UNIVERSIDADE FEDERAL DO ESTADO DO RIO DE JANEIRO
ESCOLA DE INFORMÁTICA APLICADA
CURSO DE BACHARELADO EM SISTEMAS DE INFORMAÇÃO

IMPLEMENTAÇÃO DO MÉTODO MAC KNIGHT PARA ELICITAÇÃO DE REQUISITOS NA METODOLOGIA ARIS

Victor Manaia Gonçalves Chaves

Orientadoras:
Renata Mendes de Araujo, D. Sc.
Fernanda Araujo Baião, D. Sc.

Março de 2008

IMPLEMENTAÇÃO DO MÉTODO MAC KNIGHT PARA ELICITAÇÃO DE REQUISITOS DE SOFTWARE NA METODOLOGIA ARIS

Projeto de Graduação apresentado à Escola de Informática Aplicada da Universidade Federal do Estado do Rio de Janeiro (UNIRIO) para obtenção do título de Bacharel em Sistemas de Informação

Victor Manaia Gonçalves Chaves

Orientadoras:
Renata Mendes de Araujo, D. Sc.
Fernanda Araujo Baião, D. Sc.

IMPLEMENTAÇÃO DO MÉTODO MAC KNIGHT PARA ELICITAÇÃO DE REQUISITOS DE SOFWARE NA METODOLOGIA ARIS

Aprovado em ____/_____/_____

BANCA EXAMINADORA

_____

_____
Nome e Assinatura do(a) professor(a) orientador(a)

_____

_____

_____
Nomes e Assinaturas dos demais Membros da Banca

O(s) autor(es) deste Projeto autoriza(m) a ESCOLA DE INFORMÁTICA APLICADA da UNIRIO a divulgá-lo, no todo ou em parte, resguardados os direitos autorais conforme legislação vigente.

Rio de Janeiro, _____de_____ de 2008

_____
Nome e Assinatura do aluno

A Deus, que antes de tudo deu-me saúde e força pra trilhar a caminhada,
Aos pais e irmãos, que se esforçaram para o meu crescimento,
Aos amigos, pela força nos momentos de dificuldade
Aos professores, especialmente Fernanda Baião, Flávia Santoro, Renata Araújo e Alexandre Andreatta, pelas oportunidades de crescimento e ajuda nos momentos de dúvida.

Resumo

A alta competitividade do mercado atual exige que as organizações se preocupem com a qualidade de seus produtos e serviços. Neste contexto, os sistemas de informação têm sido de fundamental importância, seja na automação dos processos de trabalho ou no apoio à execução de suas atividades, de maneira a permitir que bons resultados sejam gerados. No entanto, para que estes sistemas auxiliem as organizações, é essencial que sejam bem especificados, de forma que seus requisitos estejam alinhados ao negócio. Porém, especificar estes sistemas não é uma tarefa trivial. Muitas vezes, a falta de entendimento sobre o negócio impede que as necessidades existentes sejam percebidas durante a elicitação dos requisitos.

Desta forma, Débora Mac Knight (MAC KINGHT, 2004) propôs um método que considera um modelo de negócio, previamente elaborado, e guia a equipe de desenvolvimento de sistemas na análise das diferentes informações deste modelo, buscando a identificação das necessidades do negócio e dos requisitos de software que atendem a essas necessidades.

Entretanto, as tarefas do método são executadas manualmente, já que não foi desenvolvido apoio computacional. Com este, torna-se viável: a execução do método em larga escala, a integração das informações levantadas pelo método ao próprio modelo de negócio, além da facilidade de acesso a informações sobre as execuções anteriores do método. Torna-se possível ainda a utilização do método sob a perspectiva do processo, formando uma visão integrada de todas as necessidades do negócio e requisitos de software identificados.

Desta forma, este trabalho se propõe a implementar uma ferramenta de apoio computacional ao método Mac Knight, aplicado à metodologia ARIS e utilizando a ferramenta *ARIS Business Architect*. Com isso, a execução do método pode ser facilitada, permitindo a manutenção da documentação e do histórico de solicitações anteriores, além da plena utilização da metodologia, dos seus objetos e da semântica dos relacionamentos. Para avaliar a implementação do método, foi elaborado um estudo de caso a partir da elicitação de requisitos realizada para um sistema de apoio ao processo de seleção do programa de pós-graduação em informática da UNIRIO. Os resultados obtidos demonstraram que a implementação obteve êxito e pode ser aplicada em larga escala, em grandes organizações.

Índice

Capítulo 1 - Introdução

1.1 Motivação

O mercado mundial vem se tornando cada vez mais competitivo. A grande quantidade de opções de produtos e serviços disponíveis, a velocidade e a dinâmica das informações fazem com que os consumidores busquem os produtos e serviços de mais alta qualidade, a preços mais acessíveis. Desta forma, as organizações devem se adaptar, oferecendo o que satisfaça a necessidade dos clientes, com preços competitivos e no menor tempo possível. Para isso, uma boa administração e controle do negócio tornou-se, mais do que um diferencial, uma necessidade para a sobrevivência das organizações.

Assim, são necessários processos de trabalho bem projetados, que possibilitem bons resultados e baixos custos, tanto para a organização quanto para seus clientes. É fundamental que as organizações conheçam e compreendam com mais precisão a si próprias, quais são suas funções e como são executadas, sua posição diante do mercado, suas possibilidades e suas metas.

Porém, compreender o funcionamento de uma organização, sem o apoio de um modelo ou uma estruturação que ajude a lidar com a sua complexidade, é uma tarefa quase inviável. A modelagem de negócio surgiu nesse contexto, com a finalidade de auxiliar as organizações a compreenderem melhor seu próprio negócio. Ao desenvolver o seu modelo de negócio, a organização nota o aumento de sua eficiência, além da descoberta de informações que possibilitam a tomada de decisões.

Através da compreensão das informações, é possível perceber pontos em que ocorrem problemas que afetam os clientes, aumentam os gastos, desperdiçam o trabalho dos funcionários, e conseqüentemente diminuem seus lucros. Com a modelagem, é possível também que a organização perceba oportunidades de melhoria em seus métodos de trabalho, de forma que o negócio seja executado mais eficientemente.

Outra vantagem que se obtém desse conhecimento é o auxílio no desenvolvimento de ferramentas computacionais para auxiliar no trabalho. Grande parte do conhecimento necessário para a execução das atividades em uma

organização é igualmente necessário no momento do desenvolvimento de sistemas de informação que visam dar apoio ou automatizar tais atividades. A organização precisa conhecer com precisão seu próprio negócio, tanto para projetar processos de trabalho eficientes, quanto para desenvolver sistemas que atinjam seus objetivos.

A descoberta (ou elicitação) dos requisitos de um sistema é a primeira e mais importante tarefa do seu desenvolvimento e pode determinar seu sucesso ou fracasso. Os resultados da elicitação de requisitos serão usados em futuras análises e especificações das funções dos sistemas. Se os requisitos não estiverem alinhados às necessidades do negócio, levarão a um sistema que não atenderá às expectativas de quem o solicitou.

Buscando associar o modelo de negócios à elicitação de requisitos, Mac Knight (MAC KNIGHT, 2004) propôs um método para elicitação de requisitos a partir do modelo de negócio da organização. Assim, sua idéia é utilizar os modelos de negócio da organização como base para a elicitação de requisitos, através de uma série de etapas e passos. Ao final da execução, é obtido o Documento de Requisitos de Software (DRS), que detalha as necessidades do sistema demandado. No trabalho, o método é concebido para ser utilizado em qualquer notação, metodologia ou software para modelagem de negócio, e a execução das etapas do método e conseqüente criação do DRS são feitas de forma manual.

1.2 Problema

Diante da execução manual do método, Mac Knight (MAC KNIGHT, 2004) sugere como contribuição para seu trabalho que o método seja automatizado, de forma que a geração do DRS possa ser apoiada através da construção de uma ferramenta.

Além disso, o trabalho original é genérico, uma vez que pode ser aplicado a diversas metodologias para modelagem de negócios, o que limita a documentação das informações, uma vez que qualquer dado a ser utilizado deve existir nas diferentes metodologias.

Assim, este trabalho procura oferecer suporte computacional ao método proposto em (MAC KNIGHT, 2004), aplicado à metodologia ARIS de modelagem de processos de negócio, que vem se destacando e tem sido amplamente utilizada e

aceita no mercado, inclusive em organizações de grande porte e órgãos governamentais brasileiros. Assim, além da possibilidade de automação da geração do documento de requisitos de software, é possível aumentar a semântica da modelagem, através da utilização dos elementos específicos da metodologia em questão.

Desta forma, a questão a ser tratada neste trabalho é a ***implementação do método Mac Knight*** (MAC KNIGHT, 2004) ***para elicitação de requisitos de software a partir do modelo de negócio sobre a metodologia ARIS, utilizando a ferramenta ARIS Business Architect***.

1.3 Enfoque da Solução

O presente trabalho baseia-se na argumentação apresentada por Mac Knight (MAC KNIGHT, 2004) de que é possível automatizar o método proposto, e desta forma identificar e documentar os requisitos de um sistema de informação a partir de uma solicitação de apoio computacional às atividades de uma organização e levando em consideração informações existentes em seu modelo de negócio.

A partir do conjunto de informações disponíveis no modelo de negócio e do conjunto de informações desejável em documentos usuais de especificação de requisitos de software, Mac Knight estabeleceu uma sistemática, dividida em sete etapas. Cada etapa do método deve considerar partes do modelo de negócio, de onde são extraídas informações relevantes ao sistema.

Neste trabalho, foi construído um conjunto de artefatos para fornecer apoio computacional às etapas do método, utilizando os conceitos da metodologia ARIS e a dinâmica da ferramenta *ARIS Business Architect*.

Destacam-se na implementação do método: a manutenção da rastreabilidade das informações de maneira gráfica, como no mapa de requisitos – que ilustra o relacionamento entre necessidades, requisitos e sistemas; a manutenção do histórico de elicitações anteriores e documentação das mesmas na própria modelagem; e a ampliação da semântica de representação de informações, com o uso de símbolos que representam os conceitos de requisitos funcionais e não funcionais.

Por fim, foi realizado um estudo de caso com o objetivo de avaliar a implementação do método, ilustrando sua execução num cenário real, validando os resultados esperados do método.

1.4 Estrutura do trabalho

No capítulo 2 são apresentados os conceitos que norteiam este trabalho, como definição e classificações de requisitos de software. Posteriormente, é apresentada a Engenharia de Requisitos, com suas fases, atividades e relevância. Em seguida são apresentados conceitos referentes à modelagem de negócio. São tratados ainda conceitos e processos para elicitação de requisitos e finalmente é apresentado o processo de elicitação de requisitos a partir do modelo de negócio.

No capítulo 3 é apresentada a metodologia ARIS. Inicialmente são apresentadas as razões e vantagens que levaram à escolha desta para a adaptação do método. Em seguida, a metodologia ARIS é detalhada, com seus modelos e diagramas. É apresentada na seqüência a ferramenta *ARIS Business Architect* e suas funcionalidades.

O capítulo 4 tem o objetivo de apresentar como foi concebida a implementação do método Mac Knight (MAC KNIGHT, 2004). Nele, são detalhadas as customizações realizadas na ferramenta para a plena utilização do método.

No capítulo 5 é apresentada a execução do método utilizando a ferramenta, ilustrada com o estudo de caso, que exemplifica a aplicação do método e se propõe a validar as adaptações realizadas para utilização na ferramenta. Para isso, é considerado como cenário o processo de seleção de mestrado do Programa de Pós-graduação em Informática da UNIRIO.

Por fim, a conclusão do trabalho é apresentada no capítulo 6 juntamente com as suas contribuições e possíveis trabalhos futuros.

O trabalho possui ainda dois anexos: no primeiro encontra-se o resultado do levantamento de requisitos para a customização da ferramenta, enquanto no segundo é exibido o documento de requisitos de software referente ao estudo de caso realizado.

Capítulo 2 - Conceitos

Neste capítulo serão apresentados os conceitos nos quais este trabalho se apóia. Serão contempladas questões sobre requisitos de software, elicitação de requisitos e métodos para executá-la, modelos de negócio, e a apresentação do método Mac Knight (MAC KNIGHT, 2004) para elicitação de requisitos a partir de um modelo de negócio genérico.

2.1 Requisitos de Software

A palavra "Requisito" é definida como **uma condição necessária para que certo objetivo seja alcançado**. No contexto de sistemas de informação, os requisitos não representam somente as declarações do que o sistema deve fazer, mas sim do que deve fazer para atender às tarefas executadas em uma organização. Dorfman e Thayer (DORFMAN, THAYER *et al*, 1990) definem como:

"(1) Uma capacidade de software exigida pelo usuário para que este resolva um problema de forma a atingir um objetivo. (2) Uma capacidade de software que o sistema deve possuir para satisfazer um contrato, uma especificação, um padrão ou outra exigência formalmente documentada."

Assim, os requisitos de um sistema definem os serviços que o sistema deve oferecer e as restrições aplicáveis à sua operação. É de extrema importância que não ocorram erros durante o levantamento de requisitos, pois estudos indicam que quando só detectados depois do software implementado, esses erros são até 20 vezes mais caros de se corrigir que qualquer outro tipo de erro.

2.1.1 Classificação de Requisitos

Tradicionalmente, os requisitos de software são classificados em funcionais e não funcionais.

Os requisitos funcionais definem como o sistema se comporta, através das declarações das funções que o sistema deve oferecer, de como o sistema deve reagir a entradas específicas e de como deve se comportar em determinadas situações, ou ainda

define restrições explícitas do que o sistema não deve fazer (SOMMERVILLE, 2003).

Os requisitos não funcionais são as restrições sobre serviços ou funções oferecidos pelo sistema - incluindo restrições de tempo e restrições no processo de desenvolvimento, padrões, e qualidades globais de um software. Surgem conforme a necessidade dos usuários, em razão de restrições do orçamento, de políticas organizacionais, pela necessidade de interoperabilidade com outros sistemas e outras necessidades do negócio (SOMMERVILLE, 2003).

2.1.2 Níveis de Requisitos

Para gerenciar a complexidade dos requisitos de software e ao mesmo tempo facilitar a troca de informações entre os diferentes tipos de colaboradores do desenvolvimento, é comum criar diferentes níveis de descrições dos requisitos. Cada nível descreve informações sobre as características do sistema, porém sob pontos de vistas de diferentes tipos de pessoas.

Segundo GOTTESDIENER (GOTTESDIENER, 2002), é possível considerar dois níveis de requisitos: os requisitos do usuário, que são declarações de alto nível sobre as funções que o sistema deve fornecer e as restrições sobre as quais deve operar; e os requisitos do sistema, que estabelecem detalhadamente funções e restrições, do ponto de vista do sistema, especificando de maneira consistente e completa o que o sistema deve fazer.

Segundo SOMMERVILLE (SOMMERVILLE, 2003), é interessante ainda considerar um nível superior a esses dois níveis de requisitos, o nível dos requisitos de negócio, onde são encontradas as descrições das necessidades que existem no negócio executado pelos usuários.

Assim, os três níveis de requisitos podem ser observados sob o ângulo de uma pirâmide: no topo estão as necessidades; no centro, encontram-se características do sistema; e na base estão os requisitos propriamente ditos. Podemos considerar um nível da camada inferior como desdobramento dos níveis superiores, com maior de detalhamento à medida que os níveis aproximam-se da base da pirâmide. A figura abaixo mostra a pirâmide com os diversos níveis.

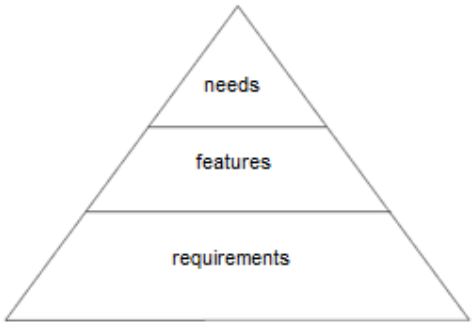


Figura 2.1 Níveis de requisitos (LEFFINGWELL, WIDRIG, 2000)

2.1.3 Documento de Requisitos de Software – DRS

Depois de levantados, tais requisitos devem ser documentados. Para isso, utiliza-se freqüentemente o Documento de Requisitos de Software. Segundo a definição do IEEE, o Documento de Requisitos de Software é a declaração oficial do que é exigido dos sistemas, descrevendo assim as características do produto de software. Os pontos essenciais, que devem ser definidos neste documento, são as funcionalidades, as interfaces externas, restrições, critérios de qualidade e de desempenho exigidas do sistema (IEEE, 1998).

Existem diversos padrões para o Documento de Requisitos de Software, cada qual desenvolvido por diferentes organizações de grande porte, como o Departamento de Defesa dos Estados Unidos e o IEEE, dentre outros.

Segundo Mac Knight (MAC KNIGHT, 2004), como o Documento de Requisitos de Software possui diferentes tipos de leitores – entre funcionários, gerentes e engenheiros -, é interessante manter diferentes versões, para que se possam documentar os requisitos específicos em seus diferentes níveis, de forma que seus leitores sejam capazes de compreendê-lo. Por exemplo, em uma versão descreve-se o sistema de acordo com a visão dos requisitos de usuário, enquanto em outra o mesmo é descrito de acordo com o nível dos requisitos do sistema.

Embora cada padrão defina uma estrutura ideal do documento, estes não devem ser considerados como modelo absoluto para todas as organizações. Cada uma deve estabelecer, de acordo com as suas particularidades, os itens e informações que são considerados importantes para o seu Documento de Requisitos de Software. O

modelo também pode variar de acordo com o tipo de software que está sendo desenvolvido e da abordagem de desenvolvimento.

2.2 Engenharia de Requisitos

A engenharia de requisitos propõe um conjunto de técnicas para criar e manter as informações sobre o sistema. Dessa forma, é estabelecido um processo que compreende as atividades de elicitar, documentar, analisar e validar os requisitos de um sistema, a fim de criar o Documento de Requisitos se Software. A gerência de requisitos é outra atividade a ser considerada, na medida em que os requisitos podem sofrer alterações ao longo do tempo.

2.2.1 Etapas

Segundo Pressman (PRESSMAN, 2001), o processo da engenharia de requisitos pode ser dividido em cinco principais etapas: o reconhecimento do problema, a interpretação das necessidades que demandam o sistema, a modelagem, a especificação e a validação dos requisitos que atendem essas necessidades.

As etapas de reconhecimento do problema e interpretação têm o objetivo de obter o conhecimento do domínio do problema e identificar as necessidades dos futuros usuários. Nas etapas de modelagem e especificação, o conhecimento adquirido é descrito através de modelos conceituais, de forma a propiciar uma melhor compreensão das propriedades que o sistema deve possuir. Esses modelos devem ser então analisados, de forma a identificar se o problema está representado e entendido coerentemente. Caso necessário, devem-se resolver as possíveis inconsistências e omissões. Finalmente, na validação dos requisitos, as informações obtidas anteriormente são confirmadas. O processo termina com os requisitos especificados no Documento de Requisitos de Software. O nível deste documento varia de acordo com o nível de detalhe das análises e validações realizadas.

2.2.2 Relevância

Segundo Reubenstein e Waters (REUBENSTEIN, WATERS, 1991), os requisitos estão associados aos principais problemas do desenvolvimento de sistemas, normalmente pela dificuldade para conseguir um entendimento comum

entre usuários e desenvolvedores, que leva à definição de requisitos incompletos ou inconsistentes, gerando especificações que não refletem as reais necessidades dos usuários.

Os resultados da má definição de requisitos vão desde atrasos no cronograma e aumento de custos - pois haverá retrabalho para atualização da documentação e modelos do sistema que tenham de ser alterados em função da correção ou levantamento de novos requisitos -, até obter um sistema que soluciona o problema errado.

Conclui-se, portanto, que as atividades da engenharia de requisitos são fundamentais para o desenvolvimento de bons sistemas. Entretanto, cada equipe deve adaptar os processos da engenharia conforme suas necessidades, em função das características do sistema, dos usuários, da equipe, da organização, do tipo do sistema e da tecnologia a ser utilizada, uma vez que não é possível definir um processo ideal para todos os projetos.

2.3 Modelagem de Negócio

Segundo Eriksson e Penker (ERIKSSON, PENKER, 2000), a modelagem de negócio é um conjunto de conceitos, modelos e técnicas com o objetivo de desenvolver o modelo de negócio de uma organização. Este modelo é resultado de uma abstração da mesma, que considera as suas características essenciais, do ponto de vista de seu negócio; indicando o ambiente no qual a organização está inserida e como a organização age em relação a este ambiente.

Segundo bibliografia da Engenharia de Software, o objetivo de um modelo de negócio é responder às seis perguntas clássicas sobre a organização: O que é feito? Quem faz? Quando? Onde? Por quê? Como? Através das respostas a essas perguntas, é possível obter uma visão ampla sobre a organização e seu negócio.

Várias são as vantagens que motivam uma organização a criar o seu modelo de negócio. Através do modelo, é possível, por exemplo, identificar pontos críticos e planejar melhorias na organização. O modelo pode representar a forma como o negócio da organização se comporta atualmente (*as-is*) ou como poderia se comportar, caso fossem feitas mudanças (*to-be*).

Além disso, o modelo de negócio pode ser utilizado para o desenvolvimento de sistemas, pois facilita a unificação do conhecimento dos envolvidos ajudando na melhoria do entendimento do sistema.

2.3.1 Conceitos dos modelos de negócio

Abaixo serão apresentados os conceitos do modelo de negócio utilizados nas diversas metodologias. Apesar de serem apresentados individualmente, não estarão isolados no modelo de negócio – encontrando-se em descrições textuais ou entremeados uns aos outros.

- **Interações de Negócio**: Visão das interações entre a parte interna e o ambiente externo à organização, definindo o escopo de negócio. Através das interações dos componentes da organização representados, observa-se como a organização age em relação ao ambiente que está inserida.
- **Estrutura Organizacional**: Representação da estrutura da organização, através de suas unidades organizacionais, papéis, grupos ou funções assumidos por seus trabalhadores. Sua representação costuma ser feita de forma hierárquica, auxiliando na sua visualização e entendimento.
- **Localização Geográfica:** Representação das diversas localidades onde a organização atua e executa seus processos de negócio; através de associações, explicita os relacionamentos entre as localidades e as diversas unidades organizacionais.
- **Objetivos de Negócio:** Representam as metas e os objetivos a serem atendidos pelos processos de negócio da organização. Esta informação é importante para que se saiba a razão pelas quais as tarefas e processos são executados.
- **Processos de Negócio:** Conjunto de atividades estruturadas e métricas destinadas a resultar em um produto específico para um determinado cliente ou mercado.
- **Objetos de Negócio:** Representação dos conceitos de tudo que é manipulado, produzido ou modificado nos processos - como papéis, informações ou produtos.

- **Eventos:** Os eventos são disparos de serviços ou estados atingidos após a execução das atividades, e que surgem da execução de processos ou do mundo exterior à organização.
- **Atividades:** As atividades são as etapas dos processos, que mostram como um determinado processo é executado, e seu fluxo de trabalho. Geralmente estão associadas aos papéis ou unidades organizacionais que as executam, os objetos de negócio que manipulam, e as ferramentas que utilizam, bem como os resultados que cada atividade gera.
- **Regras de Negócio:** definem ou restringem algum aspecto do negócio. Sua intenção é afirmar a estrutura do negócio ou controlar ou influenciar o seu comportamento.

2.3.2 Abordagens para a modelagem de negócio

A forma de representação dos conceitos do modelo de negócio pode variar, dependendo da abordagem utilizada para sua criação. Existem várias abordagens para a modelagem de negócio, que procuram apresentar as características das organizações e seus processos segundo enfoques específicos. Para isso, utilizam uma determinada notação e linguagem.

De uma forma geral, as informações que estão sendo representadas em cada abordagem são as mesmas, ou seja, os conceitos representados são os apresentados anteriormente, variando somente o conjunto de modelos que representam cada conceito e a linguagem utilizada para representar cada modelo. Dentre as existentes, podemos citar:

- **UML**: A *Unified Modeling Language* é uma linguagem visual e orientada a objetos, desenvolvida pela OMG (*Object Management Group*), para modelagem de sistemas.
- **Abordagem Rational:** A abordagem da Rational utiliza a UML, com os estereótipos descritos pela OMG, e propõe uma forma de utilizar os diagramas existentes na UML para representar conceitos do negócio.
- **Abordagem Proforma:** Essa abordagem não se fixa a uma linguagem específica, permitindo a utilização de várias, como a própria UML, ProGuide, IDEF, Core, OMT, etc.

- **Abordagem ARIS:** Criada por Scheer, considera que o modelo de negócio seja constituído de modelos e descrições sobre o negócio. Como principal vantagem, destaca-se a variedade de modelos, separando a informação e permitindo sua melhor visualização e entendimento. Além dessa, a possibilidade da customização da ferramenta, visando adaptar-se à realidade de cada organização. Diante disso, este projeto a utiliza a abordagem ARIS na implementação do método e no estudo de caso, que será apresentado em detalhes no próximo capítulo. (SCHEER, 1998)

2.4 Método de Mac Knight para Elicitação de Requisitos de Software a partir do modelo de negócio.

O método para elicitação de requisitos de software a partir do modelo de negócio foi proposto por Débora Mac Knight em sua tese de mestrado, em 2004 (MAC KNIGHT, 2004).

2.4.1 Objetivo

O método tem como objetivo servir de guia para a equipe de desenvolvimento de software durante a elicitação de requisitos. Seu uso é indicado em situações onde a automação de tarefas e serviços necessários à organização são solicitados, auxiliando os engenheiros de software a obterem os requisitos do novo sistema.

Um aspecto que recebe especial atenção do método é a colaboração entre os engenheiros de software e pessoas da organização. Essa colaboração se dá através de reuniões, onde o envolvimento de pessoas que possuem os conhecimentos necessários a cada etapa do método permite complementar o entendimento do problema e da solução, fazendo com que o resultado final seja o mais completo possível. As etapas propostas no método servem como uma guia, indicando as tarefas necessárias para a identificação de cada informação.

Outro aspecto levado em consideração é a rastreabilidade dos requisitos identificados pelo método. A relação entre cada necessidade de negócio e cada funcionalidade ou restrição do sistema deve ser mantida, permitindo a identificação do que deve ser alterado caso o negócio sofra alguma alteração.

Por fim, é esperado que o método seja capaz de considerar o modelo de

negócio de uma organização, descrito em qualquer notação. Para tanto, deve-se identificar os conceitos sobre a estrutura organizacional, processos de negócio, atividades, objetivos de negócio, eventos e localização geográfica. O documento final do método lista os requisitos de usuário, estando as demais etapas do desenvolvimento do software fora do escopo do método.

2.4.2 Detalhamento do método

O método é divido em duas fases: a visão geral do problema, e a visão geral da solução. Cada fase é composta por etapas e passos.

2.4.2.1 Fase 1: Visão Geral do Problema

A primeira fase do método engloba as três primeiras etapas do método, e tem o objetivo de dar aos engenheiros de software uma visão do que é o negócio, da organização, e quais partes destes deverão ser consideradas para atender à solicitação feita.

2.4.2.1.1 Etapa 1: Identificar o contexto da solicitação

O objetivo desta etapa é definir o escopo da área de negócio envolvido com a solicitação. Espera-se como premissa que já exista um modelo de negócio, definido em qualquer linguagem ou notação.

No primeiro passo, deve-se analisar o modelo de processos com o objetivo de identificar os processos envolvidos com a solicitação, baseando-se na sua descrição. Existindo subprocessos, estes também devem ser avaliados.

No segundo passo, analisa-se cada processo identificado, buscando dentre as atividades que o compõe, aquelas que devem ser consideradas no contexto da solicitação.

O resultado da primeira etapa é a lista de processos e atividades dentro do contexto da solicitação.

2.4.2.1.2 Etapa 2: Identificar necessidades dos processos

O objetivo dessa etapa é especificar as necessidades existentes em cada atividade dos processos no contexto da solicitação.

No primeiro passo, identificam-se as dificuldades existentes em cada atividade dos processos, através de entrevistas com os executantes das mesmas. Dificuldades são quaisquer obstáculos que devem ser superados para a execução das atividades. No segundo passo, analisam-se as dificuldades com o objetivo de se entender suas causas.

No terceiro passo são identificados os resultados insatisfatórios gerados pelos processos, através de uma análise que busca avaliar se as atividades são executadas da forma adequada e se produzem resultados satisfatórios - percebendo pontos de melhoria nos processos e problemas que podem gerar impactos em clientes, além de questões como segurança e velocidade de processamento.

Dessa forma, o resultado da segunda etapa é a lista de necessidades de negócio existentes em cada atividade dos processos.

2.4.2.1.3 Etapa 3: Identificar os impactos das necessidades

O objetivo dessa etapa é obter uma visão dos impactos das necessidades identificadas.

No primeiro passo, identificam-se as conseqüências de cada necessidade – como a existência da atividade afeta as atividades e processos.

No segundo passo, analisa-se cada uma das conseqüências para identificar os objetivos de negócio afetados.

No terceiro passo, identificam-se as pessoas envolvidas com o negócio que sofrem as conseqüências das necessidades existentes, através da analise dos papéis que se relacionam com os objetivos que sofrem os impactos das conseqüências. Por fim, devem-se relacionar as pessoas afetadas com os objetivos do passo anterior.

Já no quarto passo, definem-se quais necessidades serão atendidas pelo sistema, e as prioridades de cada uma. .

Assim, o resultado da etapa é uma tabela que mostra as necessidades existentes relacionadas com suas conseqüências, objetivos, pessoas afetadas e prioridades.

2.4.2.2 Fase 2: Visão Geral da Solução

A segunda fase do método tem o objetivo de criar a visão da solução que atenda às reais necessidades da solicitação, identificando funcionalidades e restrições

do sistema. Assim, no final desta fase, os engenheiros de software terão um resumo das principais características do sistema, de acordo com o que o negócio exige. Ao mesmo tempo, no final da fase, a organização será capaz de perceber as mudanças que ocorrerão em seus processos com a criação do sistema.

2.4.2.2.1 Etapa 4: Identificar as funcionalidades e restrições do sistema

O objetivo dessa etapa é identificar as principais funcionalidades e restrições que o sistema deverá possuir para atender às necessidades verificadas anteriormente. No primeiro passo, os próprios analistas devem identificar as funcionalidades e restrições do sistema para que atenda aos processos dentro do contexto, que podem ser restrições ou funções específicas que o sistema deverá prover.

No segundo passo, deve-se analisar cada necessidade de negócio, agora em reunião com os gerentes, com o objetivo de identificar o que deve ser feito para que seja suprida, listando necessidades, funcionalidades e restrições adicionais.

O resultado da quarta etapa é a lista de todas as funcionalidades e restrições identificadas para o sistema. Essa lista torna possível a relação das atividades e processos com as funcionalidades e restrições geradas no sistema.

2.4.2.2.2 Etapa 5: Identificar as fronteiras do sistema

O objetivo desta etapa é identificar os pontos em que o sistema trocará informações com outros recursos do negócio.

No primeiro passo, é feita uma busca nos modelos pelos papéis que executam as atividades identificadas, buscando saber se continuarão sendo executadas por estes ou se passarão a ser totalmente executadas pelo sistema. Caso o sistema apenas apóie estas atividades, haverá então a interação por parte destes papéis, sendo necessária a definição dos usuários e permissões de cada um.

No segundo passo, avalia-se a possibilidade de haver interação com outros sistemas, identificando se ela será executada com apoio de algum outro sistema e, principalmente, se ela continuará a ser apoiada por este sistema após a implantação do novo. Em caso positivo, deve-se verificar se as informações manipuladas são relevantes, de modo que se identifique se o novo sistema deve interagir com o antigo ou não.

No terceiro passo, analisam-se os acessos das atividades aos meios de armazenamento. Se o meio de armazenamento for computacional deverá haver interação - especificando seu conteúdo, na forma de descrição, e os dados dos registros que serão armazenados. Caso o meio de armazenamento seja manual e suas informações sejam relevantes ao sistema, o trabalho de desenvolvimento do sistema deverá incluir o desenvolvimento deste meio de armazenamento.

Dessa forma, o resultado da quinta etapa é a lista de recursos que interagirá com o sistema, e suas especificações (usuários, permissões e dicionário de dados), além do diagrama que represente o contexto de utilização do sistema.

2.4.2.2.3 Etapa 6: Identificar os impactos do novo sistema

O objetivo desta etapa é avaliar como o novo sistema interfere no negócio da organização. O surgimento do sistema gerará mudanças na forma como os processos serão executados, e algumas atividades poderão ser executadas automaticamente, enquanto outras passarão a ser apoiadas pelo sistema.

O primeiro passo é analisar o modelo de atividades e identificar quais atividades serão re-alocadas a novos executantes e quais serão feitas pelo novo sistema, de acordo com as funcionalidades definidas.

No segundo passo, devem-se listar todos os meios de armazenamento que interagirão com o sistema com o objetivo de ter essa informação facilmente disponível. Devem-se listar também os papéis que interagirão com o sistema.

Como resultado dessa etapa, portanto, temos estas listas, que representam o conjunto de mudanças que ocorrerão no modelo de processos, representando as alterações causadas ao processo com o surgimento do novo sistema.

2.4.2.2.4 Etapa 7: Gerar Documento de Requisitos de Software

A última etapa do método tem a finalidade de documentar as informações relevantes ao sistema no Documento de Requisitos de Software. Para isso, tem como entrada os resultados de todas as etapas anteriores.

2.4.3 Considerações sobre o método

Ao final da execução do método são obtidos os requisitos de software. Estes representam as características de alto nível para o novo sistema, o que inclui as

descrições de suas funcionalidades e suas restrições. É possível notar que o método considera que os requisitos funcionais e não funcionais sejam identificados a partir da análise do modelo de negócio, em reuniões com as pessoas da organização.

A partir da análise do Documento de Requisitos de Software gerado, é possível observar que o relacionamento entre as necessidades de negócio de cada atividade considerada e as funcionalidades e restrições definidas para o sistema foi mantido. Com isso, é possível saber a razão da existência de cada requisito e, caso alguma atividade no processo sofra alteração é sempre possível avaliar os impactos que esta alteração pode ter nos requisitos do sistema. Dessa forma, o método fornece recursos para a gerência de requisitos, atividade que deve ser executada ao longo de todo o ciclo de vida do novo sistema.

Capítulo 3 - Metodologia ARIS para a Modelagem de Processos de Negócio

Neste capítulo serão apresentados os motivos que levaram à escolha da metodologia ARIS para a implementação do apoio computacional ao método Mac Knight; além das principais características da metodologia, seus modelos e objetos disponíveis para utilização na modelagem do negócio, e algumas de suas funcionalidades.

3.1 Escolha da metodologia a ser utilizada

No início deste trabalho, existiam as propostas de criação de uma ferramenta ou a customização de uma já existente. A criação de uma ferramenta permitiria que todo o levantamento de requisitos e desenvolvimento atendessem criteriosamente às necessidades do método, mas demandaria grande esforço e não teria a abrangência que foi obtida com a customização de uma metodologia. Quando da opção pela customização, foi necessária a escolha de uma ferramenta para aplicação do método. Assim, a metodologia ARIS – com a utilização da ferramenta *ARIS Business Architect* - foi escolhida por uma série de fatores, apresentados a seguir.

A primeira razão para a escolha foi o uso no mercado. A metodologia em questão vem ocupando lugar de destaque em grandes corporações, especialmente nas governamentais - como BNDES e PETROBRAS. Como um dos objetivos deste trabalho é oferecer uma aplicação prática do método, a escolha de uma ferramenta de grande utilização no mercado na atualidade pode popularizar seu uso, e aumentar sua utilização em cenários reais.

Junto a isso, merece destaque a facilidade de utilização e personalização. A ferramenta *ARIS Business Architect* possui uma interface dividida em módulos, que dividem intuitivamente as tarefas de modelagem, administração, visualização da estrutura e criação de relatórios através de scripts. No módulo de administração, podem ser executadas diversas tarefas de personalização da ferramenta, extremamente válidas para a proposta deste trabalho, possibilitando a implementação do método Mac Knight e ampliando a possibilidade de documentação.

3.2 A metodologia ARIS

Todas as companhias têm processos de negócio, independente de seu tamanho ou segmento. Processos eficientes habilitam a companhia a fornecer produtos e serviços de qualidade, reagindo rapidamente às mudanças do mercado. Pensando em aperfeiçoar e suportar essas atividades, A.W. Scheer criou a metodologia ARIS (SCHEER, 1998)

A sigla ARIS – *Architecture of Integrated Information Systems* - define uma arquitetura para descrever processos de negócio. A metodologia ARIS de Modelagem de Processos de Negócios está fundamentada na utilização de uma grande variedade de modelos e objetos através dos quais os processos de negócio de uma dada organização podem ser representados e analisados, com a utilização da ferramenta *ARIS Business Architect*.

A metodologia é orientada ao processo, e as informações sobre os processos e atividades podem ser descritas textualmente, através do preenchimento das propriedades de objetos e modelos; através de uma tabela, gerada através do editor de matrizes ou por script, ou ainda graficamente. A vantagem deste último é demonstrar rapidamente a informação através de símbolos padronizados para os diferentes elementos – denominados de objetos.

Através da ferramenta, podem ser descritos processos e estruturas empresariais, estruturas de sistemas de aplicação e de produtos e serviços gerados na execução dos processos.

O conceito para o desenvolvimento da metodologia ARIS está fundamentado na integração dos processos de negócio. O primeiro passo para o desenvolvimento desta arquitetura foi a criação de um modelo que contivesse as principais características para se descrever um processo de negócio. O resultado criado foi extremamente complexo, sendo necessária a divisão em partes para interpretar o todo. Desta forma criou-se uma divisão em visões

Tais visões são inter-relacionadas, de forma que os modelos possam ser analisados holisticamente e sem redundâncias. Assim, podem ser agrupados em cinco partes: Organização, Função, Dados, Processos e Produtos. Essa divisão também é conhecida como *ARIS House*. (SCHEER, 1998)

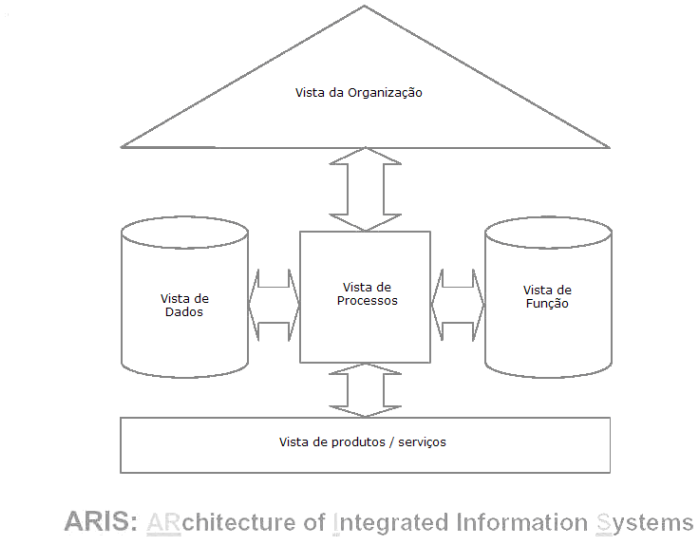


Figura 3.1 Vistas dos modelos da Metodologia ARIS (SCHEER, 1998)

3.3 A Ferramenta ARIS

Baseado na metodologia ARIS, o *ARIS Toolset* foi criado por A.W. Scheer em 1992, e é comercializado pela IDS Scheer. É composto por um conjunto de ferramentas que apóiam as atividades de gestão de processos de negócio, através da criação e manutenção dos modelos de processo de negócio da organização, podendo ser facilmente customizadas para necessidades individuais.

Assim, unidades organizacionais, dados de objetos, sistemas de aplicação e cadeias de informações podem ser representadas em eventos e funções. Textos, apresentações, vídeos e links de Internet podem ser integrados à representação gráfica como informações adicionais. O detalhamento de processos também pode ser realizado em todos os níveis hierárquicos.

A geração de modelos de forma fácil otimiza os projetos, que podem ser representados de forma gráfica e através de descrição textual. As opções de navegação configuram uma apresentação amigável dos dados, sendo possível inclusive a produção e manutenção de documentação física – através da geração automatizada de documentos e relatórios.

Dentre os objetivos alcançados com o uso da ferramenta destacam-se: a visão geral dos processos, através de uma navegação simples e intuitiva; participação do usuário, na medida em que o mesmo participa da evolução dos processos; aumento da qualidade da documentação, através do armazenamento e utilização dos dados na

própria ferramenta, ou através da geração de relatórios automatizados por script; e reutilização da informação, através da aplicação dos conceitos de definição de objeto – que guarda as propriedades do objeto, como uma espécie de definição de uma classe – e ocorrência de objeto – instâncias da definição do objeto, equivalente às instâncias de uma classe.

3.4 Funcionalidades

As funcionalidades descritas a seguir referem-se à versão 7.0.1 do ARIS. Nesta versão, a linguagem utilizada para criação de scripts é o Java Script, e traz uma série de melhorias na interface gráfica em relação à versão anterior. Além disso, possui uma nova funcionalidade, o denominado *Matrix Editor*, que permite a visualização dos relacionamentos entre objetos.

3.4.1 Objetos

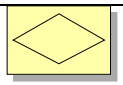
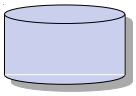
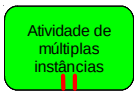
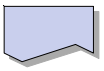
A unidade básica de toda a modelagem ARIS são os objetos. Segundo a IDS Scheer (SCHEER AG, 2006), um objeto descreve uma realidade e é gravado como uma definição na base de dados, podendo ser descrito em detalhes através do preenchimento de seus atributos.

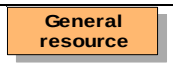
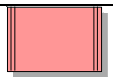
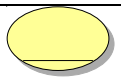
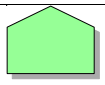
A metodologia trata os objetos como definições ou ocorrências. De maneira semelhante a uma classe, que tem sua definição e suas instâncias, uma definição de objeto pode ter uma ou mais ocorrências. Assim, as diversas ocorrências de um mesmo objeto podem compartilhar propriedades, também denominados atributos.

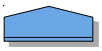
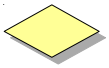
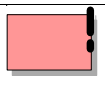
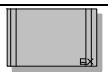
Atributos de objetos dão informações adicionais sobre o mesmo, que podem ser visualizadas pela interface gráfica, ou através dos *reports*. Cada definição possui ao menos um nome, um identificador– único para cada definição existente no banco de dados – e um tipo, mas podem ter outros atributos preenchidos pelo usuário, como descrição, autor, data de criação, e outros que podem ser personalizados pelo usuário através do filtro metodológico.

Cada definição possui um tipo associado – que pode ser função, operador lógico, papel, evento, dentre outros - e que pode ser personalizado através da alteração do símbolo no filtro metodológico.

A tabela abaixo ilustra os principais objetos utilizados na modelagem, uma breve descrição sobre cada um e uma representação gráfica do mesmo.

Nome	**Semântica**	**Sintaxe**
Link para outro modelo	Este símbolo está associado a qualquer elemento que possua um modelo associado a ele.	
Agregação	Relacionamento entre entidades com atributos ou relacionamentos próprios.	
Arquivo	Representa uma informação (documento, relatório, planilha, etc..) disponibilizada em arquivos armazenados em meio eletrônico, gerada por ou utilizada em uma atividade.	
Atividade	Constitui uma etapa de uma seqüência que precisa ser executada para que um processo seja realizado.	
Atividade de múltiplas instâncias	Representa uma atividade que pode ser executada várias vezes em um processo, de acordo com a sua granularidade. A granularidade determina o critério de repetição da atividade. Diferente de um loop, no qual o processo só prossegue depois que o loop terminar, uma atividade de múltiplas instâncias define que a partir daquele ponto o processo possui múltiplas instâncias. Cada instância é executada paralelamente, seguindo, de forma independente uma da outra, o curso do processo.	
Atributo	Representa uma propriedade de um conceito	
Conceito	Entidade conceitual identificada no domínio da modelagem realizada.	
Documento	Representa uma informação (documento, relatório, planilha, etc..) disponibilizada em arquivo físico ou impressa em papel, gerada por ou utilizada em uma atividade.	

Nome	Semântica	Sintaxe
E	Operador lógico que representa: - quando dividir o fluxo: que todos os caminhos precisam ser percorridos, ou seja, todas as atividades destino devem ser executadas; - quando unir o fluxo: que todos os caminhos devem ser percorridos antes de iniciar a atividade seguinte, ou seja, todas as atividades origem devem ser executadas.	
Equipamento	Representa um equipamento físico utilizado na execução de uma atividade.	
Evento	Representa uma circunstância ou status que propicia o início, uma etapa ou final processo, relevante para o entendimento do mesmo	
Grupo	Grupo, comitê ou célula de pessoas e/ou unidades responsáveis pela execução de uma tarefa no processo.	
Indicador de desempenho	Representa um indicador de desempenho para avaliar a atividade.	
Informação	Representa um conjunto de dados (digitais) consumidos ou gerados por uma atividade	
Integração com outros sistemas	Representa a integração do sistema do qual está sendo feito o levantamento de requisitos com sistemas externos	
Interface de processo	Representa uma ligação entre dois processos.	
Localização	Local ou ambiente onde está situada uma unidade organizacional.	
Objetivo	Objetivo associado ao processo ou da atividade.	

Nome	Semântica	Sintaxe
Ou	Operador lógico que representa: - quando dividir o fluxo: que pelo menos um dos caminhos precisa ser percorrido, ou seja, no mínimo um dos eventos destino deve ocorrer, porém podem ocorrer outros; - quando unir o fluxo: que pelo menos um dos caminhos percorridos é suficiente para iniciar a atividade seguinte, ou seja, no mínimo um dos eventos origem deve ocorrer.	
Ou exclusivo	Operador lógico que representa: - quando dividir o fluxo: que apenas um dos caminhos será percorrido, ou seja, apenas um dos eventos destino ocorrerá; - quando unir o fluxo: que apenas um dos caminhos percorridos é suficiente para iniciar a atividade seguinte, ou seja, apenas um dos eventos origem precisa ocorrer.	
Posto de trabalho	Representa o posto de trabalho responsável pela execução ou apoio a uma atividade.	
Processo	Representa um processo intermediário ou final de uma seqüência da cadeia de valor.	
Processo Inicial	Representa um processo inicial de uma seqüência da cadeia de valor. Pode representar também um processo superior, a partir do qual existe uma seqüência de processos a ele subordinados.	
Produto Físico	Representa um produto físico ou um serviço disponibilizado em determinada etapa de um processo	
Relacionamento	Relacionamento existente entre entidades.	
Risco	Risco que algo impacte a execução da atividade.	
Sistema	Representa um sistema de informação que apóia a execução ou executa uma atividade.	
Termo	Descreve uma expressão que necessita de explicação.	

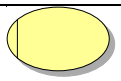
Nome	Semântica	Sintaxe
Unidade organizacional	Representa uma área da organização.	

Tabela 1 Lista de objetos da metodologia ARIS

3.4.2 Modelos

Segundo a IDS Scheer (SCHEER AG, 2006), modelo é uma maneira simplificada de representar a realidade empresarial através da descrição ou representação gráfica. Para modelar os processos de negócio de uma organização, com suas atividades, papéis, unidades organizacionais, objetivos, produtos e demais elementos da modelagem, são utilizados uma série de diagramas.

Cada um destes modelos é baseado em um tipo de modelo, que está associado a uma visão específica no *ARIS House*, e também podem ser personalizados através do filtro metodológico. Na metodologia se destacam os seguintes modelos:

3.4.2.1 Value-Add-Chain

Cadeia de Valor Agregado, ou VAC, representando o fluxo dos processos em alto nível, relacionados aos objetivos da execução dos processos;

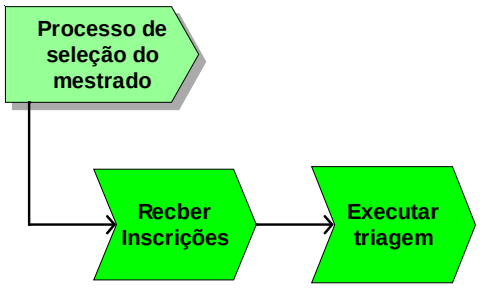


Figura 3.2 Exemplo de modelagem no VAC

3.4.2.2 Objective Diagram

Diagrama de Objetivos, ou DO, mostra em detalhes os objetivos da execução de cada etapa do processo;

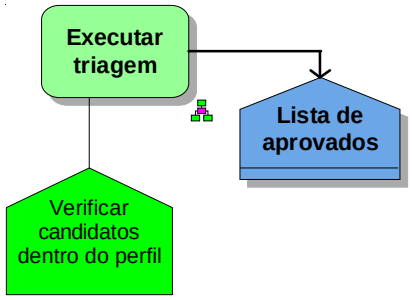


Figura 3.3 Exemplo de modelagem do diagrama de objetivos

3.4.2.3 Organizational Chart

Organograma, mostrando as diversas unidades organizacionais, papéis e grupos, com a possibilidade de contextualização por localidade geográfica;

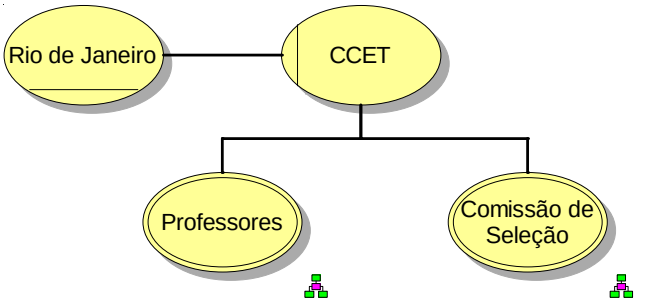


Figura 3.4 Exemplo de modelagem do organizational chart

3.4.2.4 Application System Diagram

O diagrama de sistema de aplicação mostra como um sistema se relaciona com outros sistemas e com elementos que o compõem, como bases de dados, documentos, telas, etc.

3.4.2.5 eERM

O Diagrama de Entidades e Relacionamentos define como entidades se relacionam, explicitando questões de relacionamentos e herança, etc.

3.4.2.6 Function Allocation Diagram -FAD

O Diagrama de Alocação de Função, ou FAD, oferece um detalhamento de uma atividade do processo, mostrando os recursos que interagem com esta, além dos

objetos de entrada e saída – como informações de documentos, riscos, equipamentos, dentre outros.

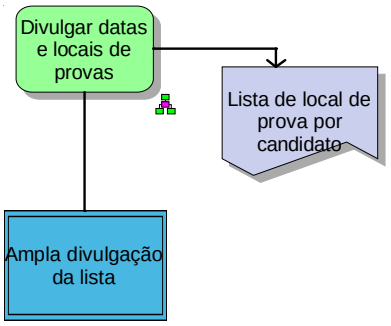


Figura 3.5 Exemplo de modelagem do FAD

3.4.2.7 EPC

Diagrama que detalha as diversas atividades de um processo, detalhando o seu fluxo de execução e os estados atingidos antes e depois da execução de uma atividade ou de uma seqüência destas. Existem variantes, como o *eEPC*, *eEPC Column Display* e o *eEPC Row Display*, digramas que detalham as atividades de um processo separadas por papel ou unidade organizacional, divididas em raias.

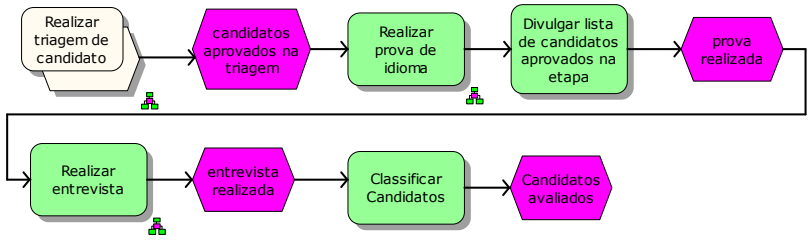


Figura 3.6 Exemplo de modelagem do EPC

3.4.2.8 Technical Terms Model

Diagrama de termos técnicos, usado para listar todas as definições de termos do modelo. Costuma ser utilizado alternativamente como dicionário de dados.

3.4.3 Grupos

Grupos são os diretórios onde o ARIS mantém diagramas e modelos. Assim como os objetos e modelos, possuem atributos default e outros que podem ser preenchidos pelos usuários. Além disso, o *ARIS Business Architect*, mantendo a

funcionalidade do *ARIS Toolset*, permite a definição de privilégios de acesso por usuário ou por grupos para cada diretório. Toda base de dados tem como início um grupo denominado ‘Grupo Principal’, onde estão armazenados subdiretórios, modelos e objetos.

3.4.4 Conexões

Segundo a IDS Scheer (SCHEER AG, 2006), uma conexão representa um relacionamento entre dois objetos. Dessa forma, a conexão sempre vai de um objeto origem a um objeto destino. Dependendo dos objetos ligados, podem existir mais de um tipo de conexão disponível e mais de uma orientação do relacionamento, alterando origem e destino das conexões.

Assim como os demais elementos, uma conexão tem um tipo associado, e demais atributos, que podem ser definidos manualmente, além da possibilidade de criação de novos atributos e alterações nos tipos de conexões através de filtro metodológico.

Através das conexões é possível percorrer o fluxo de um diagrama, bem como obter os objetos relacionados a um objeto selecionado, sendo os objetos pais aqueles cujo relacionamento tem destino no objeto selecionado; e objetos filhos aqueles cujo relacionamento tem origem no objeto selecionado.

3.4.5 Relacionamento

Os relacionamentos – *assigments* na metodologia IDS Scheer (SCHEER AG, 2006) – estabelecem uma conexão entre uma ocorrência de um objeto num modelo qualquer e outro modelo, que disponibilizará maiores informações sobre o objeto de origem. Existe apenas um tipo de relacionamento, mas os modelos disponíveis para realizar este relacionamento dependem do objeto de origem. As opções de relacionamentos disponíveis podem ser editadas no filtro, mas não há possibilidade de criação de novas combinações de relacionamentos além dos já definidos no ARIS.

3.4.6 Filtros

O filtro metodológico é um arquivo com uma série de definições que influenciam, dentre outras coisas, em quais modelos, objetos e relacionamentos estarão disponíveis para utilização, quais atributos estarão disponíveis para cada elemento do ARIS, quais símbolos serão atribuídos aos objetos, e quais os *assigments p*ossíveis entre os objetos e modelos. Os filtros definem o domínio da modelagem na medida em que delimitam os recursos disponíveis, orientam como a informação será armazenada e definem como será o padrão da simbologia dos objetos.

3.4.7 Templates

Os *templates* são uma série de definições de como serão exibidos os modelos e objetos criados. Assim, os *templates* definem cor de fundo dos modelos, local da exibição do nome e outros atributos, e tamanho da imagem dos objetos. É possível ainda aumentar o contorno dos mesmos, incluir efeitos de sombreamento e outros recursos gráficos.

3.4.8 Reports

Dentre as funcionalidades do *ARIS Business Architect* está a criação de relatórios personalizáveis. Para isso, são utilizados scripts, que manipulam as informações modeladas e geram relatórios de acordo com o que foi programado.

Segundo os arquivos de ajuda da ferramenta, a linguagem utilizada na programação destes relatórios é o *ARIS Script*, linguagem própria baseada em *Java Script* e com algumas funcionalidades *Java*, que oferece além da maioria dos métodos das linguagens de base, suporte ao acesso das informações de objetos, modelos, pastas e bases de dados através de alguns métodos internos.

Cada objeto manipulado possui uma série de métodos para acesso às suas propriedades, sendo possível a criação de novos diagramas, ocorrências do objeto e alteração de suas propriedades. Merece destaque a manutenção do nome da maioria dos métodos e constantes da versão anterior para a versão 7. As constantes - que guardam informações sobre cores, tipos de modelos e objetos – foram reunidas em uma classe chamada *constants*, já que o *Java Script* não permite o uso de constantes. Nesta versão a criação automatizada de scripts e o editor de interfaces – que gerava o

código das interfaces desenvolvidas de maneira gráfica- não estão mais disponíveis. Destaca-se, entretanto, a organização e estruturação dos scripts dentro do próprio ARIS, além da criação da pasta *Common Files*, onde são armazenadas bibliotecas com métodos e imagens que podem ser utilizadas pelos demais scripts, bastando apenas importar a biblioteca ou a imagem correspondente no script onde se deseja utilizá-los.

Cada script tem um contexto específico para ser iniciado, que pode ser um ou mais filtros, uma ou mais bases de dados, uma ou mais pastas, um ou mais modelos e um ou mais objetos selecionados. Na versão 7.x todos os contextos possuem a mesma extensão de arquivo, e existe a possibilidade de escolha de mais de um contexto para iniciar o script.

Os relatórios podem ser gerados em diversos formatos, de acordo com as opções disponibilizadas pelo desenvolvedor do script. As extensões disponíveis são *.doc (documento de texto), *.xls (planilha eletrônica), *.txt (arquivo de texto), *.rtf (arquivo de texto formatado), *.htm (arquivo de página HTML), *.xml (arquivo XML) e *.pdf (documento sem possibilidade de edição).

A geração dos relatórios é feita de forma simples: após selecionados os elementos de onde se deseja criar o relatório, basta clicar com o botão direito sobre os mesmos e selecionar a opção de *report*. Serão exibidos os scripts cujo contexto seja o mesmo dos elementos selecionados. Em seguida é solicitado o nome e o formato do arquivo de saída. Finalmente, é gerado o report.

O *ARIS Business Architect* já vem com uma série de scripts prontos para utilização. Em sua maioria, esses scripts recolhem informações sobre objetos e modelos selecionados, possibilitando o levantamento de informações personalizadas, selecionadas através de janelas exibidas durante a execução. Além desses, estão disponíveis alguns scripts de auditoria, como usuários e suas permissões para grupos selecionados, últimas alterações realizadas, objetos com mesmo nome, dentre outras.

Capítulo 4 - Proposta de apoio computacional para a aplicação do método Mac Knight sobre a metodologia ARIS

Neste capítulo será apresentado como foi concebida a customização da metodologia ARIS para oferecer apoio computacional à execução do método Mac Knight.

4.1 A concepção da aplicação do método

. A implementação segue as mesmas etapas e passos do método, a fim de produzir os mesmos resultados ao final de cada etapa. Para promover a implementação, foram elicitados um conjunto de requisitos de apoio ao método. A lista de requisitos e a maneira como foram implementados é apresentada no anexo I. Da lista de requisitos levantada para a implementação do método, todos os requisitos foram considerados viáveis de serem implementados. Ocorreram apenas alterações em formatos de documento de saída e na nomenclatura de alguns itens, buscando adequar o modelo à metodologia ARIS e à respectiva nomenclatura.

Em seguida, foi criado o modelo de dados, que complementa o anexo I, mostrando como os novos objetos criados se relacionariam. A criação da ferramenta de apoio computacional seguiu o processo de desenvolvimento interativo e incremental, e foi sendo atualizado à medida que partes da ferramenta eram finalizadas. Como no ARIS script a programação é estruturada e no contexto do projeto visava apenas ~~a~~à geração de relatórios, os testes foram feitos executando os scripts, e verificando o que deveria ser alterado para que funcionasse corretamente.

4.2 Premissas para utilização do método

Durante a customização para implementação do método na ferramenta, foi necessária a criação de algumas premissas para o processo de elicitação dos requisitos, atendendo à dinâmica de funcionamento da ferramenta.

A primeira premissa surgiu para atender à primeira etapa do método, quando deve ser identificado o domínio do problema, selecionando os objetos no contexto

da solicitação. Verificou-se, então, a necessidade de um local onde pudesse ser registrado se o processo ou a atividade em questão estariam no escopo da demanda atual. Originalmente a proposta foi a criação de um atributo nos objetos, onde seria informado o nome do sistema que está sendo desenvolvido. Entretanto, como o nome do sistema pode ainda não ter sido definido, optou-se pela alteração da propriedade para um *check box*, que deve ser marcado quando um objeto for considerado dentro do contexto da solicitação. Dessa forma, apesar do sistema manter histórico de levantamentos anteriores, os relatórios são gerados apenas para os objetos marcados no contexto da solicitação atual.

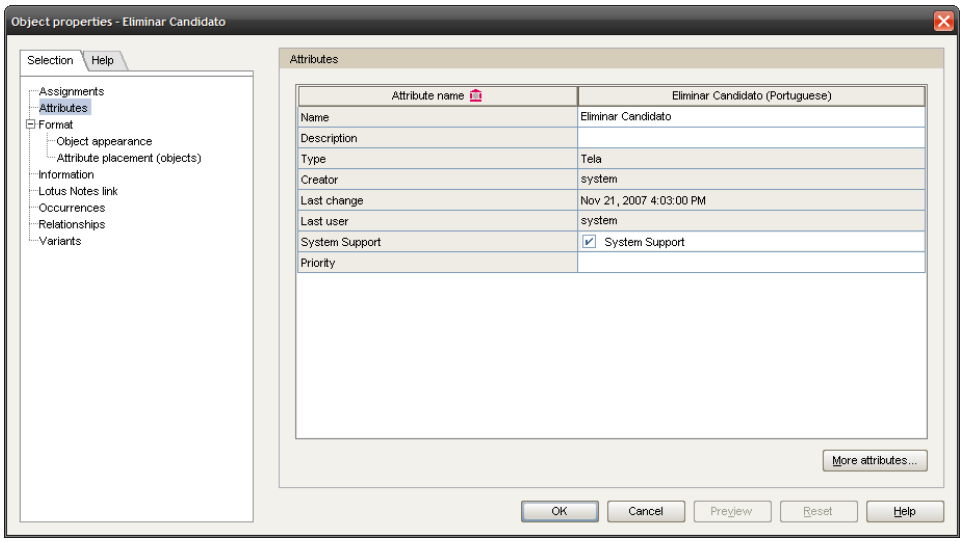


Figura 4.1 Exemplo de objeto com a propriedade System Support marcada

A segunda premissa surge quando se decidiu pela criação de um diagrama único para manter as informações de todos os requisitos de sistemas de todos os levantamentos realizados. Foi criado então, através da alteração do filtro, o diagrama de requisitos (*requirements diagram*). Na estrutura de pastas criada para a modelagem de negócio, recomenda-se a criação de apenas um diagrama, que ~~denomina-se~~se denomina *mapa de requisitos*, sendo possível através dele manter a rastreabilidade entre necessidades, requisitos funcionais e não funcionais e sistemas.

A figura 4.2 ilustra o mapa de requisitos para o primeiro levantamento feito na modelagem. Nele, podemos ver no alto as necessidades; no meio, os requisitos funcionais e não funcionais que surgem a partir das necessidades; e em baixo o

sistema que será desenvolvido, ligado aos requisitos que deverá atender. Caso a modelagem já tivesse sido utilizada para um levantamento anterior, teríamos outras necessidades, que poderiam resultar tanto nos requisitos já existentes como em novos requisitos, e que estariam relacionados com a representação de outro sistema.

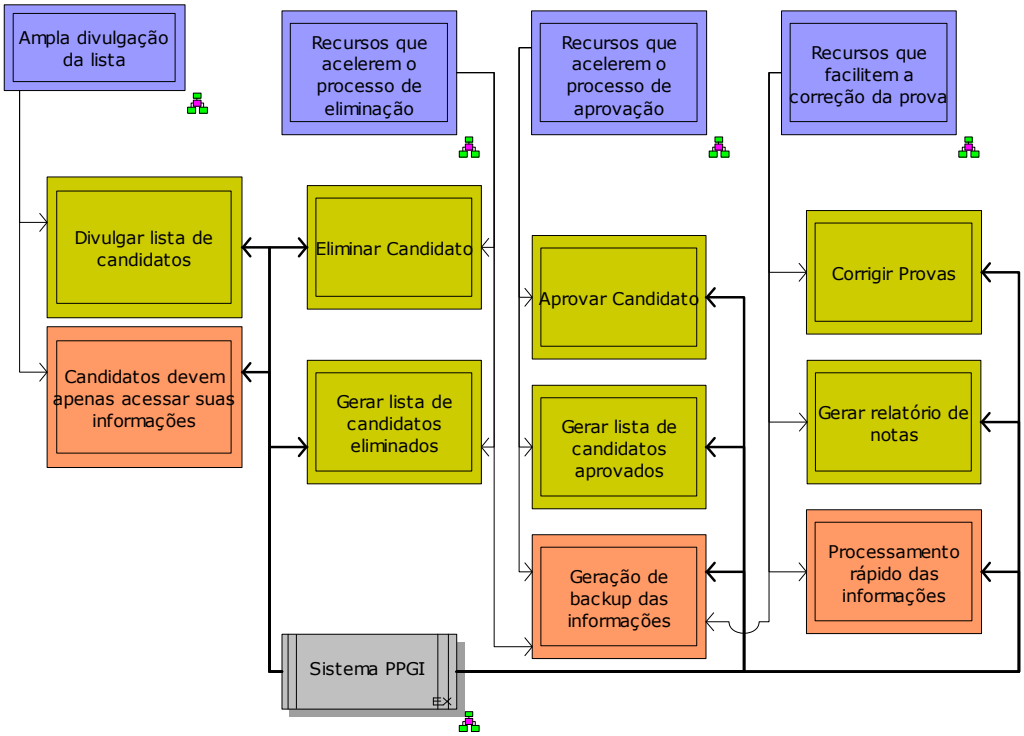
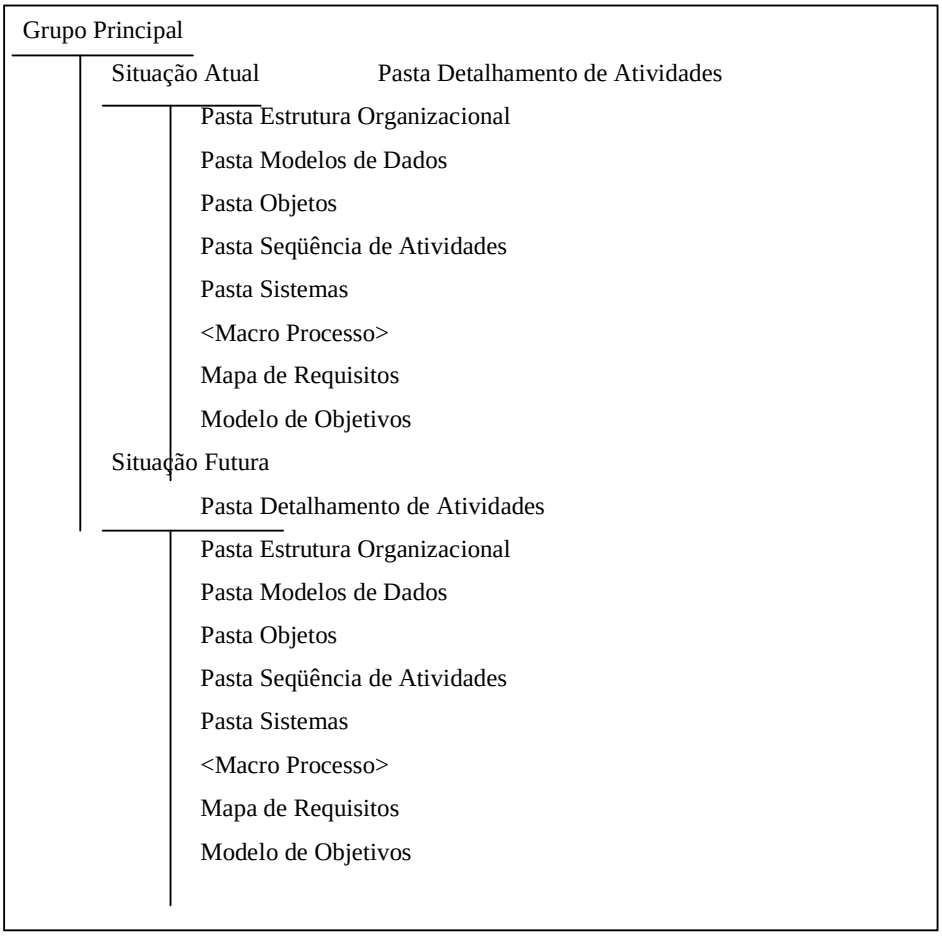


Figura 4.2 Exemplo de mapa de requisitos, com requisitos, necessidades e o sistema a ser atendido

Finalmente, a última premissa surge da necessidade de um padrão de organização da modelagem para que esta fique mais clara e intuitiva, além de facilitar a execução dos scripts de geração de relatórios. Foi proposta uma estrutura de organização dos modelos e objetos em grupos, apresentada abaixo. A execução dos scripts não exige que a estrutura seja seguida, mas a mesma é aconselhada, pois foi desenvolvida com o objetivo de isolar os diagramas dos objetos, além de isolar os níveis de detalhamento dos modelos.



A proposta é criar, a partir de uma pasta raiz, denominada ‘Grupo Principal’, toda a estrutura de modelos e objetos. Abaixo da pasta raiz, podem ser criadas as pastas ‘Situação Atual’ e ‘Situação Futura’, que representam, respectivamente, como o negócio da organização se comporta atualmente (*as-is*) ou como poderia se comportar, caso fossem feitas mudanças (*to-be*).

Dentro de cada uma destas situações, devem ser criadas as seguintes pastas: Detalhamento das atividades – onde ficarão armazenados os FADs e informações de detalhamento dos processos; Estrutura Organizacional – onde ficarão armazenados os diagramas com informações sobre a estrutura organizacional; Modelos de Dados – onde serão criados os modelos de dados; Objetos – onde ficarão armazenados todos os objetos criados na modelagem; Seqüência de Atividades – onde ficarão informações sobre o detalhamento dos processos, como os eEPCs e VACs de

detalhamento; e opcionalmente a pasta Sistemas, onde ficarão armazenados os modelos de detalhamento dos sistemas.

Dentro de cada pasta situação, serão criados obrigatoriamente: um diagrama do tipo VAC, que conterá o fluxo principal do processo, ou apenas os processos em alto nível executados na realidade modelada; o Mapa de Requisitos, onde serão mapeados as necessidades, os requisitos funcionais e, de modo opcional, os requisitos não–funcionais, além dos sistemas que atendem estes requisitos; além do Modelo de Objetivos, que define os principais objetivos dos processos

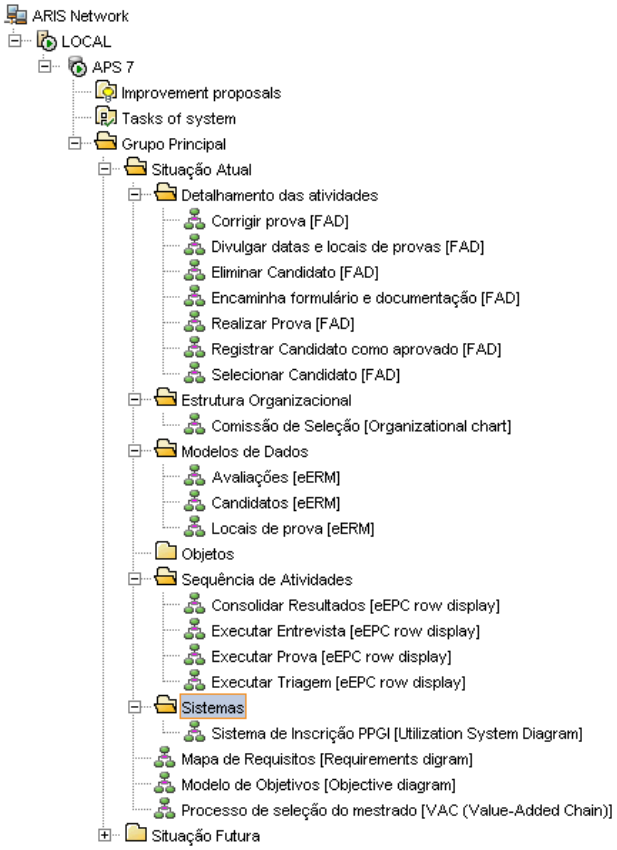


Figura 4.4 Exemplo de estrutura de organização de diretórios recomendada

4.3 A implementação do método

A implementação do método foi constituída em algumas fases, a saber: criação de novos símbolos, sobrecarga de objetos existentes culminando na criação de novos objetos, sobrecarga de tipos de modelo resultando em novos modelos, criação de novas possibilidades de relacionamento entre objetos e modelos, criação de filtro e *template* e criação dos scripts para geração dos relatórios.

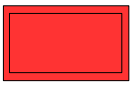
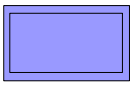
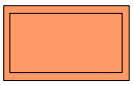
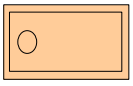
Todas as alterações realizadas foram definidas em Inglês, já que este é o idioma original da ferramenta. Entretanto, foi configurado adicionalmente o nome em português, para que os nomes todas as alterações apareçam de acordo com o idioma considerado na modelagem.

4.3.1 Criação de Novos Símbolos

A adaptação do Método Mac Knight para utilização da ferramenta demandou a criação de novos símbolos, já que alguns dos objetos necessários não tinham correspondentes na lista de objetos ARIS.

Foi utilizada então uma ferramenta da IDS Scheer para criação de símbolos, o *ARIS Symbol Editor*. Através desta ferramenta, é possível a criação de novos símbolos para utilização, que são exportados para *.wmf ou para uma extensão própria, e na seqüência importadas para o *ARIS Business Architecht*.

Desta forma, foram criados cinco novos símbolos, listados na tabela abaixo: necessidade (*need*), dificuldade (*difficult*), requisito funcional (*functional requirement*), requisito não-funcional (*non-functional requirement*), e usuário do sistema (*user*).

Nome	**Semântica**	**Sintaxe**
Dificuldade (criado no projeto)	Dificuldade para a execução de uma atividade	
Necessidade (criado no projeto)	Requisitos em alto nível, provenientes do negócio que irão definir ou restringir aspectos dos sistemas de informação.	
Requisito Funcional (criado no projeto)	Derivação das necessidades, que são reflexos de necessidades da organização que levaram ao desenvolvimento do sistema	
Requisito não-funcional (criado no projeto)	Derivação das necessidades, que são reflexos de restrições no desenvolvimento das atividades relacionadas ao sistema.	
Usuário do sistema (criado no projeto)	Um usuário que utilizará o sistema que está em desenvolvimento ou já está desenvolvido	

Entretanto, apenas a criação dos símbolos não foi suficiente, sendo necessária a alteração do filtro para definir seu pleno funcionamento, que será tratada adiante.

4.3.2 Criação de Novos Tipos de Modelo

Como a maioria dos tipos de modelo existente no padrão ARIS atendia às necessidades, foram necessárias poucas alterações. Inicialmente o objetivo era a criação destes diagramas, mas como a ferramenta só permite a sobrecarga de novos tipos, baseados em modelos já existentes, foi necessária uma pesquisa para descobrir qual dos existentes seria o melhor para atender às necessidades existentes.

Dessa forma, foram criados: o Diagrama de Relacionamentos (*requirements diagram*) e o Diagrama de utilização do Sistema (*Utilization System diagram*).

4.3.3 Criação de novos Atributos de Objetos

Para que a adaptação do método fosse bem-sucedida, foi necessária a criação de algumas propriedades no ARIS. Entretanto, assim como objetos e modelos, o ARIS só permite a sobrecarga de propriedades já existentes. Nesse sentido, a ferramenta disponibiliza uma série de atributos livres para que sejam personalizados, denominados *free attributes*. Tais atributos ainda têm um tipo padrão: texto, booleano, data, dentre outros. Para a adequação ao trabalho, foram criadas as seguintes propriedades, e seus respectivos tipos: *Causes e Effects*, do tipo texto, e associados ao objeto dificuldade; *Permission*, do tipo texto, associado ao objeto usuário do sistema; *Priority*, do tipo texto, associado aos objetos referentes a requisito; *Problem e Recommendation*, do tipo texto, e relacionados a todos os objetos que possam sofrer alterações, após a definição do novo sistema; e *System Support*, do tipo booleano, aplicável a todos os objetos para identificar quando o mesmo for definido como no contexto da solicitação.

4.3.4 Criação de Relacionamentos

Novamente a estrutura padrão do ARIS foi praticamente suficiente para adequá-lo ao método. Assim, foi necessária a criação de apenas alguns relacionamentos.

Entretanto, como o ARIS não permite a criação, foi pensada a possibilidade de sobrecarga de relacionamentos, como ocorreu com símbolos e tipos de modelo. Mas como só é possível alterar a definição dos relacionamentos já existentes no ARIS, foi necessária então a utilização de objetos que ainda não tivessem sido sobrecarregados ou que não sejam freqüentemente utilizados nas modelagens – pois seus relacionamentos seriam alterados -, além de permitir que os relacionamentos desse novo objeto com os já utilizados fizessem sentido.

Dessa forma, o símbolo dificuldade teve de ser alterado. Os outros símbolos nos quais os demais se basearam tinham relacionamentos que faziam sentido – dada a similaridade com os símbolos nos quais se basearam, não sendo necessária sua alteração.

4.3.5 Criação de Filtro Metodológico

A criação do filtro metodológico foi a tarefa que demandou maior esforço, pois é nele que estão concentradas a maioria das informações relacionadas a modelos, objetos, relacionamentos e propriedades disponíveis para utilização.

Para criar um novo filtro, utiliza-se a console de administração do *ARIS Business Architect*. Para a criação de um novo filtro é necessário um processo com treze passos, onde são configuradas informações de objetos, símbolos, relacionamentos, atributos, dentre outras informações.

Alguns dos itens permaneceram com a configuração padrão, pois não necessitavam de configurações específicas para a implementação do método, uma vez que tratam da ordem de exibição de objetos e atributos.

4.3.6 Criação de *Template*

O *template* define configurações de exibição dos objetos e modelos, como cor de fundo de objetos e modelos, tamanho de objetos e disposição de propriedades deste.

Como não havia configurações específicas para os objetos criados ou informações pertinentes para o projeto, o filtro foi configurado de maneira semelhante aos já disponibilizados pela ferramenta.

4.3.7 Criação dos Scripts

Durante o levantamento de requisitos para a customização da ferramenta, verificou-se que a melhor maneira de produzir os resultados de cada etapa seria através da criação de scripts que gerassem relatórios com as informações solicitadas.

Além disso, foi cogitada a hipótese de automação de alguns passos do método através do desenvolvimento de scripts. Uma destas tarefas seria a criação automática de requisitos funcionais a partir das necessidades existentes, mas como não seria possível determinar se uma necessidade seria transformada em apenas um requisito, ou ainda se deveria ser de fato considerada, a tarefa permaneceu sendo desenvolvida pelos engenheiros de software.

Outra proposta que não foi desenvolvida é a criação automática de usuários do sistema baseado nos papéis que executam as atividades modeladas, na medida em que podem existir apenas grupos na modelagem, ou ainda um papel existente na modelagem pode dar origem a mais de um usuário, ou mesmo não ser necessária a criação de usuário, nos casos onde o sistema passará a realizar a atividade.

Desta forma, foram demandados os seguintes scripts:

- **Relatório de Processos e Objetivos Envolvidos:** script que lista informações de cada processo ou atividade no contexto da solicitação, seus objetivos e informações de seus modelos relacionados;
- **Relatório Objetos Organizacionais envolvidos:** script que lista informações do(s) executante(s) de cada atividade dentro do contexto da solicitação cujo relacionamento com a atividade seja do tipo "*Carries out*";
- **Relatório de Dificuldades:** script que gera lista de dificuldades por atividade e lista completa de dificuldades marcadas, com suas causas;
- **Relatório de Necessidades:** script que lista cada processo, suas dificuldades e necessidades; uma lista de necessidade por atividade e uma lista detalhada de necessidades, com impacto e prioridade no atendimento;
- **Relatório de Requisitos Funcionais e Relatório de Requisitos não funcionais:** scripts que listam informações sobre os requisitos

funcionais e não funcionais associados às necessidades que atentem às atividades envolvidas, respectivamente;

- **Relatório de Requisitos por Necessidade:** script que lista requisitos funcionais e não funcionais por necessidade e atividade;
- **Relatório de Informações do Sistema**: script que lista nome e descrição do sistema a ser desenvolvido, seu diagrama de utilização, a lista de usuários e permissões de acesso, as interações com outros sistemas e as informações dos clusters, listando os diagramas de entidades e relacionamentos, quando existirem;
- **Relatório de Impactos da Criação do novo sistema:** script que lista o impacto da criação do sistema nas atividades, nos papéis existentes e nos sistemas que interagem com o sistema que será criado;
- **Relatório de Sistemas de Apoio:** script que gera lista de sistemas que interagem com o sistema a ser desenvolvido ou que apóiam as atividades deste;
- **Relatório de Geração do DRS:** script que gera o documento de requisitos de software a partir das informações geradas anteriormente;
- **Apagar Propriedade *System Support*:** script que altera a propriedade *system support* de todos os objetos marcados, deixando-os desmarcados ao final da execução.

Como alguns desses scripts compartilhavam métodos ou tinham métodos que faziam a mesma tarefa com nomes diferentes, foram criadas bibliotecas com estas funções, de forma a tornar o código mais claro e padronizado. Assim, foram criadas bibliotecas, a saber: *libArvoreN.js* – funções para montar e percorrer a árvore de informações geradas; *libPG2* – funções genéricas, como procurar objetos em listas, verificar repetições de objetos, dentre outras; e *style tools.js* – funções referentes à formatação de estilos de texto e inicialização do arquivo de saída.

Capítulo 5 - A Execução do Método

Neste capítulo será apresentada a execução do método Mac Knight utilizando o apoio computacional customizado na ferramenta *ARIS Business Architecht,* ilustrado com o estudo de caso realizado no contexto da disciplina de graduação em Análise e Projeto de Sistemas da UNIRIO, onde foi modelado o processo de seleção do Programa de Pós-Graduação em Sistemas de Informação (PPGI) e aplicado o método a fim de obter a lista de requisitos de software em alto nível, buscando a avaliação da implementação do método na ferramenta.

Por se tratar de um processo obtido como parte dos requisitos para uma disciplina, a modelagem do processo resultante é simples, mas suficiente para validar a utilização do método, avaliando se os requisitos elicitados estariam de acordo com as verdadeiras necessidades existentes.

5.1 Contexto do Estudo de Caso

O Programa de Pós-Graduação em Informática da UNIRIO oferece cursos de mestrado em informática, especificamente em Sistemas de Informação, tornando este programa pioneiro nesta linha em universidades brasileiras.

Segundo o seu site, o programa se destaca pelo fato de apresentar como área de concentração a pesquisa em Sistemas de Informação, e tem como objetivos: aprofundar os conhecimentos científicos e técnico-profissionais de seus alunos na área de Sistemas de Informação; formar profissionais com conhecimento técnico e organizacional para a modelagem, o desenvolvimento, a seleção, a implantação e a gestão de Sistemas de Informação em empresas; possibilitar aos seus alunos o desenvolvimento de habilidades para a pesquisa e para a docência no ensino superior na área de Sistemas de Informação, e; desenvolver pesquisas de ponta na área de Sistemas de Informação que possam contribuir para o desenvolvimento social e econômico da região em que a UNIRIO se insere.

O processo de seleção do programa consta de fases, a saber: análise de documentação, provas de inglês e entrevistas com os professores da linha de pesquisa escolhida. Todas as tarefas são realizadas manualmente, e apenas os resultados de

cada etapa são transferidos para meio eletrônico para divulgação no site do programa.

A proposta do trabalho dos alunos que desenvolveram o estudo de caso foi de aumentar a automação das etapas do processo de seleção. Assim, a solicitação pode ser generalizada como ***automatizar o processo de avaliação dos candidatos a uma vaga no PPGI***.

5.2 Execução do método Mac Knight na ferramenta

Neste tópico serão apresentados as etapas e passos da execução do método customizado no ARIS, segundo o processo de derivação utilizado na proposta original. Assim como no trabalho original, a proposta deste também está dividida em duas etapas - visão geral do problema e visão geral da solução, com etapas e passos.

5.2.1 Fase 1: Visão geral do problema

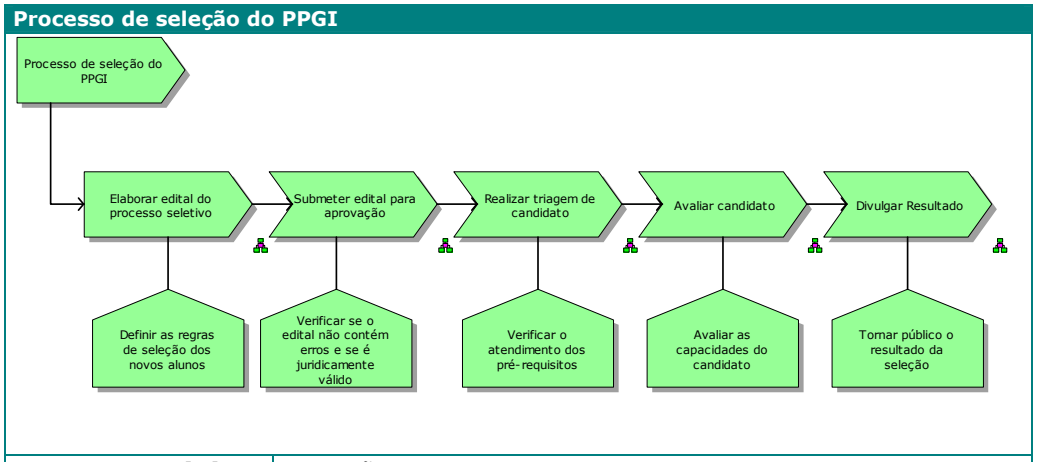
5.2.1.1 Primeira etapa: Identificar o Contexto da Solicitação

O objetivo desta etapa é delimitar a área do negócio envolvida com a solicitação. Mac Knight (MAC KNIGHT, 2004) sugere que sejam realizadas reuniões com os solicitantes, para que a área do negócio envolvida fique clara a todos.

Esta etapa utiliza o modelo de processos e atividades criado na ferramenta *ARIS Business Architect* como entrada, resultando em um relatório com a lista de processos e atividades no contexto da solicitação, bem como os objetivos destes.

No primeiro passo, o modelo de processos deve ser analisado, em reunião com os solicitantes, visando identificar os processos de alguma forma envolvidos com a solicitação, partindo do VAC mais genérico, e caso existam subprocessos, examinar até o mais específico. Para cada objeto do tipo processo que for considerado dentro do contexto da solicitação, seu atributo *system support* deve ser marcado.

Para identificar os processos de negócio envolvidos com a solicitação buscou-se, no modelo de processos, aqueles que deveriam ser automatizados. Analisando-se os grupos de processos e suas descrições chegou-se no processo “Avaliar Candidato”. Assim, este teve a propriedade *system support* marcada. O modelo de processos é mostrado na tabela abaixo.



Processo / Atividade	**Descrição**
Avaliar candidato	O processo avalia diversas habilidades e competências dos candidatos, a fim de verificar aqueles que têm o perfil de aluno desejado. A etapa é composta de prova de idioma (inglês) e entrevista com a comissão de seleção

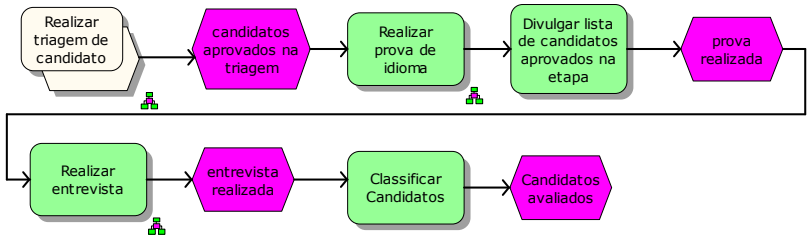
Tabela 2 Modelo de Processos "Processo de seleção do PPGI" e lista com nome e descrição do processo selecionado

Os demais processos do diagrama dizem respeito à publicação do edital, a triagem dos candidatos e à divulgação do resultado. Como a solicitação tratava apenas da avaliação dos candidatos, os demais processos não foram considerados. Esta decisão foi tomada através do entendimento sobre a finalidade de cada processo, o que foi obtido pela análise do modelo de processo.

No segundo passo, analisa-se cada processo identificado individualmente para definir melhor o domínio do desenvolvimento do sistema. Por isso, devem-se analisar as atividades de cada processo também com o objetivo de verificar se pertencem ao contexto da solicitação. Assim, para cada objeto do tipo atividade que for considerado dentro do contexto da solicitação, seu atributo *system support* deve ser marcado.

O processo "Avaliar Candidato", ilustrado pela tabela abaixo, é responsável pela etapa de avaliação dos candidatos. Analisando-o, identificou-se a atividade que corresponde à prova de idioma. As outras atividades correspondem às etapas de divulgação dos aprovados na prova de idioma, realização de entrevistas e classificação dos candidatos, mas que não foram consideradas no escopo da solicitação. Como no passo anterior, apenas a atividade selecionada teve a propriedade *system support* marcada.

Avaliar candidato

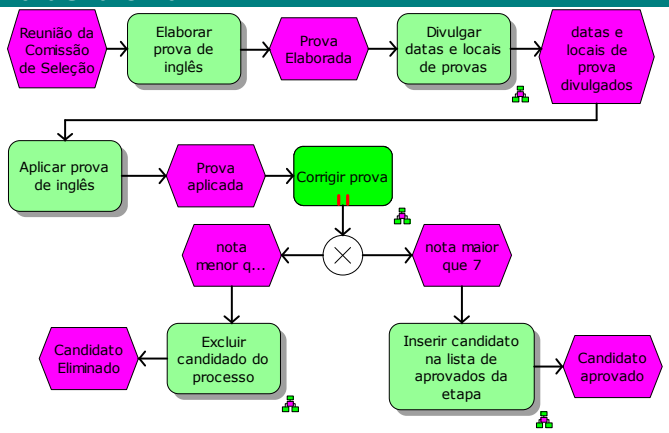


Processo / Atividade	Descrição
Realizar prova de idioma	Etapa que avalia o conhecimento de língua estrangeira dos candidatos através da aplicação de uma prova

Tabela 3 Subprocesso "Avaliar Candidato" e nome e descrição da atividade selecionada

Analisando-se então o modelo de atividades da atividade "Realizar prova de idioma", mostrado na tabela abaixo, definiram-se quais atividades fazem parte do contexto da solicitação. Para isso, avaliaram-se quais atividades poderiam ser automatizadas pelo novo sistema. Foram consideradas as atividades "Divulgar datas e locais de provas", "Excluir candidato do processo", "Corrigir prova" e "Inserir candidato na lista de aprovados da etapa", que também tiveram sua propriedade *system support* marcada.

Realizar prova de idioma



Processo / Atividade	Descrição
Divulgar datas e locais de provas	Atividade que torna público a data e local de realização da prova para todos os candidatos aprovados na fase de triagem
Excluir candidato do processo	Atividade de exclusão do candidato do processo de seleção do PPGI, finalizando o processo dos candidatos eliminados
Corrigir prova	Atividade de correção da prova do candidato, executada para cada candidato inscrito. Logo, se repete tantas vezes o número de candidatos inscritos.
Inserir candidato na lista de aprovados da etapa	Atividade que consiste na inserção do candidato na lista de candidatos aprovados na etapa

Tabela 4 Modelo de atividades da atividade "Realizar prova de idioma" e lista com nome e descrição das atividades selecionadas

Abaixo seguem as tabelas com os detalhamentos das atividades, com seu nome e descrição. É válido ressaltar nesses diagramas a aparição do grupo "Comissão de Seleção", conforme visto nas figuras, além dos documentos gerados. As atividades "Elaborar prova de inglês" e "Aplicar prova de Inglês" não foram consideradas, pois precisam ser executadas manualmente; além da primeira já ser apoiada por ferramentas computacionais, não sendo conveniente a sua realização no sistema.

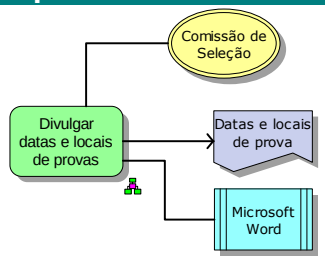
Divulgar datas e locais de provas	
	
Processo / Atividade	**Descrição**
Divulgar datas e locais de provas	Atividade que torna público a data e local de realização da prova para todos os candidatos aprovados na fase de triagem

Tabela 5 Detalhamento da atividade "Divulgar datas e locais de provas" e lista com nome e descrição da atividade.

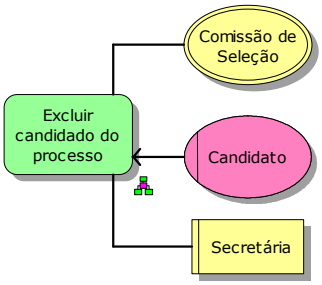
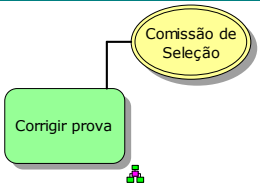
Excluir candidato do processo	
	
Processo / Atividade	**Descrição**
Excluir candidato do processo	Atividade de exclusão do candidato do processo de seleção do PPGI, finalizando o processo dos candidatos eliminados

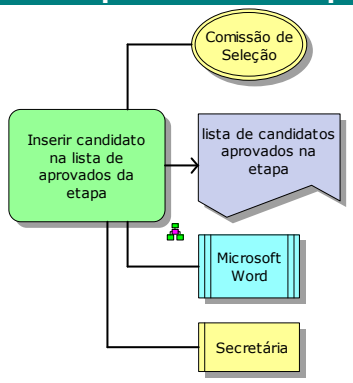
Tabela 6 Detalhamento da atividade "Excluir candidato do processo" e lista com nome e descrição da atividade.

Corrigir prova	
	
Processo / Atividade	**Descrição**

Atividade	
Corrigir prova	Atividade de correção da prova do candidato, executada para cada candidato inscrito. Logo, se repete tantas vezes o número de candidatos inscritos.

Tabela 7 Detalhamento da atividade "Excluir candidato do processo" e lista com nome e descrição da atividade.

Inserir candidato na lista de aprovados da etapa



Processo / Atividade	Descrição
Inserir candidato na lista de aprovados da etapa	Atividade que consiste na inserção do candidato na lista de candidatos aprovados na etapa

Tabela 8 Detalhamento da atividade "Excluir candidato do processo" e lista com nome e descrição da atividade.

Ao final, o script *Relatório de Processos e Objetivos Envolvidos* deve ser executado, a fim de listar os processos e atividades considerados dentro do contexto da solicitação. Além disso, o script lista os objetivos relacionados a cada processo e atividade marcados.

No processo de seleção do PPGI, a execução do script gerou todas as tabelas deste item – com imagem e informações sobre os processos e atividades selecionados, além da lista de objetivos associados ao processo, cuja tabela está listada abaixo.

Processo/ Atividade	Objetivo	Descrição
Avaliar candidato	Avaliar as capacidades do candidato	Avaliando as capacidades do candidato é possível perceber aqueles cujo perfil e interesses estão de acordo com a linha do orientador.

Tabela 9 Detalhamento da atividade "Excluir candidato do processo" e lista com nome e descrição da atividade.

5.2.1.2 Segunda etapa: Identificar Necessidades dos Processos

O objetivo dessa etapa é especificar as dificuldades existentes na execução das atividades, e as conseqüentes necessidades existentes na execução das atividades e dos processos no contexto da solicitação.

A etapa tem como entradas a modelagem de processos e atividades, além do modelo de localização e do modelo organizacional. Fornece como resultado um relatório com a lista detalhada de necessidades, a lista necessidades por atividade; além de um relatório com lista de dificuldades detalhadas, e uma lista de dificuldades por atividade, com as respectivas causas

Inicialmente, é necessário consultar os executantes das atividades. Dessa forma, pode ser utilizado o relatório do fim da etapa anterior ou pode ser novamente executado o script *Relatório de Processos e Objetivos Envolvidos*, que gerará a lista de executantes por atividade, cujo relacionamento com a atividade seja do tipo *carries out*.

A execução do script para o processo de seleção resultou na tabela abaixo:

Divulgar datas e locais de provas		
Atividade	**Objeto - Tipo do Objeto**	**Descrição**
Divulgar datas e locais de provas	Comissão de Seleção - Group	Grupo composto por docentes do PPGI, escolhidos através da reunião do colegiado
Excluir candidato do processo		
Atividade	**Objeto - Tipo do Objeto**	**Descrição**
Excluir candidato do processo	Comissão de Seleção - Group	Grupo composto por docentes do PPGI, escolhidos através da reunião do colegiado
Corrigir prova		
Atividade	**Objeto - Tipo do Objeto**	**Descrição**
Corrigir prova	Comissão de Seleção - Group	Grupo composto por docentes do PPGI, escolhidos através da reunião do colegiado
Inserir candidato na lista de aprovados da etapa		
Atividade	**Objeto - Tipo do Objeto**	**Descrição**
Inserir candidato na lista de aprovados da etapa	Comissão de Seleção - Group	Grupo composto por docentes do PPGI, escolhidos através da reunião do colegiado

Tabela 10 Objetos organizacionais por atividade, com descrição dos mesmos

No primeiro passo são identificadas as dificuldades existentes em cada atividade dos processos. As dificuldades são quaisquer obstáculos que as pessoas devem transpor para executar as atividades, daí a necessidade em consultar os executantes de cada atividade, pois são eles que possuem a prática do que é feito diariamente. Para cada dificuldade encontrada deve ser criado um objeto *difficult*,

associado à atividade onde houver dificuldade. Caso a dificuldade esteja associada a todo um processo, o objeto que o representa deve conter um *assigment* para um diagrama do tipo FAD, e neste deve ser associado um objeto do tipo *difficult* com o processo – sendo este representado por um objeto do tipo *function* no diagrama criado.

No estudo de caso, as dificuldades existentes no processo foram descritas a partir de reuniões com o grupo que realizou o trabalho. Foi criado então um objeto do tipo *difficult* para cada dificuldade, nos *FADs* referentes a cada atividade selecionada, já que não existem modelos de detalhamento para processos, e que todas as atividades já possuíam modelos de detalhamento.

No segundo passo, as dificuldades são analisadas de forma a entender a causa de cada uma e, assim, identificar as necessidades do negócio que estão por trás delas. Deve-se analisar se a atividade que contém uma dificuldade está sendo executada da forma que deveria, se possui as informações suficientes para ser executada, etc. – de forma a concluir qual é a necessidade que não está sendo suprida no processo e, por isso, fez a dificuldade surgir. Assim, as causas que levam à ocorrência da dificuldade devem ser armazenadas no atributo *causes* da respectiva dificuldade.

No final desta etapa, o script *Relatório de Dificuldades* deve ser executado a fim de gerar a lista as dificuldades criadas por processo ou atividade, com suas respectivas causas.

Assim, no caso do processo seletivo do PPGI, foram levantadas as causas de cada uma das dificuldades encontradas, preenchendo os respectivos atributos de cada uma. Feito isso, foi executado o script *Relatório de Dificuldades*, que gerou o relatório das dificuldades, chegando aos resultados apresentadas na tabela abaixo:

Nome	Descrição	Causa(s)
Informar todos os candidatos	A grande quantidade de candidatos pode impedir que todos tomem conhecimento das datas e locais de provas	Dificuldade de comunicação com alguns candidatos
Grande número de candidatos para lançar eliminação	A grande quantidade de candidatos que são eliminados pode causar demora na conclusão das etapas do processo, atrasando o cronograma	Número grande de candidatos para processamento manual
Curto espaço de tempo para	O calendário apertado dificulta a correção das	Calendário do processo de seleção tem etapas de

correção	provas, exigindo esforço excessivo dos professores	curta duração
Número grande de provas para correção	O grande número de candidatos pode atrasar a correção das provas e o conseqüente atraso no cronograma	Número grande de candidatos para processamento manual
Grande número de candidatos para lançar aprovação	O grande número de candidatos demanda tempo e esforço da comissão de seleção para geração da lista nominal de aprovados da etapa	Número grande de candidatos para processamento manual

Tabela 11 Descrição das dificuldades

No terceiro passo, são identificadas as necessidades do negócio que não estão sendo supridas atualmente. Para isso, deve-se partir da lista de dificuldades encontradas, e com o auxilio da verificação do fluxo de atividades, verificar onde existem necessidades de negócio - criando-se um objeto do tipo *need* para cada uma, e relacionado com as respectivas atividade e dificuldade que o originaram.

Existe ainda outro tipo de necessidade cujas conseqüências não são dificuldades, mas resultados insatisfatórios gerados pelos processos. Dessa forma, no quarto passo cada processo deve analisado novamente, avaliando se as atividades são executadas de forma adequada e se produzem resultados satisfatórios. Percebendo questões que podem ser melhoradas e/ou problemas que podem estar gerando impactos em clientes ou em outros processos; ou ainda critérios de segurança, velocidade de processamento, ou padrões adotados pela organização, também deve ser criado um objeto do tipo *need*, e relacionado à atividade onde foi identificada a necessidade.

No caso do PPGI, para cada dificuldade e suas causas, foram identificadas as necessidades que deviam ser atendidas para que se eliminassem as dificuldades, representadas através da criação de um objeto do tipo *need* no detalhamento da atividade – FAD. Os diagramas atualizados com os novos objetos encontram-se abaixo:

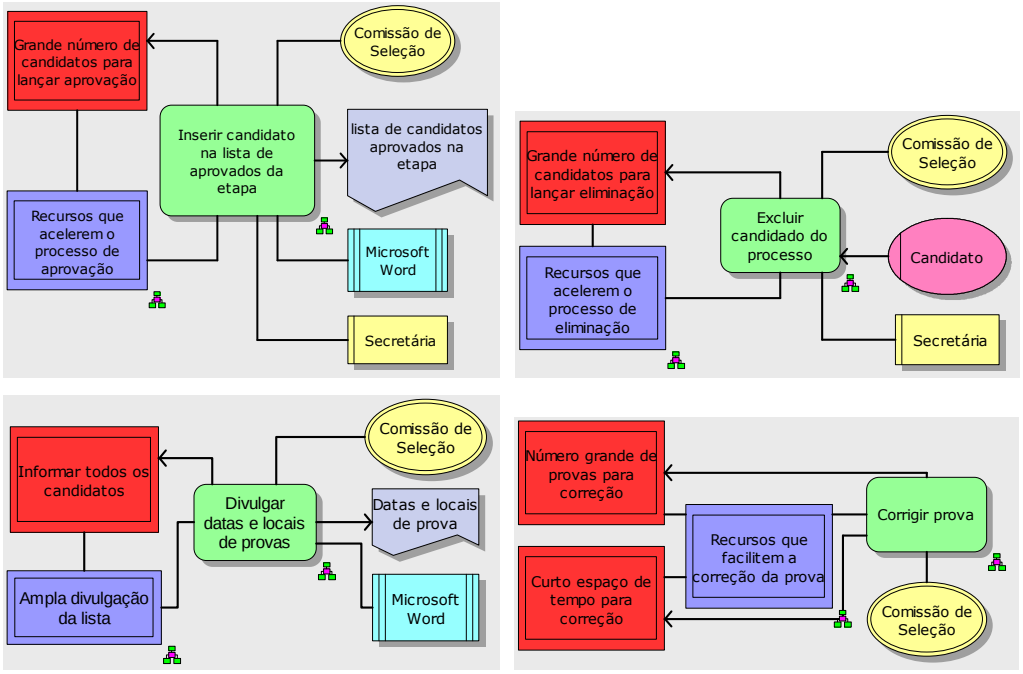


Figura 5.1 Da esquerda para a direita e de cima para baixo: detalhamento do processo 'Inserir candidato na lista de aprovados da etapa', 'Excluir candidato do processo', 'Divulgar datas e locais de provas' e 'Corrigir prova'

Finalmente, o script *Relatório de Necessidades* deve ser executado a partir do VAC principal da modelagem, a fim de gerar então o relatório de necessidades por processos e atividades no contexto da solicitação.

O resultado da execução do script no estudo de caso está listado na tabela abaixo. É importante notar o relacionamento existente entre cada necessidade listada com a dificuldade que a originou e com a atividade que possui essa dificuldade. Esse relacionamento está representado através das linhas da tabela, e graficamente, através das ligações entre os objetos.

Processo(s)	Dificuldade(s)	Necessidade(s)
Divulgar datas e locais de provas	Informar todos os candidatos	Ampla divulgação da lista
Excluir candidato do processo	Grande número de candidatos para lançar eliminação	Recursos que acelerem o processo de eliminação
Corrigir prova	Curto espaço de tempo para correção	Recursos que facilitem a correção da prova
	Número grande de provas para correção	Recursos que facilitem a correção da prova
Inserir candidato na lista de aprovados da etapa	Grande número de candidatos para lançar aprovação	Recursos que acelerem o processo de aprovação

Tabela 12 Lista de Necessidades por Dificuldade

5.2.1.3 Terceira etapa: Identificar as conseqüências de cada necessidade

O objetivo desta etapa é obter uma visão geral dos impactos das necessidades identificadas. A entrada é a lista de necessidades do negócio, além dos modelos de atividades e de objetivos modelados no ARIS. A saída é o relatório com a lista de necessidades, com o impacto do não-atendimento das mesmas e a prioridade no atendimento de cada uma.

No primeiro passo, devem-se identificar as conseqüências de cada necessidade encontrada na etapa anterior, isto é, a forma com que a atual falta do atendimento de cada uma afeta as tarefas que são realizadas. Para isso, deve-se analisar como a existência da necessidade afeta o resultado de cada atividade ou do processo como um todo, por exemplo, atrasa ou inviabiliza uma atividade, possibilita que um resultado errado seja gerado, expõe informações a pessoas indevidas, etc. As conseqüências encontradas para cada necessidade devem ser armazenadas na propriedade *impact* do objeto que representa este necessidade.

Assim, identificaram-se as conseqüências de cada necessidade que não estava sendo suprida no processo de seleção do PPGI. Para isso, avaliou-se como cada necessidade afetava as atividades e o processo como um todo, sendo preenchido o atributo *impact* de cada necessidade com a informação

O segundo passo consiste em identificar os objetivos de negócio e os processos atingidos pelas conseqüências das necessidades. Como a informação já está disponível no relatório de processos e atividades no contexto da solicitação, não há necessidade de execução de nenhuma tarefa específica neste passo. Caso a informação não esteja disponível, o script *Relatório de Processos e Objetivos Envolvidos* deve ser novamente executado.

No terceiro passo, o script *Relatório de Necessidades* é executado para gerar a lista de dificuldades e necessidades dos processos no domínio da solicitação, além das listas de necessidades por processo e de necessidades e os impactos do não atendimento de cada uma.

No caso do PPGI, o script foi executado, gerando as informações listadas anteriormente. Entretanto, o campo prioridade da lista de necessidades ainda aparece vazio, pois essa informação será preenchida no próximo passo.

No quarto passo, é preciso definir quais necessidades serão atendidas pelo sistema. Analisam-se os impactos causados pelas necessidades existentes através do resultado do passo anterior, de forma que se definam as prioridades de cada uma. Esta informação deve ser armazenada no atributo *priority* de cada objeto necessidade existente. Recomenda-se o uso dos termos ALTA, MÉDIA e BAIXA. Uma vez identificadas as prioridades, é preciso definir o conjunto de necessidades que será atendido. Aquelas que não forem atendidas na solicitação atual devem ter o atributo *priority* vazio, além da propriedade *System Support* desmarcada – sendo assim desconsideradas nos relatórios de necessidades.

Todas as necessidades identificadas no estudo de caso foram atendidas, logo, em nenhuma delas a propriedade *System Support* foi desmarcada. Além disso, como ficou decidido que todas teriam prioridade alta, todas tiveram o campo *priority* preenchidos com o valor 'ALTA'.

Finalmente, é executado novamente o script do passo 2 para obter o relatório com todas as informações completas dos requisitos que serão atendidos na solicitação atual.

Finalmente, o script foi executado novamente, completando a tabela de necessidades, listada abaixo. As demais tabelas encontram-se no anexo II, no documento de requisitos de software gerado.

Nome	Descrição	Impacto	Prioridade
Ampla divulgação da lista	São necessários recursos que permitam e facilitem a ampla divulgação dos resultados	Candidato não ser informado da data e do local da prova	ALTA
Recursos que acelerem o processo de eliminação	São necessários recursos que automatizem o processo, acelerando o processo de eliminação do candidato	Geração de erros durante a correção, atraso na entrega dos resultados	ALTA
Recursos que facilitem a correção da prova	São necessários recursos que facilitem a correção da prova	Atraso na entrega dos resultados, erros na correção das provas	ALTA
Recursos que acelerem o processo de aprovação	São necessários recursos que automatizem o processo, acelerando o processo de criação da lista	Erros durante o lançamento dos resultados	ALTA

Tabela 13 Lista detalhada das necessidades

5.2.2 Fase 2: Visão Geral da Solução

5.2.2.1 Quarta etapa: Identificar Funcionalidades e Restrições do Sistema

O objetivo desta etapa é identificar as principais funcionalidades e restrições - que serão convertidos em requisitos funcionais, e opcionalmente, em não funcionais, de forma que atendam às atividades do processo de negócio e às necessidades identificadas na fase anterior. A entrada é a modelagem realizada no ARIS e o relatório com a lista de necessidades a serem atendidas, obtida na fase anterior. A saída é o relatório de requisitos funcionais e não funcionais, bem como informações detalhadas destes e das atividades e necessidades relacionadas.

Antes do início da etapa, deve-se verificar se já existe um mapa de requisitos na modelagem atual. Nele será representado graficamente o desdobramento das necessidades em requisitos funcionais e não funcionais, bem como quais destes serão atendidos por quais sistemas – caso já tenha sido realizada a elicitação para um sistema utilizando a modelagem anteriormente. Caso ainda não tenha sido criado, um modelo do tipo *Requirements Diagram* deve ser inserido na pasta raiz da modelagem, com o nome de 'Mapa de Requisitos'.

Como esse era o caso da modelagem do PPGI, foi criado um diagrama do tipo *Requirements Diagram*, nomeado 'Mapa de Requisitos'.

No primeiro passo, deve-se analisar cada necessidade de negócio visando identificar o que deve ser estabelecido para que seja suprida. Essa análise deve ser feita considerando-se o que deve ser acrescentado no sistema para que se atenda à necessidade em questão.

Assim, no mapa de requisitos devem ser inseridos todos os objetos necessidade considerados na solicitação atual, copiando cada objeto deste tipo relacionado às atividades e colando no mapa de requisitos. Em seguida, devem ser adicionados os requisitos funcionais e não funcionais, originados a partir das necessidades, e que por essa razão devem ser associados a estes objetos. Vale ressaltar que um requisito já criado pode ser conseqüência de mais de uma necessidade, e que, portanto pode estar relacionado a mais de uma necessidade ou sistema.

Ao final da etapa, a execução dos scripts *Relatório de Requisitos Funcionais e Relatório de Requisitos não funcionais* geram as listas de requisitos funcionais por necessidades e a lista de requisitos não funcionais por necessidade, respectivamente.

Assim, no caso do PPGI foram inseridas no mapa de requisitos todas as necessidades a serem atendidas na solicitação. Em seguida, foram analisadas, a fim de descobrir os requisitos necessários funcionais e não funcionais necessários ao sistema. A execução dos scripts gera as listas de encontram-se abaixo.

Processo	Requisitos Funcionais	Descrição
Divulgar datas e locais de provas	Divulgar da lista de candidatos	A divulgação da lista de locais e provas deve ser feita a todos os candidatos, seja através do envio de emails ou solicitação de contato telefônico
Excluir candidato do processo	Gerar lista de candidatos eliminados	O sistema deve gerar automaticamente a lista de candidatos eliminados a partir das informações sobre o resultado de cada etapa das provas
Excluir candidato do processo	Eliminar Candidato	O sistema deve apresentar mecanismos para conclusão do processo do aluno, mantendo apenas nos registros no banco de dados
Corrigir prova	Gerar relatório de notas	O sistema deve gerar automaticamente o relatório de notas baseado na correção de provas realizada
Corrigir prova	Corrigir Provas	O sistema deve prover soluções automatizadas para correção da prova, preferencialmente através do sistema de leitura óptica
Inserir candidato na lista de aprovados da etapa	Gerar lista de candidatos aprovados	O sistema deve gerar automaticamente a lista de candidatos aprovados a partir das informações sobre o resultado de cada etapa
Inserir candidato na lista de aprovados da etapa	Aprovar Candidato	O sistema deve apresentar mecanismos para conclusão do processo do aluno, possibilitando a troca de informações com o sistema de matrícula da UNIRIO

Tabela 14 Lista de requisitos funcionais

Processo	Requisitos Não funcionais	Descrição
Divulgar datas e locais de provas	Candidatos devem apenas acessar suas informações	Os candidatos devem ter acesso apenas às suas informações cadastrais e aos seus resultados no processo, mantendo o sigilo de informações
Excluir candidato do processo	Geração de backup das informações	O backup deve ser gerado para que nenhuma informação seja perdida em caso de sinistro ou outro problema qualquer
Corrigir prova	Geração de backup das informações	O backup deve ser gerado para que nenhuma informação seja perdida em

		caso de sinistro ou outro problema qualquer
Inserir candidato na lista de aprovados da etapa	Geração de backup das informações	O backup deve ser gerado para que nenhuma informação seja perdida em caso de sinistro ou outro problema qualquer

Tabela 15 Lista de requisitos não funcionais

No segundo passo, a lista de necessidades é novamente avaliada, desta vez em reunião com os gerentes do negócio, e baseado no resultado do relatório do passo anterior, verificar a partir das necessidades existentes outros requisitos que devem ser acrescentados para complementar a lista. Para cada novo requisito, um objeto do tipo *need* deve ser criado e relacionado com a necessidade que o originou.

Finalmente, o script *Relatório de Requisitos por Necessidade* é executado, a fim de gerar a lista completa de requisitos. Assim, é possível manter a informação de quais necessidades do processo geraram quais requisitos funcionais ou não funcionais do sistema.

No segundo passo não foram adicionados novos requisitos na modelagem do estudo de caso. Foi executado então o script, que gerou a tabela abaixo:

Atividades	Necessidades	Requisitos Funcionais e não funcionais
Divulgar datas e locais de provas	Ampla divulgação da lista	Divulgar da lista de candidatos
		Candidatos devem apenas acessar suas informações
Excluir candidato do processo	Recursos que acelerem o processo de eliminação	Gerar lista de candidatos eliminados
		Eliminar Candidato
		Geração de backup das informações
Corrigir prova	Recursos que facilitem a correção da prova	Gerar relatório de notas
		Corrigir Provas
		Geração de backup das informações
Inserir candidato na lista de aprovados da etapa	Recursos que acelerem o processo de aprovação	Gerar lista de candidatos aprovados
		Aprovar Candidato
		Geração de backup das informações

Tabela 16 Lista de requisitos por necessidade e atividade

5.2.2.2 Quinta etapa: Identificar as Fronteiras do Sistema

O objetivo desta etapa é identificar os pontos em que o sistema trocará informações com outros recursos do negócio. É preciso identificar os usuários, outros sistemas e meios de armazenamento que interagirão com o sistema.

Inicialmente, deve ser criado um objeto do tipo *Application System*, com o nome do sistema que está sendo solicitado. Neste objeto, preencher a descrição com as informações iniciais do sistema, como: demandante, descrição em alto nível, dentre outras informações. Deste objeto, clicar com o botão direito sobre o mesmo e selecionar a opção *New... > Assigment*; selecionado a opção *New Model* e o tipo de modelo *Utilization System Diagram* na janela que será aberta. Neste modelo criado estarão todas as informações sobre o sistema que será criado.

Foi inserido no mapa de requisitos do estudo de caso um objeto do tipo *Application System*, nomeado de 'Sistema PPGI', e que representa o sistema que atenderá à solicitação. Em seguida, foi criado um *assigment* deste objeto para um diagrama do tipo *Utilization System Diagram*, que armazenará as informações sobre as fronteiras do novo sistema.

No primeiro passo executa-se o script *Relatório Objetos Organizacionais* para obter a lista de papéis existentes na modelagem. Com essa lista, verificar se as atividades que são executadas por eles passarão a ser totalmente executadas pelo sistema ou serão somente apoiadas pelo sistema. Se forem totalmente executadas pelo sistema, o papel em questão não deverá interagir com o sistema; porém, se forem somente apoiadas, então haverá interação.

Assim, uma vez identificados os papéis que interagirão com o sistema, definem-se os usuários e quais as permissões necessárias a cada um. Para cada usuário identificado, deve ser criado um objeto do tipo *User* no diagrama de utilização do sistema; armazenando as informações de permissões no atributo *Permission*, e; relacionando-o ao objeto que representa o sistema.

A execução do script no estudo de caso gerou a lista de objetos organizacionais que executam as atividades, cujo resultado já foi apresentado na Tabela 10. Com essa lista, foram pensados os usuários que interagiriam com o sistema. Para cada usuário, foi criado um objeto do tipo *User*, onde as informações sobre permissões de acesso foram preenchidas no atributo *Permission.*

No segundo passo, avaliam-se possíveis interações com outros sistemas. Para cada atividade que será atendida, analisar se é executada com apoio de algum outro sistema e se continuará sendo após a implantação do novo sistema. Para cada sistema identificado, avaliar se as informações manipuladas são relevantes, de forma a definir se o novo sistema deverá interagir ou não com o antigo. Em caso positivo, criar um objeto do tipo *System Interface*, nomeando o objeto com o nome do sistema com o qual será realizada a integração no diagrama de utilização do sistema e associando os dois objetos.

O sistema de matrículas da UNIRIO é o único que foi identificado, uma vez que sua função é de matricular os alunos classificados. Assim, foi criado um objeto do tipo System Interface para representá-lo, associando-o com o objeto Sistema PPGI, representando a troca de informações entre os sistemas.

No terceiro passo, analisam-se os acessos das atividades aos meios de armazenamento. Se o meio de armazenamento acessado por uma atividade for computacional, deverá haver interação deste com o sistema. E, se for um meio de armazenamento manual e suas informações forem relevantes ao sistema, então o trabalho de desenvolver o sistema englobará o trabalho de criar esse meio de armazenamento também. Para cada meio de armazenamento que interage com o sistema, deve-se criar um objeto do tipo *cluster*, especificando seu conteúdo na propriedade *description* do objeto criado. Em seguida, relaciona-se o objeto recém-criado àquele que representa o sistema; e finalmente, em cada objeto cluster pode ser realizado um *assigment* deste para um modelo do tipo eERM, onde será criado o diagrama de entidades e relacionamentos referente às informações do banco.

Em seguida, executar o script *Relatório de Informações do Sistema* a partir do VAC principal da modelagem. Será gerado um relatório com nome e descrição do sistema a ser desenvolvido, seu diagrama de utilização, a lista de usuários e permissões de acesso, as interações com outros sistemas e as informações dos clusters, listando os diagramas de entidades e relacionamentos, quando existirem.

Assim, foram analisados os meios de armazenamento a serem gerenciados pelo sistema e as informações que serão registradas sobre eles. Essas informações foram obtidas através dos objetos de negócio, manipulados nas atividades, e suas

descrições. Para cada um foi criado um objeto do tipo *Cluster*, e na sua descrição foram descritas as informações que este deve manipular.

Finalmente, para cada *cluster* foi criado um *assigment* deste para um modelo do tipo *eERM*, e neste último foi definido um modelo de entidades e relacionamentos, onde são listados também os atributos de cada uma dessas.

Ao final da etapa foi executado o script, que gerou o diagrama de utilização, bem como lista de sistemas de apoio, usuários e clusters, além dos modelos de entidades e relacionamentos associados aos clusters. A figura abaixo ilustra o diagrama de utilização do sistema. As demais informações de clusters, usuários e outros sistemas encontram-se no Documento de Requisitos de Software, no anexo II.

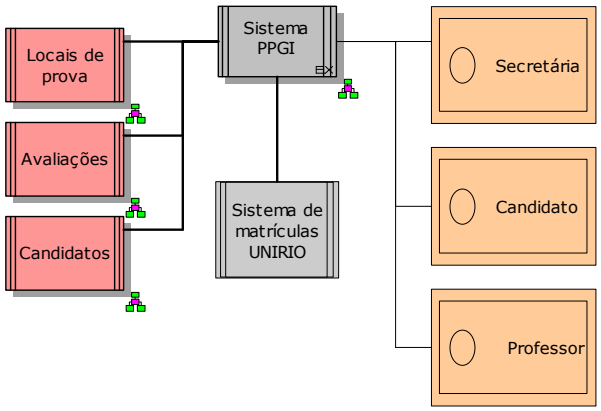


Figura 5.2 Diagrama de utilização do sistema, referente ao Sistema PPGI

5.2.2.3 Sexta etapa: Identificar os Impactos do Novo Sistema

O objetivo desta etapa é avaliar como o novo sistema interfere no negócio da organização. A entrada é o modelo de atividades e o relatório gerado na fase anterior, com o diagrama de utilização e as funcionalidades e restrições do sistema. A saída é o relatório das mudanças que deverão ser feitas no modelo de atividades, refletindo as alterações causadas ao processo com o surgimento do novo sistema. Como o surgimento do sistema gerará mudanças na maneira como o processo é executado, tais alterações devem ser identificadas para que seja possível, posteriormente, atualizar o modelo de atividades.

O primeiro passo é analisar o modelo de atividades e identificar quais atividades serão re-alocadas a novos executantes. As atividades que passarão a ser executadas pelo sistema não serão mais de responsabilidade de seus ex-executantes.

Além disso, verificar ainda alterações na maneira como os sistemas auxiliares apóiam as atividades. Para cada alteração, clicar com o botão esquerdo sob o objeto correspondente – papel ou sistema, e selecionar a opção *Change Management > Submit Proposal*. Preencher, na janela que será aberta, o campo *New Proposals*, com as alterações que devem ser realizadas no modelo.

Assim, as atividades onde foram identificadas oportunidades de melhorias no estudo de caso tiveram a propriedade *proposals* editadas através da opção *Edit Proposal* do item *Change Managent* do menu *Edit*. O mesmo processo se deu para oportunidades de melhoria de objetos organizacionais e sistemas que apóiam a execução das atividades.

No segundo passo, executar o script *Relatório de Sistemas de Apoio*, que lista informações dos sistemas e dos meios de armazenamento que interagem com o sistema a ser desenvolvido, com a finalidade de manter estas informações facilmente acessíveis. Os gerentes devem validar o resultado.

Finalmente, executar o script *Relatório de Impactos da Criação do Novo Sistema*, que lista o impacto da criação do sistema nas atividades, nos papéis existentes e nos sistemas que interagem com o sistema que será criado.

No estudo de caso, a execução do script *Relatório de impactos da criação do novo sistema* listou as propostas de alteração, cujo resultado é apresentado no anexo II. Também é apresentado no anexo o resultado da execução do script *Relatório de sistemas de apoio*, que gerou a lista dos sistemas e dos *clusters* existentes e que serão criados.

5.2.2.4 Sétima etapa: Gerar Documento de Requisitos de Software

A última etapa do método tem a finalidade de documentar as informações relevantes ao sistema no Documento de Requisitos de Software. Para isso, esta etapa utiliza como entrada a modelagem realizada e as informações obtidas durante a execução do método; e como saída, é obtido o DRS.

O primeiro passo consiste na execução do script *Relatório de Geração do DRS*, que criará o Documento de Requisitos de Software baseado nas informações modeladas. O script deve ser executado a partir do VAC principal da modelagem. Após a criação do documento, o item 1.1 (descrição da solicitação de automação)

deve ser preenchido manualmente, pois trata de informações que não foram modeladas, e portanto, fogem ao escopo deste trabalho.

Na primeira parte do documento gerado é listada a visão geral do problema. Esta parte do DRS é composta de: uma descrição da solicitação inicial; a lista dos processos e atividades considerados para o desenvolvimento do sistema bem como o respectivo modelo; os objetivos relacionados aos processos no contexto da solicitação; os papéis e departamentos envolvidos nas atividades; a lista de dificuldades por atividade, bem como a descrição detalhada de cada uma; lista de necessidades de negócio por dificuldade e por atividade; e descrição dos impactos das necessidades.

Na segunda parte do documento é listada a visão geral da solução, com as funcionalidades e restrições do sistema. Nesta parte estão: a lista de requisitos por necessidade; a lista detalhada de requisitos funcionais e a lista de requisitos não funcionais; as informações sobre o sistema que está sendo desenvolvido, seu diagrama de utilização, a lista de usuários e permissões e a lista de relacionamentos com outros sistemas; os dicionários de dados, e os modelos de dados dos clusters identificados; além da lista de mudanças que o novo sistema impactará no modelo de negócio atual.

O Documento de Requisitos de Software do estudo de caso foi gerado a partir da execução do macro-processo ‘Processo de Seleção PPGI’, e pode ser observado no Anexo II - Documento de Requisitos de Software

Ao final do método, quando a levantamento de requisitos for concluído, executa-se o script *Apagar Propriedade 'System Support'*, para limpeza da propriedade *System Support* - que diferencia os objetos considerados no contexto da solicitação - a fim de que num próximo levantamento não sejam consideradas atividades, processos ou demais objetos utilizados em um levantamento anterior.

A execução do script na modelagem realizada pelos alunos desmarcou a propriedade de todos os objetos do modelo, permitindo a reutilização para uma nova elicitação de requisitos

5.2.3 Considerações

Este capítulo apresentou o detalhamento da execução do método Mac Knight aplicado à metodologia ARIS. Ao final da sua execução do método, obtêm-se a lista dos requisitos funcionais e não funcionais levantados a partir da análise do modelo de negócio, em reuniões com as pessoas da organização.

Através do estudo de caso, foi possível perceber que o apoio computacional desenvolvido atende ao método. O estudo de caso, apesar de tratar de uma realidade simples, mostrou-se suficiente para validar o trabalho proposto.

Finalmente, vale destacar que o método não necessita ser seguido rigorosamente em suas etapas e passos, sendo possível a mudança de ordem de alguns passos ou a não execução dos mesmos.

Capítulo 6 - Conclusão

Este trabalho apresentou uma proposta de customização da ferramenta ARIS para a implementação do método de extração de requisitos de software a partir de um modelo de negócio, proposto por Débora Mac Knight (MAC KNIGHT, 2004).

A viabilidade da implementação proposta foi verificada através de um estudo de caso a partir de um modelo de negócio real, onde se pôde observar que a ferramenta facilita tanto a geração de relatórios quanto a manutenção da informação, seja na própria ferramenta ou nos relatórios gerados; e amplia a possibilidade de documentação além dos itens propostos no método Mac Knight, dada a utilização de uma metodologia e a criação de estruturas complementares.

O apoio computacional fornecido ao método torna válida a possibilidade de uso em grandes organizações, cujos modelos do negócio costumam ser extremamente complexos, repletos de informações, e voláteis. Nesses ambientes, a geração e manutenção manual da documentação dos requisitos do negócio e de software se tornaria inviável, pois despenderia grande tempo e esforço para análise de toda a modelagem, e gerência das mudanças. Funcionalidades nativas do ARIS, como o Matrix Editor, aliadas às funcionalidades desenvolvidas neste trabalho tendem a possibilitar escalabilidade nos modelos e levantamento de requisitos.

Ainda, a adoção de uma metodologia de modelagem do negócio para aplicação do método não só atende o método Mac Knight da forma como ele foi proposto originalmente, atuando neste ponto como apoio na derivação dos requisitos originados por uma demanda de desenvolvimento de um sistema, como amplia sua possibilidade de utilização, tornando possível a visão global das necessidades do negócio e de cada processo, onde um mesmo modelo de processos agrega os requisitos originados de diversas demandas de desenvolvimento de sistemas.

A implementação do método sobre metodologia e ferramenta específicas e já existentes também trouxe benefício. A adoção da metodologia ARIS enriquece o modelo de negócio, uma vez que se utiliza de uma gama maior de objetos com semântica específica que antes não podiam ser considerados, dada a independência de metodologia proposta pelo método original.

Entretanto, devem ser destacados os riscos da implementação utilizando uma metodologia específica. Por conta da adoção de uma definição que não é universal, caso a metodologia caia em desuso, ou seja, descontinuada, corre-se o risco de grande parte da modelagem ser perdida – caso nenhuma outra ferramenta consiga importar as informações armazenadas na ferramenta. Entretanto, como o *ARIS Business Architect* possui a opção de exportação do formato XML, adotado como padrão de interoperabilidade por grande parte das ferramentas atuais na área de TI, este risco reduz-se bastante.

Outra limitação está no fato do método não fazer parte das etapas tradicionais de desenvolvimento de software, sendo necessária adaptação do processo para sua utilização.

Além disso, existiria grande possibilidade de todo o método customizado para a ferramenta ser perdido, dado que cada ferramenta também possui uma maneira própria de representar seus elementos, e principalmente de configurar como os objetos, modelos e relacionamentos estão disponíveis para utilização. Mas dada a utilização e a aceitação da metodologia ARIS no mercado e seus quase vinte anos de utilização, o risco desse cenário se tornar real é relativamente baixo.

6.1 Contribuições do Trabalho

Apesar da utilização de um método já existente como base, este trabalho contribui para as áreas de modelagem de negócio e desenvolvimento de sistemas. Através de uma seqüência de etapas e das configurações realizadas na ferramenta, é possível analisar o modelo de negócio e identificar características necessárias ao sistema, adicionando informações neste modelo de negócio relativas aos requisitos. Estes fatos são importantes, pois a presença desta seqüência de etapas guia o analista de sistemas no entendimento tanto do negócio quanto dos requisitos que vão sendo descobertos. Mesmo que o método não seja seguido em todas as suas etapas e passos, o simples fato de ser utilizado ajuda as equipes a discutirem, avaliarem, documentarem e então decidirem se seguirão inteiramente ou parcialmente o método.

A estrutura criada, além de atender ao método Mac Knight, permite a ampliação da documentação, uma vez que ofereceu a possibilidade de registro de

requisitos não funcionais na modelagem, não considerados por Mac Knight no escopo de seu trabalho.

Além disso, a implementação realizada neste trabalho torna extremamente simples a geração e manutenção de documentação dos resultados das etapas do método. Enquanto todo o trabalho era gerado manualmente no método Mac Knight, neste trabalho toda a documentação necessária é gerada automaticamente, de forma dinâmica.

Uma última contribuição deste trabalho reside na criação de um mapa de requisitos, cujas vantagens estão: na manutenção da rastreabilidade entre requisitos e sistemas – onde a visualização gráfica permite uma compreensão mais fácil dos relacionamentos; na manutenção do histórico de sistemas desenvolvidos, com informações de quais requisitos e quais processos e atividades cada sistema atende; e ainda na facilidade de realização de alterações nos processos, modelos e informações – sejam estas correções ou melhorias.

6.2 Trabalhos Futuros

Uma oportunidade de trabalho futuro está na criação de um método que elicite os requisitos não funcionais, uma vez que este trabalho apenas possibilitou sua representação e coleta de informações para geração de relatórios, não fundamentando definindo um método para sua utilização. Inicialmente, sua utilização deve ser opcional, e realizada de maneira análoga aos requisitos funcionais.

Entretanto, a documentação na ferramenta foi estendida aos requisitos não funcionais. Vale destacar que não está no escopo deste trabalho o levantamento de requisitos não funcionais, e que o recurso está disponível apenas por considerar que já existem estudos em desenvolvimento na UNIRIO, no sentido de elaborar um método para elicitação dos requisitos não funcionais

De forma relacionada, outra proposta trataria da rastreabilidade entre requisitos funcionais e não funcionais, não abordada neste trabalho já que foi apenas estendido o suporte à documentação de requisitos não funcionais.

Como o trabalho, seguindo a proposta original, considera como escopo apenas uma demanda, não foram implementados mecanismos para tratar da evolução dos requisitos que foram incorporados ao longo de elicitações para os diferentes

sistemas relacionados ao modelo de processos. Dessa forma, essa seria outra possibilidade de trabalho futuro.

Destaca-se ainda como trabalho futuro a possibilidade a elaboração de novos estudos de caso, considerando ambientes reais de desenvolvimento de sistemas utilizando o conjunto de soluções proposto por este trabalho a fim de avaliar o uso real e sistemático do método, já que a proposta não foi adotada num ambiente corporativo – com uma solicitação formal de desenvolvimento de software aos analistas. Apesar de prover um conjunto ferramental funcional, ainda são identificados pontos de melhoria, que não foram incluídos no escopo do projeto. Dentre estes pontos, está a criação automática de requisitos a partir das informações de necessidades e dificuldades coletadas, automatizando a tarefa atualmente manual de verificação de dificuldades e necessidades e criação dos respectivos requisitos.

Além disso, a criação de mecanismos ferramentais que gerem visualização gráfica dos relacionamentos entre atividades, dificuldades, causas e conseqüências das dificuldades. Atualmente, o relacionamento entre atividades e dificuldades é visualizado em uma tela e as propriedades causas e conseqüências em outra.

Outra oportunidade está no estudo e implementação de mecanismos para gerência de mudança nos requisitos e gerência de configuração de sistemas. Dessa forma, seria possível manter versionamento e localizar alterações nos requisitos ao longo do tempo.

Referências Bibliográficas

(ARAUJO, BORGES, 2003) ARAÚJO, Renata; BORGES, Marcos. **Modelagem de Processos**. Rio de Janeiro, 2003. Sistemas de Workflow, Bacharelado em Ciência da Computação, UFRJ.

(ARAUJO, MKNIGHT, BORGES, 2005) ARAUJO, R. M. ; MKNIGHT, D. ; BORGES, Marcos Roberto da Silva . **A Systematic Approach for Identifying System Requirements from the Organization´s Business Model**. In: Simpósio Brasileiro de Sistemas de Informação, 2005, Florianópolis. II Simpósio Brasileiro de Sistemas de Informação, 2005

(BPTRENDS, 2008) **Glossary: Aris**. Disponível em http://www.bptrends.com/resources_glossary.cfm?letterFilter=A&wordid=1585CED3-1031-D522-34E687A7B24EAAAC. Acesso em 25/01/2008.

(CARVALHO, TAVARES, 2002) CARVALHO, Ana Elizabete; TAVARES, Helena Cristina. **Visão Geral sobre Requisitos**. Tematec, Ano VIII, Nº 60, 2002.

(DORFMAN, THAYER et al, 1990) DORFMAN, M.; THAYER, R. H., **Standards, Guidelines, and Examples of System and Software Requirements Engineering**, IEEE Computer Society Press, 1990.

(ERIKSSON, PENKER, 2000) ERIKSSON, H.-E.; PENKER, M., **Business Modeling with UML: Business Patterns at Work**. New York: Wiley Publishers, 2000.

(GOTTESDIENER, 2002) GOTTESDIENER E., **Requirements by Collaboration**, Addison-Wesley, 2002

(IEEE, 1998) Software Engineering Standards Committee of the IEEE Computer Society, **IEEE Recommended Practice for Software Requirements Specifications,** 1998.

(LEFFINGWELL, WIDRIG, 2000) LEFFINGWELL D., WIDRIG D., **Managing system requirements**, Addison-Wesley, 2000.

(MAC KNIGHT, 2004) MAC KNIGHT, Débora; Araújo, Renata; Borges, Marcos. **Elicitação de Requisitos a partir do Modelo de Negócio**. Dissertação de Mestrado, Programa de Pós-Graduação em Informática, NCE/IM-UFRJ, Rio de Janeiro, 2004.

(MIRANDA, ARAUJO, BORGES, 2007) MIRANDA, I. ; ARAUJO, R. M. ; BORGES, Marcos Roberto da Silva . **Discovering Group Communication Requirements**. In: Workshop Iberoamericano de Ingeniería de Requisitos y Ambientes de Software, 2007, Isla Margarita. IDEAS´07: X Workshop Iberoamericano de Ingeniería de Requisitos y Ambientes de Software, 2007. p. 107-120.

(PRESSMAN, 2001) PRESSMAN, R.S., Software Engineering: a practitioner's approach, 5ª edição, McGraw Hill, 2001.

(PROFORMA, 2000) PROFORMA., **Enterprise Application Modeling – Vision and strategy for the ongoing development of ProVision Workbench**. Proforma Technical White Paper by Proforma Corporation, 2000.

(PUNTAR, DUARTE, 2007) PUNTAR, Sérgio;. DUARTE, Diego. **Documento de Requisitos de Software do Sistema PPGI**. 2007. Análise e Projeto de Sistemas, Bacharelado em Sistemas de Informação, UNIRIO.

(REUBENSTEIN, WATERS, 1991) REUBENSTEIN, H.B.; WATERS, R.C., **The Requirements Apprentice: Automated Assistance for Requirements Acquisition**, IEEE Transaction on Software Engineering, vol 7, n° 3, p.226-240, 1991.

(SANTOS, CAMEIRA, CLEMENTE, CLEMENTE, 2008) SANTOS, Rafael Paim C.; CAMEIRA, Renato Flórido; CLEMENTE, Armando Augusto; CLEMENTE, Rafael Gomes. **Engenharia de Processos de Negócios: Aplicações e Metodologias.** Rio de Janeiro, 2002. In: ENEGEP, 2005, Porto Alegre. XXV ENEGEP, 2005

(SCHEER AG, 2006) SCHEER AG, IDS. **Aris Toolset:** A Solução Ideal para E-Business Engineering. 2006. Disponível em http://www2.ids-scheer.com/sixcms/detail.php/17870. Acesso em 25/01/2008.

(SCHEER, 1998) SCHEER, A.W., **ARIS -Business Process Frameworks**, 2 ed., Springer Verlag, Berlin, 1998.

(SOMMERVILLE, 2003) SOMMERVILLE, I., **Engenharia de Software** , Addison-Wesley, 2003.

Anexos

Anexo I: Levantamento de requisitos e Modelo de dados para customização da ferramenta *ARIS Business Architect*

1. Levantamento de Requisitos

Etapa	**Passo**	**Requisitos**	**Solução**
Etapa 1: Identificar o contexto da solicitação	Passo 1	Analisar descrição dos processos	Disponibilizar a propriedade *Description* nos processos, para que o usuário tenha acesso de leitura e escrita, escrevendo a descrição do processo.
		Identificar os processos envolvidos no contexto da solicitação	Disponibilizar a propriedade *System Support* nos processos, para que o usuário marque a propriedade nos processos envolvidos no contexto da solicitação.
	Passo 2	Analisar descrição das atividades	Disponibilizar a propriedade *Description* nas atividades, para que o usuário tenha acesso de leitura e escrita, escrevendo a descrição da atividade.
		Identificar as atividades envolvidas no contexto da solicitação	Disponibilizar a propriedade *System Support* nas atividades, para que o usuário marque a propriedade nas atividades envolvidas no contexto da solicitação.
	Ao final	Gerar relatório com processos, subprocessos, atividades e objetivos selecionados	Criação de script que, para cada processo no contexto da solicitação liste seu nome e descrição; os subprocessos a as atividades relacionadas, com as respectivas descrições; além dos objetivos de negócio relacionados
Etapa 2: Identificar necessidades dos processos	Passo 1	Identificar executantes de uma atividade	Criação de script que liste – caso exista(m) – o(s) executante(s) de cada atividade dentro do contexto da solicitação. Listar apenas aqueles papéis cujo relacionamento seja "Carries out"

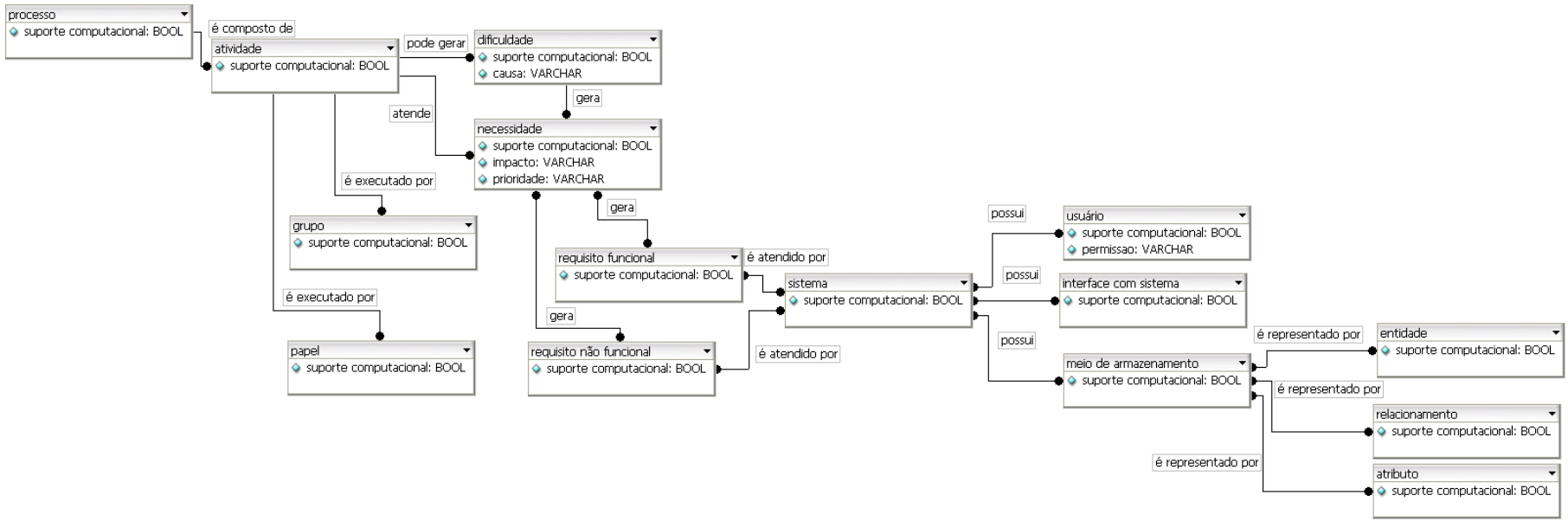
		Identificar dificuldades existentes na execução de cada atividade dentro do contexto	Para cada atividade onde seja encontrada dificuldade: criar, caso não exista, um FAD; nele, criar um objeto *difficult* para cada dificuldade encontrada.
	Passo 2	Identificar causas das dificuldades encontradas anteriormente	Criar a propriedade *causes*, dentro do objeto dificuldade, a fim de que possam ser preenchidas as suas causas
		Identificar necessidades de negócio que não são supridas atualmente	Criar um objeto *need*, onde serão representadas as necessidades
	Passo 3	Identificar resultados insatisfatórios gerados pelos processos	Criar um objeto *need*, onde serão representadas as necessidades
		Necessidades encontradas devem ser listadas e relacionadas com as atividades que as originaram	Criação de script que liste cada processo, suas atividades, dificuldades e necessidades
	Ao final	Gerar relatório com informações obtidas nesta fase	
Etapa 3: Identificar as conseqüências de cada necessidade	Passo 1	Identificar o impacto do não atendimento das necessidades existentes	Criar uma propriedade *impact* no objeto *need*, onde será armazenado o impacto do não atendimento de uma necessidade
	Passo 2	Identificar os processos prejudicados com o não atendimento das necessidades, relacionados com os objetivos do negócio que são afetados	Executar um script que liste os processos considerados no contexto da solicitação, relacionados aos seus objetivos
	Passo 3	Identificar as pessoas envolvidas com o negócio que sofrem as conseqüências do não atendimento das necessidades	Executar um script que gere a lista de papeis relacionados às atividades selecionadas.

	Passo 4	Definir quais necessidades serão atendidas pelo sistema	Criar a propriedade *priority* no objeto necessidade, atribuindo valores de acordo com a prioridade no atendimento da necessidade. Para as que não forem atendidas, deixar vazio e desmarcar a propriedade system support
		Identificar as prioridades das necessidades a serem atendidas	
	Ao final	Gerar relatório com informações obtidas nesta fase	Criação de script que gere a lista de dificuldades e necessidades dos processos no domínio da solicitação, além das listas de necessidades por processo e de necessidades e os impactos do não atendimento de cada uma
Etapa 4: Identificar as funcionalidades e restrições do sistema	Passo 1	Identificar as funcionalidades do sistema, para que este atenda aos processos dentro do contexto	Para cada funcionalidade encontrada, criar um objeto do tipo *functional requirement*, preenchendo seu nome e sua descrição
		Identificar as restrições do sistema, para que este atenda aos processos dentro do contexto	Para cada restrição encontrada, criar um objeto do tipo *functional requirement*, preenchendo seu nome e sua descrição
	Passo 2	Acrescentar funcionalidades e restrições nas atividades para que atendam às necessidades levantadas	Complementar a lista de funcionalidades e restrições criadas nos passos anteriores, criando outros objetos conforme a necessidade.
	Ao final	Gerar lista de funcionalidades e restrições do sistema	Criar script onde, sejam listadas funcionalidades e restrições por conjunto de atividades e necessidades.
Etapa 5: Identificar as fronteiras do sistema	Passo 1	Identificar os papéis que participam das atividades no contexto do sistema	Executar script que liste todos os papéis, grupos e unidades organizacionais relacionados com as atividades selecionadas.
		Definir quais papéis interagirão com o sistema e as respectivas permissões	Criar um objeto do tipo *user*, com a propriedade *permission* para cada usuário identificado.

	Passo 2	Identificar interações com outros sistemas através da descrição das atividades	No diagrama de sistema, criar uma ocorrência de sistema e ligar ao objeto que representa o sistema que está sendo criado para cada sistema que, após definido que deve interagir com os antigos.
		Definir se novo sistema deve interagir com os antigos	
	Passo 3	Representar o acesso das atividades aos diversos meios de armazenamento	Criar um objeto do tipo File para cada meio de armazenamento que uma atividade fará contato. A ligação será feita do símbolo do sistema com os diversos meios de armazenamento
		Para cada meio de armazenamento, especificar seu conteúdo e os dados que serão armazenados	Criar um *assigment* de cada representação do meio de armazenamento para um eERM, e criar os diversos dados que serão armazenados, e seus relacionamentos
	Ao final	Gerar lista de recursos que interagirá com o sistema e suas especificações (usuários, permissões e dicionário de dados).	Criar Script para listar os usuários do sistema e suas respectivas permissões, gerar o dicionário de dados do sistema, e relacionar os recursos que interagem com o sistema com suas especificações
Etapa 6: Identificar impactos do novo sistema	Passo 1	Identificar atividades que serão executadas pelo sistema e alterações nos processos com a criação do sistema	Identificar alterações que devem ser realizadas nos processos ou nas atividades em função da criação do sistema. Ao final um script deve listar onde existe necessidade de alteração
	Passo 2	Listar meios de armazenamento que interagirão com o sistema, incluindo também os já existentes e os que serão criados.	Criar script que liste os meios de armazenamento que interagirão com o sistema, incluindo os que já existem e os que serão criados

		Listar os papéis que interagirão com o sistema, com os respectivos usuários e permissões	Criar script que liste os papéis, os respectivos usuários e permissões
Etapa 7: Gerar Documento de Requisitos de Software	Passo 1	Obter descrição da solicitação inicial	O usuário preencherá manualmente após a geração do documento de requisitos de software. No documento deve ser exibida uma mensagem solicitando que seja informada a solicitação inicial.
		Geração do documento de requisitos de software	Execução de um script que execute a geração de um relatório com o resultado de todas as etapas anteriores
	Ao final	Apagar propriedade marcada	Script que apague a propriedade *system support*, deixando todos os objetos desmarcados

2. Modelo de dados para os objetos criados para atender aos requisitos



O modelo acima apresentado lista apenas as entidades e os atributos criados e manipulados pela ferramenta, de forma a explicitar o seu funcionamento, ficando fora do escopo do modelo os objetos e atributos do ARIS.

Vale explicitar ainda que na notação gráfica utilizada no modelo, o relacionamento entre duas entidades segue a seguinte semântica: uma entidade que possua um relacionamento cuja cardinalidade seja N (muitos) é representada por um circulo escuro. Um exemplo é o relacionamento entre processo e atividade, onde cada processo "é composto de" varias atividades, e cada atividade compõe somente um processo. Neste caso, como a entidade atividade tem cardinalidade N, é exibido o círculo escuro.

Anexo II: Documento de Requisitos de Software

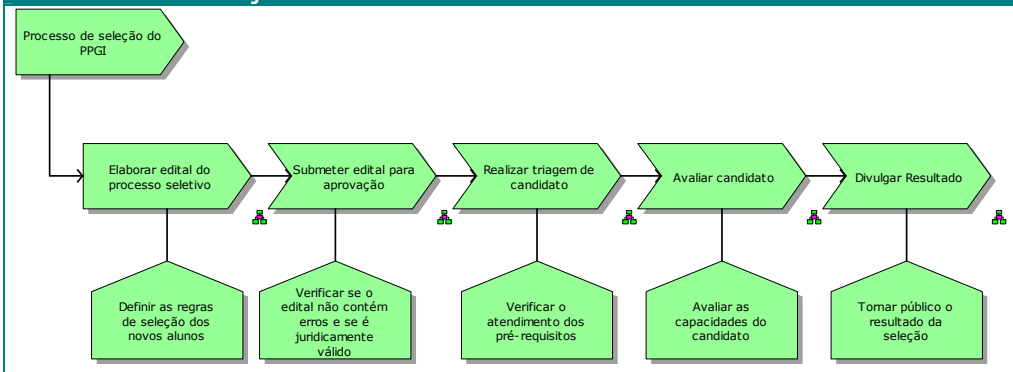
Visão geral do problema

Descrição da solicitação de automação

<Escreva uma breve introdução sobre o problema e o contexto da solicitação de automação>

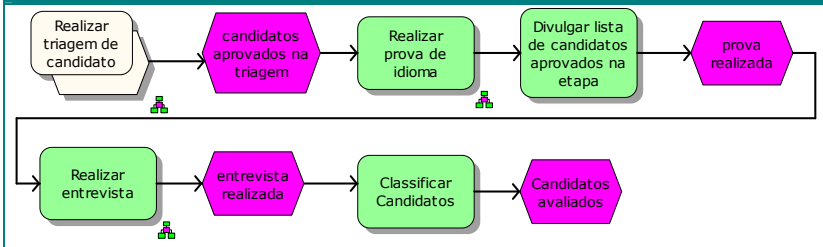
Processos de negócio envolvidos

Processo de seleção do PPGI



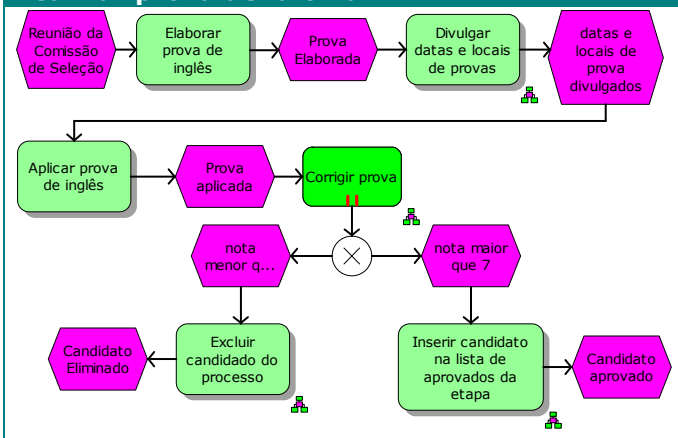
Processo / Atividade	Descrição
Avaliar candidato	O processo avalia diversas habilidades e competências dos candidatos, a fim de verificar aqueles que têm o perfil de aluno desejado. A etapa é composta de prova de idioma (inglês) e entrevista com a comissão de seleção

Avaliar candidato



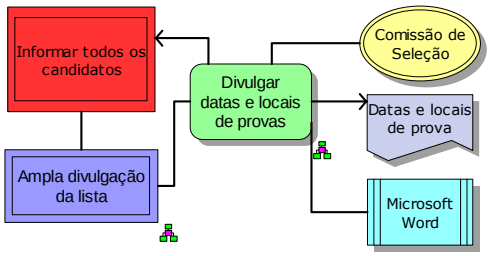
Processo / Atividade	Descrição
Realizar prova de idioma	Etapa que avalia o conhecimento de língua estrangeira dos candidatos através da aplicação de uma prova

Realizar prova de idioma



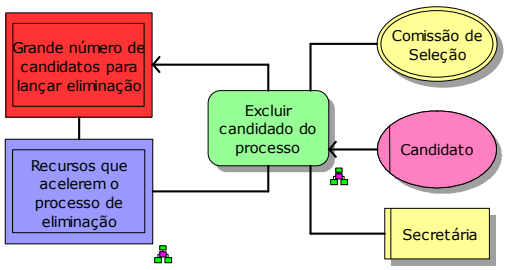
Processo / Atividade	Descrição
Divulgar datas e locais de provas	Atividade que torna público a data e local de realização da prova para todos os candidatos aprovados na fase de triagem
Excluir candidato do processo	Atividade de exclusão do candidato do processo de seleção do PPGI, finalizando o processo dos candidatos eliminados
Corrigir prova	Atividade de correção da prova do candidato, executada para cada candidato inscrito. Logo, se repete tantas vezes o número de candidatos inscritos.
Inserir candidato na lista de aprovados da etapa	Atividade que consiste na inserção do candidato na lista de candidatos aprovados na etapa

Divulgar datas e locais de provas



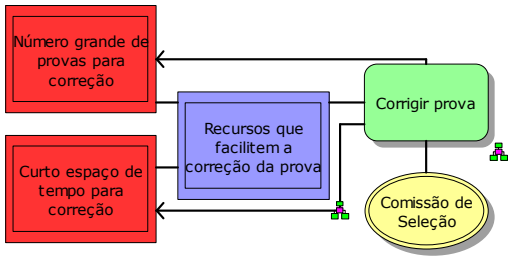
Processo / Atividade	Descrição
Divulgar datas e locais de provas	Atividade que torna público a data e local de realização da prova para todos os candidatos aprovados na fase de triagem

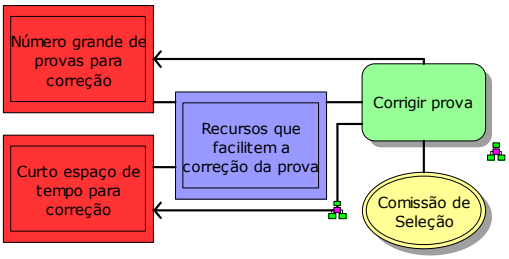
Excluir candidato do processo



Processo / Atividade	Descrição
Excluir candidato do	Atividade de exclusão do candidato do processo de seleção do PPGI,

processo	finalizando o processo dos candidatos eliminados

Corrigir prova	
	
Processo / Atividade	**Descrição**
Corrigir prova	Atividade de correção da prova do candidato, executada para cada candidato inscrito. Logo, se repete tantas vezes o número de candidatos inscritos.

Inserir candidato na lista de aprovados da etapa	
	
Processo / Atividade	**Descrição**
Inserir candidato na lista de aprovados da etapa	Atividade que consiste na inserção do candidato na lista de candidatos aprovados na etapa

Objetivos relacionados aos processos envolvidos

Processo/ Atividade	**Objetivo**	**Descrição**
Avaliar candidato	Avaliar as capacidades do candidato	Avaliando as capacidades do candidato é possível perceber aqueles cujo perfil e interesses estão de acordo com a linha do orientador.

Papéis e departamentos envolvidos

Divulgar datas e locais de provas		
Atividade	Objeto - Tipo do Objeto	Descrição
Divulgar datas e locais de provas	Comissão de Seleção - Group	Grupo composto por docentes do PPGI, escolhidos através da reunião do colegiado
Excluir candidato do processo		
Atividade	Objeto - Tipo do Objeto	Descrição
Excluir candidato do processo	Comissão de Seleção - Group	Grupo composto por docentes do PPGI, escolhidos através da

		reunião do colegiado
Corrigir prova		
Atividade	Objeto - Tipo do Objeto	Descrição
Corrigir prova	Comissão de Seleção - Group	Grupo composto por docentes do PPGI, escolhidos através da reunião do colegiado
Inserir candidato na lista de aprovados da etapa		
Atividade	Objeto - Tipo do Objeto	Descrição
Inserir candidato na lista de aprovados da etapa	Comissão de Seleção - Group	Grupo composto por docentes do PPGI, escolhidos através da reunião do colegiado

Dificuldades na execução dos processos

Descrição das dificuldades

Nome	Descrição	Causa(s)
Informar todos os candidatos	A grande quantidade de candidatos pode impedir que todos tomem conhecimento das datas e locais de provas	Dificuldades no meio de comunicação com candidatos
Grande número de candidatos para lançar eliminação	A grande quantidade de candidatos que são eliminados pode causar demora na conclusão das etapas do processo, atrasando o cronograma	Dificuldades no meio de comunicação com candidatos
Curto espaço de tempo para correção	O calendário apertado dificulta a correção das provas, exigindo esforço excessivo dos professores	Calendário do processo de seleção tem etapas de curta duração
Número grande de provas para correção	O grande número de candidatos pode atrasar a correção das provas e o conseqüente atraso no cronograma	Número grande de candidatos
Grande número de candidatos para lançar aprovação	O grande número de candidatos demanda tempo e esforço da comissão de seleção para geração da lista nominal de aprovados da etapa	Grande número de candidatos

Lista de dificuldades por Atividade

Processo	Dificuldade	Descrição
Divulgar datas e locais de provas	Informar todos os candidatos	A grande quantidade de candidatos pode impedir que todos tomem conhecimento das datas e locais de provas
Excluir candidato do processo	Grande número de candidatos para lançar eliminação	A grande quantidade de candidatos que são eliminados pode causar demora na conclusão das etapas do processo, atrasando

		o cronograma
Corrigir prova	Curto espaço de tempo para correção	O calendário apertado dificulta a correção das provas, exigindo esforço excessivo dos professores
	Número grande de provas para correção	O grande número de candidatos pode atrasar a correção das provas e o conseqüente atraso no cronograma
Inserir candidato na lista de aprovados da etapa	Grande número de candidatos para lançar aprovação	O grande número de candidatos demanda tempo e esforço da comissão de seleção para geração da lista nominal de aprovados da etapa

Descrição das Necessidades que serão supridas

Lista de Necessidades por Dificuldade

Processo(s)	Dificuldade(s)	Necessidade(s)
Divulgar datas e locais de provas	Informar todos os candidatos	Ampla divulgação da lista
Excluir candidato do processo	Grande número de candidatos para lançar eliminação	Recursos que acelerem o processo de eliminação
Corrigir prova	Curto espaço de tempo para correção	Recursos que facilitem a correção da prova
	Número grande de provas para correção	Recursos que facilitem a correção da prova
Inserir candidato na lista de aprovados da etapa	Grande número de candidatos para lançar aprovação	Recursos que acelerem o processo de aprovação

Lista de Necessidades por Atividade

Processo	Necessidade	Descrição
Divulgar datas e locais de provas	Ampla divulgação da lista	São necessários recursos que permitam e facilitem a ampla divulgação dos resultados
Excluir candidato do processo	Recursos que acelerem o processo de eliminação	São necessários recursos que automatizem o processo, acelerando o processo de eliminação do candidato
Corrigir prova	Recursos que facilitem a correção da prova	São necessários recursos que facilitem a correção da prova
Inserir candidato na lista de aprovados da etapa	Recursos que acelerem o processo de aprovação	São necessários recursos que automatizem o processo, acelerando o processo de criação da lista

Descrição dos impactos das necessidades

Nome	Descrição	Impacto	Prioridade
Ampla divulgação da lista	São necessários recursos que permitam e facilitem a ampla divulgação dos resultados	Candidato não ser informado da data e do local da prova	ALTA
Recursos que acelerem o processo de eliminação	São necessários recursos que automatizem o processo, acelerando o processo de eliminação do candidato	Geração de erros durante a correção, atraso na entrega dos resultados	ALTA
Recursos que facilitem a correção da prova	São necessários recursos que facilitem a correção da prova	Atraso na entrega dos resultados, erros na correção das provas	ALTA

Recursos que acelerem o processo de aprovação	São necessários recursos que automatizem o processo, acelerando o processo de criação da lista	Erros durante o lançamento dos resultados	ALTA

Visão geral da solução

Funcionalidades do Sistema

Lista de Requisitos por necessidade

Atividades	Necessidades	Requisitos Funcionais e não funcionais
Divulgar datas e locais de provas	Ampla divulgação da lista	Divulgação da lista de candidatos
		Candidatos devem apenas acessar suas informações
Excluir candidato do processo	Recursos que acelerem o processo de eliminação	Gerar lista de candidatos eliminados
		Eliminar Candidato
		Geração de backup das informações
Corrigir prova	Recursos que facilitem a correção da prova	Gerar relatório de notas
		Corrigir Provas
		Geração de backup das informações
Inserir candidato na lista de aprovados da etapa	Recursos que acelerem o processo de aprovação	Gerar lista de candidatos aprovados
		Aprovar Candidato
		Geração de backup das informações

Lista de Requisitos Funcionais

Processo	Requisitos Funcionais	Descrição
Divulgar datas e locais de provas	Divulgação da lista de candidatos	A divulgação da lista de locais e provas deve ser feita a todos os candidatos, seja através do envio de emails ou solicitação de contato telefônico
Excluir candidato do processo	Gerar lista de candidatos eliminados	O sistema deve gerar automaticamente a lista de candidatos eliminados a partir das informações sobre o resultado de cada etapa
Excluir candidato do processo	Eliminar Candidato	O sistema deve apresentar mecanismos para conclusão do processo do aluno, mantendo apenas nos registros no banco de dados
Corrigir prova	Gerar relatório de notas	O sistema deve gerar automaticamente o relatório de notas baseado na correção de provas realizada
Corrigir prova	Corrigir Provas	O sistema deve prover soluções automatizadas para correção da prova, preferencialmente através do sistema de leitura óptica
Inserir candidato na	Gerar lista de candidatos	O sistema deve gerar automaticamente a lista de

lista de aprovados da etapa	aprovados	candidatos aprovados a partir das informações sobre o resultado de cada etapa
Inserir candidato na lista de aprovados da etapa	Aprovar Candidato	O sistema deve apresentar mecanismos para conclusão do processo do aluno, possibilitando a troca de informações com o sistema de matrícula da UNIRIO

Lista de Requisitos Não funcionais

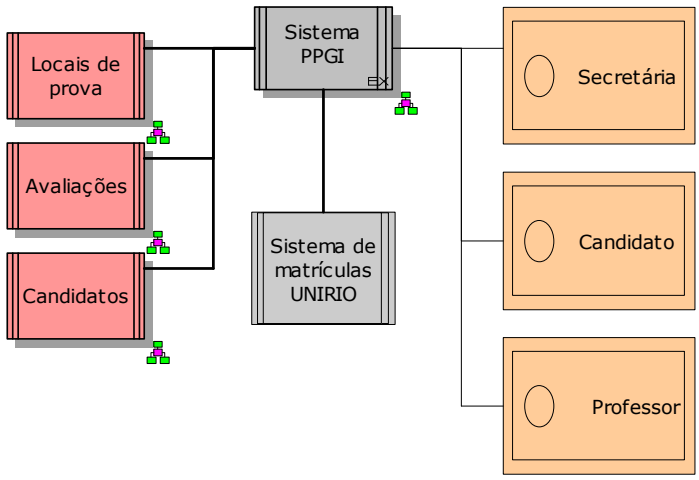
Processo	Requisitos Não funcionais	Descrição
Divulgar datas e locais de provas	Candidatos devem apenas acessar suas informações	Os candidatos devem ter acesso apenas às suas informações cadastrais e aos seus resultados no processo, mantendo o sigilo de informações
Excluir candidato do processo	Geração de backup das informações	O backup deve ser gerado para que nenhuma informação seja perdida em caso de sinistro ou outro problema qualquer
Corrigir prova	Geração de backup das informações	O backup deve ser gerado para que nenhuma informação seja perdida em caso de sinistro ou outro problema qualquer
Inserir candidato na lista de aprovados da etapa	Geração de backup das informações	O backup deve ser gerado para que nenhuma informação seja perdida em caso de sinistro ou outro problema qualquer

Fronteiras e usuários do Sistema

Informações do Sistema

Nome	Descrição
Sistema PPGI	Sistema de controle do processo seletivo do PPGI

Diagrama de utilização do sistema Sistema PPGI



Usuários e Permissões

Usuário	Descrição	Permissão
Secretária	A secretária manipula informações referentes à abertura e conclusão dos processos de seleção, além de encaminhar os selecionados para a matrícula	Registrar notas, cadastrar candidatos, acessar resultado, criar e finalizar processos de seleção dos candidatos
Professor	Manipulam e gerenciam todas as informações do processo seletivo	Avaliar candidato, selecionar candidato, gerar relatórios, acessar informações dos candidatos, incluir data e local de prova, divulgar agenda de entrevistas
Candidato	Acessa os resultados do processo e visualiza suas informações de cadastro	Consultar informações, preencher formulário de inscrição

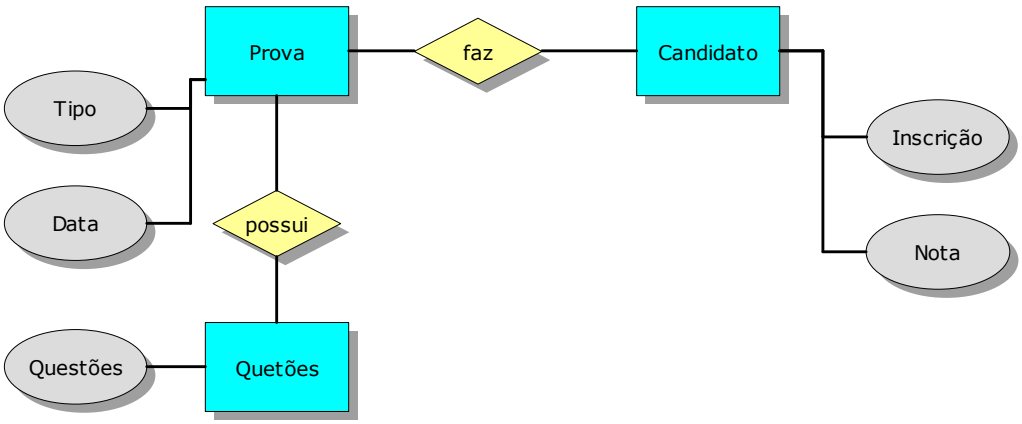
Relacionamentos com outros sistemas

Sistema	Sistema Associado	Descrição
Sistema PPGI	Sistema de matrículas UNIRIO	Sistema que gerencia as informações de matrícula dos alunos, além de notas, CR e disciplinas já cursadas. A proposta é que o sistema transfira as informações dos aprovados para o sistema de matrículas.

Dicionário de Dados

Meio de Armazenamento	Descrição
Avaliações	
Locais de prova	
Candidatos	

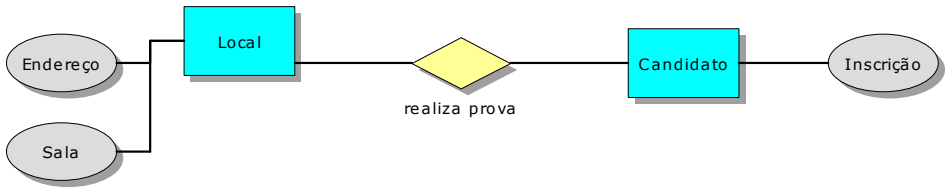
Modelo de dados do cluster Avaliações



Nome	Tipo	Descrição
Prova	Entity type	Avaliação do candidato
Data	ERM attribute	Data de realização da prova
Gabarito	ERM attribute	Gabarito de uma questão
Questões	Entity type	Questões da prova

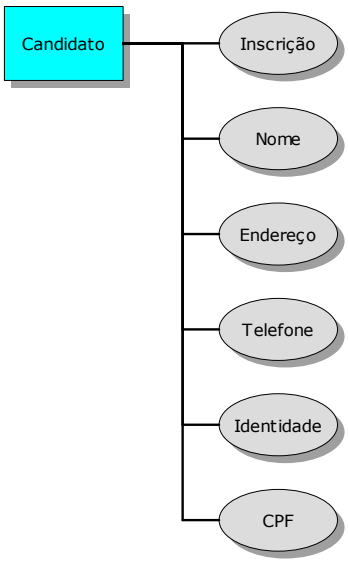
Inscrição	ERM attribute	Número de inscrição do candidato
possui	Relationship type	Relacionamento entre prova e questões
faz	Relationship type	Relacionamento que representa a execução de alguma atividade
Tipo	ERM attribute	Tipo de prova
Nota	ERM attribute	Conceito atingido pelo candidato
Candidato	Entity type	Pessoa que se candidata a uma vaga no processo seletivo

Modelo de dados do cluster Locais de prova



Nome	**Tipo**	**Descrição**
Sala	ERM attribute	
Candidato	Entity type	
Inscrição	ERM attribute	
Local	Entity type	
Endereço	ERM attribute	
realiza prova	Relationship type	

Modelo de dados do cluster Candidatos



Nome	**Tipo**	**Descrição**
Telefone	ERM attribute	

Nome	ERM attribute	
Identidade	ERM attribute	
Inscrição	ERM attribute	
Candidato	Entity type	
Endereço	ERM attribute	
CPF	ERM attribute	

Impactos gerados pelo sistema no negócio

Impactos da criação do novo sistema nos processos

Atividade	Descrição	Impacto
Corrigir prova	Atividade de correção da prova do candidato, executada para cada candidato inscrito. Logo, se repete tantas vezes o número de candidatos inscritos.	A prova será corrigida pelo sistema e não mais pela comissão de seleção

Meios de armazenamento existentes e que serão criados

Meio de Armazenamento	Descrição	Impacto
Avaliações		
Locais de prova		
Candidatos		

Impactos nos papéis que interagem com o sistema

Objeto Organizacional	Descrição	Impacto
Comissão de Seleção	Grupo composto por docentes do PPGI, escolhidos através da reunião do colegiado	No processo "Executar prova", o grupo deve ser alterado pelo papel secretaria