

4 Uso das Representações

Este Capítulo descreve as diferentes representações na modelagem do SimulES ao longo das diferentes evoluções e versões que o jogo já teve; umas abordagens baseadas em linguagem natural de léxicos e cenários e outra abordagem de orientação a metas através do *framework i**, base para a representação das intencionalidade dos atores. Ao longo do Capítulo, será apresentada a visão do uso de léxicos e cenários nas primeiras versões do jogo e sua evolução e uso na segunda representação com o *framework i**, apresentaremos seus modelos SD e SR, além de novas técnicas, métodos e conceitos que estendem o supracitado *framework*: o modelo SA (*Strategic Actor*), situações de dependências estratégicas (*SDsituations*) e painéis de intencionalidade (diagramas IP).

4.1. Contextualização

O jogo SimulES conta com diferentes versões que estão acompanhadas com sua respectiva representação ao longo de sua vida, a primeira versão do jogo proposta aparece no trabalho de Figueiredo [5]. Esta versão é identificada como a versão 1.0 do jogo, para a representação do jogo foi utilizado léxicos e cenários.

A versão 1.0 foi a base para a evolução do SimulES apresentada em [4] léxicos e cenários propostos nesta versão (versão 2.0) foram utilizados para realizar nossa atividade na UERJ como foi descrito no Capítulo 3, além disso, esta versão também foi utilizada para o entendimento do domínio e análise da representação de léxicos e cenários ao longo deste trabalho. Finalmente esta versão foi base dos primeiros modelos intencionais do jogo apresentados por Napolitano em [44], sendo esta última a versão 3.0 do jogo.

Nosso trabalho apresenta a versão 4.0 do jogo que também será chamada de SimulES-W.

4.2. Representação de Requisitos em Linguagem Natural

4.2.1. Léxico Ampliado da Linguagem (LAL)

LAL (*Language Extended Lexicon*) [23] é uma técnica para descrever os símbolos de uma linguagem. Usada nas diferentes evoluções do Simules, é útil pois permite a identificação de palavras ou frases do contexto social, conhecido como o Universo de Informação (UdI).

Conforme [23], para o Simules o LAL, na fase de requisitos, tem fornecido uma organização explícita de conceitos e relações de contexto, o fator crítico de sucesso na elicitacão, modelagem e validacão dos conceitos relacionados ao UDI.

Ao longo das evoluções do Simules [5],[4],[44] tem sido utilizados léxicos para descrever os símbolos do contexto do jogo.

Conforme o ilustrado na Figura 21 que corresponde à versão 2.0 do Simules, por exemplo, o símbolo jogador é considerada uma palavra chave dentro do ambiente de Simules, nesta versão do LAL o jogador pode assumir dois papéis o jogador da vez e adversário.

Nome	jogador
	Participante do jogo Simules
	Tem por objetivo vencer o jogo
Noção	jogador pode assumir papel de jogador da vez
	jogador pode assumir papel de adversario
	jogador da vez possui adversários
Classificacão	Sujeito
	jogador JOGA RODADA DE AÇÖES
	jogador JOGA RODADA DE CONCEITOS
	jogador INTEGRA ARTEFATOS EM UM.MODULO
Impacto	jogador TRATA PROBLEMAS
	jogador contrata engenheiro de software
	jogador demite engenheiro de software

Figura 21 – Representacão no CEL do léxico usando a palavra chave “Jogador” v 2.0 [4].

Na terceira versão do jogo Simules apresentada no trabalho [44] novamente as palavras foram analisadas, classificadas em sujeito, verbo, estado e objeto, e uma nova versão foi apresentada, isso deu origem a um léxico mais depurado.

Na Figura 22 é ilustrado o refinamento do símbolo jogador (versão 3.0 do Simules), além de apresentar papéis do jogador eles foram detalhados ainda mais para dar maior entendimento ao símbolo, que é vital no desenvolvimento do jogo, pois ele é um ator na maioria dos cenários.

O LAL apresentado no trabalho [44] serviu de base para o início da implementação do protótipo, já o léxico evoluído após do desenvolvimento do SimulES-W está detalhado.

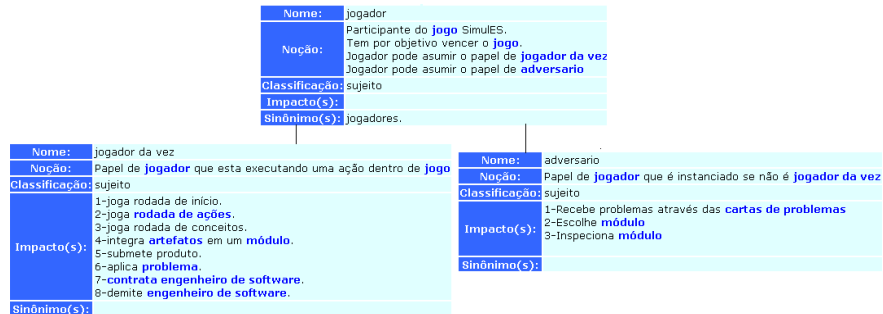


Figura 22 – Representação no CEL do léxico usando a palavra chave “Jogador” v 3.0 [44].

4.2.2. Cenários

Cenários tem sido empregados especificar o comportamento do SimulES [5], [4], [44], pois sua estrutura permite a descrição das diferentes situações do jogo com a vantagem que os cenários estão ligados diretamente ao LAL usando os símbolos nele definidos.

Na Figura 23 podemos ver os cenários definidos para o jogo na versão 2.0 [4], uma seqüência de execução foi determinada e um cenário que estabelece essa seqüência foi estabelecido, esse cenário é nomeado como *Dinâmica do SimulES* e é conhecido como cenário integrador.

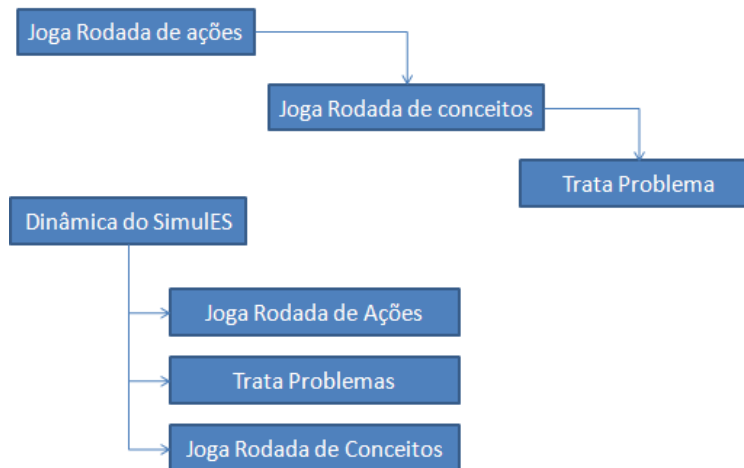


Figura 23 – Exemplo do comportamento dos cenários do jogo v 2.0 [4].

4.2.3. C&L

O C&L é um software para edição de símbolos de léxico e de descrição de cenários [22]. Sendo um software que disponibiliza um ambiente em que usuários podem interagir para construir, manter, evoluir e gerenciar projetos contendo cenários e símbolos do léxico. Tem sido possível usá-lo para modelar os diferentes elementos do jogo Simules.

O primeiro protótipo do C&L foi construído pela PUC-Rio no 2002 e atualmente é o resultado da evolução de protótipos construídos por estudantes da graduação, mestrado e doutorado do Departamento de Informática da PUC-Rio ao longo destes anos. O C&L foi desenvolvido desde o início com a filosofia de software livre e seu código fonte está disponível de forma aberta.

A Figura 24 representa a interface gráfica da ferramenta C&L utilizada nas diferentes etapas de modelagem do Simules.

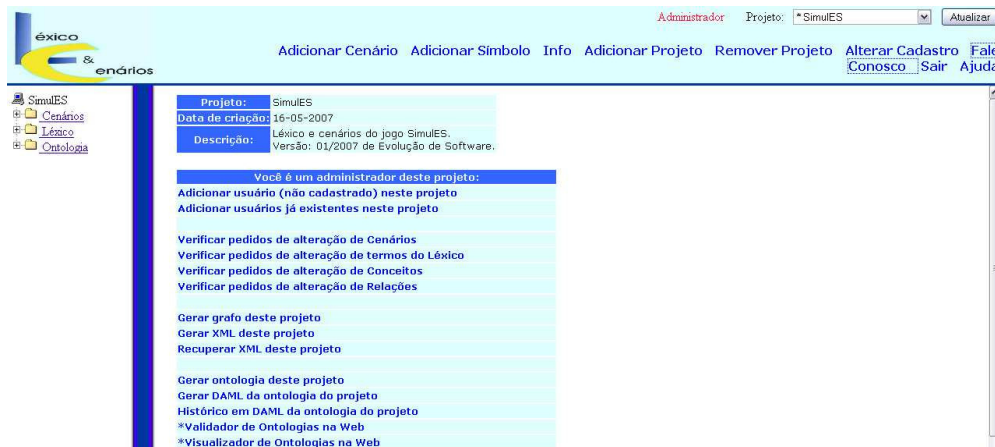


Figura 24 – Interface do C&L usado para modelar elemento do jogo Simules.

4.3. Representação Intencional dos Requisitos

A intencionalidade pode ser definida como a motivação ou interesses dos atores em uma organização. Essas motivações ou interesses são representados e modelados através de metas. A modelagem intencional baseia-se na visualização do contexto organizacional e das relações de dependência entre os atores. Além disso, esta modelagem permite-nos ver como os atores dependem uns dos outros

para que os objetivos destas organizações sejam alcançados, os recursos sejam disponíveis, as tarefas sejam feitas e as metas flexíveis cumpridas razoavelmente.

4.3.1. Framework i^*

Como foi apresentado no capítulo de introdução o *framework i^** é habitualmente usado para modelar contextos organizacionais baseados nos relacionamentos entre os atores. Portanto, modelar em i^* tem permitido obter uma melhor compreensão dos relacionamentos dos jogadores e demais atores do jogo nas etapas iniciais de desenvolvimento.

Atores em i^* são participantes ativos do contexto organizacional, em nosso caso o ambiente do jogo, além disso, em i^* é possível representar como os atores estão obrigados a alcançar objetivos através do exercício das suas competências e conhecimentos, este último um dos motivos da escolha desta modelagem, pois queremos criar modelos que refletem a interação entre os jogadores.

O *framework i^** propõe dois modelos, o modelo SD (por sua sigla em inglês *Strategic Dependency*) e o modelo SR (por sua sigla em inglês *Strategic Rationale*), modelos intencionais iniciais do SimULES são apresentados no trabalho [44] e foram obtidos através do uso do método ERi^*c [11].

4.3.2. O Método ERi^*c

O método Engenharia de Requisitos Intencional – ERi^*c [11] assim como o *framework i^** está baseado no conceito de intencionalidade. O método propõe a construção dos modelos i^* através de seis etapas. Este método fornece algumas contribuições importantes para processo de construção, como as situações de dependência estratégica (*SDsituations*), a técnica AGFL (*Agent Goals From Lexicon*) para elicitare metas dos atores, painéis de intencionalidade (diagramas IP), entre outras contribuições.

Além disso, é um método simples de empregar no qual o engenheiro de requisitos é auxiliado através das etapas para criar os modelos i^* ocultando um pouco a sua complexidade para que pessoas menos familiarizadas com este tipo de modelagem também possam fazer uso dele, baseado no conceito de intencionalidade também reflete as motivações e interesse dos atores.

Para explicar as etapas do método *ERi*c*, esta dissertação irá utilizar os modelos iniciais da modelagem intencional proposta para o jogo SimulES em [44] que corresponde à versão 3.0 do jogo. A Figura 25 apresenta um esquema simplificado do método.

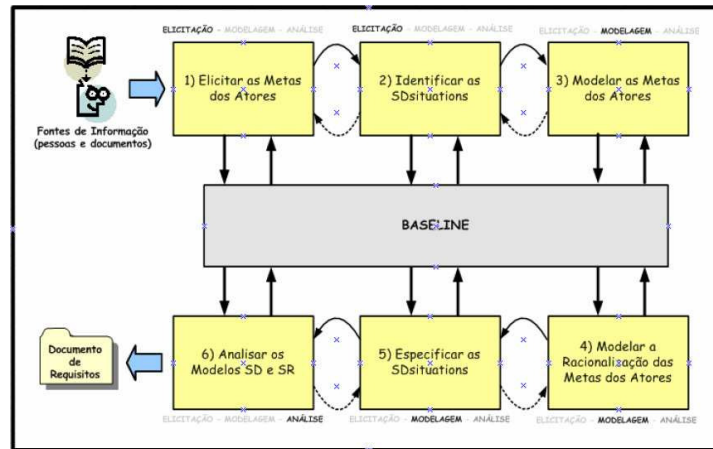


Figura 25 – Visão geral do método *ERi*c* [11].

A Figura 25 ilustra as etapas do método, que foram seguidas para a primeira versão dos modelos intencionais [44]. (i) Na primeira etapa, *elicitar metas dos atores*, usando a técnica AGFL, é feita a elicitação a partir do LAL. (ii) Na segunda etapa, *identificar as SDsituations*, define blocos de dependência que compartilham intencionalidade situacional. (iii) Na etapa três, *modelar as metas dos atores*, apresenta os painéis de intencionalidade, ou diagramas IP, modelando em um diagrama apenas metas concretas e flexíveis e suas relações para cada um das *SDsituations* identificadas. (iv) Na quarta etapa do método, *modelar a racionalização das metas dos atores*, acontece o detalhamento de metas concretas e flexíveis o que deriva na criação de modelos SR tomando como base os diagramas IP, assim como também os modelos SD. A quinta etapa, *especificar as SDsituations*, descreve as situações de dependência estratégica através da técnica de cenários [18]. E a sexta etapa usa a técnica da análise *i* diagnoses* para verificar modelos SD e SR.

4.3.3. Modelagem Intencional do SimulES

Procurávamos uma modelagem para a representação da elicitación e validación posterior onde fosse possível ganhar em rastreabilidade além de poder aplicar conceitos de transparência. O uso da modelagem intencional auxilia a transparência. A análise das representações usadas nos outros jogos nos mostra que o uso da modelagem intencional é inovador.

4.3.3.1. Elicitar as Metas de Atores

A primeira etapa proposta pelo método divide-se em três atividades:

- (i) Preparar o LAL – léxico estendido da linguagem
- (ii) Definir AGFL – metas dos agentes vindas do léxico
- (iii) Refinar as metas

4.3.3.1.1. Preparar o Léxico Ampliado da Linguagem

Para construir o LAL, identificaram-se as fontes disponíveis (pessoas e documentos), segundo [11]. Para a refatoração do léxico foram seguidos os passos recomendados em [11] *construção do léxico*. A tabela a seguir, presente em [22], exhibe as regras gerais para a definição de símbolos.

Tabela 4 – Regras gerais para definição de símbolos [22].

Noção		Impacto
Sujeito	Quem é o sujeito	Quais ações ele executa
Verbo	Quem realiza, quando acontece e quais os procedimentos envolvidos	Quais os reflexos da ação no ambiente (outras ações que devem ocorrer) e quais os novos estados decorrentes
Objeto	Definir o objeto e identificar outros objetos com os quais se relaciona	Ações que podem ser aplicadas ao objeto
Estado	O que significa e quais ações levaram a este estado	Identificar outros estados e ações que podem ocorrer a partir do estado que se descreve

Esta prática melhora o entendimento do contexto e ajuda no refinamento do vocabulário, a Figura 26 apresenta um exemplo do léxico do SimulES revisto em [44] a partir de [4].

Nome:	dado
Noção:	Cubo utilizado para sortear aleatoriamente números entre 1 e 6.
Classificação:	objeto
Impacto(s):	1- jogador joga o dado. 2- jogador obtêm resultado do lançamento do dado .
Sinônimo(s):	dados.

Figura 26 – Exemplo de símbolo do LAL para SimulES v 3.0 [44].

O foco desta atividade é descobrir a motivação (o porquê) de cada ação. Segundo [2], quando uma ação muda de estado, esta ação define metas concretas. Quando uma ação fornece uma “qualidade” a um estado, esta ação define metas flexíveis. Nesta atividade foram identificados os atores e as metas relativas a eles a partir dos impactos dos símbolos pertencentes ao LAL.

Segundo [23] no LAL os símbolos do tipo sujeito ou aqueles que praticam ações sobre objetos são bons candidatos a atores. Na versão 2.0 de SimulES em [4] somente havia o ator: *Jogador*, já na versão 3.0 do SimulES apresentada em [44] incorpora o *Jogador*, *Engenheiro de Software* e *SimulES* como atores como é mostrado na Figura 27.

Nome:	jogador
Noção:	Participante do jogo simulés . Tem por objetivo vencer o jogo . Jogador pode assumir o papel de jogador da vez Jogador pode assumir o papel de adversario
Classificação:	sujeito
Impacto(s):	
Sinônimo(s):	jogadores.

Nome:	jogador da vez
Noção:	Papel de jogador que esta executando uma ação dentro de jogo
Classificação:	sujeito
Impacto(s):	1-joga rodada de início. 2-joga rodada de ações . 3-joga rodada de conceitos. 4-integra artefatos em um módulo . 5-submete produto. 6-aplica problema . 7- contrata engenheiro de software . 8-demite engenheiro de software .
Sinônimo(s):	

Nome:	adversario
Noção:	Papel de jogador que é instanciado se não é jogador da vez
Classificação:	sujeito
Impacto(s):	1-Recebe problemas através das cartas de problemas 2-Escolhe módulo 3-Inspecciona módulo
Sinônimo(s):	

Nome:	simulés_
Noção:	É um jogo educacional que simula várias situações da Engenharia de Software.
Classificação:	sujeito
Impacto(s):	1-Fornecer recursos para os jogadores
Sinônimo(s):	

Nome:	engenheiro de software
Noção:	carta do jogo que representa o profissional responsável por desempenhar ações. Possui habilidade e maturidade .
Classificação:	sujeito
Impacto(s):	1-Constrói artefatos 2-Inspecciona artefatos 3-Corrige artefatos 4-Integra artefatos em um modulo
Sinônimo(s):	engenheiros de software.

Figura 27 – Símbolos tipo sujeito v 3.0 [44].

4.3.3.1.2. Definir AGFL – Metas dos Agentes Vindas do Léxico

Uma vez identificados os atores (resultado mostrado na Figura 29), o passo seguinte é de capturar as metas dos atores a partir do LAL. Sugere [11] o uso de *templates* diferentes para símbolos do tipo sujeito, objeto, verbo e estado, levando também em consideração metas concretas e metas flexíveis. Para descobrir cada uma das metas é preciso responder o porquê.

Nas Tabelas 5, 6, 7, 8 e 9 mostramos alguns dos *templates* preenchidos, para ações concretas e flexíveis do Simules. E se pode observar a motivação descoberta em cada impacto a partir da análise do “porquê”. Pode-se identificar uma ação concreta quando esta é descrita por verbos pouco precisos como (controlar, avaliar, apurar, por exemplo.). Já ações flexíveis são pouco concretas e não se pode identificar um resultado concreto a princípio.

No caso de Simules Tabela 5, o impacto do símbolo *fornece recursos para os jogadores* ao ser analisado o “porquê” descobrimos metas que ele tem que cumprir a cada rodada. Estas metas se fazem necessárias para que ele (Simules) efetivamente possa fornecer os recursos aos jogadores. Esta mesma heurística tem que ser aplicada aos demais símbolos.

Tabela 5 – Template preenchido com metas dos símbolos do tipo sujeito [44].

TIPO: SUJEITO		<meta concreta>			ATOR	
-- impacto	resposta ao porquê?	<sujeito / objeto LAL>	seja / esteja	<verbo>		<sujeito LAL>
SIMULES						
-- fornece recursos para os jogadores.						
	Porque simules deseja que	rodada de início	seja	iniciada		
	Porque simules deseja que	rodada de ações	seja	iniciada		
	Porque simules deseja que	rodada de conceitos	seja	iniciada		
	Porque simules deseja que	recursos	sejam	disponibilizados		
	Porque simules deseja que	artefatos	sejam	comprados		
	Porque simules deseja que	cartas	sejam	compradas		

Tabela 6 – Template com metas concretas do símbolo do tipo objeto [44].

TIPO: OBJETO	<meta>			ATOR	
-- impacto resposta ao por quê?	<sujeito / objeto LAL>	seja/ esteja	<verbo>	<sujeito LAL>	
ARTEFATO					
-- Engenheiro de software constrói artefato.					
Para que	produto	seja	construído	por	Engenheiro de Software
Para que	produto	seja	empacotado	por	Jogador da vez
-- Engenheiro de software inspeciona artefato.					
Para que	artefato	seja	livre de defeitos		
Para que	produto	seja	construído	por	Engenheiro de Software
-- Engenheiro de software corrige artefato.					
Para que	artefato	seja	corrigido	por	Engenheiro de software
Para que	produto	seja	construído	por	Engenheiro de Software

Tabela 7 – Template com metas flexíveis do símbolo do tipo objeto [44].

-- impacto resposta ao por quê?	<meta flexível>		<meta concreta associada> <ator>
	<TIPO atributo de qualidade	[TOPICO] sujeito/objeto LAL	
-- Engenheiro de software inspeciona artefato	ação flexível		
Porque	qualidade	[artefato]	Artefato seja inspecionado Engenheiro de Software
-- Engenheiro de software corrige artefato	ação flexível		
Porque	qualidade	[artefato]	Artefato seja corrigido Engenheiro de Software
-- jogador usa qualidade do projeto	ação flexível		
Porque	qualidade	[projeto]	produto seja concluído jogador da vez

Tabela 8 – Template com metas do símbolo do tipo verbo [44].

TIPO: VERBO	<meta flexível>		<meta concreta associada>	<ator>
-- impacto resposta ao por quê?	<TIPO atributo de qualidade	[TOPICO] sujeito/objeto LAL		
APLICAR CONCEITO				
-- jogador da vez neutraliza problema	ação concreta		Conceito seja aplicado	Jogador da vez

Tabela 9 – Template com metas do símbolo do tipo estado [44].

TIPO: ESTADO	<meta flexível>		<meta concreta associada>	<ator>
-- impacto resposta ao por quê?	<TIPO atributo de qualidade	[TOPICO] sujeito/objeto LAL		
ARTEFATO DEFEITUOSO				
-- Engenheiro de software pode corrigir artefato	ação concreta		artefato seja corrigido	Engenheiro de Software

4.3.3.1.3. Refinar Metas

Conforme [11] após definir as metas dos agentes vindas do léxico, com o fim de facilitar seu entendimento é necessário organizá-las, para facilitar a identificação de situações de dependência.

A Tabela 10 mostra o *template* preenchido para o refinamento das metas concretas e flexíveis agrupadas pelo ator o ator SimuleS, as repetições são excluídas e as metas são ordenadas cronologicamente, segundo [11].

Tabela 10 – Template para agrupar e organizar metas por ator cronologicamente [44].

DEPENDER					DEPENDEE
SIMULES					
	rodada de início	seja	iniciada		
	rodada de ações	seja	iniciada		
	rodada de conceitos	seja	iniciada		
	recursos	sejam	disponibilizados		
	cartas	sejam	compradas	por	jogador da vez
	artefatos	sejam	comprados	por	jogador da vez

4.3.3.2. Identificar as Situações de Dependência Estratégica

Esta etapa tem três atividades:

1. Distinguir *SDsituations*
2. Reconhecer interdependências entre *SDsituations*
3. Construir diagrama de *SDsituations*

4.3.3.2.1. Distinguir *SDsituations*

Conforme [11], as *SDsituations* ou situações de dependência estratégica são uma forma de representação estruturada de uma situação, e estão formadas por um ou mais elementos de dependência no qual os atores participam através da colaboração. Ou seja, uma situação é um bloco que faz parte de um modelo intencional maior que através dos relacionamentos recíprocos de dependência estratégica entre os atores têm a capacidade de concluir uma meta situacional. Neste ponto, o objetivo em SimuleS era identificar as metas que se inter-relacionavam ou que formavam uma situação bem definida [11] isto é que apresentem forte coesão de intencionalidade.

4.3.3.2.Reconhecer Interdependências entre SDsituations

Ao observar as *SDsituations* se determinou suas interdependências físicas, lógicas ou temporais. Ou seja, neste ponto se decidiu como as situações estavam relacionadas dentro do processo, qual acontecia primeiro que outras e suas condições necessárias.

4.3.3.2.3.Construir Diagramas de SDsituations

O resultado desta atividade é a representação gráfica de todas as situações de dependência estratégica em um único diagrama [11], identificando o fator tempo e as interdependências.

A Figura 28 mostra o diagrama de *SDsituations* para SimulES v 3.0. Na Figura temos o seguinte: na atividade *Joga Rodada de Inicio*, a qual se executa uma só vez ocorre no tempo T1, após é executada *Joga Rodada de Ações* a qual está acompanhada de situações derivadas dela que podem ou não serem executadas, tudo isso acontece no tempo T2, já para o tempo T3 as atividades definidas foram *Joga Rodada de Conceitos* e de forma opcional *Tratamento de Problema*, finalmente temos *Submissão de Produto*, neste ponto se o produto é submetido a contento por algum jogador o jogo termina, caso contrario o jogo volta para a *SDsituation Joga Rodada de Ações*.

PUC-Rio - Certificação Digital Nº 0812549/CA

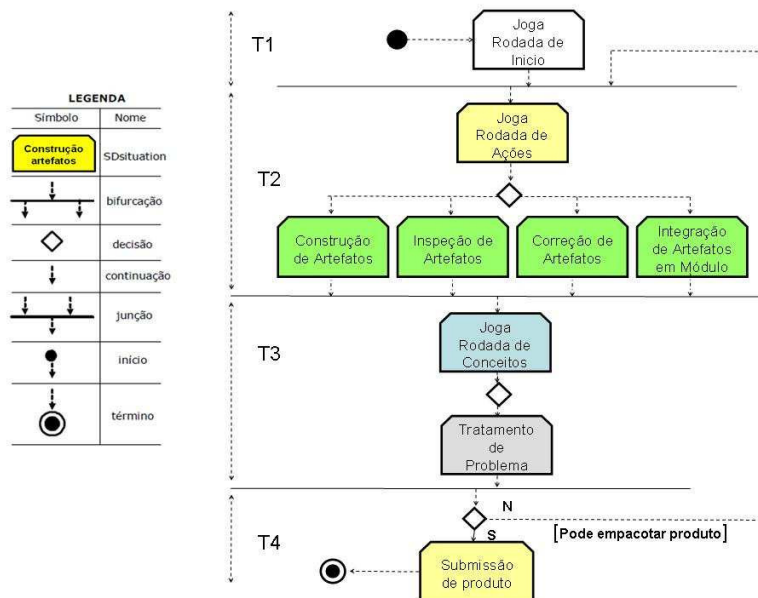


Figura 28 – Diagrama de *SDsituations* do SimulES v 3.0 [44].

4.3.3.3. Modelar as Metas dos Atores

A terceira etapa do método *ERi*c* propõe duas atividades:

1. Identificar agentes, posições e papéis
2. Criar painéis de intencionalidade

4.3.3.3.1. Identificar Agentes, Posições e Papéis

Quando as *SDsituations* são modeladas também identifica-se os atores que trabalham ou colaboram para atingir as metas ou meta daquela situação, podendo para isso, assumir posições e desempenhar papéis. Todas as metas elicítadas para cada ator dentro de cada situação estratégica devem ser examinadas para verificar se existem posições ou papéis que possam ser definidos. A Figura 29 de SimULES 3.0 apresenta aqueles símbolos tipo sujeito que foram identificados na etapa *Preparar o Léxico Ampliado da Linguagem*, podemos observar que o SimULES como fornecedor de recursos se encarrega de coordenar os recursos durante o jogo, o jogador, que são alunos participantes do jogo, pode assumir a posição de jogador da vez quando é a sua jogada e tem que executar o papel de gerente de projeto ou adversário quando está recebendo um problema, escolhendo ou inspecionando modulo de outro jogador, neste último, seu papel é de auditor e por ultimo temos os engenheiros de software que são as cartas dispostas por SimULES e que se encarregam de produzir o produto.

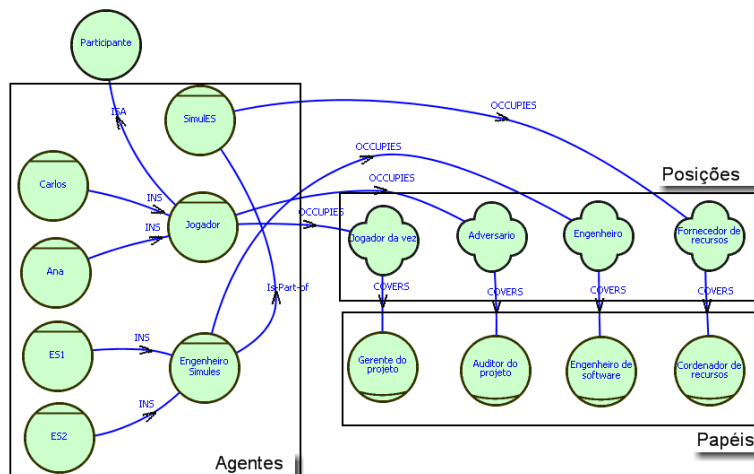


Figura 29 – modelo SA para SimULES v 3.0 [44].

4.3.3.3.2. Criar Painéis de Intencionalidade

Cada painel de intencionalidade ou diagrama IP representa uma *SDsituation*. Cada ator em um diagrama IP possui um eixo com a ordenação das metas nas quais participa, de baixo para cima, de modo que as metas iniciais fiquem na parte de baixo do eixo finais fiquem na de cima.

A Figura 30 apresenta o diagrama IP da *SDsituation Joga Rodada de Inicio* do SimuleS 3.0. Na construção deste tipo de diagrama é importante fazer a análise com um ator de cada vez, progressivamente, representando primeiramente as relações de contribuição (entre metas flexíveis), seguidas das correlações (entre metas flexíveis e metas concretas) e por fim, devem-se representar as relações de dependência entre as metas concretas dos atores. Porém na Figura 30 podemos ver que meta a ser cumprida é *rodada de inicio seja iniciada* e como responsável temos SimuleS, mas esta meta somente é alcançada se SimuleS *disponibiliza os recursos*, após vemos o Jogador da vez que é o encarregado de *iniciar o jogo*, quando *jogada de dado* é executada, sendo esta última uma pré-condição para que o *projeto seja escolhido* e *projeto seja aceito*, depois cada jogador *contrata seu engenheiro de software*.

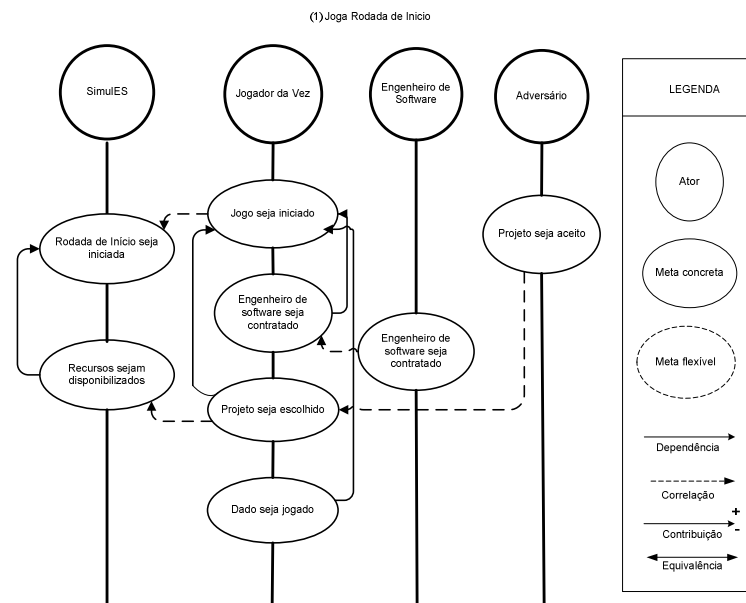


Figura 30 – Diagrama IP – Joga Rodada de Inicio do SimuleS v 3.0 adaptado de [44].

4.3.3.4. Modelar a Racionalização das Metas dos Atores

- (i) Construir Modelos SD
- (ii) Construir Modelos SR

4.3.3.4.1. Construir Modelos SD

Para usar o método nesta atividade, se retomam os artefatos elaborados previamente, bem como diagramas *IP*, *AS* e *SDsituations*. Ressaltado que para cada *SDsituations* se deve construir um diagrama *SD*. É preciso analisar cada *SDsituations* e seu correspondente diagrama *IP* identificar entre os quatro tipos de dependência disponível, ou seja, por meta concreta, por meta flexível, por recurso ou por tarefa e identificar os “*dependee*” (de quem se depende) e os “*depender*” (que depende), dependendo do grau de liberdade na relação de colaboração. Como mostra a Figura 31 com esta análise, se deriva na criação do diagrama *SD* com os atores e as dependências modeladas, é importante ressaltar que o principal insumo para a criação deste diagrama é o diagrama *IP* do qual se extraem a maioria das metas aqui representadas.

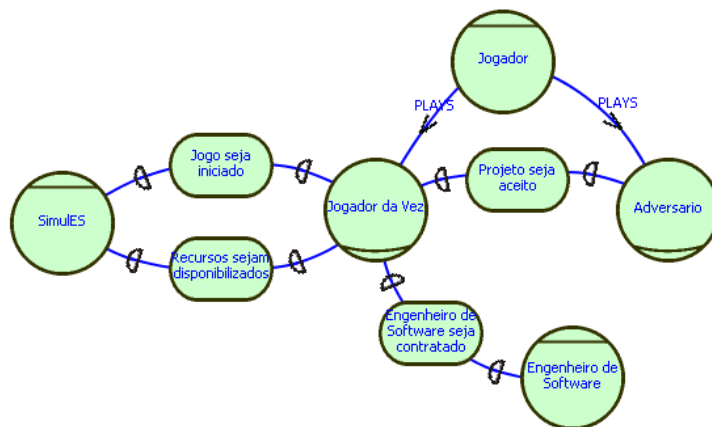


Figura 31 – Modelo SD – Joga Rodada de Início do SimulES v 3.0 [44].

Como é ilustrado na Figura 32 do SimulES 3.0 o modelo SR tem como objetivo representar as estratégias internas de cada ator, neste caso *SimulES*, *Jogador da vez* e *Adversário*. O modelo SR fornece uma descrição intencional do processo que é realizado para que a situação *Rodada de Início* seja realizada. O

modelo SR permite representar explicitamente o entendimento detalhado do raciocínio dos atores envolvidos no processo.

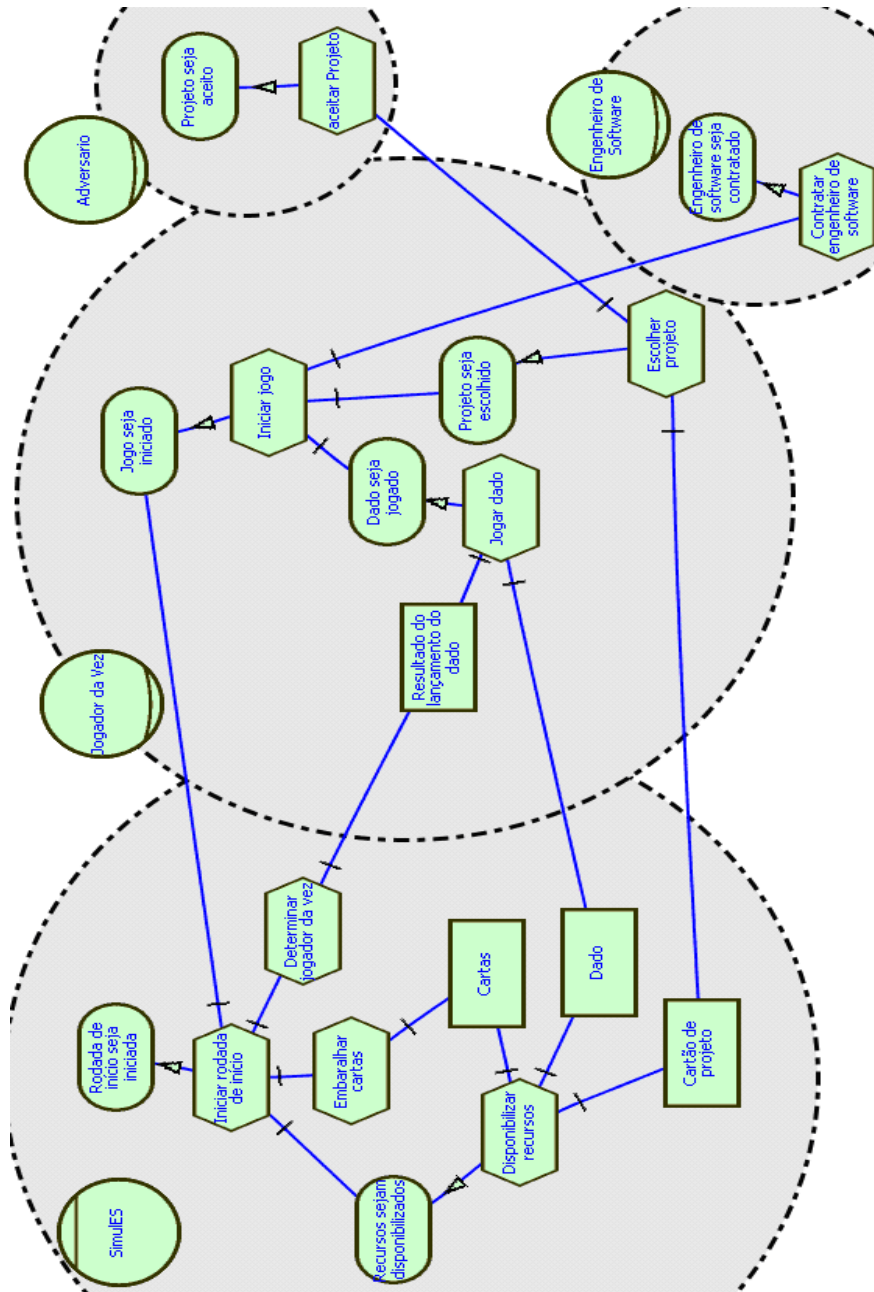


Figura 32 – Modelo SR – Joga Rodada de Início do SimulES v 3.0 [44].

4.3.3.5. Especificar as *SDsituations*

Maximizando o uso do LAL na descrição de cada *SDsituation* como é sugerido em [11], observa-se que cada símbolo é destacado em azul e a sintaxe própria da técnica de cenários foi usada para descrever e detalhar cada *SDsitutations* [44] veja um exemplo na Figura 33.

Título:	inspeciona artefato
Objetivo:	Descrever as regras da inspeção de artefatos .
Contexto:	informações do projeto no centro da mesa . jogador tem artefatos no tabuleiro.
Atores:	jogador
Recursos:	cartas , informações do projeto , tabuleiro.
Exceção:	
Episódios:	jogador escolhe artefato do tabuleiro. Se o engenheiro de software responsável pelo artefato faz a inspeção, então o mesmo gasta um ponto de tempo . Se outro engenheiro de software faz a inspeção, então esse gasta dois pontos de tempo . O artefato inspecionado é desvirado no tabuleiro.

Figura 33 – Descrição através de cenários de *SDsituation* Inspeção de Artefato do Simules v 3.0 [44].

4.4. Análise das Representações de Requisitos Usadas

Como foi apresentado ao longo do capítulo diferentes abordagens tem sido usadas para modelar SimuleS, a modelagem situacional de léxico e cenários dos trabalhos [5] e [4] e outra intencional baseada no *framework i** [44]. Estes trabalhos trazem pontos interessantes que foram explorados como base desta dissertação e que são considerados relevantes na estruturação da implementação do protótipo de SimuleS-W ou versão 4.0 do jogo. O trabalho [4] correspondente à versão 2.0 foi o ponto de partida para o entendimento do contexto e foram base para a realização de atividades de elicitação de requisitos mencionadas no Capítulo 3 na versão 3.0, que apresenta os modelos intencionais iniciais do SimuleS [44], conseguimos identificar maior detalhamento e a possibilidade de representar intencionalidade. A efetividade destes modelos será detalhada no Capítulo 6 onde mostraremos como foi aproveitada a modelagem intencional para a criação de SimuleS-W e detalharemos o refinamento da mesma.

A seguir exploraremos trabalhos relacionados ao uso da modelagem intencional, como também trabalhos relacionados com a transparência de software. A finalidade é focar a evolução do SimuleS em uma combinação de modelagem intencional apoiado em conceitos de transparência.