

Modelando, com o Método ERI*c, um Aplicativo para Gerenciamento da Fila de um Restaurante

Antonio de Padua Albuquerque Oliveira¹, Carolina dos Santos Aquino¹, Marcio Presley Farias Melo¹

¹ Universidade do Estado do Rio de Janeiro – UERJ
Rua São Francisco Xavier, 524 – 6º andar - Maracanã - Rio de Janeiro, Brazil

padua.uerj@gmail.com.br, aquinocarolina@ymail.com,
presley.melo@gmail.com

Abstract. *This paper proposes to show the application of the ERI*c method (Intentional Requirements Engineering Strategy) in modeling the requirements of a system that aims to manage the queue of a restaurant. The ERI * c method is a collaboration with Goal Oriented Requirements Engineering (GORE). An important feature of the method is the ability to reduce the complexity of iStar (i*) models used to create intentional models. For demonstration, all six steps that make up the ERI*c method were applied to the proposed case study. This article acts as a course completion project to obtain a bachelor's degree in Computer Science.*

Resumo. *Este artigo propõe-se a mostrar a aplicação do método ERI*c (Engenharia de Requisitos Intencional) na modelagem dos requisitos de um sistema que visa gerenciar a fila de espera de um restaurante. O método ERI*c é uma colaboração à Engenharia de Requisitos Orientada a Metas (GORE). Uma importante característica do método está na capacidade de reduzir a complexidade dos modelos iStar (i*), utilizados para a criação de modelos intencionais. Para demonstração, foram aplicadas todas as seis etapas que compõem o método ERI*c ao estudo de caso proposto. Este artigo atua como projeto de conclusão de curso para obtenção de grau de bacharel em Ciência da Computação.*

1. Introdução

Um planejamento detalhado é uma questão importante no desenvolvimento de qualquer software de qualidade. Por planejamento entende-se a identificação dos objetivos do software (as metas dos atores) e o detalhamento de tempo e recursos para o encaminhamento da solução. Diante de um bom planejamento torna-se possível estimar os recursos necessários para a elaboração do sistema, definir as melhores ferramentas a serem utilizadas, bem como fazer um levantamento dos requisitos necessários.

A “Etapa de Requisitos” é responsável por 40 a 60% dos problemas encontrados nos projetos de software [Leffingwell 97]. Uma elicitación falha, pode resultar em um sistema que não atinge seus objetivos, tendo como consequência um software que foge da proposta inicial e que não atende a todas as necessidades do cliente. Por esse motivo,

a etapa de requisitos é uma fase muito importante no desenvolvimento de software, devido à necessidade de identificar possíveis problemas antes de iniciar a implementação.

Engenharia de Requisitos é o processo sistemático de levantamento, entendimento, análise, documentação e gerência de requisitos [Kotonya 98]. É o ramo da Engenharia de Software que contempla as atividades do processo de software responsável por tratar os requisitos funcionais e não funcionais de um sistema a ser construído.

Nesse contexto, a Engenharia de Requisitos Orientada a Metas (*GORE*) está preocupada com o uso de metas para elicitar, elaborar, estruturar, especificar, analisar, negociar, documentar e modificar requisitos funcionais e não funcionais [Lamsweerde 01]. A orientação a metas desempenha um papel importante no processo de Engenharia de Requisitos [Lamsweerde 98]. As metas representam a intencionalidade dos atores (pessoas e grupos organizacionais) envolvidos, e a intencionalidade espelha os interesses e as decisões desses atores. Para se aproximar mais do sucesso, sistemas de software devem atender as metas dos atores de uma organização [Oliveira 08].

2. Método ERi*c

O método ERi*c (Engenharia de Requisitos Intencional) é uma contribuição a Engenharia de Requisitos Orientada a Metas (*GORE*) e se baseia no conceito de intencionalidade como apresentado pelo Framework i* [Yu 95]. O método é composto por um conjunto de técnicas úteis para auxiliar a construção de modelos *iStar*. Ao término da aplicação das etapas do método ERi*c, o engenheiro produzirá os quadros e os diagramas que compõem o documento de requisitos do método.

A Figura 1 mostra uma visão geral do método ERi*c. O mnemônico ERi*c do método tem um correspondente em inglês: *IRES - Intentional RE Strategy*.

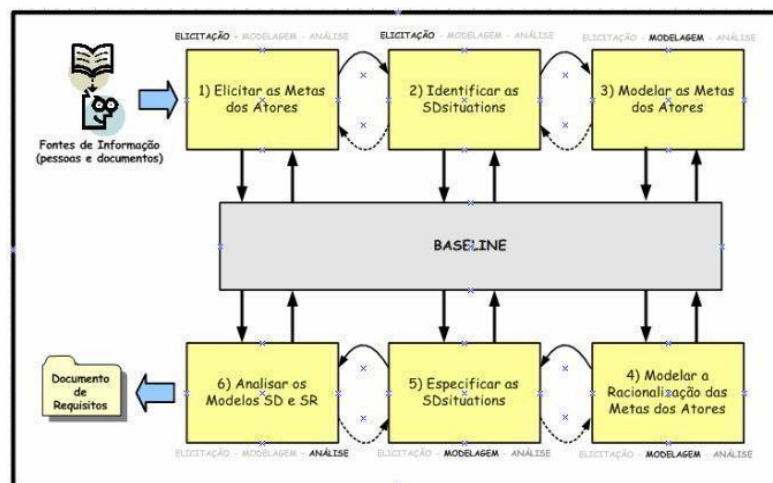


Figura 1. Visão geral do método ERi*c [Oliveira 08]

O método aborda as atividades de elicitação, modelagem e análise referenciadas por [Oliveira 08] como foram definidas e nomeadas primeiramente por [Leite 95], estas três atividades estão distribuídas pelas seis etapas do método. As atividades estão destacadas em negrito na Figura 1, indicando quais etapas estão preferencialmente

incluídas em cada atividade. As duas primeiras etapas do método envolvem a elicitación, na primeira etapa é feita a elicitación das metas e a identificação dos atores e, na segunda etapa, são identificadas as situações de dependência estratégica. As três etapas seguintes englobam a modelagem, na terceira etapa é elaborada a modelagem das metas dos atores, em seguida é modelada a racionalização das metas dos atores e, na quinta etapa, são especificadas as situações de dependência estratégica. Para concluir, na sexta etapa, é executada a análise dos modelos SD (dependências estratégicas – *Strategic Dependency*) e SR (razões estratégicas – *Strategic Rationale*).

O termo *baseline* [Leite 95], presente no centro do diagrama da Figura 1, indica que o método deve ser apoiado por ferramentas de software e por uma base de documentação, com a missão de facilitar o trabalho do engenheiro de requisitos no processo e também garantir, de modo automatizado, a rastreabilidade dos requisitos [Ramesh 01], além de permitir o armazenamento dos diagramas e dos documentos produzidos. O objetivo é que o método ERi*c colabore na qualidade final dos requisitos, e aperfeiçoar a qualidade final dos requisitos implica em modelos de requisitos mais completos, organizados e menos sobrecarregados.

O propósito desta seção foi apresentar uma visão geral do método ERi*c. Nas seções seguintes, descrevemos cada etapa do método, como mostrado na Figura 1, e apresentamos a modelagem do aplicativo do estudo de caso proposto.

3. Trabalhos Relacionados

Alguns estudos [Oliveira 08b, 17] mostram a criação de modelos intencionais utilizando o método ERi*c. Dentre estes, o artigo publicado por Oliveira, Leite, Cysneiros e Da Silva [Oliveira 17] apresenta uma ferramenta capaz de auxiliar o engenheiro de requisitos no desenvolvimento da primeira etapa do método ERi*c, usando como exemplo um sistema de pedágio (*TRC System*) para ilustrar a estratégia proposta.

Outro artigo publicado por Oliveira, Leite e Cysneiros [Oliveira 08b] apresenta todas as etapas do método ERi*c de maneira mais detalhada, através do exemplo de um sistema para apoiar a organização de conferências (*Expert Commitee*).

4. Modelagem ERi*c do Aplicativo do Restaurante

O sistema modelado instanciou o problema do gerenciamento e controle da fila de um restaurante típico. A fila de um restaurante, de um modo corriqueiro, provoca nos clientes ansiedade por estes não terem conhecimento de seu lugar na lista de atendimento e, por outro lado, provoca desconfiança por eles não entenderem como o número de pessoas (que é um requisito de otimização da disponibilidade do tamanho das mesas) afeta a ordem de chamada para a alocação dos clientes às mesas.

Os requisitos do “aplicativo” foram obtidos das metas elicitadas para o contexto organizacional do problema contemplando a visualização do cliente da evolução da fila do restaurante, o consumo de bebidas durante a espera e outros. O sistema aborda uma serie de problemas frequentes do cotidiano (problemas com requisitos específicos que podem ser adequadamente elicitados) como a fila de processos jurídicos, a marcação e o atendimento de consultas médicas, a fila de transplantes etcetera.

4.1 Elicitar as Metas dos Atores

O objetivo da primeira etapa do método é realizar a elicitación e refinamento das metas dos atores organizacionais [Oliveira 08]. Essa primeira etapa é composta por três fases: 1. Preparar LAL - Léxico Ampliado da Linguagem, 2. Definir AGFL - Metas dos Atores Vindas do Léxico e 3. Refinar as Metas dos Atores.

4.1.1. Preparar LAL - Léxico Ampliado da Linguagem

O LAL descreve a linguagem do domínio do sistema e está ancorado numa ideia muito simples: Entender a linguagem do problema, sem se preocupar com a compreensão do problema [Leite 00].

Para a elaboração do LAL, o engenheiro de requisitos primeiro identifica as fontes de informação que podem ser utilizadas. São considerados todos os documentos disponíveis, e todas as pessoas que podem fornecer informações sobre o contexto organizacional objeto do novo sistema. Os seguintes passos são recomendados [Leite 00]: (i) identificar a lista de símbolos relevantes, símbolos são as palavras ou frases peculiares e mais usadas; (ii) classificar os símbolos como: sujeito (o que pratica a ação), objeto (o que sofre a ação), estado (uma situação em um dado momento) e verbo (representa uma ação); (iii) descrever os símbolos através da noção e do impacto; (iv) verificar o LAL através de inspeção; e (v) validar o LAL com os atores do Universo de Informação.

As Figuras 2, 3, 4 e 5 mostram, respectivamente, exemplos do sistema modelado pelo trabalho, símbolos do tipo sujeito, objeto, estado e verbo. As figuras foram retiradas da ferramenta C&L [C&L 18], utilizada nessa fase.

Nome:	repcionista
Noção:	Funcionário do restaurante que realiza o primeiro atendimento ao cliente .
Classificação:	sujeito
Impacto(s):	- Repcionista aceita novos números de contato de clientes na fila . - Repcionista encerra conta da fila do cliente na fila . - Repcionista vincula reserva a uma mesa .
Sinônimo(s):	

Figura 2. Símbolo do LAL do tipo sujeito

Nome:	mesa
Noção:	Local onde cliente faz suas refeições.
Classificação:	objeto
Impacto(s):	- Mesa é vinculada a uma reserva pelo repcionista . - Garçom libera mesa.
Sinônimo(s):	

Figura 3.
LAL do tipo

Símbolo do
objeto

Nome:	cliente na fila
Noção:	cliente que aguarda mesa livre .
Classificação:	estado
Impacto(s):	- cliente na fila faz pedido . - cliente na fila acompanha situação da fila . - cliente na fila altera reserva . - cliente na fila paga conta da fila . - cliente na fila sai da fila .
Sinônimo(s):	

Figura 4. Símbolo do LAL do tipo estado

Nome:	efetuar reserva
Noção:	cliente solicita entrada na fila de espera
Classificação:	verbo
Impacto(s):	- cliente fora da fila informa dados de contato. - cliente fora da fila entra na fila de espera. - cliente na fila aguarda recebimento de alerta. - cliente na fila pode sair da fila caso desista do atendimento.
Sinônimo(s):	entrar na fila.

Figura 5. Símbolo do LAL do tipo verbo

4.1.2. Definir AGFL – Metas dos Atores Vindas do Léxico

No LAL, os atores são identificados como sujeitos, e os impactos reportam ações desempenhadas pelo ator. A finalidade dessa atividade é identificar a motivação por trás de cada ação. Uma ação que altera ou modifica um estado é chamada de ação concreta, quando a ação proporciona uma qualidade a um estado, é chamada de ação flexível.

Para definir o *template* do método AGFL, passamos por duas subatividades: Identificar os atores e extrair as metas dos atores a partir dos impactos dos símbolos. A segunda subatividade foi realizada com o auxílio da ferramenta AGFL Tool [Oliveira 17].

4.1.2.1. Identificar os Atores

Para identificar os atores a partir do LAL, verificamos, em primeiro, os impactos dos símbolos classificados como sujeito, pois eles praticam ações. Em segundo, analisamos os impactos dos símbolos classificados como objeto, porque esses símbolos sofrem ações praticadas por atores. Depois são analisados os impactos dos símbolos classificados com estado e verbo.

Portanto, os pretendentes naturais à posição de ator são os praticantes das ações sobre os objetos e os símbolos do tipo sujeito.

4.1.2.2. Extrair as Metas dos Atores a Partir dos Impactos dos Símbolos

A finalidade dessa atividade é extrair as metas a partir dos símbolos do LAL, seguindo o *template* próprio para cada tipo de ação classificada (concreta ou flexível). De acordo com o *template* e a classificação da ação, cada impacto irá gerar uma meta em um formato pré-estabelecido [Oliveira 08]. Uma meta concreta (*goal*), quando o impacto mencionar uma ação concreta, ou uma meta flexível (*softgoal*), quando o impacto mencionar uma ação flexível.

Pela subatividade de extração das metas, é possível estabelecer a posição de cada ator como “*dependor*” ou “*dependee*”. Essa posição é definida, respectivamente, como: aquele que depende (*dependor*) e aquele de quem se depende (*dependee*) para o alcance de uma meta.

Os Quadros 1, 2, 3, 4 e 5 mostram quatro metas concretas e uma meta flexível, extraídas a partir dos símbolos do LAL, cada uma seguindo o *template* pré-estabelecido de acordo com sua classificação. Os Quadros 1, 2 e 3 mostram metas em que o ator (*dependor*) depende de outros atores (*dependee*) para atingir a meta em questão.

Quadro 1. Extração de metas concretas de símbolo do tipo sujeito

	<i>goal</i>			ATOR
-- impacto resposta ao "por que?"	sujeito / objeto LAL	seja	verbo	sujeito LAL
RECEPCIONISTA				
recepcionista vincula reserva a uma mesa				
Porque recepcionista quer que	mesa	seja	ocupada	por cliente
Porque cliente quer que	reserva	seja	atendida	por recepcionista

Quadro 2. Extração de metas concretas de símbolo do tipo objeto

	<i>goal</i>			ATOR
-- impacto resposta ao "por que?"	sujeito / objeto LAL	seja	verbo	sujeito LAL
MESA				
garçom libera mesa				
Para que	mesa	seja	alocada	por recepcionista
Porque recepcionista quer que	mesa	seja	ocupada	por cliente
Porque cliente quer que	reserva	seja	efetivada	por garçom

Quadro 3. Extração de metas concretas de símbolo do tipo estado

	<i>goal</i>			ATOR
-- impacto resposta ao "por que?"	sujeito / objeto LAL	seja	verbo	sujeito LAL
CLIENTE NA FILA				
cliente na fila faz pedido				
Para que	pedido	seja	servido	por garçom
Por que garçom quer que	item do cardápio	seja	selecionado	por cliente

Quadro 4. Extração de meta concreta de símbolo do tipo verbo

	<i>goal</i>			ATOR
-- impacto resposta ao "por que?"	sujeito / objeto LAL	seja	verbo	sujeito LAL
EFETUAR RESERVA				
cliente fora da fila entra na fila de espera				
Para que	reserva	seja	desfrutada	por cliente

Quadro 5. Extração de meta flexível

	<i>softgoal</i>		
-- impacto resposta ao "por que?"	Tipo atributo de qualidade	[TÓPICO] sujeito / objeto LAL	Meta concreta associada
ATUALIZAR CARDÁPIO DA FILA			
cliente na fila acessa cardápio da fila atualizado			
Para que	boas opções	[item do cardápio]	pedido seja consumido por cliente

4.1.3. Refinar as Metas dos Atores

O propósito da atividade é agrupar as metas (*goals* e *softgoals*) por ator e organizá-las em ordem cronológica [Oliveira 08]. Para tal fim, é preciso converter as metas do tipo objeto em metas do tipo sujeito, em seguida, as metas são agrupadas por ator. Por fim, elas são

ordenadas temporalmente, colocando as metas de mais longo prazo para o final, além de serem removidas as metas redundantes provenientes do processo de elicitação.

O Quadro 6 mostra as metas refinadas do ator Cliente.

Quadro 6. Metas refinadas do ator Cliente

<i>Depender</i>						<i>Dependee</i>
CLIENTE						
comunicabilidade [fila]	reserva	seja	feita			
transparência [fila]	reserva	seja	cancelada			
rápida atualização [fila]	reserva	seja	cancelada			
rapidez no preparo [pedido]	pedido	seja	servido	por		garçom
boas opções [item do cardápio]	pedido	seja	consumido			
flexibilidade [reserva]	reserva	seja	atendida	por		recepcionista
	conta da fila	seja	quitada			
	conta da fila	seja	encerrada	por		recepcionista
limpeza [mesa]	mesa	seja	ocupada			

4.2. Identificar as Situações de Dependência Estratégica (*SDsituations*)

O objetivo da segunda etapa do método ERi*c é o refinamento e a organização das metas [Oliveira 08]. Essa etapa é composta por três subetapas: 1. Distinguir *SDsituations*, 2. Reconhecer as interdependências entre as *SDsituations* e 3. Construir o Diagrama de *SDsituations*.

Uma *SDsituation* define um bloco de elementos de dependência com intencionalidade (meta) situacional compartilhada [Oliveira 08].

4.2.1. Distinguir *SDsituations*

A Figura 6 exibe as *SDsituations* identificadas na modelagem do estudo de caso proposto com seus respectivos atores envolvidos.

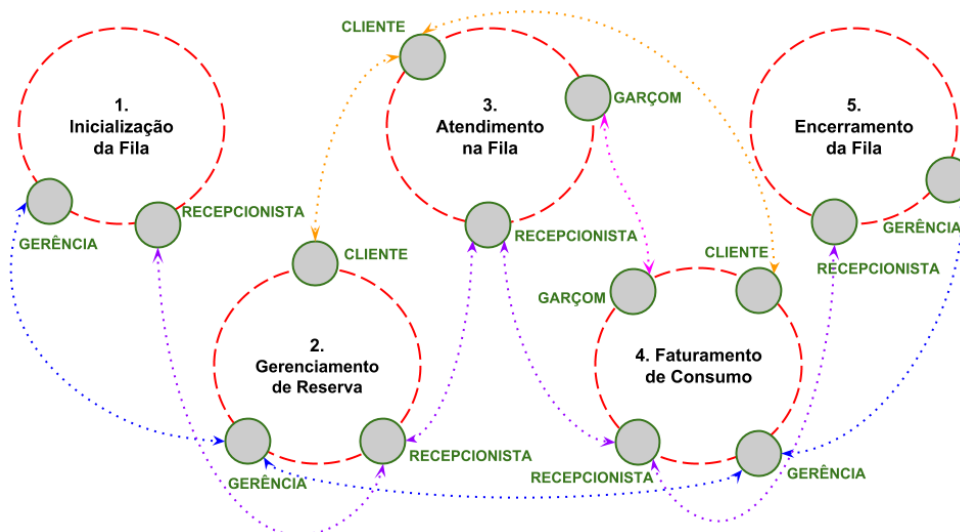


Figura 6. *SDsituations* e atores envolvidos

A atividade tem o propósito de identificar, em cada situação, um agrupamento de atores e suas metas em que ocorre uma dependência estratégica. Identificando como as metas estão ligadas entre si de forma a resultar em uma situação de negócio; uma situação estratégica de colaboração entre os atores; uma *SDsituation*.

No Quadro 7 estão identificadas as *SDsituations* para as metas refinadas anteriormente, na etapa 1.3.

Quadro 7. Identificação das *SDsituations*

<i>Depender</i>	<i>SDsituation</i>					<i>Dependee</i>
CLIENTE						
comunicabilidade [fila]	2	reserva	seja	feita		
transparência [fila]	2	reserva	seja	cancelada		
rápida atualização [fila]	2	reserva	seja	cancelada		
rapidez no preparo [pedido]	3	pedido	seja	servido	por	garçom
boas opções [item do cardápio]	3	pedido	seja	consumido		
flexibilidade [reserva]	2	reserva	seja	atendida	por	recepcionista
	4	conta da fila	seja	quitada		
	4	conta da fila	seja	encerrada	por	recepcionista
limpeza [mesa]	3	mesa	seja	ocupada		
GARCOM						
rapidez no atendimento [item do cardápio]	3	item do cardapio	seja	escolhido	por	cliente
rapidez no preparo [pedido]	3	pedido	seja	servido		
	3	item consumido	seja	registrado		
	4	item consumido	seja	cobrado		
limpeza [mesa]	3	mesa	seja	alocada	por	recepcionista
GERENCIA						
ideal acomodação [cliente]	1	fila	seja	organizada	por	recepcionista
rápida atualização [fila]	2	reserva	seja	excluída		
transparência [fila]	2	reserva	seja	excluída		
	4	conta da fila	seja	quitada	por	cliente
bom atendimento [fila]	4	cliente	seja	recompensado		
	5	reserva	seja	atendida	por	recepcionista
RECEPCIONISTA						
comunicabilidade [fila]	2	reserva	seja	feita	por	cliente
	1	fila	seja	autorizada	por	gerencia
	4	conta da fila	seja	cobrada	por	gerencia
	4	conta da fila	seja	transferida		
	4	conta da fila	seja	quitada	por	cliente
	4	conta da fila	seja	encerrada		
comunicabilidade [fila]	3	cliente	seja	notificado		
limpeza [mesa]	3	mesa	seja	alocada		
limpeza [mesa]	3	mesa	seja	ocupada	por	cliente
zerada [fila]	5	fila	seja	encerrada	por	gerencia

4.2.2. Reconhecer as Interdependências entre as *SDsituations*

Nesta subetapa, é verificado, em cada *SDsituation*, se há situações de dependência temporal, lógica ou sequencial com outra *SDsituation*, além de analisar a presença de algum paralelismo entre as *SDsituations*.

No estudo de caso proposto, identificamos que “Inicialização da Fila” é a *SDsituation* inicial para que todas as outras *SDsituations* possam acontecer. Em seguida, “Gerenciamento de Reserva” ocorre em paralelo com “Atendimento na Fila” e elas acontecem até que “Faturamento de Consumo” seja efetivado. Por fim, “Encerramento da Fila” é a última situação de dependência estratégica para o ciclo de um dia.

4.2.3. Construir Diagrama de *SDsituations*

A atividade tem como propósito a construção de um diagrama que apresente as interdependências sequenciais entre as *SDsituations*. A Figura 7 mostra o diagrama de *SDsituations* do domínio estudado.

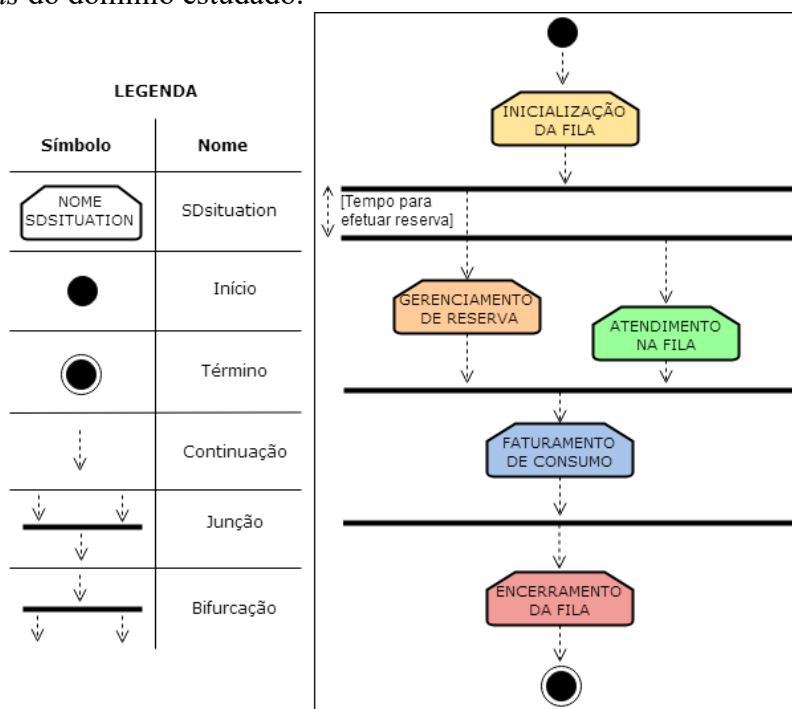


Figura 7. Diagrama de *SDsituations*

4.3. Modelar as Metas dos Atores

A finalidade da terceira etapa é a modelagem e avaliação das metas dos atores [Oliveira 08]. Esse passo é dividido em duas subetapas: 3.1. Identificar Agentes, Posições e Papéis e 3.2. Criar os Painéis de Intencionalidade.

4.3.1. Identificar Agentes, Posições e Papéis

Em cada *SDsituation*, os atores atuam reciprocamente para atingir a meta principal da situação identificada. Para esse fim, eles podem assumir posições e exercer papéis propícios para cada *SDsituation*.

Com a separação por *SDsituations* o Engenheiro de Requisitos lida com um número reduzido de metas por vez e, além disso, elas são restritas a situação, assim ele

pode perceber com mais clareza como os atores tem algum comportamento que pode ser classificado como especialização do ator, como agente, papel ou posição [Oliveira 08].

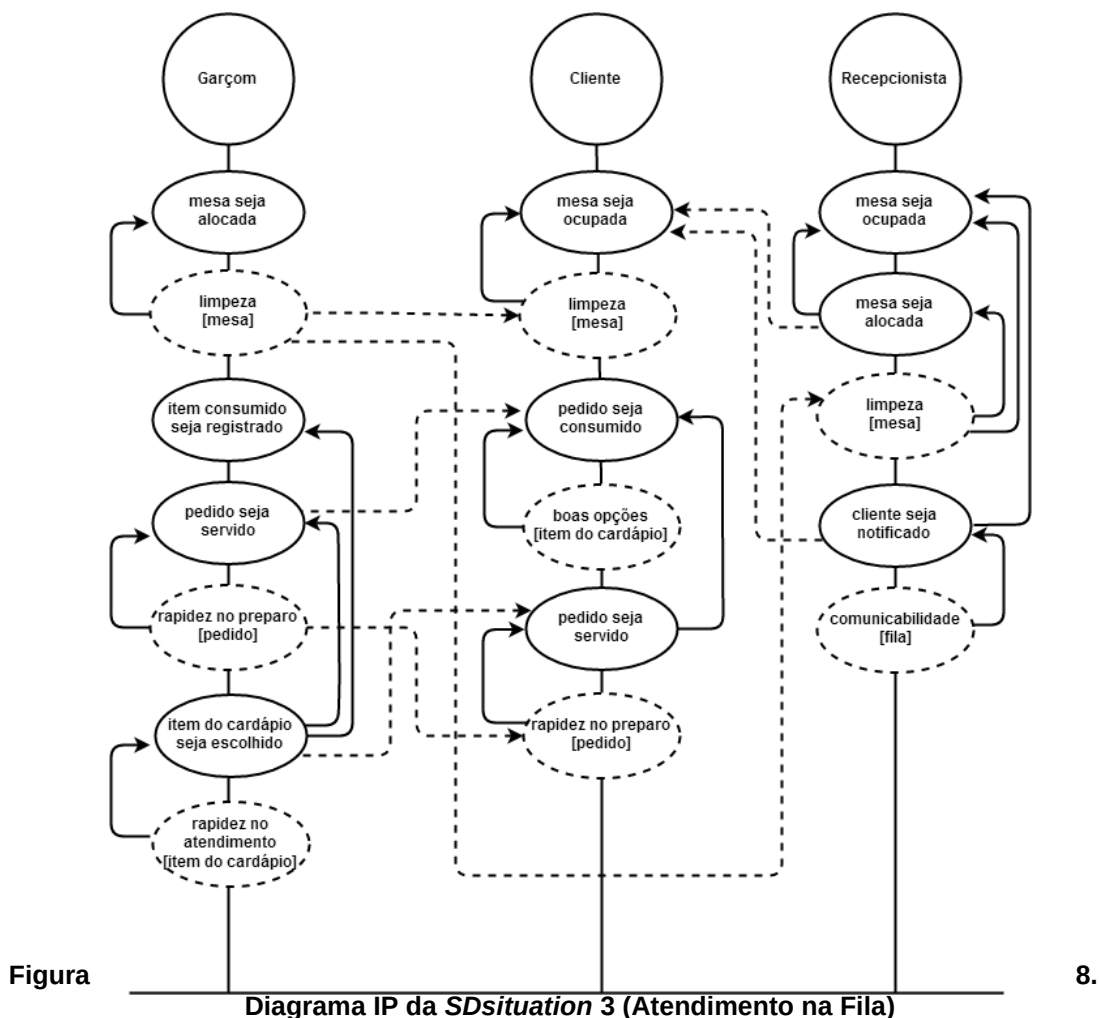
Metas exclusivas de um ator indicam um papel interpretado pelo ator. Um ou mais papéis, geralmente exclusivos a uma única *SDsituation*, podem estar relacionados a um cargo coberto (posição) por um agente da organização, que pode surgir em mais de uma *SDsituation*.

4.3.2. Criar os Painéis de Intencionalidade (Diagrama IP)

Essa atividade objetiva construir um diagrama para cada *SDsituation*. O Diagrama IP é uma redução do Modelo SR do *Framework iStar*, que será visto na etapa seguinte. Nele são considerados 3 elementos: os atores, as metas (concretas e flexíveis) e as relações entre elas. O Diagrama IP é um tipo de diagrama de transição de estados que também representa os atores. Ele é um diagrama de estados por ter estados (que são as metas) conectados pelas transições [Oliveira 07].

A principal motivação da criação do Diagrama IP é a representação da intencionalidade em um único e homogêneo diagrama [Oliveira 08].

Para exemplificar o conceito de Diagrama IP, a Figura 8 mostra um dos diagramas IP do estudo de caso estudado, acompanhado da descrição de cada item desse diagrama.



Figura

Diagrama IP da *SDsituation* 3 (Atendimento na Fila)

8.

A Figura 8 apresenta o Diagrama IP da ***SDsituation 3 – Atendimento na Fila***, neste diagrama é possível visualizar as relações de dependência estratégica entre três atores: Garçom, Cliente e Recepcionista.

Garçom tem como meta principal (ou meta fim) que ‘mesa seja alocada’, essa meta possui como atributo de qualidade a meta flexível ‘limpeza [mesa]’. Ele ainda possui as metas ‘item consumido seja registrado’, que só pode acontecer após ‘item do cardápio seja escolhido’, e ‘pedido seja servido’, essa última possui como atributo ‘rapidez no preparo [pedido]’, é item de dependência para Cliente em ‘pedido seja consumido’, e também ocorre apenas após ‘item do cardápio seja escolhido’. ‘Item do cardápio seja escolhido’ possui o atributo ‘rapidez no atendimento [item do cardápio]’, além de também ser item de dependência para Cliente.

Cliente tem como meta principal ‘mesa seja ocupada’, essa meta depende de duas outras metas do Recepcionista, sendo elas ‘mesa seja alocada’ e ‘cliente seja notificado’, e ainda possui como atributo ‘limpeza [mesa]’, essa última com dependência do Garçom. Cliente ainda possui a meta ‘pedido seja consumido’, que tem como atributo ‘boas opções [item do cardápio]’ e só ocorre após ‘pedido seja servido’, essa última meta possui o atributo ‘rapidez no preparo [pedido]’ que também depende do Garçom.

Recepcionista possui a meta principal ‘mesa seja ocupada’, essa meta tem como atributo ‘limpeza [mesa]’ com dependência do garçom, e somente acontece após as demais metas que são ‘mesa seja alocada’, também com atributo ‘limpeza [mesa]’, e ‘cliente seja notificado’, com atributo ‘comunicabilidade [fila]’.

4.4. Modelar a Racionalização das Metas dos Atores

O objetivo dessa etapa é construir o *rationale* dos atores. A quarta etapa do método ERI*c é formada por duas subetapas: 1. Construir Modelos SD e 2. Construir Modelos SR.

4.4.1. Construir Modelos SD

As etapas do Método ERI*c estão encadeadas, portanto, para construir os Modelos SD, o engenheiro de requisitos necessita do Diagrama de *SDsituations*, elaborado na etapa 4.2.3, e dos Diagramas IP, de cada *SDsituation*, construídos na etapa 4.3.2.

O propósito da atividade é definir as dependências estratégicas entre os atores, usando os critérios do *Framework iStar* [Yu 95].

O Modelo SD é construído da seguinte maneira: Para cada *SDsituation* do Diagrama de *SDsituations*, o engenheiro de requisitos deve observar as relações de dependência assinaladas no Diagrama IP e, para cada uma delas, definir entre os quatro tipos de dependência possíveis (por meta concreta, por meta flexível, por recurso ou por tarefa) qual delas é a mais indicada e vantajosa para o “*dependee*” [Oliveira 08].

Cada tipo de dependência tem uma particularidade, com relação à cooperação dos atores envolvidos. Na dependência por recurso, o *dependee* deve disponibilizar o recurso para o *dependee* utilizar. Na dependência por tarefa, o *dependee* realiza uma tarefa, seguindo as instruções definidas pelo *dependee*. Na dependência por meta concreta, o *dependee* não exerce influência sobre como o *dependee* executará a meta. Na dependência por meta flexível (*softgoal*), o *dependee* avalia e aceita se achar que a meta foi razoavelmente satisfeita.

Após a definição das dependências estratégicas, o Modelo SD pode ser construído. A Figura 9 ilustra o Modelo SD da **SDsituation 3 – Atendimento na Fila** do estudo de caso estudado, seguido de sua descrição.

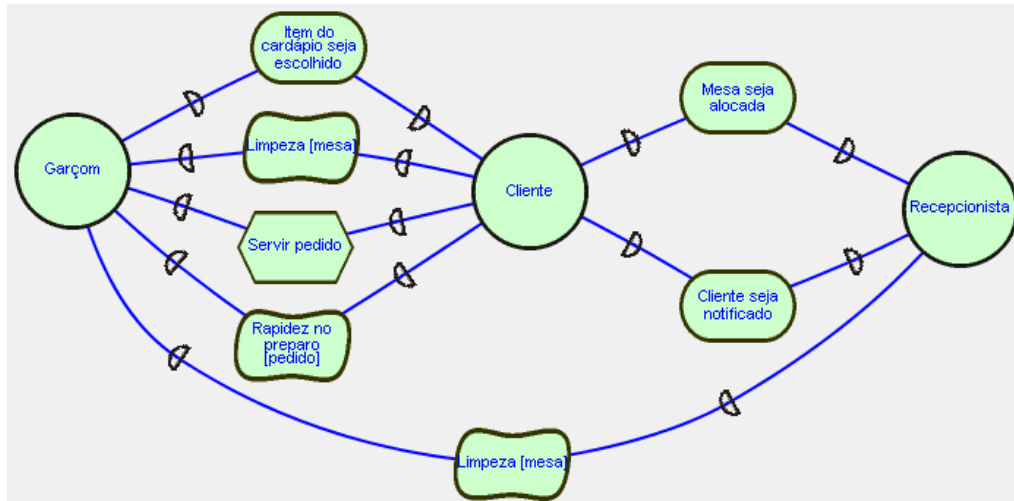


Figura 9. Modelo SD da SDsituation 3 (Atendimento na Fila)

O Diagrama IP da **SDsituation 3 - Atendimento na fila**, mostra quatro dependências estratégicas entre Garçom e Cliente, duas dependências entre Cliente e Recepcionista e uma entre Recepcionista e Garçom. As dependências foram representadas da seguinte maneira no Modelo SD, como visto na Figura 9:

Entre Garçom e Cliente:

- por meta concreta em ‘item do cardápio seja escolhido’, porque o Garçom não tem interferência na escolha dos itens pelo Cliente;
- por meta flexível em ‘limpeza [mesa]’, por ser atributo de qualidade;
- por tarefa em ‘servir pedido’, porque Cliente escolhe como será servido o pedido;
- por meta flexível em ‘rapidez no preparo [pedido]’, por ser atributo de qualidade.

Entre Cliente e Recepcionista:

- por meta concreta em ‘mesa seja alocada’, porque o Recepcionista pode realizar essa meta sem interferência do Cliente;
- por meta concreta em ‘cliente seja notificado’, porque o Cliente não tem interferência sobre como o Recepcionista realizará essa meta.

Entre Recepcionista e Garçom:

- por meta flexível em ‘limpeza [mesa]’, por ser atributo de qualidade.

4.4.2. Construir Modelos SR

O Diagrama IP, o Modelo SD e o Diagrama de *SDsituations* elaborados nas etapas anteriores do Método ERI*c, são itens fundamentais para elaboração do Modelo SR.

Primeiramente, posiciona-se todas as metas, concretas e flexíveis, que aparecem no eixo de cada ator do Diagrama IP dentro da linha limite do ator no Modelo SR. Após esse passo, é definida uma tarefa como meio para atingir cada meta concreta. Em seguida as metas são associadas à meta fim no Diagrama IP como decomposição de tarefa, ou

seja, os passos necessários para alcançar determinada tarefa. Ainda são alocadas como tarefas ou recursos as metas próprias do ator [Oliveira 08].

As associações representadas por uma linha com traço no Modelo SR indicam as decomposições de tarefas, e as associações representadas por uma seta indicam uma relação meio-fim, ou seja, a tarefa necessária para se alcançar a meta fim. A Figura 10 mostra o Modelo SR da *SDsituation 3 – Atendimento na Fila*.

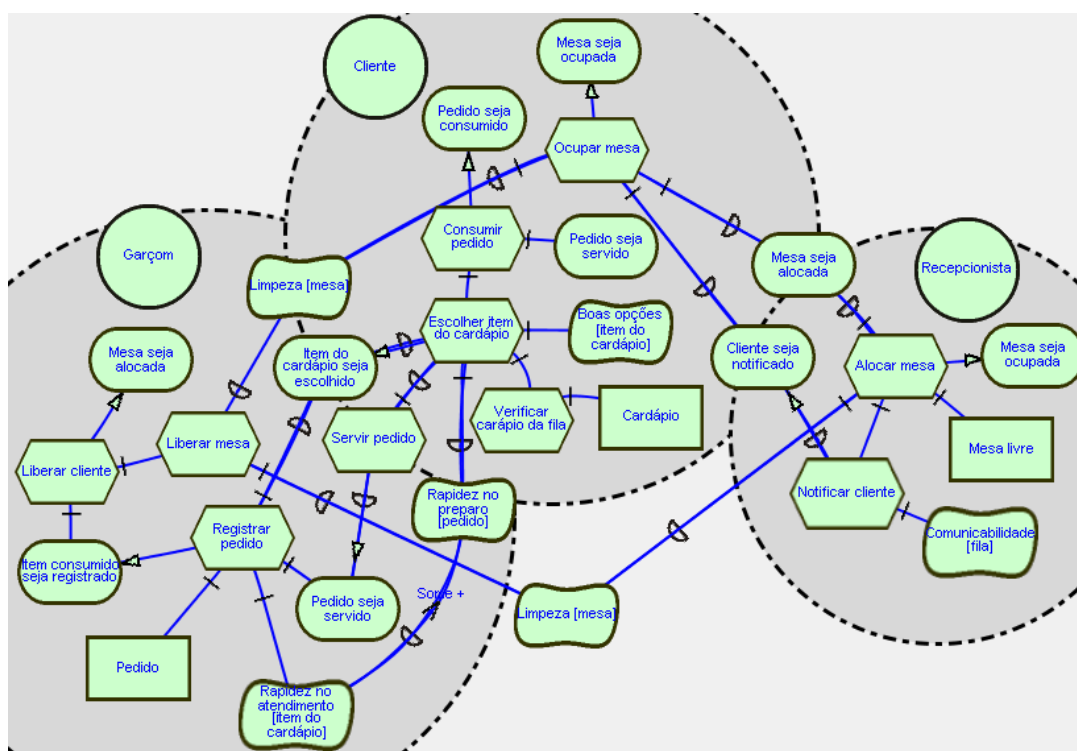


Figura 10. Modelo SR da *SDsituation 3* (Atendimento na Fila)

4.5. Especificar as *SDsituations*

A quinta etapa é composta por uma única subetapa: 1. Descrever as Situações de Dependência Estratégica. Seu objetivo, como o próprio nome indica, é a especificação das situações de dependência estratégica [Oliveira 08].

4.5.1. Descrever as Situações de Dependência Estratégica

A quinta etapa do método ERi*c consiste em descrever as *SDsituations* por meio de cenários. Para realizar essa tarefa, o engenheiro de requisitos utiliza os itens produzidos nas etapas anteriores, sendo eles o Diagrama de *SDsituations* e o Modelo SR de cada *SDsituation*.

O Quadro 8 apresenta o exemplo da especificação, em forma de cenário, da *SDsituation 3 – Atendimento na Fila*.

Quadro 8. Descrição da *SDsituation* Atendimento na Fila

Título: ATENDIMENTO NA FILA
Objetivo: <u>Cliente seja atendido</u>
Contexto:
Localização Geográfica: Restaurante
Localização temporal: Após <u>efetuar reserva</u> , enquanto aguarda <u>mesa livre</u>
Precondição: <u>Reserva</u> foi efetuada
Recursos: <u>Pedido</u> , <u>cardápio</u> , <u>mesa livre</u>
Atores: <u>Cliente</u> , <u>Garçom</u> e <u>Recepcionista</u>
Episódios: <u>Cliente faz pedido</u>
<u>Cliente</u> questiona sobre <u>item</u> do cardápio
<u>Garçom</u> responde sobre <u>item</u> do cardápio
<u>Cliente</u> escolhe <u>item</u> do cardápio
<u>Garçom atende cliente</u>
<u>Garçom</u> anota <u>pedido</u>
<u>Garçom</u> serve <u>pedido</u>
Restrição: Para <u>Recepcionista</u> alocar <u>cliente</u> a uma <u>mesa</u> , a <u>mesa</u> precisa estar limpa.
Exceção: Não há.
Metas flexíveis: Comunicabilidade [fila], boas opções [item do cardápio], rapidez no atendimento [item do cardápio], limpeza [mesa], rapidez no preparo [pedido]

4.6. Analisar a Racionalização das Metas dos Atores

A sexta e última etapa do Método ERi*c tem como objetivo a verificação dos modelos *iStar* para a melhoria da qualidade [Oliveira 08], com a finalidade de prevenir defeitos ao invés de detectá-los.

Essa última etapa é dividida em três subetapas: 1. Identificar Estruturas Canônicas, 2. Aplicar o *Framework* de Perguntas e 3. Verificar as Perguntas Respondidas.

4.6.1. Identificar Estruturas Canônicas (*SRconstructs*)

Um *SRconstruct* possui um fim que será alcançado por um ou mais meios, estes meios são as tarefas. Para não haver repetição, não são consideradas as metas concretas subcomponentes da decomposição de tarefas, porque estas são o elemento principal de análise de outro *SRconstruct*.

Para a *SDsituation 3 – Atendimento na Fila*, os *SRconstructs* existentes estão identificados abaixo, organizados por ator:

- Cliente: mesa seja ocupada, pedido seja consumido, item do cardápio seja escolhido.
- Recepcionista: mesa seja ocupada, cliente seja notificado.
- Garçom: mesa seja alocada, item consumido seja registrado, pedido seja servido.

Totalizando para o exemplo oito *SRconstructs*.

4.6.2. Aplicar *Framework* de Perguntas

Esta subetapa consiste em dividir os diagramas i^* em problemas menores, para que se faça uma análise minuciosa de cada parte do diagrama. É aplicado então um *framework* de perguntas para cada *SDsituation* e para cada *SRconstruct* da *SDsituation*.

Para melhor distribuição na página, o *framework* de perguntas aplicado à ***SDsituation 3 - Atendimento na Fila*** foi separado em três partes. O Quadro 9a contém as questões externas e os Quadros 9b e 9c contém as questões internas. O *framework* de perguntas do *SRconstruct* Mesa seja Alocada foi separado em duas partes pelo mesmo motivo, o Quadro 10a contém as questões externas e 10b contém as questões internas.

Quadro 9a. Diagnóstico da *SDsituation 3* Atendimento na Fila – Questões Externas

DIAGNÓSTICO DA SDSITUATION:

***SDsituation*: “Atendimento na fila”**

I. QUESTÕES EXTERNAS

1. Quem mais poderia colaborar com o “**Recepcionista**” para atingir “**Cliente seja atendido**”? Quanto ele pode colaborar? (completamente ou parcialmente)
 - Garçom colabora parcialmente.
2. Por que o “**Garçom**” colabora com o “**Recepcionista**” para atingir “**Cliente seja atendido**”?
 - Para agilizar o atendimento de clientes na fila.
3. Quais *SDsituations* acontecem antes de “**Atendimento na Fila**”?
 - As *SDsituations*: Inicialização da Fila e Gerenciamento de Reserva.
4. Que problemas com as *SDsituations* anteriores podem ser identificados para atingir “**Cliente seja atendido**”?
 - Gerência demora a autorizar a inicialização da fila.
 - Dificuldade no manuseio do aplicativo.
5. E se o “**Garçom**” não puder colaborar no “**Atendimento na Fila**”?
 - Garçom deve receber o devido treinamento.

Quadro 9b. Diagnóstico da *SDsituation 3* Atendimento na Fila – Questões Internas

II. QUESTÕES INTERNAS

6. Quais são os problemas dentro do “**Atendimento na Fila**”? Que tipos de problemas (precisão, deficiências, ambiguidades ou omissões) são identificados em “**Cliente seja atendido**”?
 - Travamento/lentidão do sistema
 - Recepcionistas e garçons sem o treinamento adequado.
 - Falta de comunicabilidade com o cliente.

RELAÇÃO GARÇOM - CLIENTE (DEPENDER-DEPENDEE)

7. Que detalhes o “**garçom**” necessita?
 - a) Caso: dependência de recurso - não se aplica.
 - b) Caso: dependência de meta concreta - Quais são os problemas que o “**item do cardápio seja escolhido**” tem para ser alcançada pelo “**cliente**”? (tempo, habilidade) Quando? Como?
 - Não ter disponível todas as opções do cardápio.
 - c) Caso: dependência de meta flexível - não se aplica.
 - d) Caso: dependência de tarefa - não se aplica.

RELAÇÃO RECEPCIONISTA - GARÇOM (DEPENDER-DEPENDEE)

8. Que detalhes o “**recepcionista**” necessita?
 - a) Caso: dependência de recurso - não se aplica.
 - b) Caso: dependência de meta concreta - não se aplica.
 - c) Caso: dependência de meta flexível - Quais são os problemas que o “**limpeza [mesa]**” tem para ser razoavelmente satisfeita pelo “**garçom**”? (capacidade) Existe “**limpeza [mesa]**” na conclusão de “**cliente seja atendido**”? Por que? Quem está demandando pela meta flexível?
 - Dificuldade para liberar a mesa no aplicativo.
 - d) Caso: dependência de tarefa - não se aplica.

Quadro 9c. Continuação do Diagnóstico da *SDsituation 3* Atendimento na Fila – Questões Internas

RELAÇÃO CLIENTE - RECEPCIONISTA (*DEPENDER-DEPENDEE*)

9. Que detalhes o “**cliente**” necessita?
- Caso: dependência de recurso – não se aplica.
 - Caso: dependência de meta concreta - Quais são os problemas que o “**mesa seja alocada**” tem para ser alcançada pelo “**receptionista**”? (tempo, habilidade) Quando? Como? Quanto?
 - Mesa não estar em condições apresentáveis.
 - Travamento/lentidão do sistema.
 - Receptionista com dificuldade em manusear o aplicativo.
- Quais são os problemas que o “**cliente seja notificado**” tem para ser alcançada pelo “**receptionista**”? (tempo, habilidade) Quando? Como? Quanto?
- Falta de comunicabilidade com o cliente.
- Caso: dependência de meta flexível – não se aplica.
 - Caso: dependência de tarefa – não se aplica.

RELAÇÃO CLIENTE - GARÇOM (*DEPENDER-DEPENDEE*)

10. Que detalhes o “**cliente**” necessita?
- Caso: dependência de recurso – não se aplica.
 - Caso: dependência de meta concreta – não se aplica.
 - Caso: dependência de meta flexível - Quais são os problemas que o “**limpeza [mesa]**” tem para ser razoavelmente satisfeita pelo “**garçom**”? (capacidade) Existe “**limpeza [mesa]**” na conclusão de “**cliente seja atendido**”? Porque? Quem está demandando pela meta flexível?
 - Dificuldade para liberar a mesa no aplicativo.
 - A contribuição é manter a mesa bem apresentada.
- Quais são os problemas que o “**rapidez no preparo [pedido]**” tem para ser razoavelmente satisfeita pelo “**garçom**”? (capacidade) Existe “**rapidez no preparo [pedido]**” na conclusão de “**cliente seja atendido**”? Porque? Quem está demandando pela meta flexível?
- Demora para preparar o pedido.
- Caso: dependência de tarefa – O “**garçom**” recebeu as orientações de como fazer “**servir pedido**”? Pode o “**garçom**” realizar a tarefa? (tempo, habilidade)
 - O cliente informa ao garçom como deseja que o pedido seja servido.
 - Garçom deve receber treinamento de como atender clientes.
11. Qual dependência possui a maior responsabilidade para atingir “**cliente seja atendido**”?
- A meta concreta “**mesa seja alocada**”.
- Por que?
- Porque o cliente depende enormemente dessa meta para ser atendido.

Quadro 10a. Diagnóstico do *SRconstruct* Mesa seja Alocada – Questões Externas

DIAGNÓSTICO SRCONSTRUCT:

***SRconstruct*: “mesa seja alocada”**

I. QUESTÕES EXTERNAS

12. Quem mais tem como meta “**mesa seja alocada**”?
- Ninguém.
13. Quais são as alternativas possíveis para que “**mesa seja alocada**” seja atingida? Por quê?
- Mesa será alocada via aplicativo.
14. Quais são os elementos de dependência dos *dependees*?
- Meta concreta: Item consumido seja registrado. Tarefa: liberar mesa.
15. Que tipos de problemas (precisão, deficiências, ambiguidades ou omissões) podem ser vislumbrados? E se os recursos ficarem indisponíveis? Como evitar tais problemas?
- Pedidos consumidos que ainda não foram registrados.
16. E se a “**mesa seja alocada**” for partilhada com outro ator?
- Não se aplica.
17. Que outro construto depende dessa meta? Por quê? Quanto?
- Nenhum.

Quadro 10b. Diagnóstico do *SRconstruct* Mesa seja Alocada – Questões Internas**II. QUESTÕES INTERNAS (PARA CADA TAREFA MEIO)**

18. Quais são os problemas com a tarefa “**liberar cliente**”? Por quê?
- Dificuldade do garçom em liberar clientes pelo aplicativo.
19. Para a tarefa “**liberar cliente**” que componentes são necessários para atingir “**mesa seja alocada**”?
- Tarefa: Liberar Mesa. Meta concreta: Item consumido seja registrado.
- a) Caso: recurso – não se aplica.
b) Caso: meta flexível – não se aplica.
c) Caso: subTarefa – Pode o ator “**garçom**” fazer “**liberar mesa**”? (tempo, habilidade)
- Sim, desde que a mesa esteja limpa.
20. Existe algum detalhe omitido na operacionalização? De que tipo? Por quê? Como? Quanto?
- O garçom precisa estar atento na liberação da mesa, para que outro cliente seja acomodado rapidamente.
21. Há falta de algum recurso? De que tipo? E se o recurso não estiver disponível?
- Não há.

4.6.3. Verificar as Perguntas Respondidas

A partir dos diagnósticos aplicados na subetapa anterior, é possível elaborar o Quadro de Metas x Problemas. O Quadro consiste em identificar quais metas são atingidas pelos problemas observados nas *SDsituations* e *SRconstructs*. Com o Quadro em mãos é possível visualizar mais facilmente deficiências do sistema e problemas do negócio, podendo assim reparar esses erros, antes de desenvolver a aplicação.

O Quadro 11 é a legenda de identificação das metas do Quadro 12. O Quadro 12 mostra as Metas x Problemas elaborado para a *SDsituation 3 – Atendimento na Fila*, a partir das perguntas respondidas no passo anterior. As metas concretas estão numeradas e as metas flexíveis estão identificadas por letras.

Quadro 11. Legenda para identificação das metas no Quadro Metas x Problemas

# Metas	Nomes das Metas (<i>SDsituation 3</i> Atendimento na Fila)
1	Mesa seja ocupada
2	Pedido seja consumido
3	Item do cardápio seja escolhido
a	Boas opções [item do cardápio]
4	Mesa seja ocupada
5	Cliente seja notificado
b	Comunicabilidade [fila]
6	Mesa seja alocada
7	Item consumido seja registrado
8	Pedido seja servido
c	Rapidez no atendimento [item do cardápio]
d	Rapidez no preparo [pedido]
e	Limpeza [mesa]

Quadro 12. Quadro Metas x Problemas da *SDsituation 3* Atendimento na Fila

Problemas	Metas →	1	2	3	a	4	5	b	6	7	8	c	d	e
(1) Interface que dificulte o manuseio		x				x	x	x	x			x		
(2) Travamento/lentidão do sistema		x				x	x	x	x			x		
(3) Dificuldade do garçom ou recepcionista no manuseio do aplicativo		x				x	x	x	x			x		
(4) Gerência demora a autorizar a inicialização da fila		x				x								
(5) Recepcionistas e garçons sem o treinamento adequado										x	x	x	x	x
(6) Falta de comunicabilidade com o cliente		x				x	x	x						
(7) Não ter disponível todas as opções do cardápio				x	x						x			
(8) Demora em servir o pedido			x								x			
(9) Demora em liberar mesas		x				x			x					
(10) Garçom troca ou toma nota dos pedidos erradamente			x								x			
(11) Garçom não registra pedidos consumidos ou registra erradamente										x				
(12) Falta, em estoque, do item do cardápio escolhido ou dos ingredientes necessários				x	x							x		

5. Conclusões

O objetivo do artigo foi apresentar a aplicação do método Engenharia de Requisitos Intencional (ERi*c). Para demonstração e melhor entendimento do método, aplicamos todas as etapas do método ERi*c a um sistema que tem como objetivo o gerenciamento da fila de espera de um restaurante. Mostramos também neste artigo, os diagramas e quadros desenvolvidos ao longo do processo.

A elaboração de modelos i* pode ser muito complexa [Pastor 11], por isso o método ERi*c surgiu como uma alternativa para simplificar a produção desses modelos dividindo seu desenvolvimento em etapas menos elaboradas. O propósito foi de concluir a última etapa do método com os modelos i* prontos e com uma análise detalhada das possíveis deficiências da aplicação desejada, sejam elas falhas do sistema em si ou problemas do negócio. Sendo possível então tomar alguma medida para reparar tais problemas, antes que atinjam de fato a aplicação durante a implementação ou produção do sistema.

Para exemplificar, analisando os dados obtidos no Quadro Metas x Problemas da *SDsituation 3 – Atendimento na Fila* (Quadro 12), é possível verificar as metas com maior incidência de problemas e os problemas que atingem mais metas, ambos em destaque no Quadro 12. As metas 1 e 4, são atingidas por seis problemas cada. O problema 5 atinge cinco metas, os problemas 1, 2 e 3 atingem seis metas e o problema 6 atinge quatro metas cada. Nota-se que os problemas 1, 2 e 3 são problemas que atingem o sistema, já os problemas 5 e 6 são problemas que afetam o negócio. O levantamento dos problemas com maior incidência e das metas mais atingidas em todas as *SDsituations* são

importantes, pois permitem ao engenheiro de requisitos saber quais áreas do sistema devem ter a correção priorizada, e para que, nesse caso, o restaurante busque uma solução para os problemas de negócio identificados, antes que afetem o atendimento.

Analisando as demais etapas do método notamos a importância de um amplo conhecimento do estudo de caso abordado para elaboração do léxico ampliado da linguagem na primeira etapa, visto que o LAL é a base para o desenvolvimento das etapas seguintes. A ferramenta *AGFL Tool* [Oliveira 17] nos ajudou a extrair as metas dos atores a partir dos impactos dos símbolos, tornando mais ágil o refinamento das metas dos atores. As *SDsituations* identificadas na segunda etapa ajudaram a melhor entender e dividir o funcionamento do sistema em blocos com mesma intencionalidade. Podendo assim agrupar as metas de acordo com os blocos identificados. Na terceira etapa foram elaborados os Diagramas IP, para desenvolvê-los utilizamos os quadros e diagramas produzidos nas etapas anteriores. O Diagrama IP nos permitiu visualizar melhor as relações de dependência estratégica entre as metas de cada ator. Os modelos SD e SR foram construídos na quarta etapa do método, o modelo SD direcionou a construção do modelo SR que é bastante complexo, o que torna seu desenvolvimento mais trabalhoso. Como o objetivo do método *ERi*c* é auxiliar a construção de modelos *i**, nesta etapa o próprio método fornece um passo a passo detalhado para construir os modelos SD e SR, utilizando como base os diagramas elaborados nas etapas anteriores. Ao especificar as *SDsituations* na quinta etapa, temos uma descrição detalhada que facilita o entendimento de cada *SDsituation*.

A análise do quadro Metas x Problemas, obtido na última etapa do método, mostra sua relevância na verificação das falhas do sistema. Essa análise também enfatiza a importância das etapas anteriores, pois os diagramas produzidos são necessários para elaboração do quadro final e para compor o documento de requisitos. Ressaltando assim a importância de todas as etapas do método *ERi*c*. Durante a aplicação do método, foi possível observar também uma menor complexidade e maior clareza na elaboração dos modelos *i** [Pastor 11] seguindo os *templates* oferecidos pelo método. Concluímos assim que os resultados alcançados com a modelagem do aplicativo do estudo de caso apresentado, mostram que o método *ERi*c* é eficiente e consegue atingir o objetivo a que se propõe.

Oliveira, em [Oliveira 08b], fornece um guia mais detalhado das etapas do método *ERi*c*, e foi uma importante fonte para o artigo corrente.

Para compor o projeto de conclusão de curso modelamos os quadros e diagramas de todas as *SDsituations* do estudo de caso proposto. Contudo, apenas alguns deles foram apresentados no artigo devido ao limite de páginas. Nosso trabalho oferece como contribuição, a descrição de um novo exemplo da aplicação do método *ERi*c*, bem como um conjunto de quadros e diagramas úteis, capazes de auxiliar em uma futura implementação do sistema de fila de espera transparente em um restaurante. Todos os quadros e diagramas desenvolvidos e não presentes neste artigo estão disponíveis no repositório *ERicMethod* do *GitHub* [ERicMethod 19].

Referências:

[C&L 18] Cenários e Léxicos - PUC-Rio - Disponível em: <http://pes.inf.puc-rio.br/cel/>. Acessado em: Janeiro/2018.

- [ERicMethod 19] Repositório disponível em: <https://github.com/ERicMethod> Acessado em: Abril/2019.
- [Kotonya 98] Kotonya, G., & Sommerville, I. (1998). *Requirements engineering: processes and techniques*. Wiley Publishing.
- [Lamsweerde 98] Van Lamsweerde, A., Darimont, R., & Letier, E. (1998). Managing conflicts in goal-driven requirements engineering. *IEEE transactions on Software engineering*, 24(11), 908-926.
- [Lamsweerde 01] Van Lamsweerde, A. (2001). Goal-oriented requirements engineering: A guided tour. In *Requirements Engineering, 2001. Proceedings. Fifth IEEE International Symposium on* (pp. 249-262). IEEE.
- [Leffingwell 97] Leffingwell, D. (1997). Calculating your return on investment from more effective requirements management. *American Programmer*, 10(4), 13-16.
- [Leite 95] Leite, J. C. S. D. P., & Oliveira, A. D. P. A. (1995). A client oriented requirements baseline. In *Requirements Engineering, 1995., Proceedings of the Second IEEE International Symposium on* (pp. 108-115). IEEE.
- [Leite 00] Leite, J. C. S. D. P., Hadad, G. D., Doorn, J. H., & Kaplan, G. N. (2000). A scenario construction process. *Requirements Engineering*, 5(1), 38-61.
- [Oliveira 07] Oliveira, A. D. P. A., Leite, J. C. S. D. P., Cysneiros, L. M., & Cappelli, C. (2007). Eliciting multi-agent systems intentionality: from language extended lexicon to i* models. In *Chilean Society of Computer Science, 2007. SCCC'07. XXVI International Conference of the* (pp. 40-49). IEEE.
- [Oliveira 08] Oliveira, A. D. P. A. (2008). *Intentional Requirements Engineering: A Method for Requirements Elicitation, Modeling, and Analysis*. 261 p (Doctoral dissertation, Doctoral Thesis–Computer Science Department, PUC-Rio-Rio de Janeiro).
- [Oliveira 08b] Oliveira, A. D. P. A., Leite, J. C. S. D. P., & Cysneiros, L. M. (2008). Método ERi* c-Engenharia de Requisitos Intencional. In *WER*.
- [Oliveira 17] Oliveira, A. D. P. A., Leite, J. C. S. D. P., Cysneiros, L. M., & Da Silva, W. G. S. (2017). Eliciting Goals and Softgoals-How to Perceive the Intentionality at the Beginning of the Journey. In *iStar* (pp. 31-36).
- [Pastor 11] Pastor, O., Estrada, H., & Martinez, A. (2011). The strengths and weaknesses of the i* framework: an experimental evaluation. *Social Modeling for Requirements Engineering. Cooperative Information Systems series. MIT Press, Cambridge*.
- [Ramesh 01] Ramesh, B., & Jarke, M. (2001). Toward reference models for requirements traceability. *IEEE transactions on software engineering*, (1), 58-93.
- [Yu 95] Yu, E. (2011). Modelling strategic relationships for process reengineering. *Social Modeling for Requirements Engineering*, 11, 2011.