5 Conceptual Model and Extended System for Digital Interactive Storytelling Supported by Plan Generation and Recognition

5.1. Chapter Preface

The main goal of this paper is to describe the extensions made to a conceptual model of a genre and how support for it was implemented in LogTell-R. This work builds on the LOGTELL system [Pozzer 2005] and our previous experiments with plan recognition [Karlsson et al. 2006a] and discusses the theoretical foundations of our new model.

This paper was co-authored with Antonio Furtado and at the time of this thesis writing is to be submitted for journal publication.

5.2. Introduction

Storytelling is a central aspect of human life and culture. Schank [1990] writes that humans understand the world in terms of stories, and often approach problem solving and new ideas by adapting stories they have previously collected and somehow indexed for future reference. Arguably, the human brain has a natural affinity not only for enjoying narratives, but also for creating them [Wallis 2007]. Storytelling and narrative are therefore fundamental to human experience.

The dynamic generation of stories is, consequently, an interesting problem. On the other hand, allowing the prospective audience to take part in the composition process has a wide appeal. Of course, in order to enable the audience in shaping such stories, it is necessary to provide adequate forms of interaction, which is tantamount to combining stories with game-like features.

Interactive storytelling (IS) as a research area can be described as a form of digital entertainment that brings together techniques and tools for the creation, visualization, and control of interactive stories through digital means. In its broadest view, it is hardly a new area. Already in the 1970s, the development of a computer program capable of automatically generating stories was seen as a worthy goal to be pursued; the most famous of those systems was Tale-Spin [Meehan 1977], which generated stories from a simulation of characters pursuing specific goals.

Nonetheless, the field remains unsettled, still presenting many open issues for research. The key issue in interactive storytelling is the generation of stories that are both coherent and interesting; while also taking into account authors, autonomous characters, and an audience interacting during their generation.

A wide range of different approaches has attempted to reach the goal of integrating storytelling and interactive entertainment (narrative and gameplay), such as works of interactive fiction and adventure games, but none of those did fully match the expectations, either by focusing too much on puzzle-solving or by restricting user interaction options to some narrow set of choices.

A reason for the inadequacy of most efforts so far is that IS is such a broad problem, whereas current systems tend to concentrate in a restricted experience in which a user is allowed to take part. This experience can happen differently every time it is "played", but it usually consists of the same restricted storyline or is presented in a "too open" storyworld. For example, the story is always about a detective who must collect evidences and solve a murder, or it is always about a visitor to a couple with relationship problems.

While agreeing that this choice is justifiable as an appropriate attempt to simplify the problem space and maximize impact of design decisions, we feel that more generative approaches are in order if the field is to become truly successful.

One of the main open problems in the generative craft of stories resides in the transfer of knowledge between human authors and the computer. Any story generation system needs access to some knowledge base that describes how stories may be generated and told. Different techniques can be used for the generation of stories depending on the model utilized. The differences among these techniques can be regarded as differences in 'style' among the respective story generation systems.

While it is probably not possible to define an ideal general model for good stories, since the quality of a story depends heavily on tastes and expectations of those watching or participating in it, one of the goals of IS is to find models that might be used in specific situations, scenarios, or genres; thus fulfilling the expectations of a good percentage of the interested audience.

We recognize that sound methods to organize and combine events must be considered in order to confer enough dramatic power to narratives. Most storytelling model approaches consider stories formed by sequences of events constructed by using the inherent temporal characteristics of each event or simple causality relations, but these are clearly not enough to guarantee interesting and varied stories.

Some applications make use of special narrative functions that enforce narrative principles in the context of the whole story (like raising tension, introducing dilemmas, or some mechanism to "move the story forward"), but while these efforts are interesting in improving user engagement they still do not guarantee diversity of experience. A possible avenue for the development of more generative systems that can create interesting stories is related to the definition of a genre. Once a genre is specified with some rigour in a constructive way, it becomes possible not only to determine whether a given plot is a legitimate representative of the genre, but also to generate such plots.

How to characterize a literary genre is a much debated problem, which can be approached with useful results by combining models drawn from both Literary Theory and Computer Science, an ability of obvious relevance to Storytelling theory and practice. However, this is not an easy effort, as witnessed by a number of researchers who have been using computers to try to model and compare stories at least since the 60s [Dundes 1965].

To tackle the problem of creating a more generative model to support the production of stories, we have drawn on previous experience with event relations in the LOGTELL system [Pozzer 2005], in turn motivated by the four master tropes identified by semiotic research [Burke 1969], namely: metaphor, metonymy synecdoche, and irony. By offering mechanisms derived from these event relations, we intend primarily to augment the expressiveness of the narrative model. The conjecture that the interplay of the syntagmatic, paradigmatic, antithetic and meronymic relations already permits an ample coverage is reinforced by the connection between these relations and the four master tropes; which, as Culler [2009] points out, are the basic rhetorical structures by which we make sense of experience.

This paper describes a conceptual model used to represent plots in a given genre together with an analysis of event relations (along with some related work), and presents extensions to better support these relations and enhance the space of possibly interesting stories in the context of a generative system for digital interactive storytelling.

As this work builds on top of a previous prototype from our research group (LOGTELL), we first describe its architecture and some experience exploring its utilization, after which we explain how to handle event relations in order that interesting and coherent stories can be created.

Based on this discussion, we outline how an intended genre (to whose conventions the plots must conform) can be modelled. Following that, we describe LogTell-R, an extended version of LOGTELL. This new system uses plan recognition along with plan generation in the creation of stories. We then show how the new system works on the basis of the identified event relations, also indicating what is done to narrow the gap between the conceptual model and the implementation while helping to construct the possible plots. Finally, we offer some remarks on the results obtained.

5.3.Story Generation Systems

Even though story generation systems (SGSs) have common motivations, they do not necessarily try to solve exactly the same problem. To begin with, systems using different fixed knowledge bases may be constrained to generate stories in completely different genres and themes. Also, SGSs employ different strategies to configure their story models, which can be broadly classified as character-based, plot-based, and user-experience models.

In character-based models, the storyline results from the interaction among virtual agents. The main advantage of a character-based model is the ability for anytime user intervention, which means that the user may always interfere with the ongoing action of any character in the story. Although powerful in terms of interaction, such an extreme interference level may lead the plot to uncharacteristic situations. Additionally, there is no guarantee that narratives emerging from the interaction between agents will create interesting dramas.

By contrast, in plot-based models, characters should follow more rigidly defined rules, specifying the intended plot structures. In a pure plot-based approach, user intervention is far more limited, but it is usually easier to guarantee coherence and a measure of dramatic power.

SGSs with user-experience models, a newer category focused on the user rather than on the plot or on the characters, can tackle a wide range of problems by addressing individual user preferences. The means employed for this purpose include user profiling, identifying user moods, and measuring inherent interestingness of pieces of stories.

We feel that an approach that fits in between plot-based and character-based is the best option, given that plots and characters are interlocking elements that cannot exist without each other [Glassner 2004]. Specifically, we try to conciliate both plot-based and character-based modelling. User-experience concerns might be added at later stages; as long as we can guarantee that enough variety of stories can be generated, possibly while considering changes in dramatization/presentation of story events.

Even though it is argued that the key aspect of an interactive narrative is the story representation used to encode the author's vision of the possible narrative experiences, called a story space [Magerko 2007], little attention is given to what qualities a story space should have, especially with respect to relevant relations between events.

Most approaches to plot-influenced storytelling models consider only stories formed by sequences of events constructed by using the inherent temporal characteristics of each event, or by simple causality relations, but these are clearly not enough to guarantee interesting and varied stories. Other approaches use dilemma-inducing or tension-raising events, paying no attention to the relations between events in the plot model.

Indeed few efforts dedicate enough attention to relations between events. Three relevant exceptions are [Pinhanez 1997], [Lang 1999], and [Nakasone and Ishisuka 2007]. Pinhanez [1997] deals with a temporal model of events, providing mechanisms to handle their interlacing temporal relations.

Building on the same interval algebra as Pinhanez, the Joseph system [Lang 1999] describes a grammar-based model for simple narratives. Lang complains that the syntax given in the usual story grammars does not entirely rule out the construction of "non-stories", and that their semantic annotations are not formalized rigorously enough. In contrast, he claims that Joseph provides a formal framework to relate story components to one another. In Joseph a story is modelled as having two sub-components, a setting and an episode list, both of which have temporal intervals associated with them. The information about the relationships among the elements of the event list is specified in the grammar rules. The Joseph system is (to our knowledge) the first system constructed from an explicit, formal model for stories.

Arguing that not only temporal relationships are important for the structure of stories, but also relations determined by the rhetorical context of those events, Nakasone and Ishisuka [2007] introduce ISRST, an ontology model based on the organization of events using a subset of relations proposed by Rhetorical Structure Theory (RST). Events in the world are not isolated, but interconnected through some kind of relation. In ISRST, a relation is a rhetorical binding between two entities, which refers to a specific rhetorical function [Nakasone and Ishisuka 2007].

They further claim that even though the way events are linked in any story is consistent with the set of RST relations, "its use is impractical for the purpose of content creation, since most human beings make use of a more limited set of relations to construct and remember stories" [Nakasone and Ishisuka 2007]. To match this limitation, a reduced set of relations is considered: a) Background – one event happens before another, but there is no causality; b) Cause - implies causality; c) Purpose - reflects the necessity for one event to be shown before another can be shown; d) Result - an event is a direct consequence of another; e) Contrast - shows some kind of conflict between events; f) Solutionhood - provides a way to define how a Contrast relation will be solved; g) Elaboration - an event gives more details about another; h) Evaluation - states a final conclusion about an event; and i) Sequence - establishes a temporal link between two events. While interesting, their ideas do not seem to be implemented in a concrete system yet.

As highlighted by Nakasone and Ishiuka [2007], content creation needs support and abstractions/simplifications. Having authorial tools in place to make the underlying complexity of the knowledge base transparent to authors is a very desirable goal, especially if the representation uses complex logic models or specialized programming languages. There is not much perspective for a system that places a too great burden on potential authors.

For example, although Façade has been a successful experience, its architecture requires a great effort from the prospective authors. It uses four different content languages and took two years just for authoring a game that has only one scene, two characters, and takes about 20 minutes to complete [Mateas and Stern 2003].

5.4.LOGTELL Plot Composition Model

As this work builds on top of a previous prototype developed by our research group (LOGTELL), we proceed to briefly describe the system, its plot composition model, and the structures used to generate stories. For details on the initial version of the system, see [Pozzer 2005, Ciarlini et al. 2005]. Experiments with the use of LOGTELL and various enhancements to its architecture are reported in [Karlsson et al. 2006a, Camanho et al. 2008, Ciarlini et al. 2008, Camanho et al. 2009].

In summary, LOGTELL is a logic-based tool for the interactive generation and dramatization of stories via the use of a plan-generation system. The main difference between LOGTELL's conceptual model and other similar planning systems (like Mimesis [Riedl and Young 2004]) is that its model does not assume the existence of one goal for the story as a whole. Instead, at the beginning of the plot and after each planning phase, we use goal-inference rules to consider new goals induced, for the various characters, by situations arising from the part of the plot so far generated.

The design of LOGTELL was influenced by different notions borrowed from Literary Theory and Narratology, besides methods and techniques taken from Artificial Intelligence and Computer Graphics.

From Narratology studies the designers took, in particular, the distinction of three levels in literary composition: *fabula*, *story*, and *text* [Bal 1997]. At the *fabula* level, the characters acting in the narrative are introduced, as well as the narrative plot - corresponding to a chronological sequence of events. The *story* level concerns a different representation specific to these events and how they are narrated to an audience, be it through a temporal re-ordering of the events, the use of narrative techniques, or the use of different points of view. The last level (*text*) relates to the medium used to tell the story, such as natural language, movie, virtual environment, or any other communication medium.

When breaking down the problem of interactive storytelling into its smaller sub-parts, a similar separation is observed between the story generation and narration phases [Karlsson et al. 2009b]. In SGSs, this breakdown means that different modules handle the two parts. In discussing the conceptual model behind the story space in LOGTELL, we are interested here only in the *fabula* level, which is the job of LOGTELL's Plot Manager. The narrative plot manipulated by this module consists of a partially-ordered set of events.

Here we should recall that plot managers, in general, are interactive story generators that receive as input an already created piece of story (initially empty) and present as output a suggestion on how to continue the story. Their main function is to effect corrections so the story plot can develop correctly. If some inconsistency is detected in the input, the manager intervenes in order to ensure that the original goal can be reached, satisfying all the necessary constraints [Roberts and Isbell 2007].

In order to model a chosen genre – fairy-tales for instance – to which the plots to be composed should belong, it is necessary to specify at least: a) what can exist at some state of the underlying storyworld; b) how states can be changed; and c) the factors driving the characters to act. Accordingly, the context for the creation of stories in the original version of LOGTELL [Pozzer 2005] comprises three levels of conceptual modelling, wherein three schemas are successfully structured to provide the underlying knowledge base.

5.4.1.Modelling Storyworld

In order to create any kind of story, a description of the mini-world where the narrative takes place is necessary; i.e. the static aspects of a literary genre need to be specified.

In LOGTELL this description takes the form of a static schema: a set of facts (state), introducing the characters and their initial situation, as well as the description of the scenarios and other static features needed for the generation of stories. The static schema specifies, in terms of the Entity-Relationship model [Batini et al. 1992], the entity and relationship classes and their attributes. An entity is anything of interest by itself, material or abstract, animate or not.

In LOGTELL's simple example, characters and places are entities. Besides entity identifiers, other attributes may characterize the entity instances. The entity class character admits person and dragon as specializations. Furthermore, princess, knight, and magician specialize person.

The attributes of characters are *name*, which serves as identifier, *strength,* which indicates each character's capacity for fighting, and *nature*, which states if the character is good or evil. Places have only one identifying attribute (*name*) and a *protection* value, declaring how much they are guarded against attackers. Characters are pairwise related by relationship *affection*. Finally, two relationships associate characters with places: *home* and *current_place*.

Attributes refer to what the entities are, but there was also need to assign roles to certain entities, in order to indicate how they are expected to act. The choice of roles -*hero*, *victim*, *villain* and *donor* - is a subset of the seven *dramatis personae* proposed by Vladimir Propp.

The state of the world at a given instant consists of all facts about the existing entity instances and their properties (attributes and relationships) holding at that instant. A genre is of course compatible with an ample choice of (valid) initial states. Different initial states lead to the development of possibly very different narratives, all of which are constrained to remain within the limits of the defined genre.

5.4.2.World State Changes

The dynamic level of specification takes us from descriptions of states, as covered by static schemas, to narratives. While states are sets of facts, narratives are composed of events. An event is a transition from a valid state Si to another state Sj, which should also be valid, i.e. conforming to the established static integrity constraints. In addition, we shall require that the transition itself be valid, which means that it should obey a further set of restrictions, termed dynamic integrity constraints.

In general it is sufficient to enforce these constraints by restricting state changes to what can be accomplished by applying a limited repertoire of pre-defined domain-oriented operations. The operations must be defined in such a way that, if one is applied to a valid initial state, their execution will always result in a valid state.

A similar notion has been proposed in literary theory by Vladimir Propp [1973]. Propp identified a set of 31 functions that cover all situations in Russian fairy-tales. LOGTELL equates the notion of event with the state-change brought about by the execution of a predefined operation by some agent.

Each of the nine Propp-inspired operations in the original LOGTELL prototype represents what actions can be deliberately enacted by the participants, and is defined in terms of its pre-conditions and post-conditions. Post-conditions (or effects) consist of two sets of facts: those to be asserted and those to be denied as a consequence of executing the operation.

Integrity preservation depends on a careful adjustment of the interplay among pre-conditions and post-conditions over the entire repertoire of operations. This interplay has an even more important consequence, which is to establish a partial order for the execution of operations. It is worksome but not too hard to check how the combined interplay of pre-conditions and post-conditions in this repertoire contributes to preserve static and dynamic integrity constraints, once the validity of the postulated initial state has been verified.

The dynamic schema is thus composed of a limited repertoire of pre-defined operations (typical of the chosen genre) in which characters can take part. Being defined in terms of their pre-conditions and post-conditions, operations can be readily chained together by a plan-generating algorithm.

5.4.3. Characters in the Storyworld

It is not enough to have just a specification of world state and a set of operation to be applied over it in order to have a generative story model. It is also necessary to model what motives guide the characters' behaviour.

In LOGTELL's model, for those entity classes or roles whose instances are agents, there may exist goal-inference rules specifying (in a logic formalism) the goals that will motivate these agents when certain situations occur during a narrative.

Having, at a given initial state, applied such rules to determine intermediate short-term goals for the various agents, one is in a position to apply a suitable plan generator to start composing a plot, where each event is associated with the execution of one of the operations defined in the dynamic schema.

So, the behavioural schema consists of a set of logical rules, to infer goals to be pursued by each character, as certain situations arise in the course of plots.

5.4.4. Plot Generation and Some Remarks

The generation of a plot starts by inferring goals for the characters from the initial configuration. Given this initial input, the system uses a planner that inserts events in the plot in order to allow the characters to try to fulfil their goals. When the planner detects that all current goals have been either achieved or abandoned, the first stage of the process is finished.

The partial plot then generated is presented to the user by means of the Plot Manager. If the user accepts the plot generated so far, the process continues by inferring new goals from the situations generated in the first stage. If new goals are inferred, the planner is activated again to fulfil them. The process alternates goal-inference and plan generation until the moment the user decides to stop or no new goal is inferred.

Further exploring the possibilities of generating a large variety of coherent stories in the given "fairy-tale genre", we used in our own work not only plan-generation, but added facilities for plan-recognition as well [Karlsson et al. 2006a], thus achieving a dual plan-recognition/plan-generation paradigm [Furtado and Ciarlini 2000].

Experiments with the system seem to demonstrate that combining goal inference, plan generation/recognition, and user participation constitutes a promising strategy towards the production of plots which are both diverse and coherent.

However, even though coherence and diversity are essential for plot composition, they do not guarantee dramatic power. Sound methods to combine events must be considered in order to confer enough dramatic power to narratives. This led to the necessity to further investigate and possibly extend the model for stories and plot composition used in LOGTELL.

We do not claim that the present effort is sufficient to create a fully immersive experience. We endeavour, as a more limited but hopefully already significant objective, to explore the possibilities of generating a large variety of coherent stories by applying the plan-recognition/plan-generation paradigm over a given genre.

5.5.Relations Between Events in Plot Composition

Many narratologists agree that the most important relation between events is the causal one. Nevertheless, this is not the only relation taken into consideration when human beings try to create story patterns in their minds [Nakasone and Ishisuka 2007]. The impact on the audience very much depends on how events are combined.

It has been suggested [Furtado 2008] that at least four concerns are involved in plot composition: a) the plot must be formed by a coherent sequence of events; b) for each position in the sequence, several alternative choices should apply; c) non-trivial interesting sequences must permit unexpected shifts along the way; and d) one may need to go down to details to better visualize the events or, conversely, to summarize detailed event sequences.

These concerns have led to the identification of four relations between events that play a basic role in an interactive plan generating system, and to the introduction of a fourfold perspective of plot composition in the context of interactive storytelling. It turns out that such relations hold as a consequence of the conventions regulating the chosen narrative genre and its conceptual specification, as described in Section 5.4.

5.5.1.Reviewing the Four Event Relations

As previously discussed, LOGTELL’s conceptual model allows the production of plots which are both diverse and coherent. The plan-generator is able to align the plot events in a coherent sequence in view of the characters' objectives (and operations pre and post conditions), whenever possible coming up with more than one plot, so as to provide alternative ways to reach the objectives.

On the basis of the well-known homology between literary and linguistic structures, we feel justified to borrow Ferdinand Saussure's [1967] notions of a syntagmatic axis and a paradigmatic axis to refer, respectively to concatenating events in sequence and choosing alternatives.

It is clear from Propp's work that there is a structuring principle, a syntagmatic relation, between events in plot. This relation induces a weak form of causality or enablement between events, which justifies their sequential ordering inside the plot. Also, Lévi-Strauss stresses the importance of analysing paradigmatic relations when dealing with folklore structure [apud Dundes 2007], seeking to describe the patterns underlining the text and the possibilities to instantiate elements in those patterns.

For example, to declare that it is legitimate to continue a plot containing an abduction by placing a rescue next to it, we say that these two events are connected by a syntagmatic relation. More precisely, we can define the semantics of the two events in a way that indicates that the occurrence of the first leaves the world in a state wherein the occurrence of the second is coherent.

Similarly, in a tale the events of abduction and elopement can be seen as alternative ways to accomplish a similar kind of "villainy", whereby someone is taken away from its current place by some foreign person. Both achieve approximately (though not exactly) the same effect. To express that they play a similar function we say that a paradigmatic relation exists between the two events.

Special care must be taken when dealing with the interplay between syntagmatic and paradigmatic relations. For example: an abducted woman expects to be rescued from the villain's captivity by the man she loves. On the contrary, if she freely eloped with the seducer, she will only leave him through forceful capture. As this example suggests, the syntagmatic and the paradigmatic axes are not really orthogonal in that the two relations cannot always be considered independently when composing a plot.

These two relations were already handled by LOGTELL's Plot Manager. In previous work [Karlsson et al. 2006b], we explored extending the system to include the usage of plan-recognition (along with plan-generation) during the plot composition phase. In using plan-recognition, a library of typical plans (or hierarchy of typical plans) is introduced, against which recognition takes place.

This hierarchy consists of a conveniently structured library containing typical plans (sequences of events with partial order relationships between them), which can be adapted if necessary to specific circumstances. The typical plans (also called complex operations) stored in the library result from a composition of other possibly complex and/or basic operations (our set of Propp-inspired operations). Complex operations formed by generalization are also represented, branching down to specialized operations corresponding to alternative ways to reach the same main effects.

Also, for composing a plot, a modular strategy may look attractive (like the card's method – described by both McKee [1997] and Field [1982], a simple technique, very popular among movie script authors). One may start (along the syntagmatic axis) aligning events corresponding to large narrative units. At a subsequent stage, one moves down in depth, so to speak, to explain these broad events in terms of smaller episodes richer in detail. In fact, this "package" approach recalls the use of story models like the monomyth-inspired writer's journey [Vogler 2007] as support for the narrative.

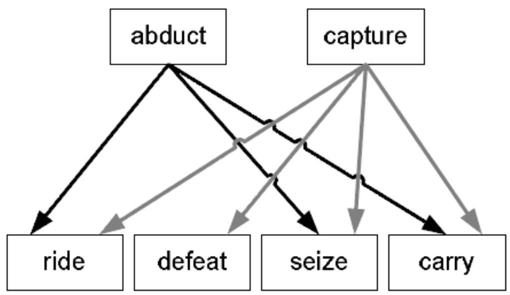

Figure 5.1: Example of decomposition via meronymic relations.

LOGTELL only dealt with the decomposition of events during its dramatization phase, when breaking down an event into sub-actions readily preparable for animation. It is interesting to note that decompositions are not fixed, being context-sensitive compilations (lower-level operations are selected as required by the current state).

Putting to use this strategy of iteratively breaking down plot events - or of assembling a hierarchy of events and generalized events - implies the existence of a third axis, which we call the *meronymic axis* (meronymy being understood as the relation between whole objects and their parts). As such, we shall call **meronymic relations** those that hold between an event and a lower-level set of events that can provide a more detailed account of the action at hand. Meronymic relations, such as those represented in Figure 5.1, open the opportunity for event decomposition or, going in the opposite direction, event summarization; the formation of mixed plots combining events of different levels can also be considered.

Though not all-encompassing, these three relations cover the generation of a wide range of stories, at a higher or lower degree of detail. On the one hand the relations are closely associated with three out of the "four master tropes" indicated by Kenneth Burke [1969]: *metonymy*, *metaphor*, and *synecdoche*, corresponding, respectively, to displacements along the **syntagmatic**, **paradigmatic**, and **meronymic** axes (metonymy is taken here in the specific sense of substitution justified by contiguity [Koch 1999]). On the other hand, they also appear relevant in the context of ontologies [Staab and Studer 2004]; in terms of ontologies, displacement along the three axes signifies the traversal of hierarchies of concepts, connected via next-to, is-a, or part-of links.

While normal plots, fully belonging to the modelled genre, can be composed exclusively on the basis of these relations alone (or even of the two first ones, if the level of detail is disregarded), the possibility to introduce unexpected turns is often desirable in order to make the plots more attractive.

Such story *twists* would be brought about through the insertion of events that in principle should not occur together in a coherent sequence (i.e. events that contradict each other's pre-/post-conditions scheme), without causing sharp discontinuities in the context. In particular, a sudden change of beliefs or feelings can give rise to surprising sequences. Also, both in fiction and in reality, things do not always proceed according to planned events. Natural phenomena, disasters, the mere passage of time, intervention of agents empowered to change the rules, supernatural or magic manifestations, etc., cannot be discounted.

Specifically for the tragedy genre, the Poetics [Aristotle 2004] distinguishes between simple and complex plots, characterizing the latter by the occurrence of *recognition* and *reversal*. Differently from reversal, which disrupts the current situation with a drastic effect on the factual context, recognition does not imply that the world itself has changed, but only the beliefs of one or more characters about the actual facts. Mechanisms to support this kind of dramatic change must be provided, one of the possibilities being to allow arbitrary user interventions at certain points (recalling the *deus ex-machina* device of the ancient theatre).

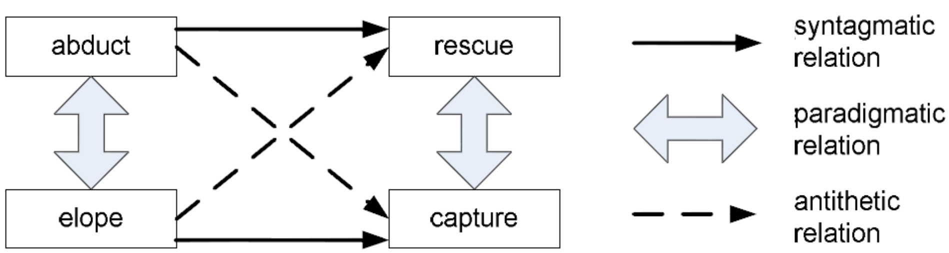

Figure 5.2: Syntagmatic, paradigmatic, and antithetic relations.

To characterize irregular event sequences, we introduced **antithetic relations**. Curiously, the fourth and last of the main tropes listed by Burke - namely irony - sounds as a deviation from the right path; leading to the transgression of rules. Figure 5.2 represents in a single diagram syntagmatic, paradigmatic and antithetic relations.

By offering mechanisms derived from these four relations - connected to the four master tropes, and following the previously mentioned proposal [Furtado 2008] - we try both to augment the expressiveness of LOGTELL's narrative model and to provide better support to authors who are less familiar with or confident in creating and telling stories.

5.5.2.Genre as Story Space

Finding an answer to the question of what constitutes a given literary genre should allow one to determine whether or not a story can be classified as belonging to it. However proving that a conceptual model **M** for specifying a genre **G**, formulated as discussed in this paper, can fully capture how the genre is understood by a literary expert can be an over-ambitious effort. And yet, nonetheless, a useful approximation is attainable, taking the form of a definition by extension. If a system is equipped with a device to generate stories according to the specified conceptual model **M**, the space of plots that can be generated might be regarded as constituting a genre $\mathbf{G}^{\mathbf{M}}$ defined by the model, where plot has the usual meaning of sequence of events. It would be left to the designer of **M** the task of refining the model specification in order to achieve increasingly closer matches between the intuitive view of **G** and $\mathbf{G}^{\mathbf{M}}$.

This is basically the role of the conceptual model presented in Section 5.4. As argued above, the conceptual model determines the genre of the stories to be generated in correspondence to the story space of a system that uses that model. It is important to stress that the types of events allowed in the genre being defined will be restricted to a fixed repertoire of actions, in our case the set of Propp-inspired operations. Such definition of genre as the set of stories that can be generated from the model is consistent with the way game designers discuss/describe genres by example [Wallis 1997].

While the story space of our example scenario refers to a simple swords-and-dragons genre, our conceptual modelling method should be able to cope with an ample variety of genres of higher complexity.

The structure of each story space thus specified is determined by the observed event relations. From an informal viewpoint (cf. Figure 5.3), events can be seen as nodes and syntagmatic, paradigmatic relations as connecting edges drawn over a plane, while meronymic relations appear as projections of events into event sequences over another plane. Antithetic relations between nodes can be seen as specifying constraints crosscutting the story space. Recall that paths formed by syntagmatic relations correspond to plots, whereas paradigmatic relations indicate alternatives – whose choice may entail, as signalled by antithetic relations, the exclusion of certain other alternatives branching from subsequent nodes.

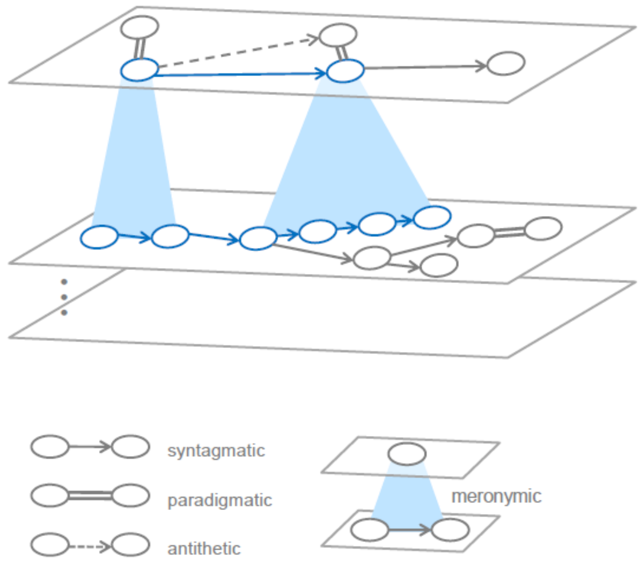

Figure 5.3: Relations between events in the story space [Furtado et al. 2009].

5.5.3.Plot Libraries

The identified relations also bring some extra benefits, since previously existing plots (no matter if composed manually or automatically) can be converted into *plot patterns* - obtained by consistently substituting variables for the constants figuring in the various parameter positions - to be kept in a library of typical plots. Plot libraries can be conveniently organized as combined is-a and part-of hierarchies, which is a convenient way to deal with plots by taking advantage of the similarity or analogy among situations.

While the part-of links result directly from meronymic relations, denoting in consequence *composition* / *decomposition*, the is-a links express the notion of *generalization* / *specialization*. If two or more events stand in a paradigmatic relation, they can be said to belong to the same paradigm, in that their main effects should coincide, although they may differ with respect to less important side-effects. For instance, we mentioned earlier that *abduction* and *elopement* as alternative forms of villainy. This justifies the introduction of a new event, which might be named *villainy* (or some more specific term), and its placement in a hierarchy above the other two one-event plots, connected by the links *abduction* is-a *villainy* and *elopement* is-a *villainy*. In words this means that *villainy* generalizes both *abduction* and *elopement* and, conversely, that the latter specialize the former.

Thus both meronymic and paradigmatic relations contribute to the hierarchical structure of plot libraries. And, once a plot library is constructed, its component plot patterns at different hierarchical levels can then be reused during the plot composition process to help create new plots.

Because our approach employs plan-generation/plan-recognition while composing a plot, we regard "typical plots" in our context as synonymous to *typical plans*.

5.5.4.Some Remarks

Finally, let us recall that we have addressed the *fabula* narrative level only, where one simply indicates which events should be included in the plots. One especially complex problem to be faced at the next level – story/narration, where the concern is how to tell the events – is how to properly convert the events into forms adequate for dramatization.

The facilities associated with the four relations are adequate for different tasks instrumental to support the production of coherent and diversified stories with more dramatic power – and increasing their ability to cause surprise. They were conceived for the interactive creation of stories, having already been tried in a storyboarding project [Ciarlini et al 2008], and may possibly be adapted in future projects to game-like interactive storytelling systems, under suitable user interfaces. These relations also bring a particularly convenient way to deal with entire plots, helping visualise their dramatic structure.

Generally speaking, the proposed model seems to offer a sound basic groundwork for continued developments. The claim that it provides an ample coverage is supported by the indicated connection between the four described relations and the four master tropes singled out by semiotic research. For the purposes of the present research, we argue that, by extending LOGTELL to accommodate all four relations, we move towards our goal of augmenting its narrative model expressiveness.

5.6.LogTell-R

After analyzing the available literature on story creation methods and story generation systems [Karlsson et al. 2009b], we noticed a lack of truly generative systems offering adequate features for modelling a given genre (or story space) and enabling the construction of knowledge bases for interactive storytelling.

In an attempt to enhance the diversity of plots that can be created by using LOGTELL's conceptual model, we seek to increase the flexibility of the story space it allows by extending the dynamic schema level, i.e. the set of operations that can transform the storyworld, in ways that take into consideration the four relations previously presented.

We focus not simply on different ways of telling pre-existing stories, but especially on the dynamic creation of plots. The extended model comprises typical events, complex events, goal-inference rules, and special events that embody narrative tools to possibly make the generated plot more attractive, while still trying to conciliate both plot-based and character-based modelling, and respecting the constraints of a genre.

As mentioned before, the Plot Manager module originally dealt only with the syntagmatic and paradigmatic axes. Meronymic relations were only handled (in a single direction) at the dramatization phase by LOGTELL, when plot events are broken down into sub-units/actions necessary for animation.

We now deal with meronymic relations by introducing complex operations in the dynamic schema. A complex operation results from the composition or from the generalization (cf. Section 5.5.3) of other operations (either complex or simple). As such, these complex operations represent abstractions whose semantic contents derive from their components, making it easier to detect inter-relations between parts of a plot. This feature makes use of the support for complex operations present in the Interactive Plot Generator algorithm (cf. Section 5.6.1) part of the original version of LOGTELL.

As the counterpart of generation, comprehension emerges as one parses a plot against this mixed hierarchy of composition and generalization, moving upward until reaching the topmost node, at which the whole plot is recognized as a valid instance of the genre. We call this "tree" the hierarchy of typical plans (Figure 5.4 in Section 5.6.1). The benefits of such structure are twofold: it helps better visualize story structure, showing how to adequately chain events into a narrative scheme (like the three-parts classic structure [Aristotle 2004, Campbell 1968, Guerra 2008]); and it helps in more easily understanding and analyzing the space of stories that can be generated. Another benefit of the structure is that it can be used in the plan recognition process, as will be discussed later.

When dealing with support for antithetic relations, since we emphasize interactive composition, we shall also consider the possibility of user interventions that can result in some sort of discontinuity being produced by ironical shifts. For example, if some binary opposition – like the "to love or not to love" dilemma – is allowed to be manipulated, then one can have plots that no longer look conventional. These disruptions can also happen as a result of natural disasters, passage of time, supernatural or magic manifestations, etc., thus opening a vast array of possibilities.

Moreover, not only mental attitudes, feelings and statements can be ironic – actions can also be ironic, typically in an unplanned, non-deliberate fashion. Irony is in fact a characteristic of certain intrigue situations that are often referred to as dramatic irony [Booth 1974]. Indeed, irony is what commonly induces antithetic relations between events which look, in principle, incompatible with each other. Mediating two such events, the until then well-behaved world must suffer a disruptive shift, whereby the truth value of certain facts or beliefs is inverted, or certain properties, such as degree of happiness, move from one extreme to the other within some ascribed value range.

LOGTELL's only example of the use of antithetic relations was the optional usage of the bewitch operation that was implemented during previous experiments; an operation typically outside the scope of the generated stories. To better support the manipulation of facts/beliefs and mediation between incompatible events, we now add support for inserting motifs [Thompson 1989] in the dynamic schema. Motifs can provide specific ways whereby antithetic relations can be handled, while preserving the properties and constraints of the dynamic schema.

A motif here is nothing more than a special "black-box" operation not to be executed by a character in pursuing its goals, but to be inserted as a special narrative tool to allow the manipulation of facts, beliefs or previously unacceptable sequences of events, quite in a *deus ex machina* way. One of the advantages of using motifs is that they often encode ingenious solutions to contradictions or dead-ends in a tale, while both fitting the given genre and being acceptable and attractive.

The changes introduced in our new prototype (LogTell-R) will be discussed in more detail in the next sub-sections. In Section 5.6.1 we discuss plot generation in face of the addition of plan recognition. In dealing with extending the conceptual model, specifically at the dynamic schema level, we discuss the creation and use of the hierarchy of typical plans in Section 5.6.2. Further details on how the system handles the addition of motifs are provided in Section 5.6.3.

LogTell-R (pronounced log-teller) is also being extended to address other requirements of interactive storytelling systems, such as player interaction during dramatization. Our approach allows us to use the same architectural infra-structure to try and handle different user roles with minimum implementation impact. A detailed description of the previous modules is not part of this paper and can be found in Pozzer's [2005] PhD thesis. More details on LogTell-R from a software architecture point of view can be found elsewhere [Karlsson et al. 2010b].

In the present version of LogTell-R, design choices were guided by a more "broad and shallow" approach with respect to some sub-modules (though not to the story model). Whereas some individual features might have benefited individually from a deeper specification and research effort, some pressing constraints such as authorability, feasibility, and interoperability were first taken into account due to time limitations, in order to achieve a prototype architecture of the system that is able to operate as a whole and to address the main issues under discussion.

5.6.1.Plot Generation

The core engine for the generation of plots is the Interactive Plot Generator (IPG), which incorporates a STRIPS-like planning system, offering a temporal modal logic formalism for the formulation of goal inference rules. A more detailed description of the planning system is given in [Ciarlini et al. 2005].

Another feature provided in IPG is a plan-recognition algorithm which can be used with a conveniently structured library (hierarchy of pre-existing typical plans), to pick a plan and adapt it if necessary to specific circumstances. The method consists of matching observed events against the plan definitions (also called complex operations) stored in the library, trying to find one or more plans of which these events may be part.

Our typical plans (complex operations) have the same syntax used for (basic) operations. If the complex operation results from a *composition* of other possibly complex and/or basic operations, the last parameters will declare any order requirements holding between them. Complex operations formed by *generalization* are also represented, branching down to specialized operations corresponding to alternative ways to reach the same main effects; clauses is_a(<more-specialized-operation>, <more-general-operation>) declare this structural link.

By making use of this feature not previously available in LOGTELL, the new infra-structure makes room for an additional option in the successive cycles of plot generation: goal-inference, planning, *plan-recognition*, and user intervention.

In LogTell-R, plot generation still happens in a step-wise fashion, starting from a given initial state. At each state, the goal-inference rules are used to induce short-term goals. As one might expect, moving along the syntagmatic axis is the primary task of the plan-generator, as it composes a coherent plot by aligning events in view of the pre- and post-conditions of predefined operations. Adding or deleting operations and changing their sequence can also be seen as manipulations regarding the syntagmatic relations.

As the paradigmatic axis has to do with possible alternatives, it can be seen as useful in a plot adaptation phase. Replacing one or more operations is one way to produce alternatives, but a more interesting way might be to ask the planner to provide alternatives for the newly inserted events upon backtracking. Yet another way of choosing alternatives is by selecting events and asking the plan-recognizer to retrieve complex operations (using the typical plan library) containing them, as will be explained in the sequel. Both these options are shown in the author flow diagram in Figure 5.5.

Exploiting the meronymic axis at different levels of detail, by using the plan recognizer, plausible alternatives may be located by essentially walking up and down over the plan hierarchy. For example, selecting lower-level events *attack* and *fight* in a plot, and asking the plan-recognizer to check them against the library, causes the system to identify these events as part of the higher-level *murder* event, and suggest the addition of the *reduce_protection* and *kill* events. If not satisfied, the user can ask for another alternative.

Also, one may wish to climb higher in the hierarchy to consider what events to insert (we note, however, that this functionality needs more development, especially from a user interface point of view). Following the previous example, if, after the system identifies the selected events as part of *murder*, the user decides that he wants a solution identifiable from the level next above, the system could then suggest an ampler set of events forming a complex *avenge* event.

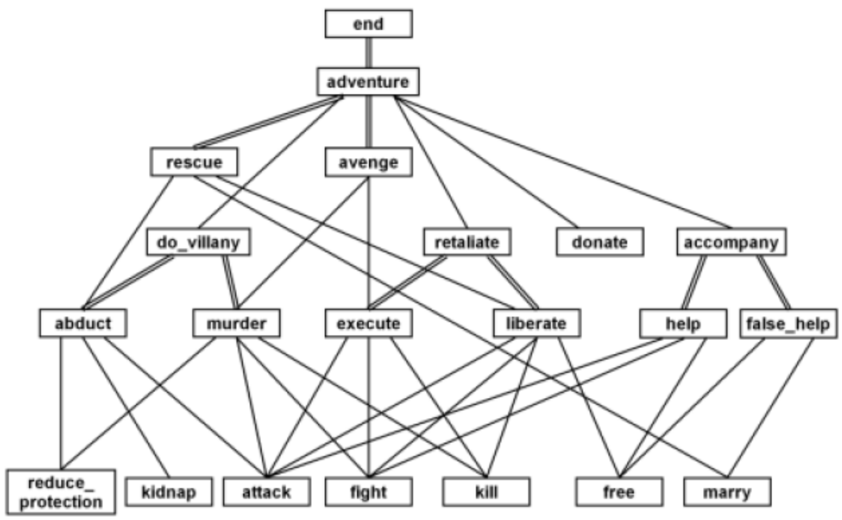

Figure 5.4: Hierarchy of typical plans used in most of our experiments.

As an additional possibility, instead of asking the system to offer suggestions, the user may simply find it helpful to visually inspect the library of typical plans associated with the genre he is dealing with at the moment (again such as that of Figure 5.4), as a clue from which he may extract inspiration, while guiding the system to compose a plot of his liking.

In order to support the presence of antithetic relations one may need to interfere in the course of plan generation so as to accommodate the presence of events that should not in principle follow each other. This could be done by inserting intermediate events, or by issuing some kind of special *directive*. Both are forms of strong interaction (more details in the user interaction section), applicable either during composition or adaptation. In [Ciarlini et al. 2009], an example is given wherein directives to alter character beliefs and certain user-controlled facts are introduced; during plan-generation, a feature is supplied to automatically detect failures involving beliefs and user-controlled facts, which assists the author to formulate the appropriate directive if he so desires.

One form of issuing directives is by way of special commands directly in the Plot Manager, with the effect of tampering with the underlying context, but this strategy has several drawbacks. The implementation would require changes in the planning system as currently used, purporting to validate plans supposed to fail in face of the inserted contradictions. Also, directly modifying events or feelings along the plot is a too explicit *deus ex machina* device, not quite acceptable to modern audiences. One additional drawback is that such manipulations would be troublesome during dramatization, as there would be nothing in the context to help showing why or how they happened at that point in the plot.

As an addendum to typical plan hierarchies, one may draw from folktales, myths, and popular culture, many of which have to offer, under the form of *motifs*, a rich repertoire of possible ingenious solutions to issues such as those arising from antithetic conflicts. The solution we propose utilizes this idea. The author can choose a motif (from a pre-defined set) to be inserted at the position in a generated plot where the respective disruptive situation holds. The motif is then inserted as a (pseudo-) operation in the plot, just like the regular operations, and its insertion is provisionally permitted by the plan-generator.

5.6.2.Creating the Hierarchy of Typical Plans

A library of typical plans can be either constructed with or without the help of an automatic system. In [Furtado and Ciarlini 2001] an algorithm is described to obtain the library through what might be called a plot-mining process. The algorithm involves the analysis of a *log* where the actual execution of events has been registered for some time, employing a previously implemented system. Basically, an attempt is made to find in the log, for each goal-inference rule, a point where the motivation situation of the rule holds, as also a later point where the corresponding goal holds. It is assumed that the sequence of events between the two points should reveal (after a filtering process) a successfully executed plan to achieve the goal. As a series of plans is being so identified, the algorithm proceeds to incrementally build the hierarchical structure, under the form of complex operations connected by is-a or part-of links to denote, respectively, generalization and composition.

Here we shall briefly sketch how the creation of the hierarchy can be described with the help of the plot manipulation algebra (PMA) proposed in [Karlsson et al. 2009a]. A 'partial' account of the construction of the structure shown in Figure 5.4 will serve as example.

Consider the operations *abduct*, *murder*, *liberate*, and *execute*. The following remarks promptly come to mind:

1. The set {*abduct*, *murder*} corresponds to the same paradigm, as forms of villainy

2. The set {*liberate*, *execute*} corresponds to a second paradigm, as forms of retaliation
3. A villainy calls for a retaliation, and thus sequences taken from the two above paradigms make sense; note that four different sequences can be formed
4. The sequence formed by (*murder*, *liberate*) is not valid, since liberate presupposes that the victim is alive

It turns out that these remarks, clearly motivated by paradigmatic, syntagmatic and antithetic relations between events can be expressed in terms of PMA. Since a systematic discussion of PMA is outside the scope of this paper (cf. [Karlsson et al. 2009a] for details), a very simplified notation will be used for convenience:

1. Union to express generalization: `do_villainy := abduct + murder`
2. Union to express generalization: `retaliate := execute + liberate`
3. Product to produce all sequences: `P1 := do_villainy * retaliate`
4. Difference to exclude invalid sequences: `P2 := P1 - murder * liberate`

The notion of being typical (from a common sense viewpoint, or by figuring in popular stories, etc.) must also be considered. Notice that, though not properly invalid, the sequence (*abduct*, *execute*) hardly qualifies as typical, and might also be excluded via difference.

Two resulting complex operations, namely *do_villainy* and *retaliate*, can immediately be added to the hierarchy of Figure 5.4, as is-a generalizations. The exclusion of the invalid and of the non-typical sequences is not represented directly in the hierarchy. To understand how it works, one must bear in mind (cf. caput of Section 5.6) that the bottom-up plan-recognition algorithm only succeeds for sequences that can be parsed by climbing along the links until the top node (*end*) is reached. Now observe that the *adventure* complex operation, just below *end*, is placed as a generalization of *rescue* and *avenge*; in words: these two are therefore the only typical adventures contemplated, the former combining, besides other complementary operations, the sequence (*abduct*, *liberate*) and the latter (*murder*, *execute*).

The PMA formalism provides two inverse operators, namely factoring and combination, associated with the meronymic dimension. For instance, *murder* is mapped by factoring into its lower-level component events by `P3 := ϕ murder`, yielding the sequence (*reduce_protection*, *attack*, *fight*, *kill*), as represented in Figure 5.4.

The transgressive *bewitch* operation is absent from Figure 5.4, a purposeful omission to indicate that it is not regarded as a regular component of typical plots.

When retrieving from the hierarchy, to extract an operation or else an entire sequence or subsequence of operations, the selection and projection PMA operators are applicable. Combination, which is the inverse of factoring, can play a very special role. One may wonder what a sequence of low-level operations, such as (*reduce_protection*, *attack*, *fight*, *kill*), signifies in a narrative. Combining them is tantamount to recognizing the sequence as a *murder* event.

As stated in [Karlsson and Furtado 2009], the direct use of PMA is not recommended, a remark applicable in fact to any logic level formalism. We circumvent the problem by using PMA inside a proof-of-concept editor for the knowledge base. This editor also allows the addition of some schema metadata (especially regarding generalizations and antithetic relations).

5.6.3. Motifs as Solutions for Inconsistencies

If we see a disruption not as a discontinuity in one context, but as an attempt to put together two originally incompatible contexts, then the notion of *blending* [Casanova et al. 2008] immediately comes to mind, as the technique or artisanship of conciliating the pending conflicts, which often requires a great deal of creativity.

Incompatibilities between events – which we have been characterizing in terms of antithetic event relations – are usually induced by some factor involving the current value of a property or the beliefs of a character. Indeed, specifically for the tragedy genre, the Poetics [Aristotle 2004] distinguishes between simple and complex plots, characterizing the latter by the occurrence of *recognition* and *reversal*, noting that recognition, unlike reversal, does not imply that the world itself has changed, but rather the beliefs of certain characters about the actual facts.

In ancient times, conflicting situations were sometimes disentangled by lowering a supernatural personage onto the stage by means of a crane – the so-called *deus ex machina* device. This is comparable, in our times, to providing support to the user of a computer-based system to impose variations to the context, thus literally acting *ex machina*, with the purpose of deviating the action from its predicted path. In any case, the justification for such arbitrary interventions is that, if the factors that cause incompatibilities can be manipulated via whatever agency external to the predefined events is involved, then one can have plots that no longer look conventional.

Yet these crude tactics do not appear to be acceptable nowadays. Fortunately, blending can be achieved in a convenient way by resorting to folktale *motifs* [Thompson 1989], which often encode ingenious solutions to contradictions or dead-ends in a tale, while seeming to fit with the conventions of genres stemming from popular culture, and looking acceptable and attractive to most audiences.

A genre can in general be seen as a contract between author and audience [Glassner 2004] – in the sense that the audience knows what to expect – even if some event almost crosses the borderline of plausibility. Folktales and myths have pooled together along the years rich repertoires of ubiquitous motifs, which now can be grafted into story spaces such as that of our example world, which is inspired by popular medieval fantasy.

By carefully choosing a motif, an author can contrive circumstances, closely congenial with the genre, whereby characters may be induced to falsely believe in something, thus fulfilling the situation that triggers a goal-inference rule that leads to the desired event. Alternatively, the selected motif can manipulate some user-controlled fact, directly changing the world state in a way not previously expected.

In order to support this kind of mediation, LogTell-R adds the possibility of inserting a motif at an appropriate position in the plot. This motif must come from a pre-defined set, and takes the form of a special kind of (pseudo-)operation, to be added to the dynamic schema of the conceptual model. We associate motifs with either a <situation, goal> specification or with a set of post-conditions. Motifs inserted in a plot should be annotated with additional information, so that the planner may distinguish them from the regular operations when generating a plot.

In our proof-of-concept experiments we selected a set of five motifs appropriate to our example: a) *Life token* – allows to do without the unrealistic assumption that characters are omniscient; b) *Love potion* (instils romantic feelings) – provides an excuse for sudden variations in amorous attachments; c) *Ordeal by fire* (to vindicate a discredited person) – restores a man's belief in his beloved's faithfulness; d) *Challenge* (proof of valour by sword or game) – demonstrates that a character deserves to be granted some favour; e) *Magical carriage* (magic displacement) – enables sudden and quick changes in location.

Motifs can also act as dramatization tools, to display to the audience a convincing explanation of why certain events must ensue. One example is the *life token* motif, where the visual aspect of an object must be transformed, for example to show how the hero knew the princess was in distress, motivating the beginning of his quest to save her. In our proof-of-concept knowledge base editor, a motif is associated with a script in Lua defining the behaviour to be followed by the involved characters when dramatizing the motif (this facility is supported by the underlying MIAGI middleware [Karlsson 2005]).

5.7. User Interaction

This section describes user interaction in LogTell-R **author mode** (i.e. via plot manager). Those who have no special talent for literary composition, like ourselves, find it difficult to invent interesting plots, but usually do not feel so uncomfortable if asked to adapt an existing plot, by introducing small modifications in a gradual fashion. The main thrust of the system consists of providing the user with efficient means for exploring coherent alternatives to the story at a given state, and guiding the plot at the level of events and characters' goals. In the role of **author**, the user has direct control over the Plot Manager whereby he can participate in the choice of the events that will figure in the plot and decide on their final sequence.

The Plot Manager, as already in the original version of LOGTELL, provides two basic forms of interaction: *weak intervention* (by using the plan generation features) and *strong intervention* (by allowing the insertion of events and situations). In LogTell-R, the system allows, in addition, the use of a plan recognition feature and the manipulation of the plan hierarchy interface to help the composition/adaptation of a plot. More details on the specific commands are presented elsewhere [Karlsson et al. 2006a]. In any case authors are still constrained to act indirectly, in the sense that any user intervention must be validated by IPG before being truly incorporated into the current plan.

In weak intervention, moving along the **syntagmatic axis** is primarily the task of the plan-generator, as it composes a coherent plot by aligning events in view of the pre- and post-conditions of the appropriate predefined operations. The author merely selects partially-generated plots that seem interesting from his perspective to proceed with the simulation.

As we are dealing with an interactive environment, the author is allowed, at any step, to further intervene, reducing thereby the characters' autonomy, while relying on the plan-generator to enforce consistency within the genre. To this purpose, certain forms of strong intervention are provided, allowing the author to indicate for insertion a goal (to be tried by the plan-generator) or even a specific operation, which the plan-generator may or may not find applicable. A more complex request is to indicate a sparse list of operations, to be filled-up until a valid plot sequence containing all operations in the list is formed, possibly interspersed with other operations.

Again at any step of plan generation, the author can ask the planner or the recognizer for alternatives. Moving along the **paradigmatic axis** gives ampler opportunity to obtain different plots than simply changing the sequence of events within the partial order requirements. Notice too that, to begin with, alternatives may result from starting from a different initial state, where different goal-inference rules may be triggered.

LogTell-R supports a form of strong intervention specifically in view of **antithetic relations**, for inserting an adequate motif (taken from a pre-defined set) at the appropriate point in the generated plot where an inconsistency is detected. This insertion is then subjected to validation by the planner. In our experiments, we used the five motifs previously discussed.

Dealing with the **meronymic axis** is a task of the plan-recognition sub-system. When using the plan-recognition feature, the author needs to mark one or more events already inserted (and/or being considered for insertion) in the Plot Manager user interface. The system will try to match these events, as observations, against the library of typical plans, in an attempt to identify one or more typical plans subsuming them.

The matching typical plan (complex operation) is highlighted over the plan hierarchy, and the author can then determine if it is an interesting option that fits the intended story. In case it is not, the plan-recognition sub-system can suggest alternatives, if any exists, that may eventually satisfy the author's preferences – which constitutes yet another way of moving along the **paradigmatic axis**. If a partial plan is found satisfactory, its component events will be inserted via the Plot Manger interface.

In LogTell-R plot composition (or adaptation) still happens in stepwise fashion, as successive cycles of goal-inference, plan generation, plan recognition, and user intervention. As we have been stressing, all user-supplied insertions are always validated before being incorporated into the plot. The entire flow is sketched Figure 5.5.

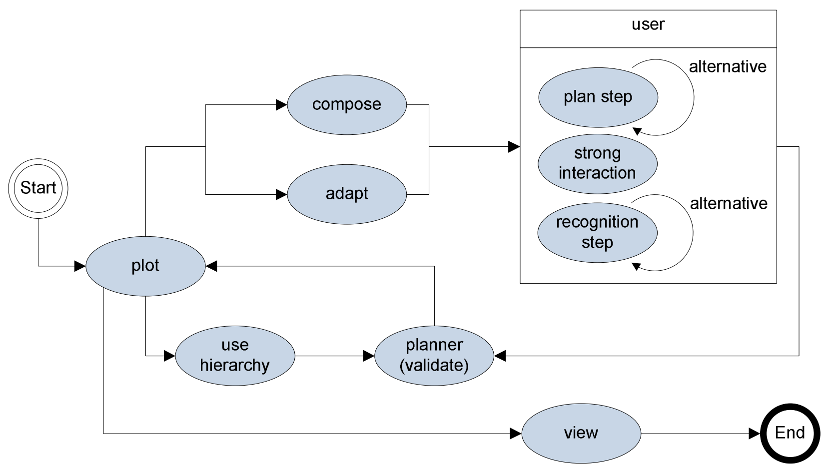

Figure 5.5: Flow of author interaction in LogTell-R's Plot Manager.

After the step-wise process terminates, it should still be possible to perform various kinds of adaptation. Those that have to do with the syntagmatic relations include adding or deleting operations and changing the sequence, if the partial order requirements imposed by the interplay of pre- and post-conditions permit.

At the adaptation phase, the ability to replace one or more operations can also be seen as a way to produce alternatives. It is important to note that if the final order has been established between a subset of the events and validated by the planner, this order can no longer change. Further enhancements to the adaptation phase can be implemented by using "snapshots" of the validated plot so the author can go back to those points and edit the story from there.

When composing a plot, the plan-generator is free to mix operations of different levels in the hierarchy. Once a plot is composed, it can also be adapted either by detailing or summarizing its constituent operations, if a multi-level typical plan library is available (however, this functionality is not fully functional in the current interface).

After all the changes are incorporated into the plot and the final ordering of events is established, one can, as in all versions of the system, watch the story unfold in a 3D world. And, if further adaptations are suggested by the visual experience, there always remains the possibility of returning to the plot-composition stage and restart the process.

5.8.Final Remarks

Most current approaches to interactive storytelling do not propose to address the problem of what constitutes a proper story space or how to model the characteristics of events pertaining to a plot, preferring to focus, perhaps too strongly, on more controlled environments and stories. We feel that more generative approaches are more promising for the field of interactive storytelling to reach a wider appeal and become truly successful. It is important to highlight, however, that we did not propose here to achieve a fully immersive experience. We endeavour, instead, to explore the possibilities of generating a large variety of coherent stories through the use of a plan-recognition/plan-generation paradigm over the conceptual model of a genre.

Our approach is oriented to finding sound methods to organize and combine events, in ways that may confer enough dramatic power to the narratives being generated. By expanding LOGTELL's conceptual model to encompass the four event relations identified in [Furtado 2008], and adding better support mechanisms for their manipulation, we strived both to augment the expressiveness of the narrative model and to provide better tools to prospective authors intent on creating and telling stories. Although the process of plot composition and adaptation could surely be enriched far beyond what is presented here, what we managed to accomplish seems to provide a sound initial basis to treat, at least, genres that exhibit a high degree of regularity.

To accomplish these goals, we focused on enhancing the flexibility of the story space spanned by the system, by extending the dynamic schema specification, especially in view of meronymic and antithetic relations. To the limited repertoire of pre-defined operations we added complex operations, as defined in [Ciarlini 1999], that can represent generalization and composition abstractions, and a genre-fitting mechanism (motifs) for user interventions aimed at the production of shifts and discontinuities in the storyworld context. Our approach is indeed centred on hierarchies of typical plans, structured by connecting complex operations via is-a and part-of links, and serving as a repository of past narrative experiences wherefrom authors may profitably draw along the composition process.

Given that, as a rule, conceptual models tend to remain too far removed from system implementations, we follow the tradition of trying to narrow the gap with the help of an intermediate logical design stage. At this stage, we use a plot manipulation algebra (PMA) [Karlsson et al. 2009a] for the creation of the hierarchy of typical plans. An added benefit of using PMA (supported by a simpler planner to check pre- and post- conditions of operations) in the knowledge base editor is that one can more easily experiment with different ways to chain the events, and use this knowledge to revise the specification so as to improve the system's story space. We are aware that effective systems should in fact be able to offer even more opportune guidance – or at least helpful clues – at each composition stage.

Also, we argue that the functionalities added in LogTell-R can be used with minor modifications in a unified system that can handle both author and player interaction modes [Karlsson et al. 2010b]. Having authorial tools in place to make the notational complexities of the knowledge base transparent to authors is a very desirable goal, primarily to reduce the burden on authors using the system, but also to guide them in the creation of interesting stories. Creating an interactive environment that behaves as expected can be a tiresome task, as it's done now mostly by trial and error. While our proof-of-concept tool to edit the knowledge base still leaves much to be improved, we feel it brought important insight, and we now have a much better grasp on the priorities for future work.

Some areas for future improvement include, among other topics: evaluation of "best plot" alternatives via heuristics; further investigation of our plot manipulation algebra (and possibly its stronger integration in the system); and better support for altering the plot hierarchy/library during plot composition.

Finally, let us recall that we have addressed the fabula level only, where one simply indicates which events should be included in the plots. A complex problem is how to deal with event translation at the next level – the story level, where the question is how to tell the events – especially when contextual disruptions are introduced via user interaction and even more if fanciful motifs are annotated in the plot.