

## 4

### Modelo de Geração de Histórias

Como mencionado no capítulo anterior, a base de todo o processo de se trabalhar com histórias interativas tem início com a especificação dos eventos e fatos que podem ser combinados para gerar o conjunto de cenas que compõe a história. Neste capítulo, inicialmente é apresentado o IPG, o módulo do sistema que dá suporte e apoio lógico à autoria e geração das histórias. Na seqüência, são apresentados aspectos relativos à geração de conteúdo interativo e diversificado, bem como detalhes da estrutura interna da base de dados usada nos exemplos de histórias abordados nesta tese.

#### 4.1

##### O IPG (Interactive Plot Generator)

Um passo inicial para a geração automática de enredos teve início com os estudos do pesquisador russo Vladimir Propp [115], que observou que, em textos literários de gêneros específicos (no caso de contos de fadas russos), é muito comum a ocorrência de eventos típicos e de padrões de encadeamento entre os eventos.

Propp sugeriu a caracterização dos textos de um determinado gênero pela associação de funções a pequenos trechos das narrativas. A ocorrência das funções ao longo de uma narrativa obedeceria sempre a determinadas seqüências, que poderiam envolver todas ou apenas parte das funções típicas do gênero. As funções proppianas seriam, então, os eventos típicos básicos da narrativa, descrevendo-se os enredos como seqüências de ocorrências das funções. O encadeamento entre os eventos é um processo que tem de respeitar uma lógica, segundo a qual um evento só ocorre na presença de determinadas condições. Cada evento, por sua vez, pode também criar as condições necessárias para a execução de outros eventos [33].

Em geral, os eventos da história não ocorrem aleatoriamente. Estes eventos ocorrem essencialmente em vista dos objetivos que levam os personagens a agirem. Por sua vez, esses objetivos surgem quando

determinadas situações se configuram, exigindo a ação dos personagens. Os enredos resultam, portanto, da interação cooperativa ou competitiva entre os personagens na busca de seus objetivos. Deve-se destacar que as situações mais interessantes, que conferem dramaticidade aos textos literários e demandam mais atenção em situações reais, decorrem justamente dos conflitos entre personagens ou dos conflitos entre diferentes objetivos de um mesmo personagem [33]. No total, Propp caracterizou 31 funções, apresentadas na Tabela 4.1.

Baseado no trabalho de Propp, Ciarlini [33] propôs um método formal para a especificação dessas funções (eventos típicos), representando-as por operações lógicas, com pré e pós-condições. Para permitir a geração semi-automática de narrativas de um certo gênero, Ciarlini [33] desenvolveu o IPG (*Interactive Plot Generator*). O IPG é um módulo de geração de enredos, implementado com o uso de técnicas de Inteligência Artificial e Bancos de Dados, por meio da linguagem Prolog. Ele é composto por dois sub-módulos: um de inferência de objetivos e um outro de planejamento. No planejamento, o IPG faz a ligação entre os eventos de acordo com a lógica, explicitada por suas pré e pós-condições. O planejador do IPG é uma extensão do planejador ABTweak [149], que tem as seguintes características:

- É hierárquico - permite estabelecimento de pré-condições mais importantes, de modo a chegar mais rápido a uma solução;
- É não-linear, ou seja, estabelece relações de ordem entre eventos apenas quando necessário, o que em geral leva a uma maior eficiência. Além disso, a busca e conciliação de vários objetivos simultâneos fica facilitada.

O IPG usa um processo de simulação para a criação de um enredo de um certo gênero, que tem como ponto de partida a descrição da situação inicial dos personagens, o modelo de comportamento atribuído a eles (especificado em termos de objetivos a serem perseguidos em situações previstas) e as alternativas que cada personagem tem para atingir seus objetivos, que são especificadas em termos das operações (ou padrões típicos) que caracterizam o gênero. Estes dois últimos caracterizam o gênero da narrativa. Uma operação descreve os fatos válidos antes e depois do evento. Para que uma operação seja executada, é necessário que suas pré-condições sejam satisfeitas, ou no próprio estado inicial, ou então pelas pós-condições de operações que a precedem.

O processo de geração dos enredos faz parte de um ciclo com múltiplos estágios, onde alternam-se fases de inferência de objetivos e planejamento.

Tabela 4.1: Lista das 31 funções típicas de contos de fadas Russos [115]

---

<i>absence</i> (1): Um dos membros da família da vítima se ausenta de casa.
<i>interdiction</i> (2): Uma proibição é feita à vítima.
<i>violation</i> (3): A proibição é violada.
<i>reconnaissance</i> (4): O vilão busca informações sobre a vítima.
<i>delivery of information</i> (5): A informação sobre a vítima é obtida.
<i>fraud</i> (6): O vilão tenta enganar a vítima para tomar posse dela ou de seus bens.
<i>complicity</i> (7): A vítima, sem querer, colabora com o vilão.
<i>villainy or lack</i> (8): O vilão causa dano à vítima (ou a membro de sua família). Pode ser, por exemplo, um rapto ou um assassinato.
<i>mediation</i> (9): O herói toma conhecimento do infortúnio.
<i>counteraction</i> (10): O herói concorda em ir em busca da vítima.
<i>departure</i> (11): O herói sai de casa.
<i>proof</i> (12): O herói é testado por um doador para saber se merece receber o objeto mágico ou um auxiliar.
<i>reaction</i> (13): O herói reage às ações do futuro doador.
<i>receipt of magical object</i> (14): O herói adquire um objeto mágico ou um auxiliar.
<i>translocation</i> (15): O herói é transportado para as vizinhanças de onde está a vítima.
<i>struggle</i> (16): O herói combate o vilão.
<i>marking</i> (17): O herói recebe uma marca.
<i>victory</i> (18): O herói vence o vilão.
<i>liquidation</i> (19): A vítima é libertada ou a causa de infelicidade é eliminada.
<i>return</i> (20): O herói retorna.
<i>pursuit</i> (21): O herói é perseguido.
<i>rescue</i> (22): O herói é resgatado da perseguição.
<i>arrival</i> (23): O herói chega em casa sem ser reconhecido.
<i>pretentions</i> (24): Um falso herói se apresenta reclamando a recompensa.
<i>task</i> (25): Uma tarefa difícil é proposta ao herói.
<i>solution</i> (26): O herói resolve a tarefa.
<i>recognition</i> (27): O herói é reconhecido.
<i>exposure</i> (28): O vilão ou falso herói é exposto.
<i>transfiguration</i> (29): O herói ganha uma nova aparência.
<i>punishment</i> (30): O vilão é punido.
<i>wedding</i> (31): O herói é recompensado, usualmente casando-se.

---

Como diversos desencadeamentos diferentes podem ocorrer, o usuário, após cada fase de planejamento, pode interagir com o sistema para selecionar as alternativas mais interessantes, na medida em que enredos parciais são criados. É importante também que o autor possa interagir, impondo

condições correspondentes a estados dos personagens ao longo da narrativa. Se o modelo de comportamento definido inicialmente não produz bons resultados, o autor deve finalmente interagir com o sistema, em um nível mais profundo, corrigindo o modelo de comportamento de seus personagens e reiniciando o processo. O usuário pode também forçar a ocorrência de eventos e especificar que algumas situações devem ser verdadeiras em certos momentos durante a narrativa. Este tipo de interação é permitido tanto no início como durante o processo de simulação.

A Figura 4.1 mostra o esquema de geração de enredos proposto por Ciarlini et al. [35]. Nesse esquema, um novo enredo pode ser obtido a partir de uma simulação completa, da instanciação de um padrão típico de acordo com informações fornecidas pelo usuário ou por um processo misto no qual usa-se simulação para adaptar um enredo típico a condições específicas.

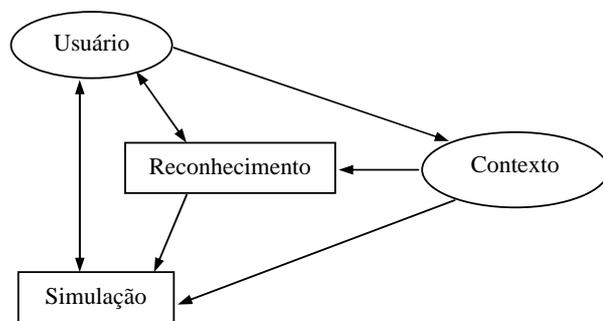


Figura 4.1: Esquema de geração de enredos.

Como o IPG trabalha com planejamento não-linear, são gerados planos compostos por um conjunto de operações parcialmente ordenadas. O exemplo apresentado na Tabela 4.2, baseado em casos das funções de Propp, ilustra esta ordenação. Para este exemplo, a operação 2 somente pode ser executada após a 1, as operações 7 e 8 somente após a 5, a operação 3 a qualquer momento, e assim por diante. A ordenação total é definida pelo usuário, que pode assumir diferentes combinações desde que respeitadas as precedências, representadas por setas na Figura 4.2.

## 4.2

### Contexto dos enredos

O IPG já foi utilizado em dois tipos de contextos: contextos de banco de dados empresariais [36], no apoio à tomada de decisões<sup>1</sup> e em contextos

<sup>1</sup>Neste caso, o IPG é usado para tratar objetivos entre uma empresa, que deseja obter mais clientes e melhorar o serviço oferecido a eles, e seus funcionários, que desejam

Tabela 4.2: Ordem parcial das operações

---

0: Init()
1: Absence_of_younger_people(princess)
2: Kidnapping_of_a_person(princess, dragon)
3: Call_for_help(hero, tsar)
4: Departure_of_seeker_hero(hero)
5: Fight_in_an_open_field(hero, dragon)
6: Out_of_earth_receipt(hero)
7: Victory_in_open_battle(hero, dragon)
8: Return(hero)
9: Reward(hero)

$1 < [2]$
$5 < [7, 8]$
$7 < [9]$
$4 < [5]$
$6 < [7]$
$8 < [9]$

---

literários, no caso contos de fadas [35]. Para a modelagem de contos de fadas, foi utilizado, com algumas adaptações, um subconjunto das funções catalogadas por Propp, cada uma associada a uma operação com pré e pós-condições. Estas condições são especificadas por um conjunto de predicados, que devem permitir que qualquer conto do gênero possa ser gerado pelo IPG.

O processo de simulação, realizado pelo IPG, por operar em um nível lógico, trabalha de forma transparente sobre qualquer tipo de operação válida e consegue gerar planos dentro dos limites estabelecidos. Com isso, novos enredos podem ser criados sem alteração do kernel do IPG. Apenas o arquivo com as especificações dos predicados, operações e objetivos deve ser reescrito.

Como ilustrado no exemplo anterior, o resultado do processo de simulação gera um conjunto de eventos, em formato texto, que possuem uma ordem temporal parcial. Neste trabalho, usa-se o IPG como um gerador de planos para alimentar um motor gráfico, onde personagens não mais lógicos, representados por atores, devem interagir “fisicamente” em um ambiente de representação gráfica. Isso traz fortes implicações tanto no contexto dos enredos, bem como nas ações realizadas pelos personagens. Simulações de Banco de Dados empresariais, usadas no apoio à decisão [33], por exemplo, são definidas por operações que não podem ser facilmente obtenção de estabilidade e ganho de promoções.

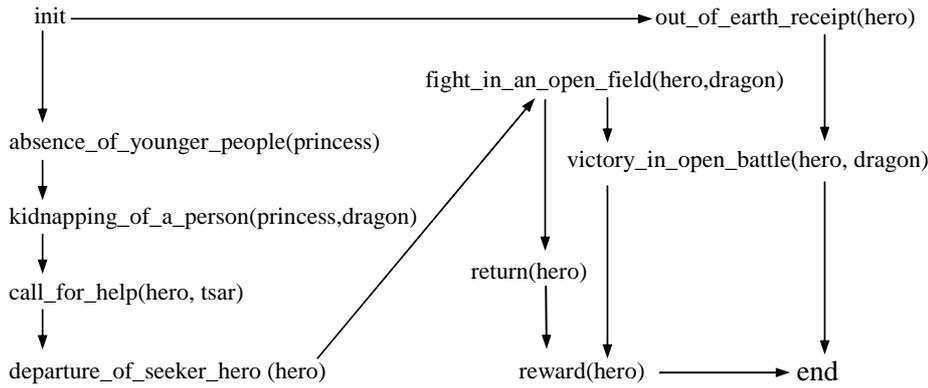


Figura 4.2: Representação gráfica de um enredo parcialmente ordenado.

expressas somente por recursos gráficos, por meio de um motor de jogos. Como exemplo, pode-se citar um evento característico deste gênero onde um personagem ganha uma promoção. A expressão de informação numérica ou textual, por meio de áudio ou janelas de diálogos, também se faz necessário para estes casos. O IPG ainda não possui recursos para geração de diálogos associados às operações de uma narrativa. Porém, já existe um protótipo, ainda rudimentar, que gera textos a partir de enredos [56]. Este recurso, entretanto, é de grande valia, mesmo em narrativas que podem ser facilmente expressas somente por meio de recursos gráficos.

Exemplos de contextos cuja representação gráfica expressa informação suficiente para o entendimento são histórias baseadas em cenas de ação, onde ocorrem interações físicas entre personagens e com o mundo que os rodeia, como no caso de histórias de contos de fadas. Como observado no decorrer desta pesquisa, muitas das 31 funções catalogadas para contextos de contos de fadas podem ser facilmente e suficientemente representadas por animações gráficas. Deste modo, este foi o gênero adotado nesta tese como base para o desenvolvimento do sistema.

### 4.3

#### Estrutura dos Enredos

Uma vez definido o gênero das histórias, partiu-se para a investigação e definição de um conjunto de operações com fácil representação gráfica e que, ao mesmo tempo, fossem suficientes para gerar enredos interessantes e com certo grau de diversidade. O objetivo foi delinear o que seria um enredo interativo, ou seja, um enredo que permita que tanto o IPG, quanto o usuário, possam disponibilizar de um grande repertório de possibilidades

na geração de histórias diversificadas e interessantes, cujo tempo de duração da representação gráfica dure um período considerável de tempo.

Como ponto de partida a esta investigação, considerou-se um exemplo tradicional de contos de fadas catalogado por Propp, como se segue:

*“Um vilão, no caso um dragão, raptar uma princesa desprotegida. O rei, em desespero, requisita ajuda a um herói que parte em busca da princesa, mata o dragão e, como recompensa, casa-se com a princesa.”* (Figura 4.2)

Para a dramatização deste simples exemplo, uma série de eventos podem ser tratados. Para que o dragão possa raptar a vítima, de forma resumida, ele deve saber onde ela se encontra, deslocar-se até ela, desviando de obstáculos, raptá-la e levá-la ao cativoiro. O rei, por sua vez, deve, de alguma forma, comunicar-se com o herói, que deve deslocar-se até o local do cativoiro, lutar com o dragão, vencê-lo, libertar a princesa e levá-la de volta para sua casa. Para que o casamento se realize, primeiramente os personagens devem ir até uma igreja. Se o casamento for realizado dentro da igreja, deve-se dispor de recursos para tratar com cenas de interiores.

Para identificar as dificuldades e melhor compreender os processos necessários para criar uma representação gráfica para esta simples história, foi desenvolvido um motor experimental. Diversas questões e problemas foram levantados. Eis os mais impactantes:

- Mecanismo de controle da duração de cada evento: a duração de cada evento pode ser fixa ou decorrente da interação entre os personagens. Em função disso, parâmetros para a determinação da finalização de uma ação devem ser descritos;
- Formas de interação entre os personagens: para este exemplo em especial, há personagens que se locomovem no chão (humanos) e um que voa (dragão). Isso traz várias conseqüências na forma como eles interagem entre si e com o ambiente;
- Mecanismo de sincronização entre a história simulada e a história que está sendo exibida graficamente: deve-se observar que a representação gráfica não consiste somente em exibir cenas, mas também na reconstrução do encadeamento de eventos e na sincronização de atributos presentes tanto no visualizador gráfico como no IPG;
- Nível de interação do usuário: o usuário deve ter o recurso de selecionar partes da história a serem visualizadas, bem como a reexibição de cenas de maior interesse;

- Nível de variação das histórias: como observado nos primeiros testes realizados, o nível de variação das histórias era relativamente pequeno. Uma questão a ser respondida era se deveriam ser incorporadas mais operações à base de dados ou se novas operações, mais específicas ao contexto, deveriam ser criadas;
- Comportamentos necessários que os personagens devem dispor: dependendo da quantidade e do nível de detalhamento das operações, comportamentos específicos para cada tipo de ação ou interação entre personagens deveriam ser implementados (como ocorre no jogo *The Sims* [48]).

Muitas destas questões são respondidas somente no Capítulo 7, que trata da implementação da ferramenta. No momento, se está preocupado em definir recursos que permitam ao IPG gerar histórias coerentes e que apresentem alto grau de diversidade e que permitam ao usuário a seleção de um grande conjunto de alternativas que conduzam ao final desejado.

Inicialmente, pode-se cogitar em especificar um grande conjunto de operações. Esta abordagem apresenta dois tipos de problemas: a nível de simulador, à medida que o número de operações cresce, cresce também o tempo de planejamento necessário para gerar a história. A nível de representação gráfica, cada nova operação implica na implementação de novas ações, bem como animações gráficas correspondentes. Em termos mais gerais, o nível de detalhamento de operações pode chegar rapidamente a níveis impraticáveis de serem representados, tanto a nível lógico como gráfico.

Visto que somente o uso de uma grande gama de operações não é uma boa solução para definir uma base que desse suporte a histórias diversificadas e com vários caminhos possíveis, procurou-se investir na definição de operações mais gerais e na agregação massiva de atributos numéricos, tanto a personagens, quanto a locais e operações. A definição de operações gerais, tais como “`go(CH,PLACE)`” (para representar a ida do personagem CH até o local PLACE) são claramente mais úteis que operações do tipo “`absence_of_younger_people(VICTIM)`” (que indica a ida da vítima a um local desprotegido). A operação “go” pode ser usada para representar os deslocamentos de todos os personagens, incluindo a ida da vítima a um local desprotegido. Consegue-se assim economizar tanto na especificação lógica do contexto quanto na implementação de recursos para a dramatização.

O conceito de locais também foi inserido na base de dados. Até então, nenhuma ação ou personagem estava associado a um local específico. Com a

inclusão deste elemento, personagens sempre estão relacionados com o local onde se encontram, o local onde residem e o local onde devem realizar as ações. Os locais possuem dois atributos numéricos: um que indica o nível de proteção e o outro para representar se o local é do bem ou do mal. Este mesmo atributo, visto sob a forma de caráter, é também associado a cada personagem. Todos os personagens possuem este atributo igual ao local onde residem, que é especificado no arquivo de inicialização dos personagens (Ver Sessão 7.3.1).

As operações mais gerais e os atributos numéricos associados aos personagens e locais permitem uma grande variedade de alternativas para a realização e combinação de ações.

Para permitir o uso de pré-condições com relações numéricas entre os atributos, o IPG suporta *constraint logic programming* [94]. Essa técnica combina os mecanismos de busca típicos de um planejador com a resolução de conjuntos de equações e inequações numéricas. Quando o conjunto de equações e inequações não tem solução para uma determinada alternativa do processo de busca, essa alternativa é descartada.

Os atributos numéricos são usados para modelar caráter, força, relações de afinidade e nível de proteção. Como apresentado no próximo capítulo, estes atributos também podem ser usados para a modelagem de *drives* e emoções.

Uma grande vantagem do uso de atributos numéricos é a possibilidade de tratar a variação contínua de suas intensidades, como resultado das pós-condições da execução de uma operação. Por exemplo, toda vez que um local é atacado, sua proteção é reduzida por um fator estipulado; quando ocorre luta entre dois personagens, suas forças são reduzidas, toda vez que a vítima é libertada, sua afeição aumenta em relação ao herói.

Muitas operações têm como pré-condições *thresholds* mínimos ou máximos para que possam ser realizadas. Para que o rapto ocorra, por exemplo, o nível de energia do vilão deve ser maior que a soma da energia da vítima com a proteção do local onde ela se encontra. Para este exemplo, o vilão poderá ter que realizar vários ataques contra o local da vítima ou então tornar-se mais forte.

Pretende-se mostrar que desta forma, pode-se construir situações complexas pela combinação de restrições numéricas, locais, personagens, níveis de proteção, relacionamento entre personagens, dentre outros. Com o uso de atributos numéricos, bem como pela estipulação de locais, pôde-se reduzir o número de operações e ao mesmo tempo, garantir a diversidade de histórias.

As operações trabalham sobre fatos previamente estipulados na base de dados do sistema. Fatos não podem ser criados durante o processo de simulação da história, porém podem ser alterados livremente pela execução das operações, como resultado de suas pós-condições. Os fatos são especificados por meio de predicados, usando a notação Prolog, como mostrado na Tabela 4.3.

Tabela 4.3: Predicados definidos na base Prolog

Predicados	Descrição
<code>character(CH,KIND)</code>	CH é um personagem com caráter do tipo KIND (natureza ou índole), que pode assumir dois valores: 1 (bem) ou -1 (mal)
<code>hero(CH)</code>	O personagem CH é um herói
<code>victim(CH)</code>	O personagem CH é uma vítima
<code>villain(CH)</code>	O personagem CH é um vilão
<code>place(PL)</code>	PL é um local
<code>home(CH,PL)</code>	O personagem CH reside no local PL
<code>current_place(CH,PL)</code>	O personagem CH está localizado em PL no dado momento
<code>protection(PL,KIND,L)</code>	O local PL tem um nível de proteção L do tipo KIND (natureza: bem ou mal)
<code>strength(CH,L)</code>	O personagem CH tem um nível de força L
<code>affection(CH1,CH2,L)</code>	O nível de afeição de CH1 por CH2 vale L
<code>alive(CH)</code>	O personagem CH está vivo

No nosso exemplo de utilização (Seção 7.9), os predicados anteriormente definidos foram usados para especificar a configuração inicial do simulador, que consiste na definição dos personagens (Marian - a vítima, Draco - o vilão, Brian e Hoel - os heróis) e seus atributos.

Na Tabela 4.4 são apresentadas as operações especificadas. Elas procuram ser genéricas, de simples representação gráfica e suficientes para gerar histórias interessantes. No Apêndice 1 são apresentadas as especificações destas operações, incluindo suas pré- e pós-condições, definidas através dos predicados listados anteriormente e de relações numéricas entre atributos.

A base de dados do IPG também contém a especificação das Regras de Inferência de Objetivos (*goals*). Elas representam os objetivos a serem atingidos pelos personagens e são usadas como meio de geração automática das histórias pelo IPG, mesmo sem a interferência direta do usuário. Cada regra representa um conjunto de fatos que um personagem passa a desejar tornar verdadeiros quando certas situações ocorrem no enredo [33]. Por meio

Tabela 4.4: Conjunto de operações (adaptadas de Propp)

Operações	Descrição
Go(CH,PL)	Locomoção do personagem CH até o local PL. Geralmente ocorre em combinação com outras operações. Faz-se necessária visto que personagens e ações estão associados a locais
Reduce_protection(PL)	Redução da proteção do local PL pela redução voluntária do número de guardas. Cada local possui um nível de proteção associado. Algumas operações têm como pré-condições que o nível de proteção do local onde a ação deve ocorrer assuma valores dentro de certos limites
Attack(CH,PL)	O personagem CH deve atacar o local PL para reduzir a proteção, lutando com os guardas do local. CH e PL devem estar associados. Para que o ataque ocorra, existe a pré-condição de que a natureza (boa ou má) de CH deve ser diferente da natureza da proteção corrente do local
Get_stronger(CH)	O personagem CH recebe poderes que aumentam a sua força
Fight(CH1,CH2)	Luta entre os personagens CH1 e CH2. Foi criada para representar as lutas entre os mocinhos (no caso os cavaleiros que fazem o papel de heróis) e o vilão (dragão). Para que a luta ocorra, o local da luta deve estar desprotegido (baixa proteção) e os personagens devem ter naturezas opostas
Kidnap(VIL,VIC)	Um personagem vilão rapta o personagem vítima. Para isso, o nível de força do vilão deve ser maior que a soma da força da vítima com a proteção do local onde ela se encontra
Kill(CH1,CH2)	O personagem CH2 é morto pelo CH1. CH1 deve ser mais forte que CH2. Devem ser também de naturezas opostas
Free(HERO,VICTIM)	O herói liberta a vítima que estava raptada. Tem como pré-condição a morte do raptor e como pós-condição o aumento da afeição da vítima pelo herói
Marry(CH1,CH2)	Casamento entre dois personagens. É resultado da premiação por uma ação realizada, neste caso a libertação. Logo, para que haja um casamento, deve haver um rapto e o vilão deve ser derrotado

delas, o autor pode modelar a estrutura geral da narrativa. Para o contexto de contos de fadas de nosso exemplo foram definidas, usando uma lógica temporal modal [34], 4 regras:

1. Se no início da história existe uma vítima, será tomada alguma medida para torná-la frágil. Segundo o elenco de operações apresentadas, isso pode ser realizado de várias formas, como por exemplo: pela redução da proteção onde a vítima se encontra, pelo ataque do vilão a este local ou fazendo com que a vítima vá a um local desprotegido;
2. Se a vítima ficar desprotegida, o vilão desejará raptá-la. Para isso, o vilão deve ir até a vítima, reduzir a proteção do local onde ela se encontra, caso seja necessário, raptá-la e levá-la para o cativeiro;
3. Se a vítima for raptada, o(s) herói(s) tentará(ão) salvá-la. Para que isso ocorra, o herói deve reduzir a proteção do local onde o vilão reside, lutar com ele e vencê-lo;
4. Se a afeição entre herói e vítima é alta, eles desejarão se casar. A principal pós-condição da libertação é o aumento da afeição da vítima pelo herói que a liberta. Neste caso, pelas operações apresentadas, a única forma do casamento se realizar é que haja um rapto seguido de uma libertação.

#### 4.4 Conclusões e Discussões

O IPG é um versátil e poderoso módulo para geração de histórias, que usa a lógica como forma de garantir a integridade e coerência das seqüências de eventos gerados por mecanismos de planejamento e inferência de objetivos, bem como pela especificação explícita de fatos pelo usuário, o que torna a geração da história um processo assistido.

Para o propósito que esta ferramenta se destina - servir como fonte de dados a um visualizador gráfico - investiu-se na especificação de uma base de dados compacta e que ao mesmo tempo fosse suficiente para gerar histórias cuja interatividade, atratividade e tempo de duração da representação gráfica fossem valores expressivos, principalmente quando comparado com outros modelos de *story engines*.

Fazendo-se uso de uma abordagem *plot-based*, o IPG apresenta características que permitem um controle rígido no direcionamento da história, mas que nem por isso limita a forma como o usuário pode interagir

com a mesma. Detalhes sobre a interação com o usuário são apresentados no Capítulo 7, que trata da implementação da ferramenta, onde são expostas as interfaces gráficas por onde o usuário interage com a história que está sendo processada.

Um recurso que ainda não foi explorado na ferramenta, mas que já está incorporado ao IPG, é a possibilidade do usuário poder, além de inserir e ordenar operações e goals, alterar atributos dos personagens. Diversas questões interessantes podem surgir deste tipo de interação. Pode-se por exemplo, transformar um herói em um vilão.

Outro recurso que pode ser mais explorado é o uso de mais de um personagem do mesmo papel ou de um personagem com mais de um papel. No exemplo, há dois heróis: Brian e Hoel. Pode haver competições entre eles para decidir quem vai salvar a princesa, ou situações mais interessantes onde um se encarrega de enfraquecer as defesas do castelo do vilão, enquanto o outro derrota o vilão e leva o prêmio. Mais situações com certeza podem surgir se for permitido que haja mais de uma vítima, mais de um vilão ou que um herói também possa ser uma vítima.

No próximo capítulo são fundamentados aspectos de agentes, comportamentos e estratégias para a implementação dos atores, cujo papel é fazer a representação dos eventos da história, gerados pelo IPG e definidas em formato texto, por meio de conteúdo gráfico animado.