resultado do refinamento de uma atividades de alto nível muito importante, exemplo ocorrido no *Cenário 4*.

Para exemplificar e apresentar mais detalhes sobre como as estratégias atuam, as Figuras 6.2 e 6.3 apresentam uma visão parcial de dois dos diversos planos de mudança detalhados e alinhados gerados. O plano da Figura 6.2 representa a alternativa que possui melhor custo em termos de tempo estimado de execução e (aproximadamente o menor) custo financeiro despendido com a alocação de recursos humanos. Já o plano da Figura 6.3 ilustra a alternativa que gera a utilização mais adequada dos humanos em termos da relação habilidades/experiência dos operadores e complexidade/impacto das atividades.



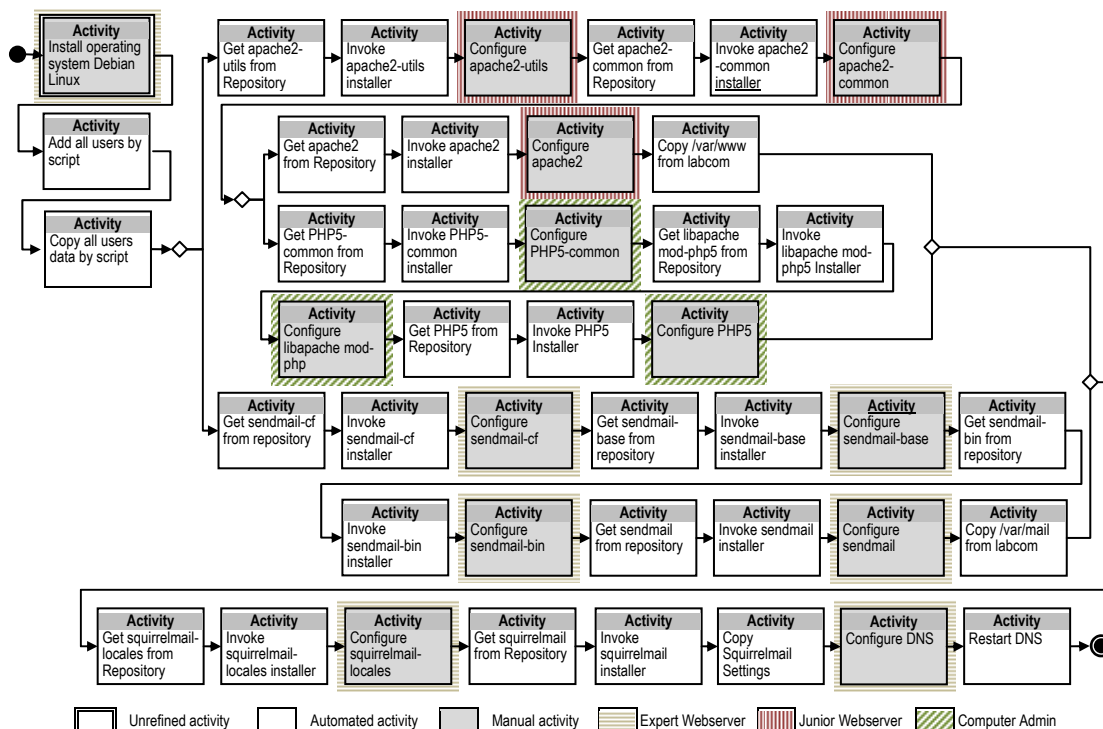


Figura 6.2. Plano de mudança do *Cenário 4* alinhado considerando a *Estratégia 3*

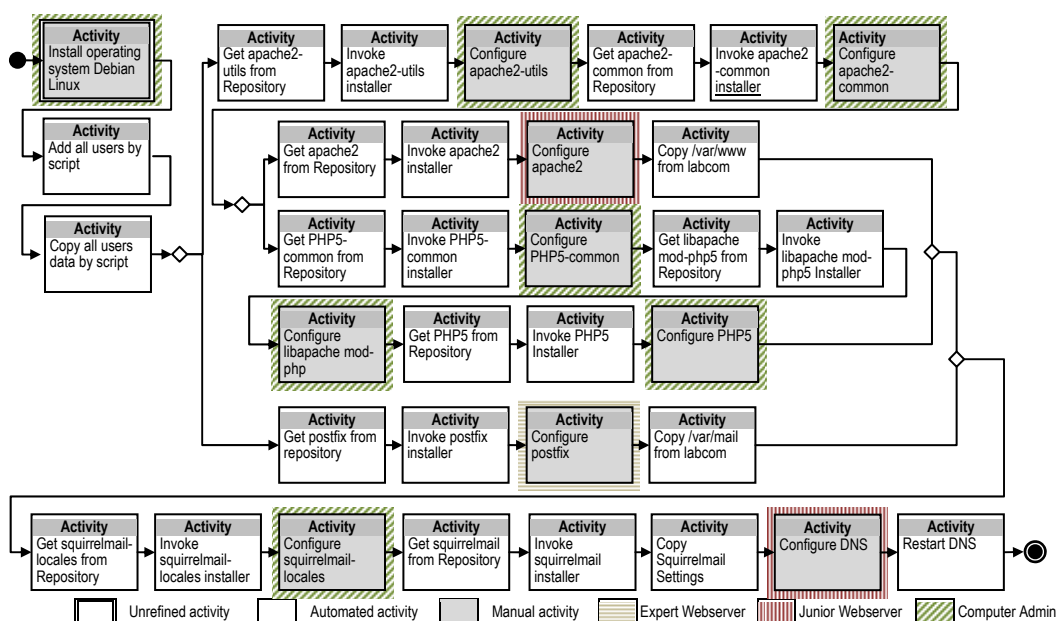


Figura 6.3. Plano de mudança do *Cenário 2* alinhado considerando a *Estratégia 5*

Note, analisando a Figura 6.2, que a *Estratégia 3* gerou alocações não alinhadas com as necessidades das atividades e a capacidade dos recursos alocados: um humano do papel *Expert Webserver Admin* foi alocado para atividades de baixo impacto, como configuração das bibliotecas do *Sendmail* (onde são necessários apenas pequenos ajustes de parâmetros). Ainda, um humano associado ao mesmo papel (*Expert Webserver Admin*) é assinalado para instalar e configurar o sistema operacional *Debian Linux*, uma tarefa que deveria ser assinalada a um humano de papel mais apropriado,

como, por exemplo, um *Computer Admin*. Em contraste, no plano da Figura 6.3, a todas as atividades foram assinalados humanos com papéis adequados.

Note, também, que apesar do plano da Figura 6.2 possuir maior número de atividades, o tempo estimado de execução das atividades é menor que o do plano gerado pela Figura 6.3. Isto se deve ao fato de que apenas 10% das atividades manuais são executadas por um *Expert Webserver Admin* – que por ser mais experiente, tende a ser mais eficiente na execução das tarefas – no plano da Figura 6.3 (1 de 10 atividades manuais representadas). Esta porcentagem aumenta para aproximadamente 53% no plano da Figura 6.3 (7 das 13 atividades manuais representadas).

### 6.3 Resumo do Capítulo

No Capítulo 6 foram apresentados dois estudos de caso para validar a solução proposta nesta dissertação. No primeiro, a ênfase foi demonstrar a importância de alinhar planos de mudança a objetivos/restrições de negócio. Por sua vez, no segundo estudo de caso, buscou-se explorar os resultados obtidos utilizando as diferentes estratégias propostas nesta dissertação. Como resultados interessantes da avaliação experimental (em ambos os estudos de caso), podem ser citadas a eficiência (baixo tempo para gerar o alinhamento, sempre na casa de dezenas de segundos) e a eficácia (geração de planos alinhados aos objetivos/restrições de negócio) da solução.

Vale ressaltar que, tendo conhecimento dos custos estimados, o gerente de TI possui informação suficiente para avaliar quão barato ou cara é uma alternativa em comparação a outra, ou qual das opções produz alocações mais adequadas (melhor utilização dos humanos). É importante enfatizar que apesar da possibilidade dos custos estimados pela nossa solução serem imprecisos, eles continuam úteis para comparar os “compromissos” (*trade-offs*) entre os planos gerados. Essa “imprecisão” se deve a fatores externos à nossa solução, tais como falta de atualização dos repositórios e alterações no ambiente (nem sempre os custos serão próximos da média calculada).

No Capítulo 7, serão apresentadas as principais conclusões, bem como os resultados obtidos pela avaliação experimental. Além disso, serão apresentados questões em aberto e possíveis investigações futuras.

## 7 CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

O planejamento de mudanças representa uma etapa fundamental para a gerência e a operação de infraestruturas de TI. No entanto, as soluções propostas anteriormente para auxiliar essa etapa não levam em consideração a geração de planos de mudança alinhados aos propósitos definidos em nível de negócio. Conseqüentemente, o planejamento e a execução de mudanças apresentarão pouca (ou nenhuma) sintonia com os objetivos/restrições de negócio demandados pelo solicitante. Por outro lado, mesmo que os planos estejam alinhados, existe a possibilidade de serem gerados planos não otimizados, ou com alocações de humanos a atividades longe do que um gerente de TI desejaria.

Portanto, a alocação adequada dos recursos humanos às atividades de mudança representa um aspecto crucial para conduzir e executar mudanças na infraestrutura de TI gerenciada. Para abordar esse problema, foram propostos modelos e algoritmos, materializados no CHANGEADVISOR, uma solução para geração - com alto grau de automação - de planos de mudança consistentes com os objetivos/restrições especificados em nível de negócio. Esta solução proporciona suporte para entender os compromissos da utilização de diferentes estratégias de alocação de humanos em tarefas de mudanças em TI, além de gerar/indicar diversas opções de planos de mudança alinhados aos objetivos/restrições de negócio.

### 7.1 Resultados obtidos

Os resultados obtidos durante a avaliação experimental, embora não exaustivos, mostram os potenciais benefícios de alinhar os planos de mudança a propósitos de negócio. Conforme apresentado anteriormente, vários planos detalhados e alinhados foram obtidos a partir de uma mesma RFC, cada um atendendo a requisitos específicos de tempo e custo de execução. Pôde-se observar, ainda, que os planos gerados foram sensivelmente diferentes tanto em forma (atividades e humanos utilizados), quanto em custos agregados (tempo e custo financeiro). Além disso, a qualidade das alocações de humanos (em relação ao que os gerentes de TI desejam) também possuiu resultados positivos, demonstrando que a automação do processo deve possuir interação com os gerentes de TI (mesmo que mínima) para que sejam realizadas alocações mais realistas e adequadas.

A geração de planos detalhados que atendam às demandas (propósitos de negócio) do solicitante mostrou-se exequível e muito eficiente. Tal pôde ser observado pelo tempo despendido na geração dos mesmos, o qual ficou na ordem de dezenas de segundos. Esse tempo certamente é de grandeza menor que o tempo que seria gasto por

um operador experiente, projetando os mesmos planos utilizando um editor de *workflow*.

Outro aspecto importante que merece destaque é a aplicabilidade da nossa solução em organizações reais. Foi concebida visando a compatibilidade e o alinhamento às boas práticas e aos processos recomendados pelo ITIL, podendo ser considerada para a gestão dos ativos de Tecnologia da Informação nos diversos níveis de uma organização. Além disso, a solução tem o potencial para atender adequadamente em vários contextos, de pequenas a grandes organizações, desde que estas sigam as práticas de gestão e respectivas recomendações do ITIL.

É natural que, para aproveitar o potencial de um sistema para o planejamento de mudanças, é preciso contar com um modelo completo e atualizado da infraestrutura de TI da organização. Estratégias e ferramentas para satisfazer esse requisito não fizeram parte do escopo desta dissertação, mas são plenamente alcançáveis. Tal convicção advém do fato de que modelo como esse é apresentado como fundamental pela ITIL, biblioteca amplamente adotada em organizações de pequeno, médio e, sobretudo, grande portes.

A partir das investigações realizadas e dos resultados alcançados o trabalho apresentado nesta dissertação foi apresentado e discutido com a comunidade em eventos científicos nacionais e internacionais de grande relevância. As críticas, de modo geral, foram extremamente positivas, o que demonstra a importância e a qualidade do trabalho desenvolvido. Algumas das sugestões feitas pelos revisores estão consolidadas nesta dissertação. Abaixo segue a relação dos três artigos aprovados no tema desta dissertação:

- **SBRC 2009 (Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos):** “CHANGEADVISOR: Alinhando o Planejamento de Mudanças em Infraestruturas de TI a Objetivos/Restrições de Negócios” (LUNARDI *et al.*, 2009a);

- **DSOM 2009 (International Workshop on Distributed Systems: Operations and Management):** “CHANGEADVISOR: A Solution to Support Alignment of IT Change Design with Business Objectives/Constraints” (LUNARDI *et al.*, 2009b);

- **NOMS 2010 (IEEE/IFIP Network Operations and Management Symposium):** “On Strategies for Planning the Assignment of Human Resources to IT Change Activities” (LUNARDI *et al.*, 2010).

## 7.2 Questões em aberto e possíveis investigações futuras

Apesar dos progressos alcançados existem diversas possíveis melhorias e investigações futuras que merecem ser citadas. Considerando investigações que podem ser realizadas no curto prazo, existe a possibilidade de estender o escopo do processo de alinhamento de mudanças para considerar objetivos/restrições de negócio diversos (como, por exemplo, custo de equipamentos envolvidos ou da indisponibilidade de serviços durante a mudança). Além disso, novos testes - com cenários mais complexos - devem ser realizados para uma melhor análise das seis estratégias propostas nesta dissertação, além de demonstrar o comportamento de cada estratégia nos novos cenários.

Como investigação futura, no longo prazo, existe a possibilidade de considerar a carga de trabalho dos operadores e, a partir disso, efetuar balanceamento de atividades (carga de trabalho) entre os humanos envolvidos na mudança. Além disso, seria importante aprimorar heurísticas para aperfeiçoar os algoritmos de alinhamento de mudanças proposto, buscando a solução ótima independentemente do *workflow* analisado. Ainda, com o objetivo de gerar estimativas de custos mais precisas, devem ser analisados outros aspectos relativos aos humanos que podem influenciar diretamente a capacidade produtiva dos administradores envolvidos e a qualidade das alocações geradas como, por exemplo, a localização geográfica e capacidade de trabalho em grupos.

## REFERÊNCIAS

- ActiveBPEL. **The ActiveBPEL Engine**. Disponível em: <<http://www.activebpel.org>>. Acesso em junho de 2010.
- BANYAHIA, H.: Costs and Productivity Estimation in Computer Engineering Economics. **Engineering Economist**, v. 41, n. 3, p. 229-241, 1996.
- BENNOUR, M.; CRESTANI, D.; CRESPO, O. *et al.* Computer aided decision for human task allocation with mono and multi performance evaluation. **International Journal of Production Research**, v43, i21, 4559-4588, 2005.
- BPEL. **Organization for the Advancement of Structured Information Standards. OASIS: Business Process Execution Language, version 2.0**. Disponível em: <<http://docs.oasis-open.org/wsbpel/2.0>>. Acesso em junho de 2010.
- BROWN, A. B.; KELLER, A. A Best Practice Approach for Automating IT Management Processes. **IFIP/IEEE Network Operations and Management Symposium (NOMS 2006)**, p. 33-44, 2006.
- CORDEIRO, W.; MACHADO, G.; DAITX, F. *et al.* A Template-based Solution to Support Knowledge Reuse in IT Change Design. **IFIP/IEEE Network Operations and Management Symposium (NOMS 2008). Proceedings...** [Sl.: s.n], p. 355-362, 2008.
- CORDEIRO, W.; MACHADO, G.; ANDREIS, F. *et al.* A Runtime Constraint-Aware Solution for Automated Refinement of IT Change Plans. **IFIP/IEEE International Workshop on Distributed Systems: Operations and Management (DSOM 2008). Proceedings...** [Sl.: s.n], p. 69-82, 2008.
- CORDEIRO, W. L. C.; MACHADO, G. S.; ANDREIS, F. G. *et al.* CHANGELEDGE: Change Design and Planning in Networked Systems based on Reuse of Knowledge and Automation. **Elsevier Computer Networks**, Vol. 53, Issue 16, p. 2782-2799, 2009.
- DIA, **Dia a drawing program**. Disponível em: <<http://projects.gnome.org/dia/>>. Acesso em junho de 2010.
- DIAO, Y. e HECHING, A. Workload Management in Dynamic IT Service Delivery Organizations. **IFIP/IEEE International Workshop on Distributed Systems: Operations and Management (DSOM 2009). Proceedings...** [Sl.: s.n], p. 123-137, 2009.
- DMTF, **Distributed Management Task Force: Common Information Model**. Disponível em: <<http://www.dmtf.org/standards/cim>>. Acesso em junho de 2010.
- DUGMORE, JENNY. **Achieving ISO/IEC 20000 - The Differences Between BS 15000 and ISO/IEC 20000**. BSI Standards. p. 124. ISBN 0580473481, 2006.
- HIBERNATE. **Relational Persistence for Java and .NET (Hibernate)**. Disponível em: <<http://www.hibernate.org>>. Acesso em junho de 2010.

ITIL. **Information Technology Infrastructure Library. Office of Government Commerce (OGC)**. Disponível em: <<http://www.ital-officialsite.com>>. Acesso em: junho de 2010.

ITIL Continual Service Improvement. **IT Infrastructure Library: Continual Service Improvement, v.3**. London: The Stationery Office, 2007, 308 p.

ITIL Service Design. **IT Infrastructure Library: Service Design, v.3**. London: The Stationery Office, 2007, 449 p.

ITIL Service Operation. **IT Infrastructure Library: Service Operation, v.3**. London: The Stationery Office, 2007, 393 p.

ITIL Service Strategy. **IT Infrastructure Library: Service Strategy, v.3**. London: The Stationery Office, 2007, 373 p.

ITIL Service Transition. **IT Infrastructure Library: Service Transition, v.3**. London: The Stationery Office, 2007, 399 p.

JAVA/SWING. **Java/Swing GUI Framework**. Disponível em: <<http://www.newt.com/java/swing.html>>. Acesso em junho de 2010.

JORGENSEN, M., SHEPPERD, M. A Systematic Review of Software Development Cost Estimation Studies. **IEEE Transactions on Software Engineering**, v. 33, n. 1, p. 33-53, 2007.

KELLER, A.; HELLERSTEIN, J. L.; WOLF, J. L. *et al.* The CHAMPS System: Change Management with Planning and Scheduling. IEEE/IFIP Network Operations and Management Symposium (NOMS 2004). **Proceedings...** [Sl.: s.n], p. 395-408, 2004.

KELLER, A. Automating the Change Management Process with Electronic Contracts. IEEE International Conference on E-Commerce Technology Workshops (CECW 2008). **Proceedings...** [Sl.: s.n], p. 99-108, 2005.

LUNARDI, R. C.; WICKBOLDT, J. A.; CORDEIRO, W. L. C. *et al.* ChangeAdvisor: Alinhando o Planejamento de Mudanças em Infra-estruturas de TI a Objetivos/Restrições de Negócios. Simpósio Brasileiro de Redes de Computadores e de Sistemas Distribuídos (SBRC 2009). **Proceedings...** [Sl.: s.n], p. 437-450, 2009.

LUNARDI, R. C.; CORDEIRO, W. L. C.; ANDREIS, F. G. *et al.* ChangeAdvisor: A Solution to Support Alignment of IT Change Design with Business Objectives/Constraints. IFIP/IEEE International Workshop on Distributed Systems: Operations and Management (DSOM 2009). **Proceedings...** [Sl.: s.n], p. 138-151, 2009.

LUNARDI, R. C.; ANDREIS, F. G.; CORDEIRO, W. L. C. *et al.* On Strategies for Planning the Assignment of Human Resources to IT Change Activities. IFIP/IEEE Network Operations and Management Symposium (NOMS 2010). **Proceedings...** [Sl.: s.n], p. 248-255, 2010.

MACHADO, G. S.; DAITX, F.; CORDEIRO, W. *et al.* Enabling Rollback Support in IT Change Management Systems. IFIP/IEEE Network Operations and Management Symposium (NOMS 2008). **Proceedings...** [Sl.: s.n], p. 347-354, 2008.

MACHADO, G.; CORDEIRO, W.; SANTOS, A. et al. Algoritmo para Geração Automática de Ações de Rollback em Sistemas de Gerenciamento de Mudanças em TI. Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC 2008). **Proceedings...** [Sl.: s.n], p. 107-120, 2008.

MOURA, A.; SAUVÉ, J.; BARTOLINI, C. Business-driven IT Management - Upping the Ante of IT: Exploring the Linkage between IT and Business to Improve both IT and Business Results. **IEEE Communications Magazine**, v. 46, n. 10, p. 148-153, 2008.

MySQL. **MySQL Query Analyzer - Improving SQL Query Performance (MySQL)**. Disponível em: <www.mysql.com>. Acesso em junho de 2010.

OASIS. **Organization for the Advancement of Structured Information Standards. OASIS: Business Process Execution Language, version 2.0**. Disponível em: <http://docs.oasis-open.org/wsbpel/2.0>. Acesso em junho de 2010.

PILLAI, K., SUKUMARAN NAIR, V. S. A Model for Software Development Effort and Cost Estimation. **IEEE Transactions on Software Engineering**, v. 23, n. 8, p. 485-497, 1997.

REBOUÇAS, R.; SAUVÉ, J.; MOURA, A. *et al.* A Decision Support Tool to Optimize Scheduling of IT Changes. IFIP/IEEE International Symposium on Integrated Network Management (IM 2007). **Proceedings...** [Sl.: s.n], p. 343-352, 2007.

RODOSEK, G. D. A Generic Model for IT Services and Service Management. IFIP/IEEE International Integrated Network Management (IM 2003). **Proceedings...** [Sl.: s.n], p. 171-184, 2003.

SHA, L., RAJKUMAR, R., e LEHOCZKY, J. P. Priority Inheritance Protocols: An Approach to Real-Time Synchronization. **IEEE Transaction on Computers** v. 39, n. 9 (Sep. 1990), 1175-1185, 1990.

SHEPHERD, M. Special Feature Distributed Computing Power: a Key to Productivity. **IEEE Computer**, v. 10, n. 11, p. 66-74, 1977.

StAX. **The Streaming API for XML**. Disponível em: <http://stax.codehaus.org>. Acesso em junho de 2010.

TRASTOUR, D., RAHMOUNI, M., BARTOLINI, C. Activity-Based Scheduling of IT Changes. International Conference on Autonomous Infrastructure, Management and Security (AIMS 2007). **Proceedings...** [Sl.: s.n], p. 73-84, 2007.

ZHENGYUAN J.; LIHUA G. Multi-criteria Human Resource Allocation for Optimization Problems Using Multi-objective Particle Swarm Optimization Algorithm. International Conference on Computer Science and Software Engineering. **Proceedings...** [Sl.: s.n], p.1187-1190, 2008.

ZHOU, L. A Project Human Resource Allocation Method Based on Software Architecture and Social Network. International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM '08). **Proceedings...** [Sl.: s.n], p.1-6, 2008.



## ANEXO A ARTIGO PUBLICADO NO SBRC 2009

Neste anexo, o artigo intitulado “ChangeAdvisor: Alinhando o Planejamento de Mudanças em Infra-estruturas de TI a Objetivos/Restrições de Negócios” é apresentado. Essa foi a primeira publicação no tema desta dissertação em eventos científicos renomados. A solução para alinhamento foi apresentada, bem como foi desenvolvido um primeiro protótipo chamado CHANGEADVISOR.

- **Título:** ChangeAdvisor: Alinhando o Planejamento de Mudanças em Infra-estruturas de TI a Objetivos/Restrições de Negócios
- **Conferência:** Simpósio Brasileiro de Redes de Computadores e de Sistemas Distribuídos (SBRC 2009)
- **URL:** <http://www.sbrc2009.ufpe.br/>
- **Período da realização do evento:** 25-29 de Maio de 2009
- **Local:** Mar Hotel, Recife, Pernambuco, Brazil
- **Disponível na Biblioteca Digital da SBC:**  
<http://www.sbc.org.br/bibliotecadigital/download.php?paper=2682>

## CHANGEADVISOR: Alinhando o Planejamento de Mudanças em Infra-estruturas de Rede e Serviços a Propósitos de Negócio\*

Roben Castagna Lunardi, Weverton Luis da Costa Cordeiro,  
Juliano Araújo Wickboldt, Guilherme Sperb Machado, Fabrício Girardi Andreis,  
Alan Diego dos Santos, Cristiano Bonato Both, Luciano Paschoal Gasparly,  
Lisandro Zambenedetti Granville

Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)  
Porto Alegre – RS – Brasil

{rclunardi, weverton.cordeiro, jwickboldt, gsmachado, fgandreis,  
adsantos, cbboth, paschoal, granville}@inf.ufrgs.br

**Abstract.** *Change planning represents a key element for the operation and management of network infrastructures and services. Its scope ranges from the high level design of a change request to the generation, either manually or automatically, of detailed plans that, if executed, will materialize the requested changes (e.g., modification of network device settings and deployment of new services). A fundamental problem is that, although correct, such detailed plans may not be necessarily aligned to requirements defined in the business level. To overcome this problem, in this paper we propose a solution for the alignment of network infrastructure and service change plans to business objectives/constraints. The solution is analyzed experimentally through a prototypical implementation of a decision support system called CHANGEADVISOR, which helps operators to understand the trade-offs between alternative change designs.*

**Resumo.** *O planejamento de mudanças é um elemento chave para a gerência e a operação de infra-estruturas de redes e serviços. Ele compreende desde a especificação em alto nível até a geração, possivelmente automatizada, de planos detalhados que, se executados, concretizarão as mudanças solicitadas (por exemplo, alteração da configuração de dispositivos de rede). Um problema fundamental é que, embora corretos, tais planos de mudança podem não necessariamente estar alinhados com os propósitos definidos em nível de negócio. Para abordar este problema, neste artigo propõe-se uma solução para alinhar o planejamento de mudanças em infra-estruturas de rede e serviços a objetivos/restrições de negócio. A solução proposta é avaliada experimentalmente por meio da implementação prototípica de um sistema de suporte à decisão denominado CHANGEADVISOR, o qual auxilia operadores a entender o "compromisso" entre planos de mudança alternativos.*

### 1. Introdução

O tamanho e complexidade crescentes das infra-estruturas de rede e serviços – também chamadas genericamente de infra-estruturas de Tecnologia da Informação (TI) ao longo do artigo – têm demandado das organizações a adoção de boas práticas e processos para assegurar o seu funcionamento correto e eficiente. Com o objetivo de auxiliar as organizações nessa tarefa, a Biblioteca de Infra-estrutura de Tecnologia da Informação (*Information Technology Infrastructure Library, ITIL*) [ITIL 2008], uma das referências mais importantes neste contexto, recomenda, entre outros processos, o gerenciamento de mudanças. Esse processo auxilia no tratamento imediato e eficiente de mudanças que venham a ser necessárias na infra-estrutura de *hardware* e *software* das organizações.

Considerado um dos elementos chave do referido processo, o *planejamento de mudanças* [Service Transition 2007] compreende desde a especificação da mudança por

---

\* Este trabalho foi desenvolvido em colaboração com a HP Brasil P&D.

um solicitante, em alto nível de abstração (ex: ampliação da capacidade de canais de comunicação, migração de serviços, implantação de novas aplicações), até a geração, possivelmente automatizada, de planos com alto grau de detalhamento (isto é, planos que compreendem atividades de baixo nível como, por exemplo, manipulação de rotas em roteadores e modificação de políticas de *firewall*). Esses planos, se executados, concretizarão as mudanças especificadas pelo solicitante na infra-estrutura gerenciada.

Um problema fundamental acerca da geração dos planos de mudança é que, a partir de uma mesma especificação em alto nível, podem ser obtidos diferentes planos detalhados. Embora corretos, esses planos não necessariamente estão alinhados com os propósitos definidos em nível de negócio para a infra-estrutura de TI. Os propósitos podem ser de natureza técnica, tais como a imposição de *restrições* quanto à indisponibilidade de serviços providos pela rede gerenciada, ou podem definir *objetivos* para otimizar a alocação de recursos durante a execução da mudança (por exemplo, minimizar o número de operadores envolvidos na manutenção de um serviço de *e-mail*). Sem levar em conta tais considerações, os planos de mudança gerados podem levar a resultados que violam políticas pré-definidas no nível de negócio (por exemplo, paradas para manutenção no serviço de *e-mail* não devem exceder 10 horas mensais).

Destacam-se três motivos que justificam a importância de alinhar planos de mudança em infra-estruturas de redes e serviços com propósitos de negócio. Primeiro, existe a possibilidade de otimização no uso dos recursos disponíveis, que normalmente são escassos e/ou caros. Segundo, tal alinhamento tem potencial para contribuir com a redução de custos como, por exemplo, o tempo despendido para completar a mudança. Terceiro – e mais importante – a possibilidade de alinhar o planejamento de mudanças a propósitos distintos permite ao operador entender os “compromissos” resultantes ao optar por um plano de mudança dentre as opções possíveis. No entanto, apesar dos potenciais benefícios, tal alinhamento tem sido negligenciado pelas investigações na área de gerência e operações que, alternativamente, têm enfatizado aspectos como a geração de planos exequíveis [Cordeiro 2008a] e otimizados [Keller 2004].

Para suprir esta lacuna, neste artigo propõe-se uma solução para alinhar o planejamento de mudanças em infra-estruturas de redes e serviços com propósitos definidos em nível de negócio. Em contraste com as propostas já estabelecidas na literatura, este artigo concentra-se na geração automatizada de planos de mudança guiada pelos objetivos/restrições definidos durante a especificação da mesma. A solução proposta é avaliada experimentalmente por meio do CHANGEADVISOR, uma implementação prototípica de um sistema de suporte à decisão que fornece apoio para o entendimento dos “compromissos” entre planos de mudança alternativos. Durante a avaliação – realizada considerando cenários baseados em casos reais – foram avaliadas a correção e a completude dos planos gerados, bem como o alinhamento dos mesmos aos propósitos de negócio definidos na fase de especificação das mudanças.

O restante do artigo está organizado como segue. A Seção 2 discute alguns dos principais trabalhos publicados na área de gerenciamento de mudanças em TI. A Seção 3 apresenta a solução conceitual para a geração de planos de mudanças alinhados a propósitos de negócio. A Seção 4 detalha o sistema CHANGEADVISOR, enquanto a Seção 5 descreve a avaliação experimental conduzida utilizando esse sistema. Por fim, a Seção 6 conclui o artigo com as considerações finais e perspectivas de trabalhos futuros.

## 2. Trabalhos Relacionados

A grande área de gerenciamento de infra-estruturas de TI tem recebido grande atenção da comunidade científica nos últimos anos. Diversos aspectos vêm sendo explorados, tais como modelos [Rodosek 2003], automação [Brown 2006] e alinhamento a propósitos de negócio [Moura 2008]. Analisando-se especificamente a subárea de

gerenciamento de mudanças, observa-se atividade semelhante. Por exemplo, passos importantes têm sido dados em tópicos como agendamento [Rebouças 2007, Trastour 2007] e *rollback* [Machado 2008]. Contudo, o tópico relacionado a *planejamento* de mudanças em TI, foco deste artigo, tem sido comparativamente pouco explorado. A seguir são discutidos os três principais trabalhos relacionados ao assunto.

Keller *et al.* [Keller 2004] propuseram CHAMPS, um sistema para automação da geração de planos de mudança que exploram um alto grau de paralelismo na execução de tarefas. Embora seja capaz de aperfeiçoar o agendamento de tarefas considerando os recursos de *hardware* disponíveis para executá-las, CHAMPS não leva em consideração a associação de tarefas a recursos humanos nem objetivos/restrições especificados pelo solicitante.

Em um trabalho recente, Cordeiro *et al.* [Cordeiro 2008a] propuseram uma solução para formalizar, preservar e reusar o conhecimento adquirido durante mudanças recorrentes em infra-estruturas de redes e serviços. A solução, baseada em *modelos de mudança*, permite que passos rotineiros (por exemplo, alterações e/ou implantações de serviços) sejam especificados em alto nível e recorrentemente instanciados (por exemplo, a partir da indicação dos serviços que serão manipulados). Embora a solução compreenda um algoritmo para a geração de planos detalhados a partir da especificação em alto nível, está fora do escopo desse algoritmo, como uma premissa simplificadoria, alinhar os planos gerados a propósitos de negócio. Em um trabalho subsequente [Cordeiro 2008b], o algoritmo de geração de planos foi aprimorado, de modo a levar em consideração as restrições técnicas impostas pelo ambiente gerenciado (por exemplo, disponibilidade de recursos como espaço em disco e memória, e permissões para a execução de ações em determinados dispositivos).

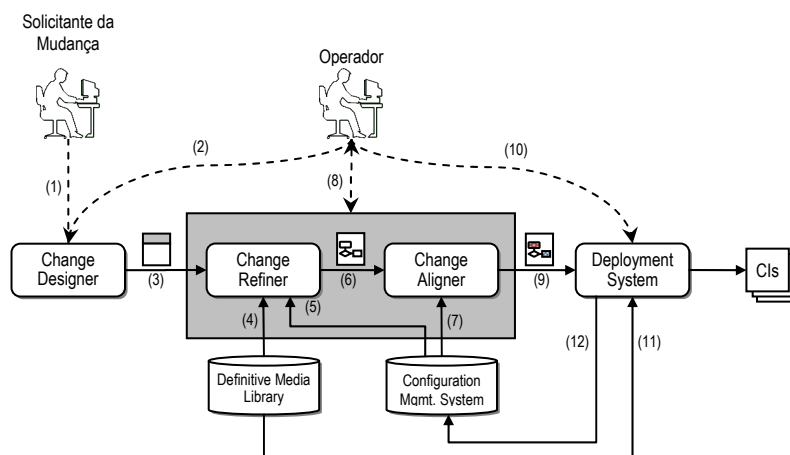
Outro conjunto de trabalhos, apesar de não tratar o problema de planejamento de mudanças, merece breve discussão por, de alguma forma, explorar alinhamento com aspectos de nível de negócio. Por exemplo, Keller [Keller 2005] introduziu o conceito de contratos eletrônicos. No trabalho em questão são propostos quatro tipos de contratos: RFCs, *Deployment Descriptors*, Políticas e Boas Práticas, e Acordos de Nível de Serviço (*Service Level Agreements, SLAs*). Esses documentos, que possuem diferentes propósitos, formatos e níveis de detalhamento, são utilizados para a especificação de objetivos/restrições e têm fundamental importância para o processo de execução de mudanças. Na mesma linha de investigação, Rebouças *et al.* [Rebouças 2007] propuseram uma abordagem para a priorização e o escalonamento de mudanças visando à minimização de custos (por exemplo, custos operacionais e perdas financeiras devido à violação de SLAs).

Em síntese, embora o planejamento de mudanças tenha sido foco de algumas investigações recentes, o alinhamento das mesmas com propósitos definidos em nível de negócio não foi abordado. Isso significa que, não raro, o planejamento e, por conseguinte, a execução de mudanças é conduzida em direção oposta, ou de forma pouco alinhada, com objetivos/restrições demandados pelo solicitante. Para lidar com este problema, as seções seguintes apresentam uma solução conceitual, além do sistema desenvolvido para suportá-la e os resultados obtidos.

### 3. Solução Conceitual

Visando oferecer suporte ao planejamento de mudanças alinhado a propósitos definidos em nível de negócio, foram introduzidos dois novos componentes na solução conceitual para o planejamento e a implantação de mudanças, proposta em um trabalho anterior [Cordeiro 2008a]. Os novos componentes – destacados na área em cinza – materializam o mecanismo para o planejamento guiado por propósitos de negócio e se encaixam adequadamente, sem modificação significativa, na solução conceitual proposta

anteriormente. A Figura 1 apresenta uma visão geral da solução estendida, destacando os seus principais componentes, atores envolvidos e interações entre esses elementos.



**Figura 1. Arquitetura conceitual da solução proposta**

O solicitante da mudança inicia um processo de mudança interagindo com o componente *Change Designer* (fluxo 1 na Figura 1) para elaborar uma requisição de mudança (*Request for Change, RFC*). A RFC descreve, em linhas gerais, quais modificações devem ser acomodadas na infra-estrutura gerenciada, os itens de configuração (*Configuration Items, CIs*) primariamente afetados (por exemplo, *firewall, switches, serviços, aplicações, etc.*), e os propósitos de negócio a serem alcançados. A RFC, no entanto, não especifica os detalhes de como a mesma deve ser materializada. Esses devem ser esboçados em um passo subsequente, por um operador, por meio da especificação de um “plano preliminar” junto ao componente *Change Designer* (fluxo 2). O plano preliminar consiste de um *workflow* de atividades (ou ações) que descrevem, em alto nível de abstração, como a mudança requisitada deve ser materializada na infra-estrutura.

A geração do plano detalhado de mudança a partir da especificação preliminar esboçada pelo operador é realizada, sem intervenção humana, pelo componente *Change Refiner* (3). Esse processamento é efetuado utilizando um algoritmo de refinamento de atividades, com base em informações sobre (pacotes de) *software*, disponíveis na *Definitive Media Library (DML)* (4), e em informações sobre a infra-estrutura de TI alvo da mudança (5), presentes no *Configuration Management System (CMS)*.

Após o refinamento da mudança, o plano resultante é consumido pelo componente *Change Aligner* (6), que possui um papel central no alinhamento do plano aos objetivos/restrições de negócio expressos na RFC. Para guiar este alinhamento, o componente *Change Aligner* obtém no CMS (7) informações sobre custos de alocação de operadores e suas habilidades (por exemplo, um operador *expert* em serviço de *e-mail* cuja hora de trabalho custa 40 unidades monetárias), bem como capacidade dos dispositivos a serem manipulados (por exemplo, poder computacional, no caso de estações de trabalho e servidores). É importante ressaltar que o processo de refinamento e alinhamento da mudança é reiterado, durante um período de tempo definido *a priori* pelo operador, visando à geração de diferentes planos detalhados que atendam aos mesmos propósitos de negócio definidos na RFC.

Posteriormente (8), os planos gerados podem ser modificados pelo operador (de modo a refletir precisamente suas necessidades), e avaliados quanto aos “compromissos” existentes entre as várias alternativas. Entre os critérios que podem servir de parâmetros para essa análise estão o tempo necessário, os custos financeiros associados e os recursos humanos que devem ser alocados para a execução dos planos.

Com base nessa análise, o operador poderá selecionar o plano mais adequado para ser executado sobre a infra-estrutura.

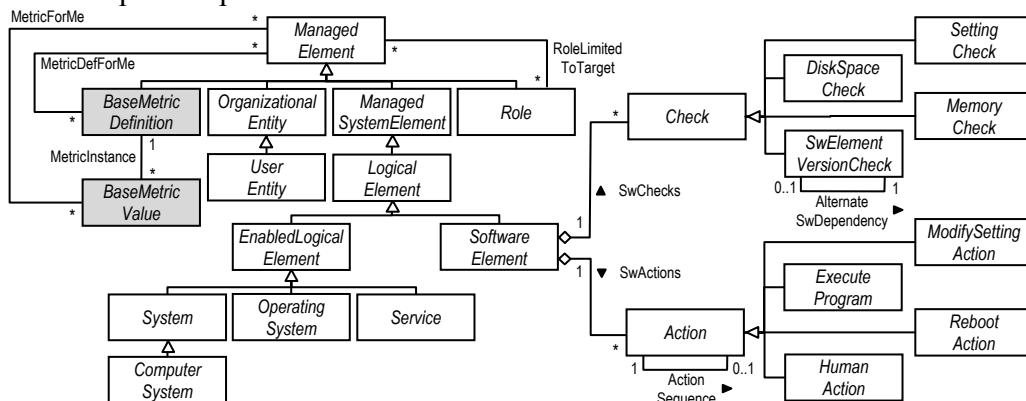
Como próximo passo do processo de mudança, o operador invoca a execução do plano selecionado (9) interagindo com o componente *Deployment System* (10). Para realizar algumas das atividades descritas no plano, tal componente pode consumir pacotes de *software* disponíveis na *Definitive Media Library* (11). Após a execução, o *Deployment System* deverá atualizar o CMS com as modificações realizadas em cada CI da infra-estrutura gerenciada (12).

Tendo apresentado uma visão geral da solução proposta, as próximas subseções têm por objetivo descrever (i) a geração de planos detalhados a partir da especificação preliminar de mudanças e (ii) o processo de alinhamento de tais planos a propósitos definidos em nível de negócio.

### 3.1. Change Refiner

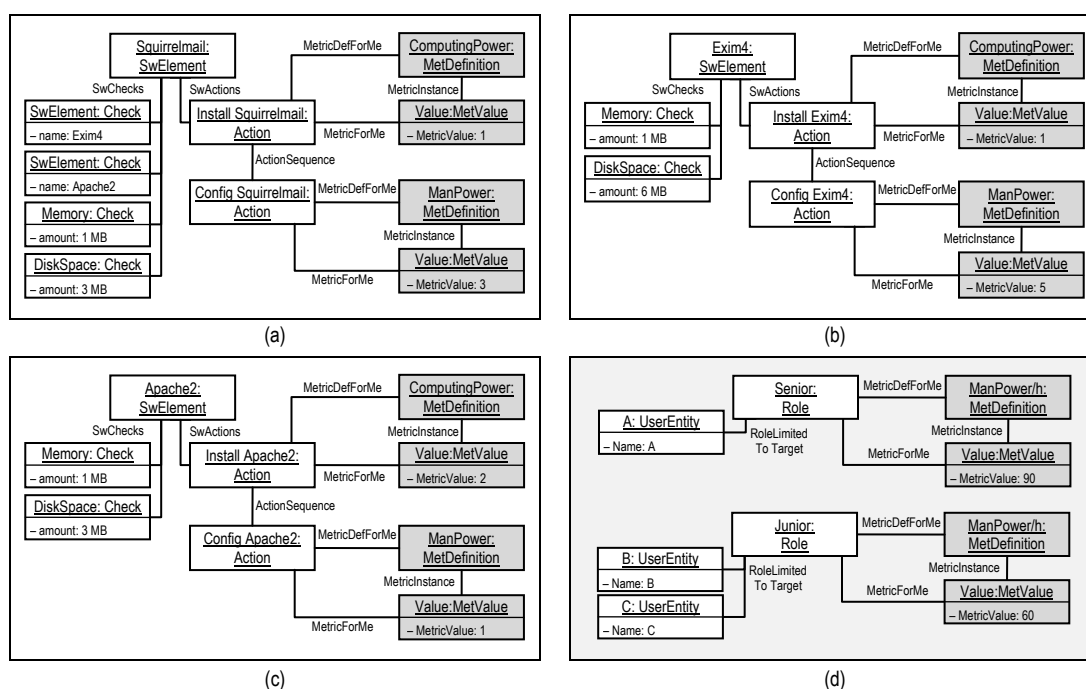
O componente *Change Refiner* desempenha um papel importante no processo de geração de um plano de mudança detalhado a partir do plano preliminar fornecido pelo solicitante da mudança. Para materializar esse componente, propõe-se um algoritmo, inspirado em um trabalho anterior [Cordeiro 2008b], que oferece suporte à geração de diferentes alternativas de planos para o processo de alinhamento. O novo algoritmo, bem como seu relacionamento com as outras entidades da solução conceitual, é brevemente descrito a seguir.

Conforme apresentado na solução conceitual, a geração de planos de mudanças detalhados requer informações sobre as entidades que compõem a infra-estrutura e seus relacionamentos, e os pacotes de *software* disponíveis para a implantação de mudanças. Com o objetivo de materializar o armazenamento dessas informações no CMS e na DML, respectivamente, é empregado neste artigo um subconjunto do *Common Information Model* (CIM) [DMTF 2008]. A Figura 2 apresenta uma visão parcial do modelo, o qual permite representar qualquer CI que compõe a infra-estrutura gerenciada. Relacionamentos tais como associações, composições e agregações, a maioria omitidas na figura por questões de legibilidade, mapeiam as dependências entre os elementos que compõem a infra-estrutura.



**Figura 2. Modelo usado para a representação da infra-estrutura de TI e da DML**

O modelo apresentado incorpora também classes *Check* e *Action*, que mapeiam informações necessárias para o cômputo de dependências e refinamento das atividades que compõem o plano preliminar. Uma instância da classe *Check* define uma condição a ser satisfeita ou uma característica requerida pelo *software* associado para que o mesmo possa evoluir para um novo estado (ex: *deployable*, *installable*, *executable* ou *running*).



**Figura 3. Instâncias de software (*SoftwareElement*) presentes na DML (a,b,c) e de Papéis (*Roles*) presentes na infra-estrutura de TI (d)**

Na Figura 3(a) são ilustradas instâncias de *Checks* associadas ao *software Squirrelmail*. As duas primeiras expressam, respectivamente, que os softwares *Exim4* e *Apache2* devem ser instalados antes da instalação do *Squirrelmail*. As duas últimas, por sua vez, expressam que o *Squirrelmail* requer 1 MB de memória e 3 MB de espaço em disco para instalação e execução, respectivamente. Cada instância da classe *Action*, por sua vez, representa uma atividade exigida para mudar o estado do *software* associado (ex: de *installable* para *executable*, isto é, um processo de instalação). Utilizando novamente como exemplo o *Squirrelmail*, são ilustradas em 3(a) duas atividades associadas ao processo de instalação do mesmo: *Install Squirrelmail*, responsável pela cópia dos arquivos binários do *Squirrelmail*; e *Configure Squirrelmail*, a qual gera as configurações necessárias para o funcionamento do serviço provido pelo mesmo.

Outro requisito importante para a geração automatizada de planos detalhados é a definição, de forma não ambígua e interpretável pelo mecanismo de refinamento, dos objetivos das atividades presentes no plano preliminar. Para tal, é utilizada uma notação, proposta em um trabalho anterior [Cordeiro 2008a], denominada *Activity Modeling Notation* (AMN). Como exemplo, considere novamente a atividade *Install Squirrelmail*. Sua representação, de acordo com a AMN, é *install SoftwareElement Squirrelmail at ComputerSystem www.example.com*. Neste exemplo, *Squirrelmail* e *www.example.com* são referências a objetos presentes na DML e no CMS, respectivamente.

O algoritmo para o refinamento de mudanças, também denominado de *change\_refinement*, recebe como entrada a especificação de mudança a ser refinada, além de dados sobre o atual estado da infra-estrutura gerenciada (oriundos do CMS) e sobre pacotes disponíveis para o processo de mudança (provenientes da DML). Para cada uma das atividades que compõe o plano preliminar, *change\_refinement* computa o conjunto de atividades que satisfazem as dependências para a execução da mesma.

Para ilustrar o processo de refinamento, considere a RFC *Migrate Mail System*, apresentada na Figura 4 (fundo em cinza na figura). Ela consiste na migração de um serviço de *e-mail* do servidor *X* para o *Y*, e tem como objetivo minimizar o tempo de execução estimado para, no máximo, 120 minutos (de modo a não comprometer as

operações da organização). O plano preliminar associado à especificação (fundo em branco) é composto por seis atividades em alto nível, que compreendem desde a instalação do sistema operacional até a migração de dados e configurações do serviço de *e-mail* localizado em *X* para *Y* (ambos servidores presentes na rede gerenciada).

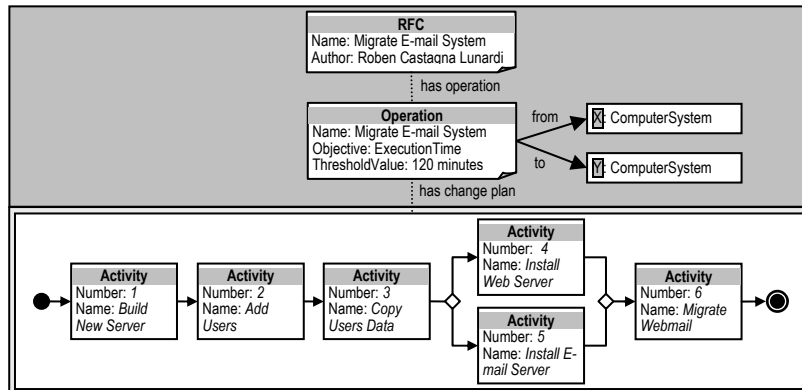


Figura 4. Exemplo de RFC e de plano preliminar de mudança para migração de servidor de *e-mail* e respectivo *front-end Web*

Focando na atividade *Install Squirrelmail* (ilustrada no plano preliminar), *change\_refinement* identifica – a partir das pré-condições (*Checks*) apresentadas na DML (Figura 3(a)) – que a execução dessa atividade depende da instalação prévia dos softwares *Apache2* (servidor Web) e *Exim4* (servidor de *e-mail*). Logo, duas novas atividades serão geradas para satisfazer as dependências de *Install Squirrelmail*: *Install Apache2* (a) e *Install Exim4* (b). Esse processo continua recursivamente, até que todas as dependências para execução das atividades do plano sejam satisfeitas. Ao final desse processo, quando um plano detalhado é obtido (por exemplo, o plano ilustrado na Figura 6), *change\_refinement* armazena o “estado atual” (isto é, as opções de refinamento percorridas para se chegar a tal plano). Esse passo é importante para que, após o alinhamento (discutido em detalhes na Subseção 3.2), seja possível gerar novos planos detalhados utilizando refinamentos alternativos. Desta forma, diferentes planos de mudança detalhados podem ser gerados para a mesma especificação inicial.

### 3.2. Change Aligner

A materialização do componente *Change Aligner*, elemento chave para o alinhamento dos planos detalhados aos objetivos de negócio, está fundamentada em três blocos construtores principais: (i) a formalização de propósitos de negócio, (ii) os modelos para representar métricas de custos/capacidades e (iii) o algoritmo para o alinhamento de mudanças em função dos propósitos especificados e custos/capacidades associados aos elementos da infra-estrutura. Cada um desses blocos construtores é detalhado a seguir.

#### 3.2.1. Objetivos/Restrições de Negócio e Métricas de Custos/Capacidades

Para que os planos de mudança produzidos pelo componente *Change Refiner* possam ser alinhados a objetivos/restrições de negócio, o(s) objetivo(s) e/ou restrição(ões) a ser(em) satisfeito(s) precisa(m) ser especificado(s) na RFC. Duas categorias de objetivos/restrições são contempladas nessa primeira iteração para solucionar o problema. São elas: o tempo necessário (*ExecutionTime*) e o custo associado (*ExecutionCost*) para completar uma mudança. Estes foram os objetivos apontados como essenciais por um grupo de operadores de rede consultados. Acompanha a definição de cada objetivo e/ou restrição um limiar (*ThresholdValue*), representando o tempo máximo admitido e/ou o custo máximo tolerado, calculado em função dos humanos alocados, para a execução da mudança. Como pode ser observado na Figura 4,



foi especificado na RFC *Migrate E-mail System* que a mudança, ao ser planejada, deve considerar um tempo máximo de execução de 120 minutos.

Outras informações indispensáveis ao processo de alinhamento são as *métricas de custo* e de *capacidades*. Enquanto as primeiras são usadas para caracterizar o esforço demandado para a execução de atividades ligadas a gerência e a operação de infra-estruturas de redes e serviços, as últimas permitem expressar uma expectativa de rendimento de humanos ou computadores para executá-las. As métricas empregadas neste trabalho, enumeradas a seguir, são adotadas e consolidadas em áreas como Sistemas Distribuídos [Shepherd 1977], Engenharia de Software [Pillai 1997] e Economia [Banyahia 1996]. Do ponto de vista de modelagem, são especificadas, e seus valores instanciados, no modelo usado para a representação da infra-estrutura de TI e da DML (ilustrado na Figura 2) por meio das classes *BaseMetricDefinition* e *BaseMetricValue*, respectivamente.

As métricas de custo são subdivididas em *ManPower* (MP) e *ComputingPower* (CP). MP é usada para quantificar o esforço humano necessário para executar uma atividade. CP, por sua vez, denota o esforço computacional requerido para executar uma atividade automatizada, *i.e.*, com nenhuma (ou desprezível) intervenção humana. Instâncias de valores para essas métricas são utilizadas para “anotar” custos em atividades associadas a elementos de *software* registrados na DML. Na Figura 3, pode-se observar, a título de ilustração, que as atividades *Install Squirrelmail* (a), *Install Exim4* (b) e *Install Apache2* (c) possuem custo de execução associado, em CP, de 1, 1 e 2 unidades, respectivamente. Já as atividades de configuração dos referidos serviços, *Config Squirrelmail* (a), *Config Exim4* (b) e *Config Apache2* (c), apresentam custo, em MP, de 3, 5 e 1 unidades.

A exemplo do que ocorre com as métricas de custo, as métricas de capacidade também se desdobram em duas: *ManPower/Hour* (MP/H) e *ComputingPower/Hour* (CP/H). Elas denotam a carga de trabalho sustentada por humanos e computadores para executar atividades. No caso de MP/H, instâncias de valores da métrica são “anotadas” junto a papéis desempenhados pelo corpo de pessoas vinculadas ao setor de TI da organização. No exemplo ilustrado na Figura 3 (d), são modelados dois papéis, *Senior* e *Junior*, com MP/H equivalentes a 90 e 60 unidades, respectivamente. A esses papéis são, então, associadas pessoas, de acordo com as funções desempenhadas por elas. Instâncias de valores da métrica CP/H, por outro lado, são vinculadas a cada um dos computadores presente na infra-estrutura de TI, refletindo o seu desempenho esperado.

É importante destacar que a investigação de formas de valoração de custos e capacidades, apesar de fundamentais para a solução proposta, não fazem parte do escopo deste artigo. Para esta finalidade, assumimos o emprego de alguma das propostas publicadas e consolidadas na literatura, muitas das quais foram recentemente compiladas por Jorgensen [Jorgensen 2007]. No caso deste artigo, adotamos o método empírico, que consiste em valorar as atividades mais simples – tanto as automatizadas quanto as baseadas em execução por humano – atribuindo custos 1 (um) CP e 1 (um) MP, respectivamente. Tomando as mesmas como base, são calculados os custos para as demais atividades. A estimativa das capacidades dos recursos computacionais (CP/H) e dos humanos (MP/H), por sua vez, é feita considerando históricos de execução das mesmas em um determinado ambiente, durante um determinado período de tempo.

### 3.2.2. Algoritmo de Alinhamento a Objetivos/Restrições de Negócio

O mecanismo de refinamento, os modelos e as métricas apresentados anteriormente formam a base sobre a qual o algoritmo para o alinhamento de planos de mudança a propósitos de negócio está constituído. O algoritmo, ilustrado na Figura 5, é denotado por *change\_alignment* (*C*, *DML*, *IT*, *H*), sendo *C* o plano de mudança conforme gerado

pelo *change\_refinement*, *DML* o repositório que contém informações sobre pacotes de *software* disponíveis, *IT* o repositório de informações sobre a infra-estrutura gerenciada, e *H* o conjunto de operadores humanos disponíveis para a condução de mudanças.

```

01 change_alignmnet (C, DML, IT, H)
02   C' ← empty change plan
03   D ← set of activities from C
04   recursive_alignment (C, C', D, DML, IT, H)
05
06 recursive_alignment (C, C', D, DML, IT, H)
07   if D is empty and cost of deploying C' comply with objective threshold, given DML, IT, then
08     return C' as solution
09   extract activity  $a_i$  from D
10   if  $a_i$  is a human activity then
11     for all h in H do
12       for all  $a_j$  in C do
13         allocable ← true
14         if  $a_i$  is parallel to  $a_j$  in C and h is allocated to  $a_j$  then
15           allocable ← false
16         if allocable is true then
17           associate human h to  $a_i$ 
18           add activity  $a_i$  to change plan C'
19           if cost of deploying C' comply with objective threshold, given DML, IT, then
20             recursive_alignment (C, C', D, DML, IT, H)
21         else
22           add activity  $a_i$  to change plan C'
23           if cost of deploying C' comply with objective threshold, given DML, IT, then
24             recursive_alignment (C, C', D, DML, IT, H)
25   return failure

```

**Figura 5. Algoritmo para geração de planos de mudança alinhados a objetivos/restrições de negócio**

Como primeiro passo do processo de alinhamento, é criado um plano de mudança “vazio”, o qual conterá o resultado deste processo (linha 01 na Figura 5). Em seguida, as atividades presentes em *C* são copiadas para o conjunto *D* (linha 02), e o alinhamento continua a partir da invocação da função *recursive\_alignment* (04).

O processo de alinhamento, conforme descrito em *recursive\_alignment*, inicia-se pela verificação se *D* (o conjunto de atividades ainda não processadas) é vazio ou não, e se o plano *C'* recebido satisfaz aos propósitos de negócio especificados na RFC (07). Caso positivo, o plano *C'* é retornado como solução. Caso contrário, os seguintes passos são executados. Inicialmente, uma atividade  $a_i$  é extraída do conjunto *D*, para processamento (09). Em seguida, é verificado se  $a_i$  requer (ou não) apoio de um operador humano para a sua execução (10). Caso  $a_i$  demande intervenção humana, o algoritmo tenta alocar o operador mais capacitado *h* (disponível em *H*, o conjunto de operadores humanos) que não esteja alocado a nenhuma outra atividade paralela a  $a_i$  (linhas 11-18). A alocação do operador mais capacitado é realizada a partir da seleção dos mesmos em ordem decrescente de Papel (*Role*) ao qual estão associados. Por exemplo, considerando os papéis *Senior* e *Junior*, nesta ordem, os operadores melhor capacitados pertencerão a *Senior*, enquanto que os demais pertencerão a *Junior*.

É importante mencionar que, para cômputo do alinhamento, a alocação dos operadores é conduzida de forma não-nominal. Em outras palavras, é especificado que um humano com papel *x* deverá executar um conjunto de atividades. A alocação efetiva (ou nominal) de operadores às atividades é postergada para a etapa de agendamento (*scheduling*) da mudança. Essa etapa está fora do escopo deste artigo, sendo abordada por diversos trabalhos relacionados [Keller 2004, Rebouças 2007, Trastour 2007].

Uma vez alocado um humano para  $a_i$ , o algoritmo verifica qual o custo de executar o plano *C'* sobre a infra-estrutura gerenciada, considerando tal alocação (linha 19). Se o custo não violar o valor de limiar (*ThresholdValue*) definido em função do objetivo especificado na RFC, o processo de alinhamento continua recursivamente (20), de modo a processar as outras atividades restantes em *D*. Vale ressaltar que a semântica do *custo* (e por conseguinte, do *limiar*) está atrelada ao objetivo/restrição associado à RFC. Por exemplo, caso seja especificada a redução do tempo de execução (*ExecutionTime*), *limiar* e *custo* são avaliados em função de *tempo*.

Alternativamente, a atividade  $a_i$  extraída do conjunto  $D$  pode não necessitar de intervenção humana (21). Nesse caso, a mesma é adicionada diretamente ao plano que será retornado como solução (22). O processamento segue recursivamente (24), caso o custo de execução de  $C'$  esteja alinhado com o limiar e objetivo definidos (23).

Caso nenhum dos humanos disponíveis possa ser alocado para  $a_i$ , ou o custo para executar  $C'$  (já com a adição de  $a_i$ ) supere o valor de limiar definido, o processo de alinhamento falhará (linha 25). Conseqüentemente, as alocações de operadores realizadas em recursões anteriores serão re-feitas, de modo a explorar novas alternativas para o alinhamento. No pior caso, *recursive\_alignment* retornará o *feedback* de que não é possível alinhar o plano detalhado  $C$  ao propósito de negócio indicado na RFC.

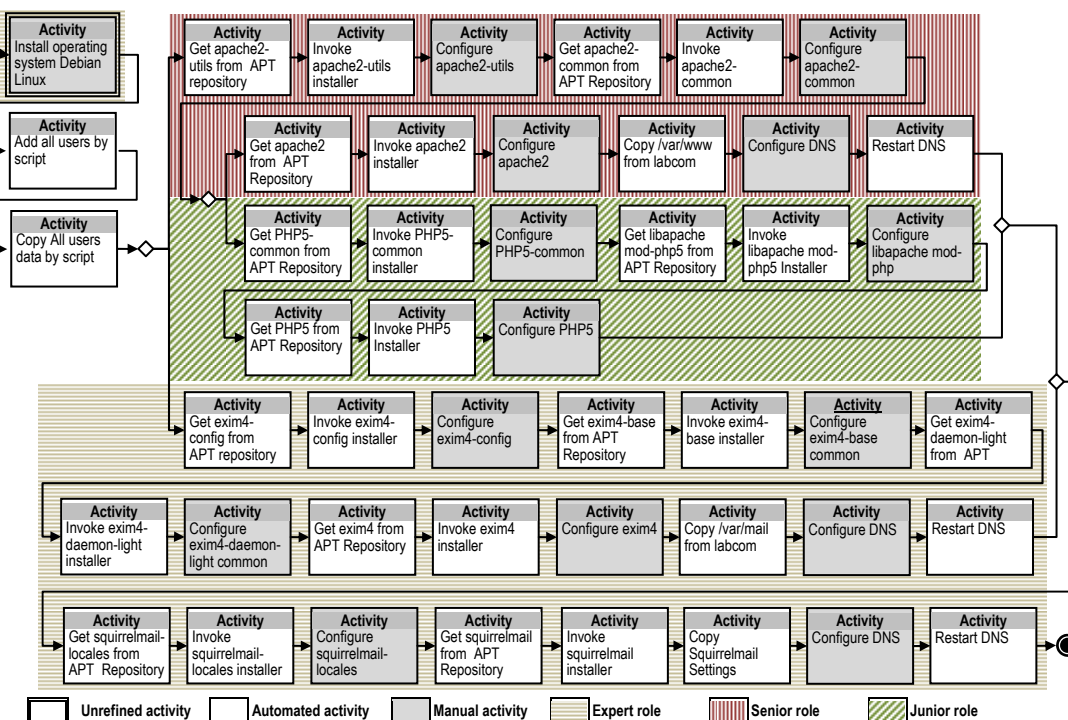


Figura 6. Plano de mudança detalhado para RFC de migração de servidor de e-mail

Para ilustrar o processo de alinhamento, considere o plano apresentado na Figura 6, gerado a partir da RFC *Migrate E-mail System* (introduzida na Figura 4). Os diferentes papéis de humanos alocados para a execução do plano são representados utilizando áreas hachuradas em torno das atividades, sendo as atividades com sombreamento cinza as que requerem intervenção humana. Nesse exemplo, há apenas um humano para cada papel definido em  $H$ , sendo *Expert*, *Senior* e *Junior*, em ordem decrescente de capacidade, os papéis utilizados. O plano apresentado obedece ao objetivo *ExecutionTime*, de acordo com o especificado na RFC. Note que as atividades relacionadas à instalação do *Exim4* (serviço de *e-mail*), e que requerem intervenção humana, possuem maior custo para serem executadas. Em função disso, essas possuem associado o melhor humano disponível (no caso, pertencente ao papel *Expert*). De forma análoga, a alocação de humanos é efetuada para as demais atividades do plano, levando em conta os paralelismos existentes, bem como os custos associados.

#### 4. Implementação

A solução para o planejamento de mudanças em infra-estruturas de rede e serviços, apresentada na seção anterior, é materializada pelo protótipo de um sistema de suporte a decisão denominado CHANGEADVISOR. Esse sistema está acomodado no contexto do

CHANGELEDGE [Cordeiro 2008a], um *framework* para a gerência de mudanças com foco no reuso de conhecimento e automação. Nesta seção serão apresentados alguns dos aspectos principais do sistema implementado, bem como as interfaces do mesmo com os componentes conceituais do *framework* CHANGELEDGE.

O sistema CHANGEADVISOR, implementado utilizando a linguagem de programação Java, materializa as funcionalidades dos componentes *ChangeRefiner* e *ChangeAligner* (destacados na área cinza da Figura 1). As especificações de mudança consumidas pelo sistema são codificadas utilizando a linguagem XML, e processadas por meio da *Streaming API for XML* (StAX). A partir das especificações recebidas, o sistema gera, como saída, planos de mudança detalhados (*workflows* executáveis) codificados em XML e respeitando o padrão *Business Process Execution Language* (BPEL) [OASIS 2008]. É importante mencionar que a escolha por BPEL se deve à grande popularidade do padrão e a sua aderência para coordenar atividades distribuídas em infra-estruturas de rede.

A manipulação de dados sobre a infra-estrutura gerenciada, além da obtenção de informações sobre os pacotes de *software* disponíveis para o processo de mudança, é feita utilizando o *framework* para mapeamento objeto/relacional Hibernate. A persistência dos dados manipulados, por sua vez, é realizada pelo sistema de gerência de banco de dados MySQL. Para desacoplar as funcionalidades do sistema CHANGEADVISOR dos aspectos relativos à persistência e recuperação de dados, foi adotado o padrão de projeto *Data Access Object* (DAO).

A comunicação entre os componentes conceituais do *framework* CHANGELEDGE e o sistema CHANGEADVISOR é feita por meio de interfaces e padrões bem definidos e entendidos por ambas as partes. Entre as interfaces existentes, merecem destaque as que comunicam o sistema com os componentes *Change Designer* e *Deployment System*. No primeiro caso, o *Change Designer* gera documentos especificados em XML, adotando como padrão o mesmo *XML Schema* que o CHANGEADVISOR. No segundo caso, os documentos BPEL gerados pelo CHANGEADVISOR são consumidos pelo orquestrador ActiveBPEL, o qual implementa as funcionalidades do *Deployment System*. O leitor interessado em informações adicionais sobre a execução dos *workflows* e a interface necessária nos CIs para sua manipulação remota poderá encontrá-las em trabalho anterior publicado recentemente pelo nosso grupo de pesquisa [Machado 2008].

## 5. Avaliação Experimental

Para avaliar a viabilidade técnica do planejamento de mudanças em TI alinhado a propósitos de negócio, foram conduzidos diversos experimentos utilizando o sistema CHANGEADVISOR. Diferentes RFCs foram submetidas, com objetivos/restrições distintos, considerando uma mesma infra-estrutura de TI como base. Por limitação de espaço, a análise é concentrada em duas dessas mudanças. Como resultado da geração de planos detalhados, foram observados o alinhamento dos *workflows* produzidos aos propósitos de negócio (caracterizando uma análise mais qualitativa da solução proposta), bem como indicadores de desempenho (análise quantitativa).

A infra-estrutura de TI empregada na avaliação é composta por 3 servidores, *A*, *B* e *C* – sendo que apenas *A* tem um sistema operacional instalado, o *Debian GNU/Linux*. Há também três papéis de operadores disponíveis para assistir a execução das mudanças, cujos custos financeiros relativos à duração de sua alocação são descritos na Tabela 1. De acordo com a Tabela 1, considerando o custo de \$40 para um operador *Expert*, se esse operador for alocado por meia hora em uma atividade, tal alocação implicará na adição de \$20 para o custo financeiro de execução da respectiva mudança. Por fim, a Tabela 2 descreve os cenários empregados na avaliação, ressaltando a finalidade das RFCs adotadas, bem como seus respectivos objetivos/restrições.

**Tabela 1. Papéis de operadores humanos disponíveis para alocação**

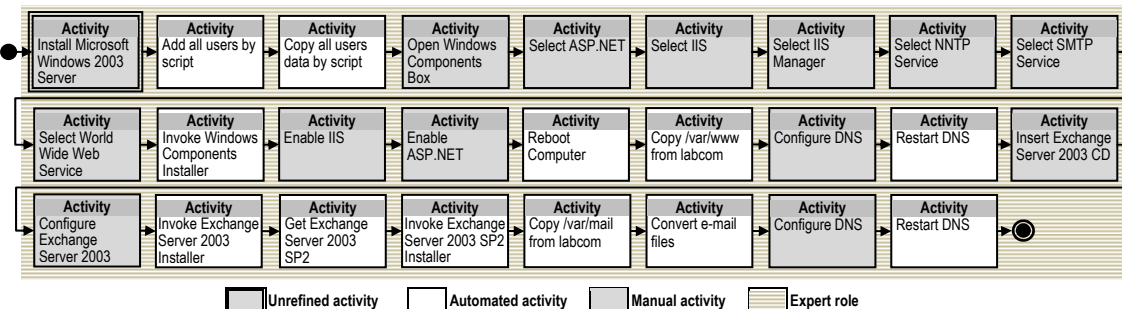
Papel	Número de operadores disponíveis	Custo de Alocação (\$ / hora)	ManPower / Hour
<i>Expert</i>	1	40	2
<i>Senior</i>	1	25	1,5
<i>Junior</i>	3	12	1

**Tabela 2. RFCs empregadas durante a avaliação experimental**

Cenário	RFC	Finalidade	Objetivo/Restrição Associada	Valor de Limiar
1	1	Migrar Serviço de <i>e-mail</i> do Servidor A para C	Diminuir Tempo Total de Implantação	120
2	1	Migrar Serviço de <i>e-mail</i> do Servidor A para C	Diminuir Custo com Alocação de Operadores	15
3	1	Migrar Serviço de <i>e-mail</i> do Servidor A para C	Nenhum	-
4	2	Instalar Aplicação Web no Servidor B	Diminuir Tempo Total de Implantação	120
5	2	Instalar Aplicação Web no Servidor B	Diminuir Custo com Alocação de Operadores	15
6	2	Instalar Aplicação Web no Servidor B	Nenhum	-

Uma visão parcial dos planos gerados para a RFC 1, nos cenários 1 e 2, é apresentada nas Figuras 6 e 7, respectivamente. Devido a restrições de espaço, o refinamento das atividades *Install Debian GNU/Linux* (Figura 6) e *Install Windows 2003 Server* (Figura 7) é suprimido em ambas as figuras. Ademais, estruturas de decisão são omitidas para facilitar a sua legibilidade. A ligação entre as atividades presentes nos planos ilustrados reflete a dependência entre os pacotes de *software* instalados, conforme identificado na DML. Por exemplo, para a instalação do *apache2* no *Debian GNU/Linux*, é necessário que as bibliotecas básicas do mesmo, providas pelo *apache2-common*, já estejam instaladas.

A partir das figuras apresentadas, observam-se as diferentes alocações de papéis para a execução das atividades *manuais*. Por exemplo, no cenário 1, a execução do plano de mudança requer três operadores com papéis distintos (um *Expert*, um *Senior* e um *Junior*) para que o objetivo especificado seja atingido. O plano do cenário 2, por sua vez, requer apenas um operador (*Expert*), de modo a satisfazer o objetivo especificado.

**Figura 7. Plano de mudança alternativo para RFC de migração de servidor de e-mail**

A Tabela 3 apresenta os custos associados à execução dos planos detalhados gerados para cada um dos cenários avaliados. Nessa tabela é possível verificar, comparativamente, que os planos detalhados são capazes de prover melhores resultados face aos objetivos/restrições especificados. Por exemplo, a RFC 1 pôde ser concretizada por dois planos distintos: um que possui tempo de execução baixo (118 min), embora apresentando alto custo de implantação (\$34,21), e outro que possui custo razoável (\$7,75), apesar de demandar mais tempo para sua execução (156 min).

A Tabela 3 apresenta, para efeito comparativo, planos alternativos não alinhados aos objetivos/restrições. Embora corretos, tais planos tendem a apresentar custos maiores em comparação às alternativas que seguem algum alinhamento. Por exemplo, o plano gerado para o cenário 3 tem custo com alocação de humanos estimado em \$45,98, valor 493% maior que o plano alinhado ao objetivo de minimização de tal custo. Tal pode ser explicado, por exemplo, pelos diferentes CIs envolvidos na mudança, em virtude dos diferentes refinamentos feitos para um mesmo plano preliminar. Esses, por sua vez, podem eventualmente implicar em atividades com diferentes tempos de duração, bem como em diferentes alocações de humanos (de papéis distintos).

**Tabela 3. Número de atividades, tempo estimado de execução e custos de alocação para os planos detalhados gerados**

Cenário	RFC	Atividades	Tempo Estimado de Execução (min)	Custo com Alocação de Humanos (\$)
1	1	69	118,75	34,21
2	1	38	156,12	7,75
3	1	68	157,08	45,98
4	2	52	75,5	18,73
5	2	39	211,31	10,42
6	2	49	250,75	28,43

Note que os valores de custos obtidos, se analisados individualmente, podem ser desprezíveis. No entanto, ao considerar a execução de diversas mudanças a longo prazo, tal alinhamento tem potencial para reduzir os custos de gerência e operação da empresa.

O desempenho do sistema CHANGEADVISOR para gerar os planos executáveis caracterizados acima é apresentado na Tabela 4. Executado em um computador equipado com um processador Intel® Core™ 2 Duo 2.33 GHz, 4.096 KB de memória *cache* e 2 GB de memória RAM, o sistema mostrou-se bastante eficiente, exigindo poucos segundos para gerar os planos. Também foi calculado um intervalo de confiança de 95% sobre os tempos medidos, considerando 10 repetições do processo de geração de planos de mudança. Conforme apresentado na Tabela 4, é esperado que o tempo de geração de planos varie minimamente para cada cenário. Esses resultados permitem observar que a solução além de gerar planos alinhados a propósitos de negócio, também tem potencial para reduzir, sensivelmente, o esforço e o tempo exigidos para tal.

**Tabela 4. Tempo despendido na geração automática dos planos detalhados**

Cenário	Tempo Total de Geração dos Planos Detalhados (ms)	Intervalo de Confiança do Processo de Geração de Planos Detalhados	
		Limite Inferior (ms)	Limite Superior (ms)
1	17.475	17.247	17.703
2	14.185	14.030	14.340
3	16.761	16.680	16.842
4	16.003	15.875	16.130
5	16.539	15.927	17.150
6	15.240	15.047	15.432

## 6. Conclusões e Trabalhos Futuros

O planejamento de mudanças representa uma etapa fundamental para a gerência e a operação de infra-estruturas de redes e serviços. No entanto, as soluções existentes para auxiliar essa etapa não levam em consideração a geração de planos de mudança alinhados aos propósitos definidos em nível de negócio. Conseqüentemente, o planejamento e a execução de mudanças apresentarão pouca (ou nenhuma) sintonia com os objetivos/restrições demandados pelo solicitante. Para abordar esse problema, neste artigo é proposto o CHANGEADVISOR, uma solução para geração de planos de mudança consistentes com os objetivos/restrições especificados em nível de negócio.

Os resultados obtidos durante a avaliação experimental, embora não exaustivos, mostram os potenciais benefícios de se alinhar planos de mudança a propósitos de negócio. Conforme apresentado anteriormente, vários planos detalhados foram obtidos a partir de uma mesma RFC, cada um atendendo a requisitos específicos de tempo e custo de execução. Pôde-se observar, ainda, que os planos gerados foram sensivelmente diferentes tanto em forma (atividades componentes), quanto em custos agregados (tempo e custo financeiro). Por fim, a geração de planos detalhados que atendem às demandas do solicitante mostrou-se exequível e muito eficiente. Tal pôde ser observado pelo tempo despendido na geração dos mesmos, o qual ficou na ordem de segundos. Esse tempo certamente é de grandeza menor que o tempo que seria gasto por um operador experiente, projetando os mesmos planos utilizando um editor de *workflow*. É natural que, para aproveitar o potencial de um sistema para o planejamento de mudanças, é preciso contar com um modelo completo e atualizado da infra-estrutura de

redes e serviços da organização. Estratégias e ferramentas para satisfazer esse requisito não fizeram parte da investigação apresentada neste artigo, mas são plenamente alcançáveis. Tal convicção advém do fato de que modelo como esse é apresentado como fundamental pela ITIL, biblioteca amplamente adotada em organizações de pequeno, médio e, sobretudo, grande portes.

Como trabalhos futuros, pretende-se (i) investigar estratégias para aperfeiçoar o algoritmo de alinhamento de mudanças proposto neste artigo, possivelmente adotando conceitos de inteligência artificial, (ii) estender o escopo do processo de alinhamento de mudanças, levando em conta objetivos/restrições de negócio diversos, e (iii) adotar heurísticas que permitam a adequação dos planos gerados a mais de um objetivo/restrição de negócio simultaneamente.

## Referências

- Banyahia, H. (1996) "Costs and Productivity Estimation in Computer Engineering Economics". *Engineering Economist*, v. 41, n. 3, p. 229-241.
- Brown, A. B.; Keller, A. (2006) "A Best Practice Approach for Automating IT Management Processes". In: *IFIP/IEEE Network Operations and Management Symposium (NOMS 2006)*, p. 33-44.
- Cordeiro, W.; Machado, G.; Daitx, F. *et al.* (2008) "A Template-based Solution to Support Knowledge Reuse in IT Change Design". In: *IFIP/IEEE Network Operations and Management Symposium (NOMS 2008)*, p. 355-362.
- Cordeiro, W.; Machado, G.; Andreis, F. *et al.* (2008) "A Runtime Constraint-Aware Solution for Automated Refinement of IT Change Plans". In: *IFIP/IEEE International Workshop on Distributed Systems: Operations and Management (DSOM 2008)*, p. 69-82.
- Distributed Management Task Force. DMTF. (2008) Common Information Model. <http://www.dmtf.org/standards/cim>.
- Information Technology Infrastructure Library. ITIL. (2008) Office of Government Commerce (OGC). <http://www.itil-officialsite.com>.
- IT Infrastructure Library: ITIL Service Transition, v.3 (2007) London: The Stationery Office, 2007, 270 p.
- Jorgensen, M.; Shepperd, M. (2007) "A Systematic Review of Software Development Cost Estimation Studies". *IEEE Transactions on Software Engineering*, v. 33, n. 1, p. 33-53.
- Keller, A.; Hellerstein, J. L.; Wolf, J. L. *et al.* (2004) "The CHAMPS System: Change Management with Planning and Scheduling". In: *IEEE/IFIP Network Operations and Management Symposium (NOMS 2004)*, p. 395-408.
- Keller, A. (2005) "Automating the Change Management Process with Electronic Contracts". In: *IEEE International Conference on E-Commerce Technology Workshops (CECW 2008)*, p. 99-108.
- Machado, G.; Cordeiro, W.; Santos, A. *et al.* (2008). "Algoritmo para Geração Automática de Ações de Rollback em Sistemas de Gerenciamento de Mudanças em TI". In: *Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC 2008)*, p. 107-120.
- Moura, A.; Sauv e, J.; Bartolini, C. (2008) "Business-driven IT Management - Upping the Ante of IT: Exploring the Linkage between IT and Business to Improve both IT and Business Results". *IEEE Communications Magazine*, v. 46, n. 10, p. 148-153.
- Organization for the Advancement of Structured Information Standards. OASIS. (2008) "Business Process Execution Language, version 2.0". <http://docs.oasis-open.org/wsbpel/2.0>.
- Pillai, K.; Sukumaran Nair, V. S. (1997) "A Model for Software Development Effort and Cost Estimation". *IEEE Transactions on Software Engineering*, v. 23, n. 8, p. 485-497.
- Rebouças, R.; Sauv e, J.; Moura, A. *et al.* (2007) "A Decision Support Tool to Optimize Scheduling of IT Changes". In: *IFIP/IEEE International Symposium on Integrated Network Management (IM 2007)*, p. 343-352.
- Rodosek, G. D. (2003) "A Generic Model for IT Services and Service Management". In: *IFIP/IEEE International Integrated Network Management (IM 2003)*, p. 171-184.
- Shepherd, M. (1977) "Special Feature Distributed Computing Power: a Key to Productivity". *IEEE Computer*, v. 10, n. 11, p. 66-74.
- Trastour, D.; Rahmouni, M.; Bartolini, C. (2007) "Activity-Based Scheduling of IT Changes". In: *International Conference on Autonomous Infrastructure, Management and Security (AIMS 2007)*, p. 73-84.

## ANEXO B ARTIGO PUBLICADO NO DSOM 2009

Neste anexo, o artigo intitulado “ChangeAdvisor: A Solution to Support Alignment of IT Change Design with Business Objectives/Constraints” é apresentado. Neste trabalho, os modelos e o protótipo sofreram algumas importantes melhorias. Além disso, essa foi a primeira publicação no tema desta dissertação em evento científico internacional.

- **Título:** ChangeAdvisor: A Solution to Support Alignment of IT Change Design with Business Objectives/Constraints

- **Conferência:** 20th International Workshop on Distributed Systems: Operations and Management (DSOM 2009)

- **URL:** <http://www.manweek.org/2009/dsom/>

- **Período da realização do evento:** 26-30 de Outubro de 2009

- **Local:** Telecom Italia Future Centre, Veneza, Itália

- **Digital Object Identifier (DOI):** [http://dx.doi.org/10.1007/978-3-642-04989-7\\_11](http://dx.doi.org/10.1007/978-3-642-04989-7_11)



## CHANGEADVISOR: A Solution to Support Alignment of IT Change Design with Business Objectives/Constraints

Roben Castagna Lunardi<sup>1</sup>, Weverton Luis da Costa Cordeiro<sup>1</sup>,  
 Fabrício Girardi Andreis<sup>1</sup>, Juliano Araujo Wickboldt<sup>1</sup>, Cristiano Bonato Both<sup>1</sup>,  
 Luciano Paschoal Gaspar<sup>1</sup>, Lisandro Zambenedetti Granville<sup>1</sup>,  
 David Trastour<sup>2</sup>, and Claudio Bartolini<sup>3</sup>

<sup>1</sup> Institute of Informatics, Federal University of Rio Grande do Sul, Brazil

<sup>2</sup> HP Laboratories Bristol, UK

<sup>3</sup> HP Laboratories Palo Alto, USA

{rclunardi, weverton.cordeiro, fgandreis, jwickboldt,  
 cbboth, paschoal, granville}@inf.ufrgs.br,  
 {david.trastour, claudio.bartolini}@hp.com

**Abstract.** Change planning represents a key element for the operation and management of Information Technology infrastructures and services. Its scope ranges from the high level design of a change request to the generation, either manually or automatically, of detailed plans that, if executed, will perform the requested changes (*e.g.*, modification of network device settings and deployment of new services). A fundamental problem is that, although correct, such detailed plans may not be necessarily aligned with the requirements defined in the business level (*e.g.*, minimization of the downtime of a given service). To overcome this problem, in this paper we propose a solution for the alignment of change plans with business objectives/constraints. The solution is analyzed experimentally through a prototypical implementation of a decision support system called CHANGEADVISOR, which helps operators to understand the trade-offs between alternative change designs.

**Keywords:** IT service management, business-driven IT management, IT change management.

### 1 Introduction

The increasing size and complexity of Information Technology (IT) infrastructures and services have demanded the adoption, by organizations, of best practices and processes in order to ensure their correct and efficient operation. Aiming at supporting organizations in such a challenging task, the Information Technology Infrastructure Library (ITIL) [1], an important reference in this context, recommends, among other processes, *change management*. This process helps to efficiently handle the necessary changes in hardware and software within organizations.

The scope of *change management* ranges from the change specification, by an initiator in a high level of abstraction, to the generation of detailed plans – those that comprehend low level activities such as modification of network device settings and

deployment of new services. These plans, once executed, tend to accomplish the changes requested by an initiator in the managed infrastructure.

One of the main problems considering the generation of change plans is that, from the same high level specification, different detailed plans can be obtained. Although correct, they may not be necessarily aligned with the purposes defined for the IT infrastructure in the business level. The purposes can either have a technical nature, such as imposing constraints regarding the unavailability of provided services by the managed IT infrastructure, or define objectives for optimizing resource allocation during the change deployment (*e.g.*, minimizing the number of human resources involved in an e-mail service maintenance procedure). When these purposes are not considered, the generated plans, if executed, may lead to results that violate policies predefined in the business level (*e.g.*, e-mail service maintenance downtime must not exceed 10 hours monthly).

We highlight three important reasons for aligning IT change plans with business purposes (also referred to as business objectives/constraints in this paper). First, there is the possibility of optimizing available resources that are usually limited and expensive. Second, the alignment has the potential of reducing costs such as the time spent to deploy changes. And finally, perhaps the most important reason, aligning change plans with distinct purposes allows the operator to understand the resulting trade-offs when one chooses a change plan among various possibilities. Despite the potential benefits, the alignment of change plans with business objectives/constraints has been neglected in investigations carried out in the area of IT service operations and management. The generation of executable [2] and optimized plans [3] are examples of issues that, alternatively, have been addressed recently.

To tackle the aforementioned problem, in this paper we propose a solution for the alignment of IT infrastructure and service change plans with business objectives/constraints. In contrast to previous investigations conducted in the area, this paper focuses on the automated generation of change plans guided by business objectives/constraints. The proposed solution is experimentally evaluated through CHANGEADVISOR, a prototypical implementation of a decision support system that provides means for the understanding of trade-offs among alternative change designs. During the evaluation, which was performed considering scenarios based on real cases, we analyzed correction and completeness of generated plans, as well as their alignment with business purposes.

The remainder of this paper is organized as follows. In Section 2 we review some key investigations in the area of IT change management. In Section 3 a conceptual solution for the generation of change plans aligned with business purposes is presented. We detail, in Section 4, the CHANGEADVISOR system and present, in Section 5, the results achieved using the system. Finally, we conclude the paper conveying final considerations and perspectives for future work in Section 6.

## 2 Related Work

The IT service management area has grabbed the attention of the scientific community in recent years. Several aspects, such as models [4], automation [5], and alignment with business purposes [6] have been explored. Specifically analyzing the subarea of

change management, it is possible to observe similar activity. For example, important investigations about scheduling [7] and rollback [8] can be cited. Nevertheless, the topic of IT change planning, which is the focus of this paper, has not been sufficiently explored, as underscored in the following two paragraphs.

Keller *et al.* [3] have proposed CHAMPS, a system for the automated generation of change plans, which explores a high degree of parallelism in task execution. Although it has proved to be able to improve task scheduling considering available hardware resources, neither the activity association to human resources nor the objectives/constraints (specified by the change initiator) have been taken into account.

In a recent paper, Cordeiro *et al.* [2] have proposed a solution to *formalize, preserve, and reuse* knowledge acquired with frequent IT changes. The solution, which is based on *change templates*, allows recurrent activities (*e.g.*, service change or implementation) to be specified and (recurrently) reused. Although this solution comprises an algorithm to generate detailed plans from high level specifications, aligning generated plans with business purposes is out of its scope. In a subsequent work [9], the mentioned algorithm has been improved to consider technical constraints imposed by the managed environment (*e.g.*, availability of resources such as disk and memory space). However, business purposes have not been considered.

Other investigations, although not designed to deal with change management, have explored alignment with the business level. One of them is Keller's [10], which introduced the concept of electronic contracts. In that research, the authors have proposed four kinds of contracts: RFCs, Deployment Descriptors, Policies and Best Practices, and Service Level Agreements (SLAs). Such contracts have different purposes, formats, and level of details. They are used for objective/constraint specification and are of great importance for the change deployment process.

Summing up, although change management has been a focus of recent investigations, they have not considered alignment of changes with specific purposes in the business level. It means that the planning and subsequent change deployment may potentially be conducted in an undesired direction, *i.e.*, may not be aligned with purposes demanded by the change initiator. In order to deal with this problem, the following sections present not only a conceptual solution, but also a system developed to support change alignment with business purposes and its evaluation.

### 3 Conceptual Solution

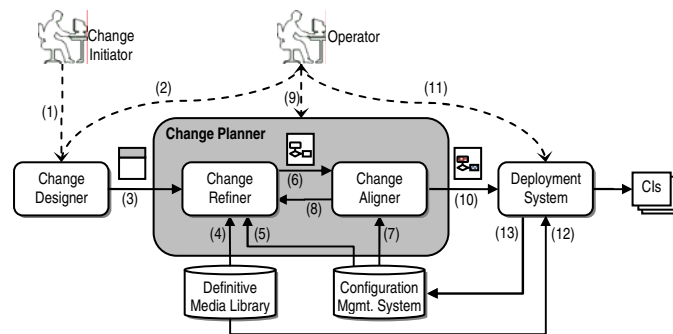
Aiming to support the generation of change plans aligned with business purposes, two new key components have been introduced to the conceptual solution for change design and planning proposed in a previous work [9]. The new components – highlighted in gray – materialize the mechanism for the planning of changes guided by business purposes and fit adequately into the previously proposed solution, without significant modifications. Fig. 1 presents an overview of the extended solution, focusing on its main components, involved actors, and their interactions.

The initiator starts a change process interacting with the component *Change Designer* (flow 1 in Fig. 1) to specify a *Request for Change* (RFC). RFCs describe what changes must be done in the managed infrastructure, the *Configuration Items* (CIs) primarily affected (network devices, computers, services, applications, etc.) and the

business objectives to be met. However, the RFC does not specify details on how the change should be executed; such details have to be specified in the following step by the design of a preliminary plan, supported by the component *Change Designer* (flow 2). This plan consists in a workflow of actions that describe, in a high level of abstraction, how the requested change should be performed in the IT infrastructure.

The generation of a detailed change plan based on the preliminary specification is then performed, without human intervention, by the component *Change Refiner* (3). The automated processing, carried out by a refinement algorithm, is based on both (i) information about software packages, available in the *Definitive Media Library* (DML) (4), and (ii) information about the target IT infrastructure, found in the *Configuration Management System* (CMS) (5).

After the change refinement, the resulting plan is forwarded to the component *Change Aligner* (6), which plays a central role in the alignment of the plan with the business objectives/constraints expressed in the RFC. In order to guide this alignment, the *Change Aligner* retrieves from the CMS (7) information about costs and skills of available human resources (e.g., an operator who is an expert in e-mail service and whose work hour costs 40 monetary units). It also gathers information about the capabilities of the hardware assets affected by the plan (e.g., computational power in the case of workstations and servers). It is important to highlight that the process of change refinement and alignment is reiterated (8) during a period of time predetermined by the operator, aiming to generate distinct detailed plans aligned to the same business purposes determined in the RFC. This is due to the fact that the problem is NP-hard, and the algorithms employed to solve this problem are heuristic-based, i.e., the techniques used take advantage of information on past experiences in order to obtain results closer to the optimal (best) solution.



**Fig. 1.** Conceptual architecture of the proposed solution

In a later moment (9), the generated plans may be modified by the operator as to reflect his/her needs, and evaluated in terms of trade-offs between the various options. Among the criteria that may guide such an analysis, we cite: expected time and/or financial costs to perform the changes and human resources required. Based on this analysis, the operator will be able to select an appropriate plan to be deployed.

Finally, in the last step of the change process, the operator invokes the deployment of the selected plan (10), interacting with the component *Deployment System* (11).

This component may consume software packages available in the *Definitive Media Library* (12) in order to be able to execute some of the activities described in the change plan. After deploying this plan, the *Deployment System* updates the CMS, adding information about modifications in each of the affected CIs (13).

It is important to mention that the conceptual solution depicted in Fig. 1 is targeted at the *design*, *planning*, and *deployment* of changes. Therefore, other phases that comprise the traditional change management process – such as the *evaluation* of the requested changes by the *Change Advisory Board* (CAB) and the *schedule* of changes into maintenance windows – are envisaged as directions and trends for long-term, future investigation in this area.

Having presented an overview of our solution, in the following subsections we describe (i) the generation of detailed plans based on preliminary specifications and (ii) the process of aligning plans with purposes determined in the business level.

### 3.1 Change Refiner

The component *Change Refiner* is fundamental in the process of generating alternative detailed change plans based on a preliminary specification. The core of this component is a refinement algorithm, inspired in a previous work [9]. As previously mentioned, the generation of detailed change plans requires not only information about the CIs (*e.g.*, hardware, software, and people) that compose the IT infrastructure and their relations, but also about the software packages available for the change deployment. Aiming to represent this information in the CMS and in the DML, respectively, a subgroup of the *Common Information Model* (CIM) is used in this paper [11]. Fig. 2 presents a partial view of the model. Relationships such as associations, compositions, and aggregations, most of them omitted in the figure for better intelligibility, express dependencies among the elements of the infrastructure.

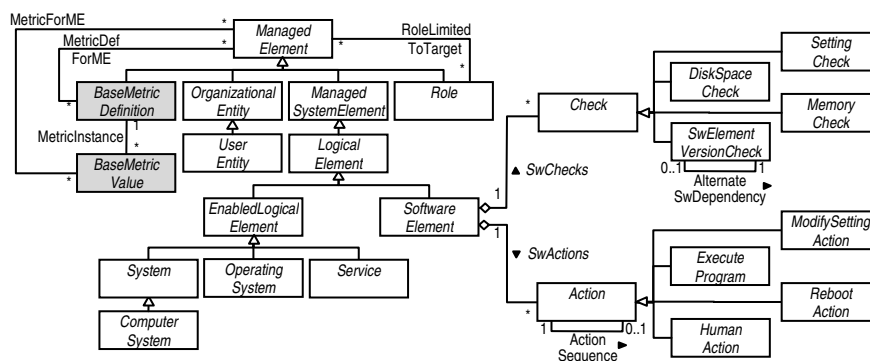


Fig. 2. Information model used to represent the IT Infrastructure and the DML

The model also incorporates the classes *Check* and *Action*, which denote necessary information for computing dependencies refining the activities of a preliminary plan. An instance of the class *Check* defines a condition to be met or a characteristic required for the associated software to evolve into a new state (*deployable*, *installable*, *executable* or *running*).

Fig. 3(a) illustrates instances of *Check* associated to the software *Squirrelmail*. The first two indicate that the software packages *Exim4* and *Apache2* must be installed before the *Squirrelmail* installation. The last two express that *Squirrelmail* requires 1MB of memory for installation and 3MB of disk space for execution. Each instance of the class *Action*, in turn, represents a mandatory activity to change the state of the associated software (e.g., from *installable* to *executable*, i.e., an installation process). Using once more the example of *Squirrelmail*, two activities associated with its installation process are illustrated in 3(a): *Install Squirrelmail*, responsible for the copies of *Squirrelmail* binary files; and *Configure Squirrelmail*, which performs the configurations necessary for the webmail to operate properly.

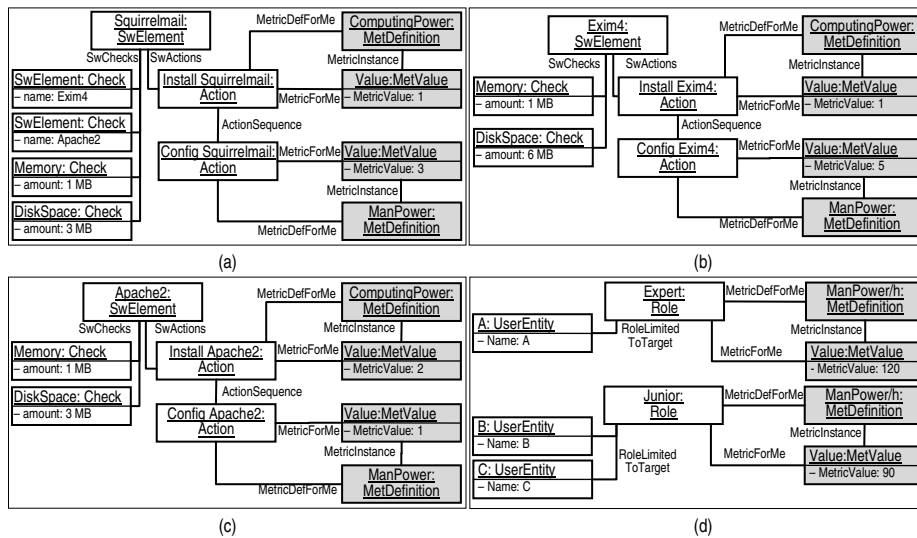


Fig. 3. Software Elements present in DML (a,b,c) and roles in the IT infrastructure (d)

The algorithm for change refinement, also known as *change\_refinement*, receives as input the change specification to be refined, along with data about the current state of the managed infrastructure (from CMS) and the software packages from the DML. For each activity composing the preliminary plan, *change\_refinement* computes the set of activities that should be executed before, so as to satisfy the requirements (e.g., installation of software dependencies) for the activity to be executable.

To illustrate the refinement process, consider the RFC *Migrate E-mail Server*, presented in Fig. 4 (background in grey). It consists in the migration of the e-mail service from server *X* to server *Y*, and aims at minimizing the deployment time estimated in, at most, 120 minutes (in a way not to compromise the normal operations of the organization). The preliminary plan associated (background in white) is composed of six high level activities that range from the installation and configuration of a new server to the migration of data from the “old” (*X*) to the “new” server (*Y*).

Focusing on the activity *Install Squirrelmail* (illustrated in the preliminary plan), *change\_refinement* identifies – through the *Checks* presented in DML (Fig. 3(a)) – that the execution of this activity depends on the previous installation of software

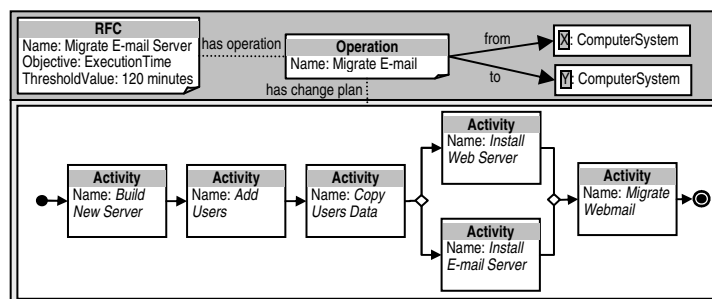


Fig. 4. Example of RFC and preliminary change plan for the migration of an e-mail server

packages *Apache2* (Web server) and *Exim4* (e-mail server). Consequently, two new activities are generated to satisfy the demands of *Install Squirrelmail: Install Apache2* (a) and *Install Exim4* (b). This process goes on recursively until all the requirements of the activities of the preliminary change plan are met. At the end of this process, when a detailed plan is obtained (Fig. 6), *change\_refinement* stores the “current state” (refinement paths traversed to generate this plan). This step is important because, after the alignment (seen in detail in the Subsection 3.2), it will be possible to generate distinct, detailed plans, from the same initial specification, using alternative refinement “paths”.

### 3.2 Change Aligner

The component *Change Aligner*, a key element for the alignment of detailed plans with business purposes, is grounded in three building blocks: (i) formalization of business objectives/constraints, (ii) information models to represent cost/capacity metrics, and (iii) algorithm for change alignment. These are described next.

#### 3.2.1 Business Objectives/Constraints and Cost/Capability Metrics

To make it possible for change plans generated by the component *Change Refiner* to be aligned with business purposes, these must be specified in the RFC. Two business purpose categories are defined in this first iteration to solve the problem: *Execution Time* and *Execution Cost* to perform the change. These objectives were indicated as the most relevant ones by a group of network/system managers (working in Brazilian IT companies) interviewed. The specification of an objective/constraint needs to be accompanied by a *Threshold Value* representing the maximum time or cost tolerated. As shown in Fig. 4, the RFC *Migrate E-mail Server* specifies that the change, when planned, should consider at most 120 minutes for deployment.

Cost and capability metrics are also essential for the alignment to be performed. Whereas the former are used to describe the effort demanded in the execution of activities related to IT service operations and management, the latter allow expressing expectation on the performance of humans and computers to execute them. The metrics employed in this paper, described next, have already been adopted and consolidated in areas such as Distributed Systems [12], Software Engineering [13], and Economy [14]. Concerning information modeling, the metrics are specified, and their values instantiated, in the IT infrastructure model (Fig. 2) by means of the classes *Base Metric Definition* and *Base Metric Value*.

The cost metrics are subdivided in *Man Power* (MP) and *Computing Power* (CP). MP is used to quantify human effort required for executing an activity. CP, in turn, denotes computational effort required for executing an automated activity, *i.e.*, with no (or minimal) human intervention. Value instances for these metrics are used to express the costs of activities associated with software elements registered in the DML. In Fig. 3, it is possible to observe, as an illustration, that activities like *Install Squirrelmail* (a), *Install Exim4* (b), and *Install Apache2* (c) have an associated execution cost, in CP, of 1, 1, and 2 units, respectively. Additionally, the activities for the configuration of these services, namely *Config Squirrelmail* (a), *Config Exim4* (b), and *Config Apache 2* (c), have an associated cost, in MP, of 3, 5, and 1 units.

Similarly to what happens to cost metrics, the capability metrics are also subdivided in two: *Man Power/hour* (MP/H) and *Computing Power/hour* (CP/H). They represent the amount of work required from humans and computers to perform the execution of activities. In the case of MP/H, instance values are associated with roles played by the board of human resources available in the IT department of the organization. In the example illustrated in Fig. 3 (d), two roles are modelled, *Expert* and *Junior*, with MP/H equal to 120 and 90 units, respectively. Human resources are then associated to these roles, according to the functions they are able to perform in the department. MP/H instance values, conversely, are attributed to each of the computers present in the IT infrastructure, reflecting its expected performance.

We highlight that the investigation about ways of valuing costs and capabilities, although fundamental for the proposed solution, are not the scope of this paper. To fill in this gap, we assume the employment of one of the approaches proposed in the literature (some of them have been compiled by Jorgensen [15]). More specifically, in this paper, we adopted an empirical method, which consists of valuing simpler activities – not only automated but also of human-based execution – assigning to them a value of 1 CP and 1 MP, respectively. Having this as a basis, costs of other activities are estimated. The estimation of computing resource capabilities (CP/H) and human ones (MP/H), in turn, is performed by analysing historic data.

### 3.2.2 Alignment of Change Plans with Business Objectives/Constraints

The building blocks previously presented form the basis upon which the algorithm for aligning change plans with business purposes is built. The algorithm, shown in Fig. 5, is denoted by *change\_alignment* ( $C, DML, IT, H$ ), where  $C$  is the refined plan generated by *change\_refinement*;  $DML$  is the repository of that contains information about software packages available;  $IT$  is the repository of information about the managed infrastructure; and  $H$  is the set of human resources available for conducting changes.

As a first step of the alignment process, an ‘empty’ change plan is created (line 1 in Fig. 5). After that, the activities contained in  $C$  are copied to the set  $D$  (line 2), and the function *recursive\_alignment* is invoked (4).

The alignment process, as described in *recursive\_alignment*, starts by verifying if both  $D$  (the set of activities which have not been processed) is empty and plan  $C$  received satisfies the business objectives determined in the *RFC* (line 7). If so, plan  $C$  is returned as a solution. Otherwise, the following steps are executed. First, an activity  $a_i$  is extracted from set  $D$  for processing (line 9). Second, the algorithm verifies if  $a_i$  requires a human to be executed (line 10). In case it does, the algorithm tries to allocate the most capable human  $h$  (available in  $H$ , the group of human resources), who has not



been allocated in any other parallel activity to  $a_i$  (lines 11-18). Human resources are allocated in a descending order in relation to the role they are associated with. Analyzing the roles *Junior* and *Expert* illustrated in Fig. 3(d), the most capable humans belong to *Expert's*.

It is important to mention that, for the alignment computation, the allocation of humans is conducted in a “non-nominal” way, *i.e.*, there is the specification that a human playing role  $x$  should execute a set of activities. The effective (or nominal) allocation of human resources to the activities is postponed to the change scheduling stage, which is out of the scope of this paper and has been investigated by several related work [3, 7].

```

01 change_alignment (C, DML, IT, H)
02   C' ← empty change plan
03   D ← set of activities from C
04   recursive_alignment (C, C', D, DML, IT, H)
05
06 recursive_alignment (C, C', D, DML, IT, H)
07   if D is empty, cost of deploying C' comply with objective threshold, given DML, IT, then
08     return C' as solution
09   extract activity  $a_i$  from D
10   if  $a_i$  is a human activity then
11     for all h in H do
12       for all  $a_j$  in C do
13         allocable ← true
14         if  $a_i$  is parallel to  $a_j$  in C and h is allocated to  $a_j$  then
15           allocable ← false
16         if allocable is true then
17           associate human h to  $a_i$ 
18           add activity  $a_i$  to change plan C'
19           if cost of deploying C' comply with objective threshold, given DML, IT, then
20             recursive_alignment (C, C', D, DML, IT, H)
21         else
22           add activity  $a_i$  to change plan C'
23           if cost of deploying C' comply with objective threshold, given DML, IT, then
24             recursive_alignment (C, C', D, DML, IT, H)
25   return failure

```

Fig. 5. Algorithm for the alignment of change plans with business objectives/constraints

Once a human is allocated to  $a_i$ , the algorithm verifies the cost of executing plan  $C'$  on the managed infrastructure, considering such allocation (line 19). If the cost does not exceed *Threshold Value* (specified in the RFC), the alignment process continues recursively (line 20), in order to process the remaining activities in  $D$ . The semantics of *cost* (and *threshold*) depends on the RFC objective/constraint. For example, if *Execution Time* is chosen, then *cost* and *threshold* are processed considering time units.

Alternatively, the activity  $a_i$  extracted from set  $D$  may not require human intervention (line 21). In this case, it is directly added to the plan that will be returned as a solution (line 22). The processing continues recursively (line 24), in case the execution cost of  $C'$  is aligned with the defined objectives (line 23).

If none of the humans available can be allocated to  $a_i$  or the cost for executing  $C'$  (added to  $a_i$ ) exceeds *Threshold Value*, the alignment process fails (line 25). Consequently, the allocation of human resources done up to this moment are recursively rolled back, and new alternatives for the alignment are explored. In the worst case, *recursive\_alignment* is not able to produce a plan aligned with the business purpose specified in the RFC, in which case a detailed feedback is returned to the operator.

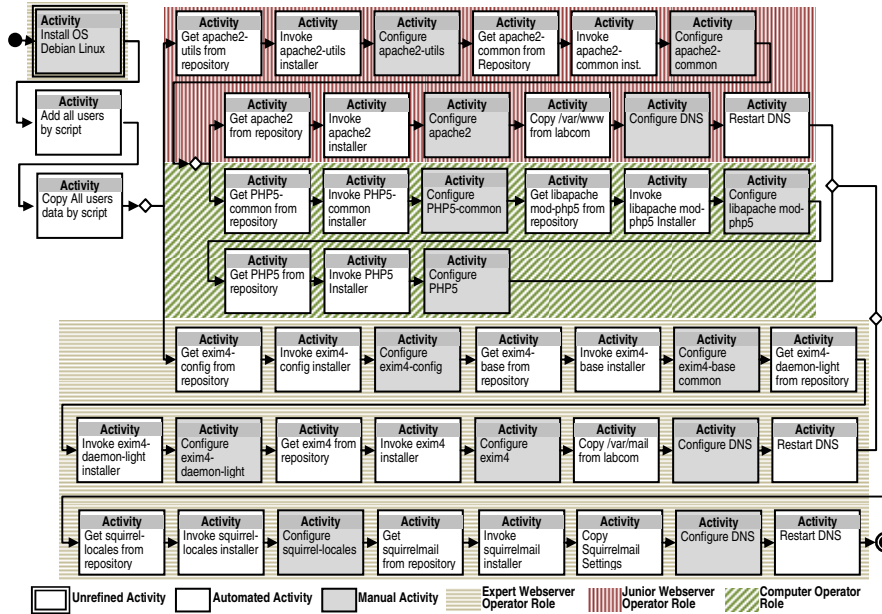


Fig. 6. Detailed change plan for the RFC Migrate E-mail Server

To illustrate the alignment process, consider the change plan presented in Fig. 6, generated for the RFC *Migrate E-mail Server* (Fig. 4). The different human roles allocated for the plan execution are represented using hachured areas around the activities, which have a grey background when human intervention is required. In this example, there is only one human associated with each role defined in  $H$ , being *Expert Webserver Operator*, *Junior Webserver Operator*, and *Computer Operator* the roles from where humans have been chosen. The presented plan meets the *Execution Time* objective, according to what was specified in the RFC. Note that the activities related to the installation of *Exim4* (e-mail system) and that require human intervention have a higher cost to be executed. Therefore, they are associated with the most capable available human (belonging to role *Expert Webserver Operator*). In the same way, the allocation of humans for other activities in the plan is done considering the existent parallelisms, as well as the associated costs.

## 4 Implementation

The solution to support alignment of change plans with business purposes presented in Section 3 has been implemented as a prototype of a decision support system called CHANGEADVISOR. This system is a key part of CHANGELEDGE [2], a framework for change management focused on the reuse of knowledge and automation. In this section, we present some of the main aspects of the implemented system, as well as its interfaces with the CHANGELEDGE components.

CHANGEADVISOR, implemented using the Java programming language, performs the functionalities of the components *Change Refiner* and *Change Aligner* (grey area of Fig. 1). The change specifications consumed by the system are coded using XML and processed through the *Streaming API for XML* (StAX). From the received specifications, the system generates detailed change plans (executable workflows), which are also coded in XML and respect the *Business Execution Language* (BPEL) [16]. The choice for BPEL is due to the widespread use of the standard and to its suitability for coordinating distributed activities in IT infrastructures. In regard to data handling and persistence (*e.g.*, about the managed infrastructure and the software packages available), both the Hibernate object-relational mapper and the MySQL database management system have been employed.

The communication between CHANGELEDGE components and the CHANGEADVISOR system is done through well-defined interfaces. Among the existing interfaces, the ones that “connect” the system with the components *Change Designer* and *Deployment System* are highlighted. In the former, *Change Designer* generates XML documents, adopting as a pattern the same XML schema as CHANGEADVISOR. In the latter, the BPEL documents generated by CHANGEADVISOR are consumed by the orchestrator ActiveBPEL, which implements the *Deployment System* functionalities. Please refer to [8] for additional information concerning the operation of the *Deployment System*.

## 5 Experimental Evaluation

In order to evaluate the technical feasibility of IT change planning aligned with business objectives/constraints, we have carried out several experiments using CHANGEADVISOR. RFCs with different objectives/constraints have been submitted, considering the same IT infrastructure as a basis. For space limitations, the analysis is focused on two of the changes.

**Table 1.** Number of humans, cost, and capacity per role

Role	Number of Available Operators	Allocation Cost (\$ / hour)	ManPower / Hour
<i>Expert Webserver Operator</i>	1	40	2
<i>Junior Webserver Operator</i>	1	25	1.5
<i>Computer Operator</i>	3	12	1

**Table 2.** RFCs used in the experimental evaluation

Scenario	RFC	Goal	Objective/Constraint	Threshold Value
1	1	Migrate Webmail from server A to C	Reduce Execution Time	120
2	1	Migrate Webmail from server A to C	Reduce Execution Cost	15
3	1	Migrate Webmail from server A to C	None	-
4	2	Install Web Application in Server B	Reduce Execution Time	120
5	2	Install Web Application in Server B	Reduce Execution Cost	15
6	2	Install Web Application in Server B	None	-

The IT infrastructure employed in this evaluation is composed by 3 servers – A, B and C – out of which only A has *Debian GNU/Linux* installed. There are also three roles available, from which humans resources may be chosen to assist change execution. Table 1 shows the number of humans playing each role, the capacity, and the allocation cost per hour (per role). Table 2 describes the scenarios employed in the evaluation, highlighting the goals of the RFCs, as well as their respective objectives/constraints.

A partial view of the plans generated for RFC 1, in scenarios 1 and 2, is presented in Figs. 6 and 7, respectively. Due to space restrictions, the refinement of *Install Debian GNU/Linux* (Fig. 6) and *Install Windows 2003 Server* (Fig. 7) is suppressed in both figures, as well as decision structures. The activities present in the illustrated plans are consistently connected and ordered, respecting the dependency information specified in the DML. Note, for example, that the installation of *apache2* in *Debian GNU/Linux* is correctly preceded by the installation of its basic libraries.

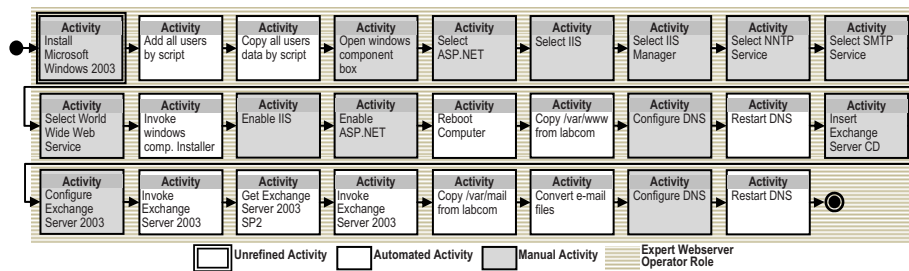


Fig. 7. Alternative change plan for the RFC Migrate E-mail Server

One can also observe in Figs. 6 and 7 that different human roles are required for the execution of different “niches” of manual activities, so as to meet the specified business objective/constraint. For example, the change plan generated for scenario 1 requires three humans, each playing a different role (one *Expert Webserver Operator*, one *Junior Webserver Operator*, and one *Computer Operator*) for the determined objective to be met. Conversely, the plan for scenario 2 requires only one operator (*Expert Webserver Operator*) to meet the determined objective.

Table 3. Estimated execution time and human allocation cost for the generated detailed plans

Scenario	RFC	Activities	Estimated Execution Time (min)	Human Allocation Cost (\$)
1	1	69	118.75	34.21
2	1	38	156.12	7.75
3	1	68	157.08	45.98
4	2	52	75.5	18.73
5	2	39	211.31	10.42
6	2	49	250.75	28.43

Table 3 presents costs associated to the execution of the generated change plans for each of the evaluated scenarios. Observe that the resulting detailed plans exhibit satisfactory results in relation to the determined business purpose. For instance, RFC 1 could be accomplished by two distinct plans: one that has a shorter execution time (118 min), although having a higher implementation cost (\$34.21), or another one which has a reasonable cost (\$7.75), in spite of demanding a longer execution time (156 min). If analyzed individually, these cost differences may be insignificant. However, assuming that many changes may be required on a daily basis within an organization, such an alignment has the potential to reduce operational costs significantly.

Table 3 also enumerates, for the sake of comparison, change plans not aligned with objectives/constraints. Although correct, such plans have higher costs if compared to the plans that follow any alignment. For example, the plan generated for scenario 3 has an estimated cost of \$45.98 with human allocation, an amount 493% higher than the cost of the aligned plan. This is explained by the different possible refinements for a same preliminary plan. If objectives/constraints are not taken into account, the exploration space gets wider, possibly leading to plans that violate business objectives/constraints.

In regard to performance, CHANGEADVISOR has required from a few hundreds of milliseconds to a few seconds to generate each aligned change plan. This is a negligible processing cost, especially if compared to the time humans would require to do the same job.

## 6 Conclusions and Future Work

Change planning is a fundamental element for the operation and management of Information Technology infrastructures and services. Existing automated solutions, however, do not take into consideration the generation of change plans aligned with objectives determined in the business level. Consequently, the planning and execution of changes will present little (or no) relation with the objectives/constraints determined by the change initiator. In order to tackle this problem, in this paper we have proposed CHANGEADVISOR, a solution for generation of change plans, which are consistent with objectives/constraints determined in the business level.

The results obtained, although not exhaustive, confirm the potential benefits of aligning change plans with business purposes. As previously mentioned, several detailed plans have been obtained from a same RFC, each of them satisfying specific execution time and cost requirements. One may also observe that the generated plans were sensibly different in form (comprising activities) as well as in associated costs (time and financial costs). Finally, the system has performed satisfactorily with respect to the time spent in the generation of the plans (kept in the magnitude of seconds). This time is certainly shorter than the time that would be spent by an experienced operator to manually design the plans using a *workflow* editor.

As future work, we intend to: (i) extend the scope of the alignment process so as to support two or more simultaneous objectives/constraints; (ii) investigate a richer set of objectives/constraints; and (iii) explore different heuristics (*e.g.*, for human resource allocation) to be used by the alignment algorithm.

## References

1. Information Technology Infrastructure Library. ITIL V3. Office of Government Commerce (OGC) (2008), <http://www.itil-officialsite.com>
2. Cordeiro, W., Machado, G., Daitx, F., et al.: A Template-based Solution to Support Knowledge Reuse in IT Change Design. In: IFIP/IEEE Network Operations and Management Symposium (NOMS 2008), pp. 355–362 (2008)
3. Keller, A., Hellerstein, J.L., Wolf, J.L., et al.: The CHAMPS System: Change Management with Planning and Scheduling. In: IEEE/IFIP Network Operations and Management Symposium (NOMS 2004), pp. 395–408 (2004)
4. Rodosek, G.D.: A Generic Model for IT Services and Service Management. In: IFIP/IEEE International Integrated Network Management (IM 2003), pp. 171–184 (2003)
5. Brown, A.B., Keller, A.: A Best Practice Approach for Automating IT Management Processes. In: IFIP/IEEE Network Operations and Management Symposium (NOMS 2006), pp. 33–44 (2006)
6. Moura, A., Sauvé, J., Bartolini, C.: Business-driven IT Management - Upping the Ante of IT: Exploring the Linkage between IT and Business to Improve both IT and Business Results. IEEE Communications Magazine 46(10), 148–153 (2008)
7. Trastour, D., Rahmouni, M., Bartolini, C.: Activity-Based Scheduling of IT Changes. In: Bandara, A.K., Burgess, M. (eds.) AIMS 2007. LNCS, vol. 4543, pp. 73–84. Springer, Heidelberg (2007)
8. Machado, G., Daitx, F., Cordeiro, W., et al.: Enabling Rollback Support in IT Change Management Systems. In: IFIP/IEEE Network Operations and Management Symposium (NOMS 2008), pp. 347–354 (2008)
9. Cordeiro, W., Machado, G., Andreis, F., et al.: A Runtime Constraint-Aware Solution for Automated Refinement of IT Change Plans. In: De Turck, F., Kellerer, W., Kormentzas, G. (eds.) DSOM 2008. LNCS, vol. 5273, pp. 69–82. Springer, Heidelberg (2008)
10. Keller, A.: Automating the Change Management Process with Electronic Contracts. In: IEEE International Conference on E-Commerce Technology Workshops (CECW 2005), pp. 99–108 (2005)
11. Distributed Management Task Force (DMTF): Common Information Model, <http://www.dmtf.org/standards/cim>
12. Shepherd, M.: Special Feature Distributed Computing Power: a Key to Productivity. IEEE Computer 10(11), 66–74 (1977)
13. Pillai, K., Sukumaran Nair, V.S.: A Model for Software Development Effort and Cost Estimation. IEEE Transactions on Software Engineering 23(8), 485–497 (1997)
14. Banyahia, H.: Costs and Productivity Estimation in Computer Engineering Economics. Engineering Economist 41(3), 229–241 (1996)
15. Jorgensen, M., Shepperd, M.: A Systematic Review of Software Development Cost Estimation Studies. IEEE Transactions on Software Engineering 33(1), 33–53 (2007)
16. Organization for the Advancement of Structured Information Standards. OASIS: Business Process Execution Language, version 2.0, <http://docs.oasis-open.org/wsbpel/2.0>

## ANEXO C ARTIGO PUBLICADO NO NOMS 2010

Neste anexo, o artigo intitulado “On Strategies for Planning the Assignment of Human Resources to IT Change Activities” é apresentado. Nesse artigo foram apresentadas as seis estratégias para alocação de humanos a atividades de TI e aprimoramentos na solução existente até então.

- **Título:** On Strategies for Planning the Assignment of Human Resources to IT Change Activities

- **Conferência:** 12th IEEE/IFIP Network Operations and Management Symposium (NOMS 2010)

- **URL:** <http://www.ieee-noms.org/2010>

- **Período da realização do evento:** 19-23 de Abril de 2010

- **Local:** Osaka International Convention Center, Osaka, Japão

- **Digital Object Identifier (DOI):** <http://dx.doi.org/10.1109/NOMS.2010.5488470>

# On Strategies for Planning the Assignment of Human Resources to IT Change Activities

Roben Castagna Lunardi, Fabrício Girardi Andreis,  
Weverton Luis da Costa Cordeiro,  
Juliano Araújo Wickboldt, Bruno Lopes Dalmazo,  
Ricardo Luis dos Santos, Luís Armando Bianchin,  
Luciano Paschoal Gasparly,  
Lisandro Zambenedetti Granville

Institute of Informatics  
Federal University of Rio Grande do Sul, Brazil

{rclunardi,fgandreis,wlccordeiro,jwickboldt,bldalmazo,  
rlsantos,labianchin,paschoal,granville}@inf.ufrgs.br

Claudio Bartolini

HP Laboratories Palo Alto, USA

claudio.bartolini@hp.com

**Abstract—***Planning* is a fundamental sub-process of the overarching Information Technology (IT) change management process, proposed by the Information Technology Infrastructure Library to help organizations to deploy and maintain IT services in an effective and efficient way. A major issue behind IT change planning and of special importance for the alignment of changes with business objectives/constraints – the adequate projection of which human resources to assign to change activities – has not been properly addressed in previous investigations. To fill this gap, in this paper we propose and analyze novel strategies for planning the assignment of human resources to change activities. These strategies explore different ways to prioritize humans to activities (*i.e.*, from the most to the less forceful or proficient humans), and to rank/cluster the activities that should be analyzed first. The novel strategies have been experimentally evaluated through CHANGEADVISOR, a prototypical implementation of a decision support system that helps IT administrators in the task of understanding the trade-offs between alternative change designs.

## I. INTRODUCTION

Well structured and managed Information Technology (IT) services are essential for the success of modern organizations. The deployment and maintenance of these services can be regarded as an optimization problem, since they are expected to have high delivery capability, conform to Service Level Agreements (SLAs), and demand as few resources as possible [1]. Intuitively, the more an organization moves towards the limits of this objective, the more “productive” (and, consequently, competitive) the organization tends to become.

In order to help organizations to achieve an elevated productivity standard, the Information Technology Infrastructure Library (ITIL) presents a set of best practices and processes for IT service deployment and management [1]. Aware that the conduction of IT changes represents significant operational costs, often dominating total cost of ownership, ITIL proposes the *change management* [2] process. Among its steps, *planning* plays a very important role in effectively

governing the IT infrastructure, since it is the moment when the change procedure to be executed and the resources (hardware, software, and staff) to be involved are delineated.

In the *change planning* realm, a number of investigations have been conducted in the recent past to *automate* the generation of detailed change plans [3][4][5], and preliminary steps have been given by our research group towards the *alignment* of change plans with business objectives/constraints [6]. A major challenge behind such alignment – the adequate planning of which human resources to assign to change activities – has not been properly addressed. In our first iteration at this problem, we have employed a simplified, greedy assignment planning strategy, where the best available human is pre-assigned for each activity of a refined change plan (traversed taking into account the order of the activities in the plan). As a consequence, the resulting plans, although correct and aligned with business objectives/constraints, are not assuredly *optimized* and do not necessarily reflect the *IT administrator’s assignment preferences*, *i.e.*, the outcome is, in this regard, unpredictable.

The importance of exploring strategies for automated planning the assignment of human resources to IT change activities is twofold. First, finding a “good” match between individuals and change activities to be performed might have a positive influence on change *throughput*, *quality*, and *cost* (all deemed very important to improve the productivity of an organization). Second, it is very hard, if at all possible, to find a “one-size-fits-all” strategy, due to the inherent *varied* nature of the changes to be performed, of the IT infrastructure in place, and of the objectives/constraints to be met.

To tackle the aforementioned problem, in this paper we propose and analyze six strategies for planning the assignment of human resources to change activities. These strategies explore different ways to prioritize humans to activities (*i.e.*, from the most to the less forceful or proficient humans), and to rank/cluster the activities that should be analyzed first. The six strategies have been experimentally evaluated through CHANGEADVISOR, a prototypical implementation of a decision support system that helps IT administrators to understand the trade-offs between alternative change designs.



The remainder of the paper is organized as follows. In Section II we discuss the related work. In Section III we revisit the building blocks upon which the strategies are implemented, while in Section IV we introduce and detail the strategies for planning the assignment of human resources to change activities. In Section V we present the experimental results obtained using the system. Finally, we conclude the paper conveying final considerations and perspectives for future work in Section VI.

## II. RELATED WORK

Several researches have been conducted in the recent past to tackle several facets of the IT change management problem. As examples, important investigations about planning [3][4][5][7], scheduling [8][9], rollback [10], and alignment of changes with business objectives/constraints [6] can be cited. Nevertheless, a critical issue – planning the assignment of human resources to IT change activities – has been barely explored, as discussed in the following paragraphs.

Keller *et al.* [3][11] were responsible for the seminal investigations on IT change management. They have focused on the automation of the process to both generate detailed change plans (for multi-system software) and schedule them. CHAMPS, a system designed by the authors to this end, was tailored to maximize the degree of parallelism in task execution. The scheduling of tasks for human beings, in addition to computers, was out of the scope of that work.

A few years later, Cordeiro *et al.* have proposed a solution, based on *templates*, to formalize, preserve, and reuse knowledge acquired with frequent IT changes [4]. Subsequently, they have proposed an algorithm to generate detailed plans from high level specifications, considering “runtime” constraints imposed by the managed environment (e.g., availability of resources such as disk and memory space) [7]. The planning of which human resources to assign to manual change activities was, as in Keller’s work, not studied.

To the best of our knowledge, the first research effort to address the problem of automatically planning how to assign human resources to IT change activities was performed by Lunardi *et al.* [6]. The results obtained with the proposed greedy assignment strategy allowed us to observe that trivial activities were, due to their location in the plan, often assigned to high skilled stakeholders, while activities of higher complexity ended up assigned to less experienced personnel. Other weakness encountered was the mismatch between the nature of the change activity to be performed and the expertise of the human assigned (e.g., an operation on a database being planned to be executed by a network operator). The same sort of limitation has also been observed in the investigation conducted by Bennour *et al.* [12], who devised a solution, based on performance objectives, to allocate humans to enterprise process activities. Although not directly connected to IT change management, the paper by Bennour is cited here for being one of the few works closely related to the general problem addressed in this paper: strategies for *automatedly* planning the assignment of human resources to activities.

Change management has been a prominent research field within the IT domain, as indicated by the previously mentioned related work. However, the existing investigations did not consider – as a simplification premise – the important role played by humans in the conduction of IT changes (a

fundamental “twist” to the problem). In the next sections we fill this gap, by revisiting the background, as well as introducing and evaluating strategies to plan the assignment of human resources to IT change activities.

## III. THE CHANGEADVISOR SYSTEM

The novel strategies for alignment of change plans with business objectives/constraints introduced in this paper are supported by CHANGEADVISOR [6], a conceptual solution for change management proposed by our research group. In this section we briefly review our previous solution, focusing on its main components, actors involved, and their interactions.

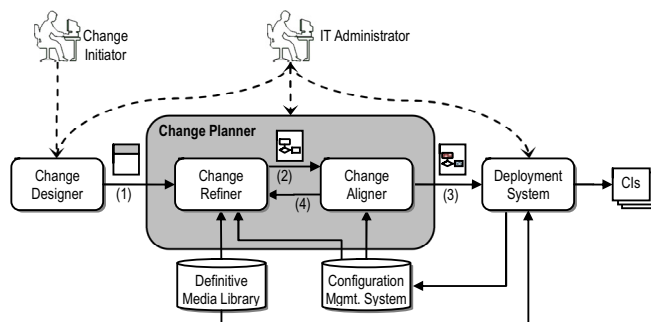


Fig.1. Conceptual Architecture of the CHANGEADVISOR system.

The change process is started with the specification of an RFC (along with business objective to be fulfilled) by a *change initiator*, using the component *Change Designer*. Subsequently, an *IT administrator* sketches a preliminary plan (containing details on how the change should be executed) with the aid of the component *Change Designer*.

The subsequent steps in such a process – as envisaged in our solution – are (i) the automated generation of a detailed plan (based on the preliminary specification) by the component *Change Refiner* (flow 1 in Figure 1), (ii) the alignment of the generated plan to the business requirements specified by the *initiator* in the RFC, by the component *Change Aligner* (flow 2), and (iii) the deployment of the detailed, aligned plan, by the component *Deployment System* (3). During this process, information about *configuration items* (CIs) that compose the IT infrastructure (e.g., hardware, software, and IT personnel) and software packages available for deployment may be consumed by the components that comprise our solution. Following the ITIL’s best practices for IT operations & management, these are envisaged to be available, respectively, in the *Configuration Management System* (CMS) and the *Definitive Media Library* (DML). Further, as the change is delivered, the *Deployment System* is responsible for updating the CMS with information about changes carried out with every affected CI.

It is important to highlight that, during the alignment of the generated plan to the business requirements specified by the *initiator* in the RFC, the component *Change Aligner* retrieves from the CMS information about (i) costs and skills of human resources available for deploying such a change (e.g., an operator who is an expert in e-mail service and whose work hour costs 40 monetary units), and capabilities of the hardware assets affected by the plan (e.g., computational power in the case of workstations and servers).

We also highlight that the refinement and alignment process is reiterated (flow 4), aiming at generating distinct, detailed

plans, aligned to the same business purposes specified in the RFC. In a later moment, the obtained plans may be evaluated in terms of trade-offs between the various options. Among the criteria that may guide such an analysis, we cite: expected time and/or financial costs to perform the changes and human resources required. Based on this analysis, the IT admin will be able to select an appropriate plan to be deployed.

Having presented an overview of our solution, in the following subsections we describe (i) the models used to express information contained in IT-related repositories and change documents, and (ii) a generalized algorithm for the alignment of change plans with purposes determined in the business level.

### A. IT-related Information Models

We adopt in the scope of our research a subset of the *Common Information Model* (CIM) [13], in order to model information contained in both the *Configuration Management System* and *Definitive Software Library* repositories. Fig. 2 presents a partial view of the model.

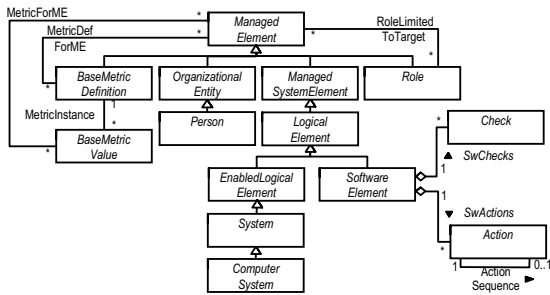


Fig 2. Partial view of the IT Information Model.

The root class of the model is *ManagedElement*. Through specialization, it is possible to represent any CI that composes the IT infrastructure, along with their relationship with other elements and with external entities (e.g., third-party services). Classes such as *Check* and *Action* represent relevant information required for managing the lifecycle of a software element (represented by instances of class *SoftwareElement*), and consumed during the generation of an actionable workflow from a preliminary specification.

Classes *BaseMetricDefinition* and *BaseMetricValue* are employed to express and quantify the *costs* and *capabilities* metrics related to activities and elements involved in the change process. On one extreme, costs describe the effort (or monetary resources) demanded in the execution of activities related to IT service operations & management. On the other extreme, capabilities allow expressing an expectation on the performance of computers to execute them.

In the scope of our research, we have initially envisaged three distinct *costs* metrics – *Required Time* (RT), *Computing Power* (CP), and *Cost per Hour* (C/H). *Required Time* quantifies human effort (in time units) required for executing a manual activity. *Computing Power* denotes computational effort required for executing an automated activity (an activity that requires minimal or no human intervention). Value instances for *Required Time* and *Computing Power* are used to express the costs of activities associated with software packages available from the DML. Finally, *Cost per Hour* quantifies the labour cost per hour of a human operator to the organization. In regard to the *capabilities* metric, we envisage

*Computing Power per Hour* (CP/H), which represents the computational capability of a *ComputerSystem* to execute activities. Please refer to our previous work [6] for additional information about the proposed metrics.

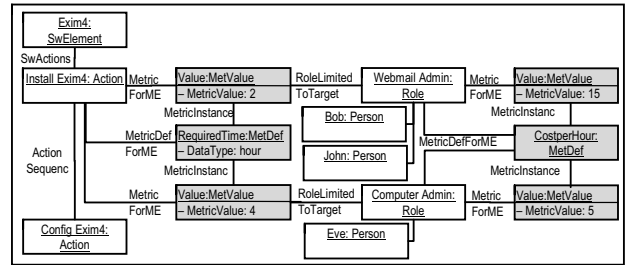


Fig. 3. Partial view of the actions associated to the *SoftwareElement Exim4*, along with their respective costs.

Fig. 3 illustrates, as a concrete example, the actions associated to the installation of a mail transfer agent (*Exim4*), available in the *Definitive Media Library*. One may note the costs – expressed in terms of *Required Time* – associated to the action *Install Exim4*. According to Fig. 3, *Install Exim4* may be executed by either a *Webmail Admin* or a *Computer Admin* operator in the organization. A *Webmail Admin*, whose *Cost per Hour* is \$15, may take approximately 2 hours to execute the installation (thus resulting in a total cost of \$30). Conversely, a *Computer Admin*, whose *Cost per Hour* is \$5, may take up to 4 hours to perform the same task (resulting in a total cost of \$20). One may also note from Fig. 3 that the organization has two operators holding the *Webmail Admin* role, and one the *Computer Admin* role.

In regard to the RFC and *Change Plan* documents, these are materialized by a model proposed in a previous work [4]. In order to support the alignment of change plans with business purposes, we extend the model to include two distinct attributes in class *RFC*: *Objective* and *ThresholdValue*. The former indicates the business objective/constraint that should be met by the change, whereas the latter indicates the upper bound tolerated regarding such an objective. Two business purpose categories are defined in this first iteration to solve the problem: *Execution Time* and *Execution Cost* to perform the change. These objectives were indicated as the most relevant ones by a group of network/system managers (working in Brazilian IT companies) interviewed. The specification of one or more objective/constraint needs to be accompanied by a respective *ThresholdValue*, representing the maximum acceptable time or financial cost.

### B. Generalized Algorithm for Alignment of Change Plans to Business Objectives/Constraints

As a first effort to align change plans with business objectives/constraints, we have proposed an initial strategy and materialized it in the form of an algorithm [6]. The strategy consisted in performing (non-nominal) human allocations to activities contained in a pre-refined change plan, and computing the overall deployment cost of the resulting aligned plan. In this paper, we go a step further by generalizing the previous algorithm, in order to support diverse strategies for alignment of changes to business requirements.

The generalized algorithm, shown in Figure 4, is denoted by *change\_alignment* (*C*, *Strategy*), where *C* is the detailed change plan; and *Strategy* is the strategy to be used during the

alignment of  $C$  to the specified objective/constraint.

As a first step of the alignment process, an ‘empty’ change plan  $C'$  is created to store the aligned plan (line 2 in Fig. 4). Next, the activities contained in detailed change plan are copied, ordered according to the criteria defined in *Strategy* (further discussed in Section IV), and stored in  $D$  (line 3). After that, function *recursive\_alignment* is invoked (4).

```

1 recursive_alignment (C, Strategy)
2 C' ← empty change plan
3 D ← set of activities from C ordered using Strategy
4 recursive_alignment(C, C', D, Strategy)
5
6 recursive_alignment (C, C', D, Strategy)
7 if D = ∅ and cost of deploying C' complies with objective then
8   return C' as solution
9 extract activity ai from D
10 if ai is a manual activity then
11   H ← set of potential operators for ai, ordered using Strategy
12   for each h in H do
13     if h is allocable to activity ai given C and C' then
14       allocate human h to activity ai
15       add activity ai to change plan C'
16       if cost of deploying C' complies with objective then
17         recursive_alignment (C, C', D, Strategy)
18 else
19   add activity ai to change plan C'
20   if cost of deploying C' comply with objective then
21     recursive_alignment (C, C', D, Strategy)
22 return failure // backtrack and try another allocation

```

Fig. 4. Algorithm for alignment of changes to business objectives/constraints.

The alignment process – as encoded in function *recursive\_alignment* – starts by verifying if  $D$  (the set of activities that have not yet been processed) is empty and plan  $C'$  has a cost lower than defined in the business objectives determined in the *RFC* (line 7). If so,  $C'$  is returned as a solution (8). Otherwise, the following steps are executed. First, an activity  $a_i$  is extracted from set  $D$  for processing (9). Second, the algorithm verifies if  $a_i$  requires a human operator to be executed (10). In case it does, a set of human operators that represent potential options to execute  $a_i$  is created, ordered following the specified *Strategy*, and stored in  $H$  (9).

After the potential options of human operator to execute  $a_i$  are identified, the algorithm tries to allocate an operator  $h$  to  $a_i$ , so that no constraints are violated (for example, a same operator may not be allocated to two or more distinct, parallel activities) (lines 11-15). Note that, during the alignment process, the allocation of humans is conducted in a “non-nominal” fashion, *i.e.*, there is the specification that a human playing role  $x$  should execute a given activity (and not *who*, in fact, must execute it). The effective (or nominal) allocation of human resources to the activities is postponed to the change scheduling stage, which is out of the scope of this paper and has been investigated by several related work [3][8][9].

Once a human is allocated to  $a_i$ , the algorithm verifies the cost of executing plan  $C'$  on the managed infrastructure, considering such allocation (line 16). If the cost does not violate the *threshold value* specified in the *RFC*, the alignment process continues recursively (line 17), in order to process the remaining activities in  $D$ . The semantics of *cost* (and *threshold*) depends on the *RFC* objective/constraint. For example, if *Execution Time* is chosen, then *cost* and *threshold* are processed considering time units.

Alternatively, the activity  $a_i$  extracted from set  $D$  may not require human intervention (line 18). In this case, it is directly added to the resulting change plan (line 19). The processing continues recursively (line 21), in case the execution cost of  $C'$  is aligned with the defined objectives (line 20).

If none of the humans available can be allocated to  $a_i$  or the cost for executing  $C'$  (added to  $a_i$ ) violates the *threshold value*, the alignment process fails (line 22). Consequently, the allocation of human resources done up to this moment is recursively rolled back, and new alternatives for the alignment are explored. In the worst case, *recursive\_alignment* is not able to produce a plan aligned with the business purpose specified in the *RFC*, in which case a detailed feedback is returned to the IT admin.

#### IV. NOVEL STRATEGIES FOR ALIGNMENT OF CHANGE MANAGEMENT PROCESSES TO BUSINESS OBJECTIVES

The conceptual solution, information models, and generalized algorithm presented in the previous section form the basis upon which the strategies for alignment of change plans to business objectives are built. In this section, we cover the proposed strategies in detail.

Given an initial state of the IT infrastructure (whose information is contained in the CMS), a set of software packages (from the DML), a number of IT operators (along with their respective roles), an *RFC* document (along with its preliminary plan), and a business objective to be satisfied by this *RFC*, finding a refined plan that optimizes the adherence of the *RFC* to the specified objective is known to be NP-hard [6]. In order to reduce the complexity of this problem, the novel strategies proposed in this paper compile two distinct heuristics, being each heuristic of the following types: (i) ordering of activities, and (ii) allocation of human operators. Combined with the algorithm presented in the previous section, the two heuristics adopted in each strategy influence, respectively, the order in which activities and human operators are processed during the alignment of changes.

Table I summarizes the strategies proposed in this paper, classified following the previously mentioned types of heuristics. These strategies implement two methods invoked during the execution of the algorithm from Fig. 4: the ordering of the activities from the change plan  $C$  (line 3), and the selection and ordering of potential human operators to execute a given activity  $a_i$  (line 11). In the following subsections, we describe each of the proposed strategies in detail, and also analyze their potentialities and limitations.

TABLE I. STRATEGIES FOR ALIGNMENT OF CHANGES, CLASSIFIED ACCORDING TO THE HEURISTICS EMPLOYED

		Ordering of Activities		
		No Ordering	Per Activity	Per Cluster of Activities
Allocation of Human Operators	Allocation from Best Human Operator to Worst	<b>Strategy A</b> – Traverse the workflow, and attempt to assign the best operator available to each activity	<b>Strategy B</b> – Order the activities per cost (in descending order); traverse the resulting list, and attempt to assign the best available operator to each activity	<b>Strategy C</b> – Cluster the activities according to a topological criteria; rank the resulting clusters per cost; traverse the resulting list, and attempt to assign the best available operator to each activity
	Allocation Based on IT Administrator's Preferences	<b>Strategy D</b> – Traverse the workflow and attempt to assign human operators to each activity, following the preferences indicated by the IT administrator	<b>Strategy E</b> – Rank activities per cost; traverse the resulting list, and attempt to assign humans to each activity, following the preferences indicated by the IT administrator	<b>Strategy F</b> – Cluster the activities; rank the resulting clusters per cost; traverse the resulting list, and attempt to assign humans to each activity, following the preferences indicated by the IT administrator

##### A. Allocation from Best Human to Worst

This strategy – introduced in a previous work [6] – represents a naïve, greedy solution for the alignment of changes to business requirements. Basically, it causes the

algorithm to process the activities contained in the change plan in their original order, and attempt to assign the best human operator available to execute them.

To illustrate, consider the function  $effort(a, r)$ , which returns the financial cost of allocating a human of role  $r$  to execute  $a$ . Consider also the change plan  $C$  illustrated in Fig. 5, and the human operators illustrated in Table II. Assuming that (i) the business objective of  $C$  is minimizing its financial cost of deployment, (ii)  $c_1 < c_2 < c_3$ , (iii) any activity may be executed by an operator of any role, (iv) all activities consume 1 hour to be executed, regardless the role of the operator allocated to it, and (v) for any allocation of an operator  $h_j$  (of role  $r_i$ ) to execute activity  $a_i$ ,  $effort(a_i, r_j) = 1 \text{ hour} \times c_j$ , the resulting allocation – according to this strategy – is  $(a_1, h_1)$ ,  $(a_2, h_2)$ ,  $(a_3, h_1)$ ,  $(a_4, h_2)$ ,  $(a_5, h_3)$ ,  $(a_6, h_3)$ ,  $(a_7, h_1)$ , and  $(a_8, h_4)$ .

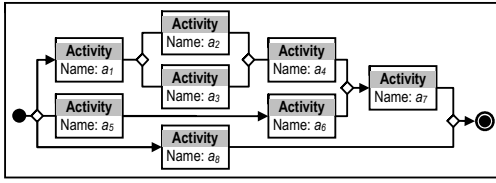


Fig. 5. Refined change plan  $C$ .

Note from the example above that the majority of allocations was performed using human operators belonging to role  $r_1$ . This is trivially explained by the fact that the cost per hour of operators of role  $r_1$  is the cheapest in the organization. Note, however, that the operator allocated to  $a_5$  is not of role  $r_1$ . This is because all operators from this role were already assigned to activities parallel to  $a_5$ . In this case, the operator from the second cheapest role available ( $r_2$ ) was chosen,  $h_3$ .

TABLE II. ROLES OF HUMAN OPERATORS AVAILABLE IN THE ORGANIZATION

Role	Human Operator	Cost per Hour (C/H)
$r_1$	$h_1$	$c_1$
$r_1$	$h_2$	$c_1$
$r_2$	$h_3$	$c_2$
$r_3$	$h_4$	$c_3$
$r_3$	$h_5$	$c_3$

The main positive aspect of this strategy is that it allocates the best human operator available to execute an activity. This characteristic has potential to increase the efficiency of the execution of each activity, and is highly effective in scenarios where a large number of “best” operators are available.

There are situations, however, in which such a greedy approach may not be convenient. For example, suppose that  $r_1$  (the role of cheapest human operators) has only one operator available, which is allocated to activity  $a_8$  (of course, assuming that  $a_8$  is the first activity in the plan). In this scenario, activities  $a_1$ ,  $a_2$ ,  $a_4$ , and  $a_7$  (which are not parallel among themselves but are parallel to  $a_8$ ) will be assigned an operator of higher cost, thus increasing the overall cost of deploying the change plan  $C$ . Another negative aspect is the possibility of “over-assignments” (when more expensive – and presumably more skilled – operators are allocated to simpler activities) and “under-assignments” (when cheaper – and presumably less skilled – operators are allocated to activities of higher complexity). In the following subsections we introduce and describe strategies that mitigate these shortcomings.

### B. Best Human to Worst with Ordering per Activity

The greedy approach of the previous strategy, though prominent, may be hampered in some scenarios due to the

processing of activities in their original order of appearance. To tackle this shortcoming, the previous strategy is extended to process the activities considering their cost of deployment. In this strategy, activities are ranked in descending order, according to their *mean effort* (detailed next). Subsequently – and similarly as in the previous strategy – the alignment algorithm processes the activities in the order they appear in the rank, attempting to assign the best human operator available to each activity.

Prior to understanding the *mean effort* of an activity, first recall that an activity may have one or more options of roles for their deployment. From the example of Fig. 3, activity *Install Exim4* may be executed by either a *Webmail Admin* or a *Computer Admin*. The *mean effort*, formalized in Eq. 1, is thus the sum of the financial costs of deploying activity  $a$  (given by function  $effort$ ) – for each option of role associated to it – divided by the number of options. Considering the example of Fig. 3, the *mean effort* of activity *Install Exim4* is \$25. In Eq. 1,  $n$  is the number of options, whereas  $r_{ka}$  represents the  $k$ -th role of an operator that may be assigned to execute  $a$ .

$$mean\_effort(a) = \frac{1}{n} \times \sum_{k=1}^n effort(a, r_{ka}) \quad (1)$$

To illustrate this strategy, consider again the refined change plan from Fig. 5 and the human operators from Table II. Now assume that (i) the business objective of  $C$  is to minimize its financial cost of deployment, (ii)  $c_1 < c_2 < c_3$ , (iii) any activity may be executed by an operator of any role, and (iv)  $m_4 < m_1 < m_7 < m_6 < m_2 < m_3 < m_5 < m_8$  (where  $m_k = mean\_effort(a_k)$ ). Under these assumptions, the resulting allocation is  $(a_8, h_1)$ ,  $(a_5, h_2)$ ,  $(a_3, h_3)$ ,  $(a_2, h_4)$ ,  $(a_6, h_2)$ ,  $(a_7, h_2)$ ,  $(a_1, h_3)$ , and  $(a_4, h_3)$ . Note in this example that activity  $a_8$  achieved highest priority in the allocation process (due to its highest *mean effort*), and thus was assigned the “best” human operator available in the organization. In contrast,  $a_4$  was ranked with lowest priority, and thus was assigned a human operator from the second “best” role available,  $r_2$  (since all operators from role  $r_1$  were already assigned to activities parallel to  $a_4$ ).

The main advantage of this strategy is that, in most of scenarios, the human allocated tends to match the capability (and even the *expertise*) demanded by the activity he/she was assigned to (thus addressing the shortcoming of *over-* and *under-assignments* of the previous strategy). Further, it is more suitable for change plans having lower degrees of parallelism, since the parallel tasks of those plans have higher probabilities of being assigned the best human available. Nevertheless, this strategy may lead to under-assignments in scenarios where all activities have a low *mean effort*.

### C. Best Human to Worst with Ordering per Cluster

In this strategy, a different approach is adopted in regard to how activities in the change plan are processed. Instead of processing activities separately, these are treated considering clusters of activities; the activities are manipulated by the alignment algorithm observing their position in the clusters.

To illustrate how the clustering of activities is performed, consider the plan of Fig. 5. As a first step, clusters are created (i) for each activity in the plan (unitary clusters) and (ii) for each flow of activities. In the example of Fig. 5, the set of clusters created in this step is  $S = \{(a_1), (a_2), (a_3), (a_4), (a_5), (a_6), (a_7), (a_8), (a_5, a_6), (a_1, a_2, a_3, a_4)\}$ . Note that activities appear at most in two different clusters (one of which is a

unitary cluster). Note also that the clusters  $(a_1, a_2, a_3, a_4, a_5, a_6, a_7)$  and  $(a_1, a_2, a_3, a_4, a_5, a_6, a_7)$  are not created. This is because created clusters must capture all parallel flows of activities contained in the plan. Observe that cluster  $(a_1, a_2, a_3, a_4)$  does not violate this constraints, since the parallel flows in it are captured by the unitary clusters  $(a_2)$  and  $(a_3)$ .

Subsequently, the newly created clusters are sorted – in descending order – considering the sum of the *mean effort* of the activities that compose them. To exemplify, consider the set of clusters  $S$ . Assuming that  $m_1 < m_2 < m_3 < m_4 < m_5 < m_6 < m_7 < m_8 < (m_1 + m_2 + m_3 + m_4) < (m_5 + m_6)$  – where  $m_k = \text{mean\_effort}(a_k)$  – the resulting sorting is  $S' = \{(a_5, a_6), (a_1, a_2, a_3, a_4), (a_8), (a_7), (a_6), (a_5), (a_4), (a_3), (a_2), (a_1)\}$ .

As a final step, each cluster is processed in order, and the activities in it are added to  $R$  (if not yet added). Subsequently,  $R$  is returned as the ordering of activities to be considered by the alignment algorithm. In the previous example, the resulting order of activities is  $R = \{a_5, a_6, a_1, a_2, a_3, a_4, a_8, a_7\}$ .

The main positive aspect of this strategy is that flows of activities within the change plan are now considered, instead of a single activity. Note that  $a_5, a_6$  and  $a_1, a_2, a_3, a_4$  form two distinct flows within the plan of Fig. 5, and thus were ranked together (according to the sum of their *mean efforts*). Such a ranking has the potential to minimize the execution cost of a flow, even decreasing the overall execution cost of the change plan. Nevertheless, this strategy may not be suitable for changes having lower degrees of parallelism (in contrast to *Strategy B*, which is more effective in such scenarios).

#### D. Allocation based on IT Administrator's Preference

The strategies introduced so far, although prominent for decreasing costs associated to changes, have not taken into account the *expertise* of the IT administrator designing the plan in regard to potential assignments of humans to activities. Based on his/her own experience, for example, the IT administrator may wish to allocate a *Web Admin* to install and configure an *Apache* server, instead of a *Network Operator*. In another situation, he/she may wish to allocate a *System Manager* to perform a small configuration in a database server, instead of a *Database Analyst*. In order to allow such an expertise to be taken advantage of, we extend *Strategy A* to prioritize, for each activity, the allocation of a human belonging to a role indicated by the administrator.

In order to illustrate how this strategy operates, consider the change plan from Fig. 5, and the human operators illustrated in Table II. Under the same assumptions adopted for the example of *Strategy A*, the resulting allocation is  $(a_1, h_1), (a_2, h_2), (a_3, h_1), (a_4, h_2), (a_5, h_3), (a_6, h_3), (a_7, h_1)$ , and  $(a_8, h_4)$ . Now consider that the IT administrator has prioritized the allocation of humans from role  $r_3$  to activities  $a_2$  and  $a_3$ . Under this new scenario, the resulting allocation is  $(a_1, h_1), (a_2, h_4), (a_3, h_5), (a_4, h_1), (a_5, h_2), (a_6, h_2), (a_7, h_1)$ , and  $(a_8, h_3)$ .

The main positive aspect of this strategy is that it offers the IT administrator the opportunity to elaborate changes that better adhere to the strategic planning for IT service delivery, using his/her own knowledge about past changes and available human resources. It is important to mention, however, that the allocation of humans based on the priorities established by the IT administrator may also lead to a number of plans that do not conform to specified business requirements. Another negative aspect is the possibility of *priority inversion*, i.e., when the IT administrator's preference is not fulfilled (due to exhaustion of

qualified human resources) for activities that account for the highest impact – in terms of financial cost or time consumed. This shortcoming, as discussed in the case of *Strategy A*, is mainly due to the processing of activities considering their original order of appearance in the plan.

#### E. IT Admin's Preference with Ordering per Activity

This strategy incorporates the heuristics of *Strategies B* and *D*, and thus allows the IT administrator designing the change plan to use his/her *expertise* to influence the assignment of humans to activities that account for the highest impact. This is an aspect of paramount importance, especially to changes having a number of activities of higher impact, in scenarios where high-quality human resources are scarce.

In spite of the previously mentioned positive aspect, this strategy may become ineffective for changes having a large number of activities with relatively low *mean effort*: in this scenario, a large fraction of the IT administrator's preferences for activities of lower impact may not be fulfilled (due to the previously mentioned exhaustion of human resources).

#### F. IT Admin's Preference with Ordering per Cluster

This strategy extends *Strategy C* by incorporating the possibility of the IT administrator to influence the assignment process. Note that the ordering of activities through clustering is not altered, since the assignment of humans to activities is performed posterior to this phase.

In contrast to *Strategy E*, activities having lower mean effort have higher probability of being assigned to a human indicated by the IT administrator. Note, however, that preferences for activities of higher impact may not be fulfilled for plans having a lower degree of parallelism.

Another negative aspect of this strategy is the higher probability of *priority inversion*. For example, consider that activity  $a_5$  accounts for the highest impact of the change plan of Fig. 5, whereas  $a_6$  accounts for the lowest. Assuming that the flow composed of these activities has the highest mean effort in the plan, activity  $a_6$  will receive higher priority for the assignment of a human operator over other activities, even though having the lowest mean effort.

## V. EXPERIMENTAL EVALUATION

In order to evaluate the technical feasibility of using different strategies for the alignment of change plans with business objectives/constraints, we have carried out several experiments using a prototypical solution of a decision support system called CHANGEADVISOR [6]. A same RFC (along with a business objective/constraint to be satisfied) has been submitted, considering the same IT infrastructure as a basis. As a result of the refinement, several alternatives of detailed plans were generated for alignment. In this paper we focus our attention on four of these plans. For the purpose of this evaluation, we have analyzed the quality of the aligned plans, considering each of the proposed strategies, in addition to performance indicators.

The IT infrastructure employed in this evaluation is composed of 2 servers, *labcom* and *noc*. The former has *Debian GNU/Linux* installed, whereas the latter has no operating system installed. Information about human resources available for deploying the change (along with their costs) is



presented in Table III.

The alternatives of detailed plans used to evaluate the proposed strategies materialize an RFC for the migration of a Webmail server from *labcom* to *noc*. The business objective specified in the RFC is to minimize the execution time to 150 minutes, and the monetary cost to \$30. Information regarding the detailed plans that are relevant for the scope of this evaluation are described Table IV.

TABLE III. HUMAN RESOURCES AVAILABLE FOR CHANGE DEPLOYMENT

Role	Number of Available Operators	Allocation Cost (Cost per Hour)
Expert Webserver Operator	1	40
Junior Webserver Operator	1	25
Computer Operator	3	12

TABLE IV. CHARACTERISTICS OF THE DETAILED PLANS GENERATED

Scenario	Activities	Software Involved
1	69	OS Debian Linux, Apache2, PHP5, Exim4, Squirrelmail
2	60	OS Debian Linux, Apache2, PHP5, Postfix, Squirrelmail
3	69	OS Debian Linux, Apache2, PHP5, Horde
4	69	OS Debian Linux, Apache2, PHP5, Sendmail, Squirrelmail

Table V presents the costs associated to the execution of each alternative plan presented in Table IV, considering each of the strategies introduced in Section IV. We have highlighted, using a gray scale, how adequate the assignment of humans to activities was (for each combination of scenario/strategy). The darker the cell is, the more adequate is the matching between activities and assigned humans. This is evaluated in terms of skill/effort required by activities, and skill/experience owned by human operators. We have also underlined the best costs achieved for each change plan, considering the business requirement specified in the RFC.

TABLE V. ESTIMATED EXECUTION TIME AND HUMAN ALLOCATION COST FOR THE GENERATED DETAILED PLANS

Strategy	Scenario 1		Scenario 2		Scenario 3		Scenario 4	
	Minutes	\$	Minutes	\$	Minutes	\$	Minutes	\$
A	107,74	22,19	86,49	15,25	120,69	21,20	83,49	<u>15,50</u>
B	107,74	22,19	83,49	15,45	<u>116,57</u>	21,00	83,49	<u>15,50</u>
C	<u>100,37</u>	<u>22,10</u>	<u>78,24</u>	<u>15,00</u>	120,69	21,20	<u>70,99</u>	15,60
D	128	23,30	109	17,60	126,58	21,50	100,5	17,55
E	118,34	21,00	92,10	15,50	131,16	18,00	93,34	13,55
F	117,9	21,20	91,72	15,60	133,07	19,40	<u>92,97</u>	17,50

According to Table V, the best cost, in terms of execution time, is achieved for the detailed plan of scenario 4 with *Strategy C* (70.99 minutes). In contrast, the best cost with assignment of human operators to activities is achieved for the plan of scenario 2, also with *Strategy C* (\$15.00). Note, however, that the alternatives of aligned plans that better take advantage of available human resources do not necessarily provide the best deployment costs. For example, the detailed plan of scenario 2, aligned using *Strategy E*, is expected to consume approximately 22 minutes more than the plan of scenario 4 aligned with *Strategy C* (which has the best execution time observed). Observe that such a comparison is valid, since we are analyzing diverse alternatives of detailed plans that, although aligned using different strategies, materialize the changes requested in a same RFC document.

In general, the strategy that has delivered aligned plans with better costs, for the majority of scenarios, is *Strategy C*. This is because reducing the cost of a flow has higher potential to decrease the overall cost of the change plan, than reducing the

cost of activities separately. It is important to mention that the best costs were not observed for *Strategies D, E, and F* because the IT administrator's preferences for assignment of humans to activities may not necessarily lead to lower costs, though yielding to change plans of higher quality.

In regard to the quality in the assignment of humans to activities, *Strategy E* has delivered aligned plans with better matching in the allocation of human resources to activities, for the majority of scenarios. Under this evaluation, *Strategy D* is significantly harmed by the *priority inversion* problem. Since activities are processed in their original order in the plan, those that account for the highest impact have higher probability of not being assigned with a human operator of a role indicated by the IT administrator. Conversely, as *Strategy F* considers flows of activities, an activity of lower impact, belonging to a flow of higher cost, has priority in the assignment than another activity of higher impact belonging to a flow of lower cost.

Figs. 7 and 8 present a partial view of alternative detailed, aligned change plans generated for the RFC considered in our analysis. The plan of Fig. 7 represents the alternative that provides the best costs in terms of execution time and (roughly in regard to) expenditure with allocation of human resources. Conversely, the plan of Fig. 8 illustrates the alternative that promotes the most appropriate usage of available human resources, in terms of matching between operators' skills and activities' complexity.

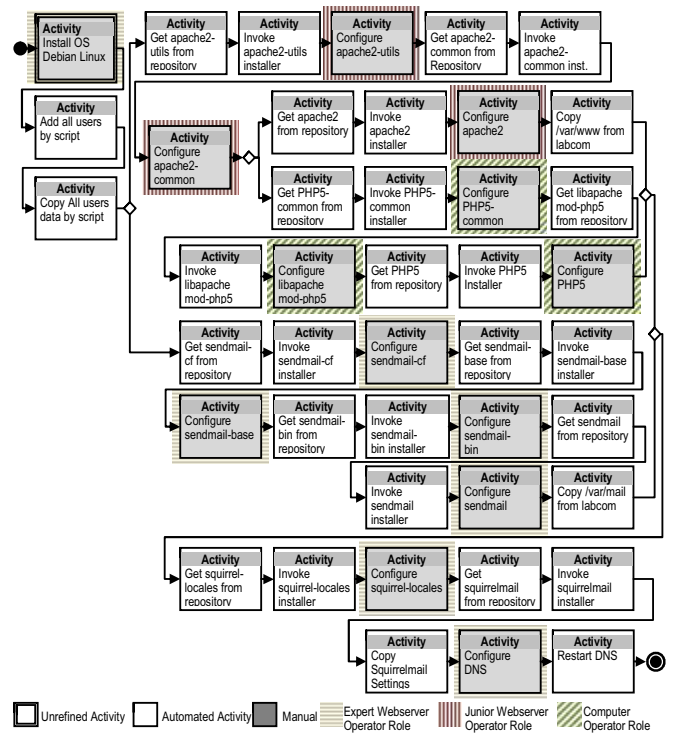


Fig. 7. Change plan of scenario 4, aligned considering *Strategy C*.

One may note from Fig. 7 that *Strategy C* has led to assignments that explore adequately the human resources available. There are, however, examples of mismatches in the assignment of humans to activities: a human of role *Expert Webserver Operator* was allocated to activities of lower complexity, such as the configuration of *Sendmail* libraries. Further, a human of same role is assigned to install and configure *Debian Linux*, a task for which assigning a human of

Computer Operator role would be more appropriate. In contrast, in the plan of Fig. 8, all activities are assigned to a human operator having the appropriate role.

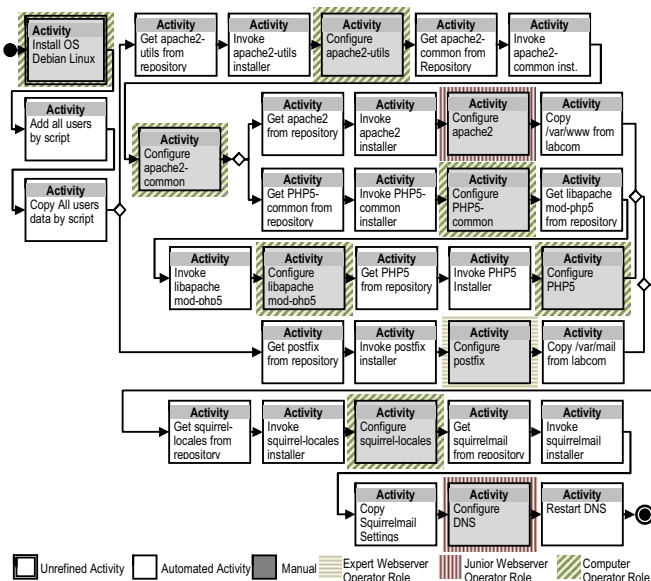


Fig. 8. Change plan of scenario 2, aligned considering *Strategy E*.

One may also note that, despite having more activities, the plan of Fig. 7 has a lower estimated execution time than the one of Fig. 8. This is because 10% of manual activities are executed by an expert operator – which is supposed to be more efficient – in the plan of Fig. 8 (1 out of 10 activities). This ratio increases to approximately 53% in the plan of Fig. 7 (7 out of 13 activities).

It is important to emphasize that the differences of cost may be insignificant, if analyzed individually; however, assuming that many changes may be required on a daily basis in the organization, such an alignment has the potential to reduce operational costs significantly. More importantly, note that even if the estimated costs are not precisely accurate (due to factors external to our solution, such as imprecise information contained in IT-related repositories), still they are useful to compare the trade-offs between generated plans. Being aware of these estimated costs, the operator has sufficient information to evaluate how relatively cheap or expensive the alternative designs are, in comparison to one another, and thus select the most appropriate plan for deployment.

In regard to performance, CHANGEADVISOR has required few seconds to generate each aligned change plan. This is a negligible processing cost, especially if compared to the time a human IT administrator would require to do the same job.

## VI. FINAL CONSIDERATIONS & FUTURE WORK

The adequate allocation of human operators to change activities represents an aspect of crucial importance for the proper conduction and delivery of changes over managed IT infrastructures. In spite of this, solutions for the automated planning of changes have not properly addressed this aspect; consequently, the detailed change plans generated by such solutions are not assuredly optimized and do not necessarily reflect the IT administrator's assignment preferences. In order to bridge this gap, in this paper we have proposed six novel strategies for planning the allocation of human resources to

change activities. These strategies are supported by CHANGEADVISOR [6], a decision support system that helps IT administrators to understand the trade-offs between alternative change designs.

The results obtained, although not exhaustive, show the trade-offs among the usage of different alignment strategies. The use of different strategies has led to the generation of detailed, aligned plans that either optimize the overall deployment cost, or promote a better usage of human resources available. In either case, the generated plans were able to satisfy the business requirement specified by the change initiator in the RFC document. Finally, the system has performed satisfactorily with respect to the time spent in the generation of the plans (kept in the magnitude of seconds). This time is certainly shorter than the time that would be spent by an experienced operator to manually design the plans using a *workflow* editor.

As future work, we intend to: (i) investigate a richer set of objectives/constraints; (ii) explore further strategies to be used by the alignment algorithm; and (iii) extend the scope of human allocation process so as to support load balancing for the operators involved in the change plan.

## REFERENCES

- [1] Information Technology Infrastructure Library. Office of Government Commerce (OGC), 2008. Url: <<http://www.itil-officialsite.com>>.
- [2] IT Infrastructure Library: ITIL Service Transition, v. 3. London: The Stationery Office, 2007, 270 p.
- [3] A. Keller, J. L. Hellerstein, J. L. Wolf *et al.*, "The CHAMPS System: Change Management with Planning and Scheduling," Proceedings of IEEE/IFIP Network Operations and Management Symposium (NOMS 2004), p. 395-408, 2004.
- [4] W. L. C. Cordeiro, G. S. Machado, F. Daitx, *et al.*, "A Template-based Solution to Support Knowledge Reuse in IT Change Design," Proceedings of IFIP/IEEE Network Operations and Management Symposium (NOMS 2008), p. 355-362, 2008.
- [5] W. L. C. Cordeiro, G.S. Machado, F.G. Andreis *et al.*, "ChangeLedge: Change Design and Planning in Networked Systems based on Reuse of Knowledge and Automation," Computer Networks, 2009. doi: 10.1016/j.comnet.2009.07.001.
- [6] R.C. Lunardi, W. L. C. Cordeiro, F.G. Andreis *et al.*, "ChangeAdvisor: A Solution to Support Alignment of IT Change Design with Business Objectives/Constraints," Proceedings of 20th International Workshop on Distributed Systems: Operations and Management (DSOM 2009), 27-28 October 2009, Venice, Italy. (to appear)
- [7] W. L. C. Cordeiro, G. S. Machado, F. G. Andreis *et al.*, "A Runtime Constraint-Aware Solution for Automated Refinement of IT Change Plans," In: IFIP/IEEE International Workshop on Distributed Systems: Operations and Management (DSOM 2008), p. 69-82, 2008.
- [8] D. Trastour, M. Rahmouni, C. Bartolini, "Activity-Based Scheduling of IT Changes," Proceedings of International Conference on Autonomous Infrastructure, Management and Security (AIMS 2007), p. 73-84, 2007.
- [9] R. Rebouças, J. Sauv , A. Moura, C. Bartolini and D. Tratour, "A decision support tool to optimize scheduling of IT changes," Proceedings of IFIP/IEEE International Symposium on Integrated Network Management (IM 2007), p. 343-352, 2007.
- [10] G. S. Machado, F. Daitx, W. L. C. Cordeiro *et al.*, "Enabling Rollback Support in IT Change Management Systems," Proceedings of IFIP/IEEE Network Operations and Management Symposium (NOMS 2008), p. 347-354, 2008.
- [11] A. Keller, "Automating the Change Management Process with Electronic Contracts," Proceedings of IEEE International Conference on E-Commerce Technology Workshops (CECW 2005), p. 99-108, 2005.
- [12] M. Bennour, D. Crestani, O. Crespo and F. Prunet, "Computer aided decision for human task allocation with mono and multi performance evaluation," International Journal of Production Research. v43 i21. 4559-4588.
- [13] Distributed Management Task Force (DMTF): Common Information Model. Url: <<http://www.dmtf.org/standards/cim>>.