PUC

ISSN 0103-9741

Monografias em Ciência da Computação
n° 30/08

A Four-sided View of Plot Composition (and the PlotBoard Tool)

Simone Diniz Junqueira Barbosa   Angelo Ernani Maia Ciarlini
Antonio L. Furtado   Marco Antonio Casanova

Departamento de Informática

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO

RUA MARQUÊS DE SÃO VICENTE, 225 - CEP 22451-900

RIO DE JANEIRO - BRASIL

Monografias em Ciência da Computação, No. 30/08 ISSN 0103-9741
Editor: Prof. Carlos José Pereira de Lucena July 2008

A Four-sided View of Plot Composition* (and the PlotBoard Tool)

Simone Diniz Junqueira Barbosa Angelo Ernani Maia Ciarlini+ Antonio L. Furtado
Marco Antonio Casanova

{simone,furtado,casanova}@inf.puc-rio.br, angelo.ciarlini@uniriotec.br

Abstract: The use of suitable methods for plot composition is a key issue for obtaining interactive storytelling systems that are able to entertain and surprise. In this paper, it is argued that the process of plot composition can be viewed under a four-sided perspective, induced by the presence of syntagmatic, paradigmatic, antithetic and meronymic relations between the constituent events. In turn, these relations are shown to be associated with the four major tropes of semiotic research. A modelling discipline is described, together with a set of facilities for interactive plot composition and adaptation, on the basis of these four relations. To accommodate antithetic relations, corresponding to the irony trope, our plan-based approach leaves room for the unplanned. To illustrate the discussion, as well as the design and use of a logic programming prototype implementation, we employ an example involving a small number of events, which, in strikingly different combinations, have been treated repeatedly in literary works.

Keywords: storytelling, plots, plan-generation, narratology, tropes.

Resumo: O uso de métodos adequados à composição de enredos é questão chave para a obtenção de sistemas de composição interativa capazes de entreter e de surpreender. Neste artigo, argumenta-se que o processo de composição de enredos pode ser visto sob uma perspectiva quadrilateral, induzida pela presença de relações sintagmáticas, paradigmáticas, antitéticas e meronímicas entre os eventos constitutivos. Por sua vez, essas relações estão associadas com os quatro tropos principais da pesquisa semiótica. Uma disciplina de modelagem é descrita, juntamente com um conjunto de facilidades para a composição e adaptação interativa de enredos, com base nessas quatro relações. Para acomodar relações antitéticas, correspondentes ao tropo da ironia, nossa abordagem baseada em planejamento abre espaço para o não-planejado. Para ilustrar a discussão, bem como o projeto e a utilização de um protótipo implementado em linguagem de programação em lógica, empregamos um exemplo envolvendo um pequeno número de eventos, os quais, em combinações marcadamente diversas, tem sido tratados repetidas vezes em obras literárias.

Palavras-chave: narração de estórias, enredos, geração de planos, narratologia, tropos.

* This work has been partly sponsored by the Ministério de Ciências e Tecnologia da Presidência da República Federativa do Brasil.

+ Unirio - Departamento de Informática Aplicada

In charge of publications

Rosane Teles Lins Castilho
Assessoria de Biblioteca, Documentação e Informação
PUC-Rio Departamento de Informática
Rua Marquês de São Vicente, 225 - Gávea
22451-900 Rio de Janeiro RJ Brasil
Tel. +55 21 3527-1516 Fax: +55 21 3527-1530
E-mail: bib-di@inf.puc-rio.br

1. Introduction

Interactive storytelling is today a source of entertainment, in which authors, autonomous characters and audience interact for generating stories. Different approaches have been proposed, some of them centered on autonomous characters, such as in [CCM], favouring interaction, and others on plot structure, such as in [GB], aiming at coherence and dramatic power. Depending on the system's emphasis and on the chosen genre, it may prove more effective to focus either on characters or on the plot (or try a combination of both, such as in [MS]). For coherence and diversity of stories, the use of automatic planning algorithms [CPFF, RY] is a promising alternative for exploring chains of events that achieve the characters' goals, or those of the story. Coherence and diversity are essential for plot composition but do not guarantee dramatic power. The impact on the audience very much depends on how the events are combined. In [Sg], for instance, an Aristotelian conception of plot is used to lead the story to a climax and then resolve it. Generally speaking conciliating interaction with coherence, diversity and dramatic power is not an easy task for a computerized storytelling system. Though still believing that automatic planning is helpful, we now recognize that sound methods to combine events must be considered in order to confer enough dramatic power to narratives.

Narratology studies [Ba] distinguish three levels in literary composition: *fabula*, *story* and *text*. At the fabula level, the *characters* (also known as *dramatis personae*) acting in the narrative are introduced, as well as the narrative *plot*, consisting of a partially-ordered set of *events*. In the present, work, we stay at the fabula level and give special attention to plots whose constituent events happen as a consequence of a predefined repertoire of actions, which we shall call *operations*, deliberately performed by the characters. Since we are interested in the interactive composition of plots, we shall also consider the possibility of user interventions through certain *directives*.

Throughout the presentation, we shall treat plot composition as a *plan generation* process, and hence the terms plot and plan shall be used interchangeably. A plan generator should be able to align the plot events in a *coherent sequence* in view of the characters' objectives, whenever possible coming up with more than one plot, so as to provide *alternative ways* to reach the objectives. But narratives are often more attractive when *unplanned shifts* are allowed to occur. This is arranged for in our proposal through the limited power given to users to interfere with the planner, causing certain discontinuities in the context, particularly concerning changes in the feelings and beliefs of characters. Finally, one should have the possibility to obtain from the planner a more *detailed account* of the events, by having them expanded into smaller grain actions.

It turns out that the need to consider these four notions – coherence, alternatives, transgressive shifts, details – informally prescribed in the above paragraph as desirable for any effective plot composition process, brings to mind four different types of relations between events: *syntagmatic*, *paradigmatic*, *antithetic* and *meronymic*, which in turn are associated with the so-called *four major tropes* of semiotic research [Bu], namely metonymy, metaphor, irony and synecdoque.

Starting from such considerations, this paper proposes a four-sided way to characterize plot composition at the fabula level. Section 2 describes the four relations between events, and points out their correspondence to the four major tropes. Section 3 outlines how we model an intended *genre*, to whose conventions the plots must conform. Section 4 sketches, over a simple example, the main features of our plan-based prototype tool. Concluding remarks are presented in section 5.

2. The four viewpoints

Early work in linguistics [Sa] has characterized two orthogonal dimensions in the structure of language, deployed along the so-called *syntagmatic* and *paradigmatic* axes. The notion is readily applicable to the events of which a narrative plot is composed, as we shall illustrate through an example to be referenced throughout the paper.

Consider four types of events, all of them having one woman and two men as protagonists: *abduction*, *elopement*, *rescue*, and *capture*. As Propp's seminal work [Pr] has demonstrated, many plots mainly consist of a villainy, i.e. of a violent action that breaks the initially stable and peaceful state of affairs, followed ultimately by an action of retaliation, which may or may not lead to a happy outcome.

Propp distinguished seven character roles (*dramatis personae*) according to the events assigned to each one's initiative: hero, villain, victim (princess), dispatcher, donor, helper, false hero. Curiously, in literary texts involving the four events above, this distribution is not unique: we called the violent initial act a "villainy", but the perpetrator of abduction, and more often of elopement, can be the hero of the narrative, and in such cases the woman's original guardian (husband, father) is regarded as the villain.

2.1. Syntagmatic relations

To declare that it is legitimate to continue a plot containing abduction by placing 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 ocurrence of the first leaves the world in a state wherein the occurrence of the second is coherent. Similarly, a plot involving elopement followed by capture looks natural, and therefore we may add that these two events are likewise related.

The syntagmatic relation between events induces a weak form of causality or enablement, that justifies their *sequential ordering* inside the plot.

2.2. Paradigmatic relations

The events of abduction and elopement can be seen as *alternative* ways to accomplish a similar kind of of villainy. Both achieve approximately – though not quite – the same effect: one man takes away a woman from where she is and starts to live in her company at some other place. There are differences, of course, since the woman's behaviour is usually said to be coerced in the case of abduction, but quite voluntary in the case of elopement. In fact, it is usual to assume that a sentence such as "Helen elopes with Paris", implies that Helen had fallen in love with Paris.

To express that abduction and elopement play a similar function, we say that there is a *paradigmatic relation* between the two events. Likewise, this type of relation is perceived to hold between the events of rescue and capture, which are alternative forms of retaliation.

And, again, there is a difference between the woman's assumed attitude, associated as before with her feelings. 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 the present example suggests, the syntagmatic and the paradigmatic axes are really *not* orthogonal in that the two relations cannot be considered independently when composing a plot. Thus, in principle, the two pairs enumerated in the previous section (abduction-rescue and elopement-capture) are the only normal combinations, the former illustrated by the Sanskrit *Ramayana* [Va] and the similarly structured Arthurian romance of *Lancelot* [Chr,FV], and the latter by the Irish *Story of Deirdre* [Mg]. Yet the next section shows that such limitations can, and even should, be waived occasionally.

2.3. Antithetic relations

While normal plots, whose outcome is fully determined, can be composed exclusively on the basis of the two preceding relations, the possibility to introduce unexpected turns is often desirable in order to make the plots more attractive – and this requires a third construct, which we chose to call *antithetic relation*. A context where a woman suffers abduction by a ravisher whom she does not love would seem incompatible with a capture event, since there should be no need to employ force to bring back the victim. So, in this sense, abduction and capture are in antithetic relation.

The mythical *Rape of the Sabines* shows what can happen as a consequence of a drastic reversal of the circumstances. King Romulus is facing a problem at the newly founded city of Rome: the population is entirely male at first. To remedy the lack, he leads his men to break into the dwellings of the Sabines and abduct their women. Sometime afterwards the Sabine warriors march against the Romans, but the women have no wish to be taken back, leaving to their countrymen no option except their capture. The Romans captors had treated them well, they had married them and made them bear children. Titus Livius identified the radical change in the women's feelings, and narrated how the seemingly inevitable confrontation ended instead with the reconciliation of the two parties [Li].

In contrast, modern history provides some distinctly regrettable examples of abduction actually followed by capture, categorized by psychiatrist Nils Bejerot as the *Stockholm syndrome*. One case in point is the abduction by a group of terrorists of Patricia Hearst, daughter of a millionaire, who ended up joining her tormentors in the practice of crimes, and was captured by the police in an apartment at San Francisco [HM].

The occurrence of elopement followed by rescue provides a much stronger case of antithetic relation. Indeed, elopement only makes sense if the victim loves the seducer, whereas, for this very motive, she would resist to any attempt to rescue her, leaving forceful capture as the only viable alternative. And yet the legendary story of *Helen of Troy*, in spite of various discordant interpretations, seems to offer a counter-example. Married to king Menelaus of Sparta, Helen fled to Troy in the company of Paris, out of her free will according to a number of versions (cf. Ovid's *Heroides* [Ov]). But, after their escapade to Troy where they married, her love feelings started to wane while the Trojan war followed its bloody course and she kept recalling the far manlier Menelaus. Homer signals repeatedly this critical change of sentiment in the *Iliad* [Ho]. Not surprisingly, her recovery by Menelaus turned from capture to rescue, as Virgil registers in the *Aeneid*. Paris was dead, and she had been delivered to Paris's brother Deiphobus. When the Greeks came out of the wooden horse and stormed the Trojan palaces, Helen herself made sure that Menelaus should win – and know that she was helping him in atonement for her previous misconduct. The shadow of Deiphobus reports the episode to Aeneas; and what better example of irony could we find than his calling Helen "this peerless wife"? [Vi].

One more example appears in the story of *Tristan and Isolde*, in Béroul's version among others [Ma,Ber]. The knight had eloped with the queen, they were living in harsh conditions in a forest. The dramatic change of their love feelings, which allowed Yseut's rescue by king Mark to be operated by a simple invitation, with no need to fight, had a very curious cause – the timely expiry date of the love potion they had drunk a few years before, when sailing from Ireland to Cornwall [Ber].

Generally speaking, if some *binary opposition* – the "to love or not to love" dilemma, in the present case – is allowed to be manipulated via some agency external to the predefined events, then one can have plots that no longer look conventional. A sort of discontinuity is produced by such radical shifts in the context. Intervening between abduction and capture, or between elopement and rescue, a sudden change of feelings can give rise to these surprising sequences. Also, both in fiction and in reality, things not always proceed according to planned events. Natural phenomena and disasters, the mere passage of time, the intervention of agents empowered to change the rules, supernatural or magic manifestations, etc., cannot be discounted.

Specifically for the tragedy genre, Aristotle [Ar] distinguished between simple and complex plots, characterizing the latter by the occurrence of *recognition* (αναγνορισισ) and *reversal* (περιπετεια). Recognition does not imply that the world itself has changed, but rather the *beliefs* of one or more characters about the actual facts. Because of a change of beliefs, a motive to be added to those enumerated in the previous paragraph, a reversal in the course of actions can take place, usually in a direction totally opposite to what was going on so far. Yet another possible external cause of both recognition and reversal in the tragic scene was the intervention of a god, who was lowered onto the stage using a crane – known, accordingly, as *deus ex machina*.

Aristotle's remarks are clearly relevant to the present discussion of plots in general. Following his lead, we shall admit state changes outside the regular regime of predefined events by allowing the user – literally acting *ex machina* (via the computer...) – to impose variations to the context (both in terms of *facts* and of *beliefs*), and thereby deviate the action from its predicted path.

This extreme device will be necessary to allow the elopement-rescue sequence. We decided, however, not to make it indispensable for abduction-capture, in order to have a chance to present a good example of *erroneous* beliefs, contradicting the actual facts. Criminal records everywhere are full of simulated abduction pacts, with the purpose of drawing a ransom from a deluded family. Conversely, a man can unnecessarily decide that capture is the only way to bring back a woman, if he mistankenly believes her to be in love with the ravisher.

2.4. Meronymic relations

Meronymy is a word of Greek origin, used in linguistics to refer to the decomposition of a whole into its constituent parts. Forming an adjective from this noun, we shall call *meronymic relations* those that hold between an event and a lower-level set of events, with whose help it is possible to provide a more detailed account of the action on hand.

Thus, we could describe the abduction of a woman called Sita by a man called Ravana as: "Ravana rides from Lanka to forest. Ravana seizes Sita. Ravana carries Sita to Lanka." And her rescue by Rama could take the form: "Rama rides from palace to Lanka. Rama defeats Ravana. Rama entreats Sita. Rama carries Sita to palace." But notice that such decompositions are not fixed, since the lower-level operations are selected as required by the current state. For instance, with respect to the rescue event, the hero may already be present at the ravisher's dwelling, or perhaps the victim is not held in captivity, respectively obviating the need for the voyage or for fighting the enemy.

Detailing is most useful to pass from a somewhat abstract view of the plot to one, at a more concrete physical level, that is amenable (possibly after further decomposition stages) to the production of a computer graphics animation [CPFF]. Mixed plots, combining events of different levels, do also make sense, satisfying the option to represent some events more compactly while showing the others in detail.

2.5. The four major tropes

It turns out that the four relations between events, described in sections 2.1 through 2.4, correspond to what Kenneth Burke [Bu] considers the four major tropes: *metonym*, *metaphor*, *irony*, *synecdoche*. In turn, it has been suggested that those rhetoric figures of speech provide models for remarkably comprehensive analyses in different areas [Cha,Wh]. They all correspond to *relations between pairs of words*, thanks to which, given two related words w1 and w2, a person can meaningfully use w1 to refer to w2.

These tropes are not defined in a uniform way by linguists, there being much disagreement, especially on the distinction between metonym and synecdoche. An excellent discussion can be found in [Cha], where many practical applications of Burke's four tropes theory are surveyed.

Metaphor [LJ, Or] and synecdoque [Cha] have to do with hierarchical structures such as those represented in ontologies [BCT]. If one concept C_1 can be metaphorically used to denote another concept C_2, the two concepts are said to be similar or analogous, and are placed under a more general concept $\hat{C}$ that subsumes both of them. C_1 and C_2 would be represented in the network with **is-a** links connecting them to $\hat{C}$. Also, one could add an **is-like** link from C_1 to C_2 [BBFC]. Clearly, metaphor is a verbal displacement along one of Saussure's axes [Sa], being thus suggestive of the *paradigmatic relation* between events.

In synecdoque, concept C_1 is used to denote concept C_2, if C_1 is a part of C_2 (which calls for another link, C_1 **part-of** C_2); the converse substitution, from whole to part, is also contemplated. The corresponding association between events is obviously what was called *meronymic relation* in the present paper.

According to [Cha], metonyms are based on various indexical relationships between concepts, notably the substitution of effect for cause. It conveys an idea of contiguity, in agreement with the s*yntagmatic relation* reviewed here, which justifies placing events in sequence.

Irony is the most intriguing of the four tropes. In [Cha], the notion is explored as follows: "Where it means the *opposite* of what it says (as it usually does) it is based on binary opposition. Irony may thus reflect the opposite of the thoughts or feelings of the speaker or writer (as when you say 'I love it' when you hate it) or the opposite of the truth about external reality (as in 'There's a crowd here' when it's deserted). It can also be seen as being based on substitution by *dissimilarity* or *disjunction*. Whilst typically an ironic statement signifies the opposite of its literal signification, such variations as understatement and overstatement can also be regarded as ironic. At some point, exaggeration may slide into irony." Disclosing paradoxes and hidden agendas in literary texts, sharp contrasts between the declared intentions and the real ones, is another source of irony, constituting a trend in critical studies known as *deconstruction* [Cu].

Not only mental attitudes, feelings and statements can be ironic – actions can also be ironic, but always in an unplanned, non-deliberate fashion [Bo]: "Irony is not limited to verbal acts, but is also a characteristic of situations that are often referred to as *dramatic irony* . . . I cannot say that I will do three ironic acts today because when I say that some act is ironic, I am asserting that it is somehow unexpected or inconsistent from my point of view, and I cannot claim this with respect to my own intentions."

Thus irony induces an *antithetic relation* between events that are, in principle, incompatible with each other, given their dependence on contexts characterized by radically opposite properties. 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 move from one extreme to the other within the ascribed value range (e.g. from helplessly weak to heroically strong).

3. A plan-based modelling approach

To model a chosen genre, to which the plots to be composed should belong, we must specify at least (to be the object of section 3.1):

a. what can exist at some state of the underlying mini-world,
b. how states can be changed, and
c. the factors driving the characters to act.

In our model, we equate the notion of event with the state change resulting from the execution of a predefined operation. Being defined in terms of their pre-conditions and post-conditions, operations can be readily chained together by a *plan-generating algorithm* [CPFF,BM] in order to achieve a given goal of some character. As a consequence, it becomes natural to equate plots (sequences of events) with plans (sequences of operations able to bring about the events). Also, to confer a degree of autonomy to the characters performing the operations, it is convenient to make their goals emerge from appropriately motivating situations.

Viewing plots as plans suggests an obvious plot composition strategy, having a plan-generator as its main engine. This and the fact that our conceptual model is expressed in Prolog makes the genre specification executable. In sections 3.2, 3.3 and 3.5, we will argue that, duly complemented by auxiliary routines, the planning strategy can effectively deal with narrative plots in view of three out of the four event relations. To accomodate antithetic relations, however, it will be necessary to leave room to the *unplanned*, as proposed in section 3.4, leading to plots that may, to a limited extent, transgress the conventions of the adpoted genre.

3.1. Conceptual schemas

We start with a conceptual design method involving three schemas: static, dynamic and behavioural, which has been developed for modelling literary genres encompassing narratives with a high degree of regularity, such as fairy tales, and application domains of business information systems, such as banking, which are obviously constrained by providing a basically inflexible set of operations and, generally, by following strict and explicitly formulated rules [FCBB]. Appendix A reproduces the complete specification used in our example.

The *static schema* specifies, in terms of the *Entity-Relationship* model [BCN], what are the entity and relationship classes and their attributes. In our simple example, `character` and `place` are entities. The attributes of characters are `name`, which serves as identifier, and `gender`. Places have only one identifying attribute, `pname`. Characters are pairwise related by relationships `loves`, `held_by` and `consents_with`. The last two can only hold between a female and a male character; thus `held_by(Sita,Ravana)` is a *fact* meaning that Sita is forcefully constrained by Ravana, whereas `consents_with(Sita,Ravana)` would indicate that Sita has voluntarily accepted Ravana's proposals. Two relationships associate characters with places: `home` and `current_place`. The state of the world at a given instant consists of all facts about the existing entity instances and their properties holding at that instant.

The *dynamic schema* defines a fixed repertoire of operations for consistently performing state changes. The *STRIPS* [FN] model is used. Each operation is defined in terms of pre-conditions, which consist of conjunctions of positive and/or negative literals, and any number of post-conditions, consisting of facts to be asserted or retracted as the effect of executing the operation. Instances of facts such as `home` and `gender`, are fixed, not being affected by any operation. Of special interest are what we call *user-controlled* facts which, although also immune to operations, can be manipulated through arbitrary *directives* (cf. section 3.4). In our example, `loves` is user-controlled.

Again for the present example, we have provided operations at two levels. The four main events are performed by level-1 operations: `abduct`, `elope`, `rescue` and `capture`. Operations at level-2 are actions of smaller granularity, in terms of which the level-1 operations can be detailed: `ride`, `entreat`, `seize`, `defeat`, and `carry`.

Our still provisional version of the *behavioural schema* consists of goal-inference (also called situation-objective) rules, belief rules, and emotional condition rules.

For the example, three goal-inference rules are supplied. The first one refers to the ravisher. In words, in a situation where the princess is not in her home and the hero is not in her company – and hence she is unprotected – the ravisher will want to do whatever is adequate to bring her to his home. The other goal-inference rules refer to the hero, in two different situations having in common the fact that the ravisher has the woman in his home: either the hero believes that she does not love the other man, or he believes that she does. In both situations, he will want to bring her back, freely in the first case and constrained in the second.

Informally speaking, beliefs correspond to the partial view, not necessarily correct, that a character currently forms about the factual context (for a formal characterization, cf. the

BDI model [CL,RG]). The belief rules that we formulated for our example look rational, but notice that they are treated as defaults, which can be overruled as will be described in section 3.4. A man (the hero or the ravisher) believes that the woman does *not* love his rival if the latter has her confined, but if she has ever been observed in his company and in no occasion (state) was physically constrained, the conclusion will be that she is consenting (an attitude that would seem too subjective to be ascertained directly in a realistic context).

The emotional condition rules refer to the three characters. A man (or woman) is happy if currently in the company of his (or her) beloved, and bored otherwise. A special condition applies to the woman: she will be *absolutely* happy if, in addition to the first motive for contentment, she has never been constrained by any of the two adversaries.

3.2. Coherent sequences

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.

For plot composition, it is convenient to proceed in a step-wise fashion, starting from a given initial state. At each state, the goal-inference rules are used to induce opportunistic short term goals from which successive plot sequences will originate.

In an interactive environment, at any step, the user, henceforward called the *Author*, should be allowed to intervene, reducing thereby the characters' autonomy, but still relying on the plan-generator to enforce consistency within the genre. To this purpose, the Author may indicate 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, possibly interspersed with others, is formed. The Author may optionally also indicate the desired goal, which would otherwise be assumed to coincide with the effects of the last operation in the list.

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. For instance, consider plot `P` below:

```
P = start => ride(Ravana, Lanka, forest) => entreat(Ravana, Sita) =>
seize(Ravana, Sita) => carry(Ravana,Sita,Lanka)
```

which can be re-ordered, to meet the Author's preferences, to produce:

```
Ps = start => ride(Ravana, Lanka, forest) => entreat(Ravana, Sita) =>
carry(Ravana,Sita,Lanka) => seize(Ravana, Sita)
```

Curiously, both the original plan `P` and the reordered plan `Ps` suggest stories that may well happen in reality or fiction. In `P`, a voluntary elopement is disguised as an abduction, whereas `Ps` can be interpreted as an overt elopement after which the seducer decides to restrict the woman's freedom.

Also, a plot can be extended with more operations if the Author supplies an additional goal in an attempt to provide a continuation.

3.3. Alternative choices

Moving along the *paradigmatic* axis gives ampler opportunity to obtain different plots than simply changing the sequence of events within the partial order requirements.

Alternatives may result, first of all, from starting from a different initial state, so that different goal-inference rules may be triggered. Notice also that more than one such rule may be ready for activation. In any case, the standard plan-generator ability to *backtrack* constitutes an expedient mechanism to offer alternative plots.

Resorting to violence, as in abduction or capture, can be certainly regarded as excessive and unnecessary when the patient of the action loves the agent, even though our specification does not invalidate their occurrence. Accordingly, if the goal-inference rules are in control and the context is not tampered with (but see section 3.4.), they will not figure in any generated plot. And yet the Author can have them as valid alternatives, simply by using the option to directly indicate a goal to the plan-generator. Such goal can be relatively non-specific, such as `current_place(Sita,palace)`, or else more restrictive, such as `(current_place(Sita,palace), held_by(Sita,Rama))` – in which case only the forceful `capture` event will result.

At the adaptation phase, the ability to replace one or more operations is a form to produce alternatives. One must bear in mind that a replacement may require another, if the Author is concerned with preserving consistency; so, replacing `abduct` by `elope` would normally imply the replacement of `rescue` by `capture`.

A particularly convenient way to deal with entire plots, rather than with individual operations, is to take advantage of the similarity or analogy among situations, inherent in the notion of paradigms. Previously existing plots, no matter if composed manually or automatically, can be converted into *plot patterns* to be kept in a Library of Typical Plots [FC2]. Plot patterns can then be *reused* to originate new plots, essentially by instantiating their variables in view of a new situation.

3.4. Shifts along the way

Until this point we restricted ourselves to planned and hence well-behaved plots. It is time now to introduce a measure of transgression, disrupting the context so as to allow the composition of plots containing events in *antithetic* relation.

The Author, as *deus ex machina*, can interfere with the plan generation discipline by issuing two kinds of *directives*, which can be applied both during composition and adaptation. One directive is `make_believe`, arbitrarily assigning a belief `B` to a character `C`, which overrules any previous belief on the same facts, either specified through the belief rules of the behavioural schema or stated by a previous application of the `make_believe` directive itself. If Sita was violently abducted by Ravana, Rama will believe (as a consequence of a belief rule) that she does not love the villain, and therefore she will gladly consent to be rescued. However, the Author is allowed to induce Rama to falsely believe the contrary, which activates a goal-inference rule leading to a forceful `capture` event.

Another directive is `vary`, which manipulates user-controlled facts, instead of mere beliefs. In our example, the only facts declared to be user-controlled are the instances of the `loves` relationship, whose Boolean value will be inverted if the directive is applied. Sita can be eloped if she currently loves Ravana, and then be willingly rescued by Rama if between these two events the Author issues the directive so as to change her feelings. But `vary` does not have to be explicitly called for. A helpful feature provided in the course of plan-generation detects failures involving user-controlled facts, in which case the Author is asked whether or not the context should be accordingly tampered with.

In other example mini-worlds, one might have different kinds of user-controlled properties, e.g. with numerical values inside a range, such as degree of strength, that the `vary` directive could change in some radical proportion.

We have just began to investigate another line, in an attempt to offer clues to an Author intent on finding ways to, at a later stage, replace the external *deus ex machina* directives by some internal narrative device with a flavour of irony, almost crossing the borderline of plausibility. Folktales, myths, and popular culture have pooled together along the years rich repertoires of *motifs* [AT], often containing ingenious solutions to the dilemmas arising from antithetic situations.

Authors have always felt free to borrow from all kinds of sources, and one can easily discover occurrences of certain motifs in the literature of different countries, with inevitable modifications required by cultural differences. For our example, we found three convenient motifs:

a. life token – an object whose aspect changes if the owner is in distress,
b. love potion – instills or stimulates romantic/ erotic feelings,
c. ordeal – to vindicate a discredited or accused person.

where (a) (indexed as E761 in [AT]) allows to do without the unrealistic assumption that characters are omniscient, e.g. explaining how Rama learned that Sita suffered abduction in the forest, (b) provides an excuse for sudden variations in amorous attachments, and (c) serves to restore the man's belief in his beloved's faithfulness. Curiously, both (b) and (c) occur in the *Tristan* romance, wherein the ordeal takes the especially ironic form of an ambiguous oath [Ber], while in the *Ramayana* Sita has to walk through the fire [Va]. In our example, we treat these motifs as black boxes, merely associating to their names a <*situation*, *goal*> specification. Thus, if the Author wants to insert motifs (simply through the mention of their names) at the positions in a generated plot where the respective situation holds, this can be asked for at the adaptation phase.

Such insertions are therefore to be regarded as provisional annotations only, which the Author should later have to unravel by mapping the events in the motifs into analogous events congenial to the genre adopted in the plot. The mappings should preserve the <situation, goal> of the motif and might require the definition of additional operations, such as communicative acts for instance. The persistence of motifs is a remarkable phenomenon, with relatively modern versions: microchip implants for (a), aphrodisiac drugs like the LSD hallucinogen for (b), and lie detectors and truth serums for (c), all of so dubious or controversial value as their primitive counterparts, but equally acceptable to the general public.

Figure 1 provides a visual representation of the three relations thus far discussed.

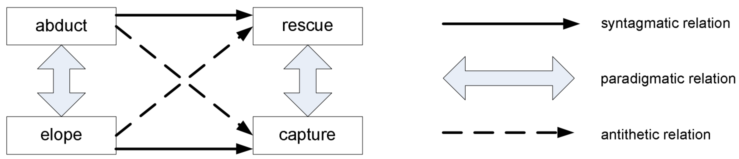

Fig 1: Syntagmatic, paradigmatic, and antithetic relations.

3.5. Down to details

Between level-1 and level-2 operations there may exist *meronymic* relations that will now be exploited (figure 2).

(a) *(b)*

Fig. 2: Meronymic relations: (a) the forceful actions and (b) the gentle actions.

The intuitive notions behind figures 1, 2(a) and 2(b) can be nearly expressed under the form of a context-sensitive grammar:

PLOT ::= VILLAINY • RETALIATION
VILLAINY ::= ABDUCT | ELOPE
RETALIATION ::= RESCUE | CAPTURE
ABDUCT • RESCUE ::= abduct, rescue
ELOPE • CAPTURE ::= elope, capture
ABDUCT • CAPTURE ::= $*$(abduct, capture)^{belief}
ELOPE • RESCUE ::= $*$(elope, rescue)^{fact}
ABDUCT • RESCUE ::= ABDUCT2 • RESCUE2
ELOPE • CAPTURE ::= ELOPE2 • CAPTURE2
ABDUCT • CAPTURE ::= $*$(ABDUCT2 • CAPTURE2)^{belief}
ELOPE • RESCUE ::= $*$(ELOPE2 • RESCUE2)^{fact}
ABDUCT2 ::= ride, seize, carry
RESCUE2 ::= ride, defeat, entreat, carry
ELOPE2 ::= ride, entreat, carry
CAPTURE2 ::= ride, defeat, seize, carry

Creating plots in hierarchic fashion is a most common practice, starting with a broad view of the events, which in the case of our example corresponds to the level-1 operators. At later stages, one would gradually decompose each event into finer grain actions, possibly along more than just two levels, to the point of coordinated physical movements, as required for displaying animated scenes [CPFF].

When composing a plot, the plan-generator is free to mix operations of the two levels, a reasonable default option considering that the Author may wish to treat some events more succinctly than others. But the Author may, on the contrary, settle for a uniform style by indicating that only one of the two levels will be used. This choice can be altered at any time, both during composition and adaptation.

Once a plot is composed, it can be adapted either by detailing or summarizing its constituent operations.

Detailing each level-1 operation `Op` in a plot into level-2 operations is treated as yet another plan generation task, taking as *situation* the instantiated pre-conditions of `Op`, and as *goal* the effects of `Op`, and using exclusively the operations in the level-2 repertoire. More than one decomposition may be possible, depending on the initial state and on the changes effected by the preceding operations.

The inverse of detailing, summarizing, is also useful. We are currently restricted to a rather limited version, which only works if the detailed plan is divisible into subsequences that can be exactly subsumed by level-1 operations. This means that the process fails if other extraneous operations intervene. In other words, `summarize(P1,P2)` succeeds if and only if `detail(P2,P1)` also does.

Figures 2(a) and 2(b) are suggestive in that they illustrate a curious symmetry in how they map the example level-1 operations into level-2 operations. The decompositions in the two figures are the same, except for the substitution of `entreat` for `seize`. This is not surprising, since a similar decomposition comes as a consequence of the paradigmatic relation between the two villainy and the two retaliation events. Notice too that, in both figures, the event corresponding to villainy only differs from the retaliation event by the presence or not of `defeat` – reflecting our observation, after surveying a number of traditional narratives, that the villain almost always resorts to some trick, avoiding a confrontation that often (though not necessarily) occurs as part of retaliation.

The decompositions suggested by the two figures are typical but not unique, since the correspondence induced by the meronymic relations is not rigidly determined, i.e. it is, so to speak, context-sensitive, depending on the current state. For instance, `abduct` can be expressed by `seize` followed by `carry` if both the victim and the ravisher are currently at the same place, but will need a preliminary `ride` if the former is in the forest and the latter still in his home.

All this suggests that it may be difficult to interpret what is happening by just observing a sequence of level-2 operations without examining the context. In this regard, the ability to summarize, identifying what level-1 operation is taking place at some point, constitutes a not so trivial form of *plan-recognition* [Ka]. Plan generation is surely more directly relevant to the composition and adaptation of plots than the recognition of plans and objectives. But the latter task becomes an asset in an interactive plan-supported game-playing environment, since each player might employ it as an aid to discover what the opponents are trying to do.

4. A prototype implementation

A very simple prototype, **PlotBoard**, was designed to experiment with the notions discussed here. Dealing with storyboarding [THA] – exclusively at the fabula level – it serves to compose plots interactively with the help of an extended version of Warren's original *Warplan* [Wa] algorithm. Implemented in SWI-Prolog (<http://www.swi-prolog.org/>), it interfaces with Java to display the events in image format.

4.1. Some features of the plan-generator

The plan generator follows a backward chaining strategy. For a fact F (or not F) that is part of a given goal, it checks whether it is already true (or false) at the current state. If it is not, it looks for an operation Op declared to add (or delete) the fact as part of its effects. Having found such operation, it then checks whether the pre-condition Pr of Op currently holds – if not, it proceeds recursively trying to satisfy Pr. Moreover, the plan generator must consider the so-called frame problem [Lo], by establishing (in second-order logic notation) that the facts holding just before Op is executed stay valid unless explicitly declared to be altered as part of the effects (additions or deletions) of Op.

Like goals, pre-conditions are denoted by conjunctions of literals and arbitrary logical expressions. We distinguish, and treat differently, three cases for the occurrence of positive or negative facts in pre-conditions:

a. facts which, in case of failure, should be treated as goals to be achieved recursively by the plan generator;
b. facts to be tested immediately before the execution of the operation, but which will not be treated as goals in case of failure: if they fail the operation simply cannot be applied;
c. facts that are not declared as added or deleted by any of the predefined operations.

Recall that the general format of a pre-condition clause is precond(Op, Pr) :- B. In cases (a) and (b), a fact F (or not F) must figure in Pr, with the distinction that the barred notation /F (or /(not F)) will be used in case (b). Case (c) is handled in a particularly efficient way. Since it refers to facts that are invariant with respect to the operations, such facts are included in the body B of the clause, being simply tested against the current state when the clause is selected.

Take, for instance (cf. appendix A.2.2), the precond clause of operation seize(M,W), where M is the agent and W the patient of the action. Clearly the two characters should be together at the same place, and, accordingly, the Pr argument shows two terms containing the same variable P to express this requirement, but the term corresponding to W is barred: /current_place(W,P), which does not happen in M's case. The difference has an intuitive justification: the prospective agent has to go to the place where the patient is, but the latter will just happen to be there for some other reason.

The proper treatment of (a) and (b) is somewhat tricky. Suppose the pre-condition Pr of operation Op is tested at a state T1. If it fails, the terms belonging to case (a) will cause a recursive call whereby one or more additional operations will be inserted so as to move from T1 to a state T2 where Op itself can be included. It is only at T2, not at T1, that the barred terms in case (b) ought to be tested, and so the test must be *delayed* until the return from the recursive call, when the plan sequence able to reach T2 will be fully instantiated.

Operations can admit more than one precond cause, so as to cope with different circumstances. This happens with the carry(M,W,P2) operation, whereby W will either freely consent to be transported to P2 by M, or will have to be forcefully held by him.

With respect to the added and deleted clauses declaring effects of operations, the plan generator also employs a barred notation, to distinguish between two cases: (a) primary effects, (b) secondary unessential effects. In case (a), if any fact F to be added by Op already holds, or already does not hold if it should be deleted, then Op is considered *non-productive* and fails to be included in the plan. In contrast, in case (b), such lack of effect would be admitted and therefore would not cause failure.

As an example, consider the clause of operation capture(M1,W) that declares as deleted the fact held_by(W,M2), as a result of M1's action to take away W from M2. Notice that the fact may or may not hold prior to capture; it will hold if W was abducted by M2, but will not hold if an elopement occurred instead – and that is why the barred notation is used for this particular deleted clause. On the contrary, the fact current_place(W,P2), where P2 is the home of M2, must necessarily be deleted by an effective execution of the operation, and so does not figure as barred.

The execution of plans is done through assert or retract commands on the facts to be, respectively, added or deleted. The plan's pre- and post-conditions are checked during the process, there being no effect in case of failure. A log(L) literal, initiated with L=start, is extended with each successful plan execution and can be usefully retrieved for a variety of purposes. On the basis of the log and of the initial state, which is saved when a session begins, it is possible to query about facts at any intermediate state. It is also possible to restore a previous state S (initial or intermediate) that has been saved, which enables simulation runs.

User interventions, necessary to achieve unplanned situations, are permitted in a limited scale through *directives* that can be either intermixed with the operations in a plan or called separately. Two of these are used in our example, one for changing loves facts, immune to the predefined operations, and the characters' beliefs, which may or may not reproduce correctly the actual facts.

To finish this partial review of the plan features, we remark that the planning algorithm plans(G,P) is called in more than one way. More frequently G is given, as the goal, and P is a variable to which a generated plan will be assigned as output. However an inverse usage has been provided, wherein P is given and G is a variable; in this case, the algorithm will check whether P is valid and, if so, assign its net effects (a conjunction of F and not F terms) to G.

4.2. The PlotBoard tool

We shall briefly describe how **PlotBoard** works, after the controlling user, here called the Author, enters the `plot` command. The diagram of figure 3 will serve to guide the description. Appendix B shows the menus attached to the nodes.

The main option is to compose the plot from scratch, in a step-wise fashion. Ideally, the Author should leave a measure of autonomy to the characters (branching into the `planner` node of the diagram). At each step (cf. the `plan step` node), one subsequence of the plot willl be generated. As if emerging from the mind of a character `C`, a goal `G` is instantiated by some goal-inference rule `(C,S,G)`, if the situation component `S` of the rule currently holds. More than one rule may be simultaneously ready for activation, and the planning algorithm may find more than one alternative subsequence able to achieve the corresponding goals (whenever the planning algorithm backtracks), as represented by the self-loop around the `plan step` node. As a subsequence is being presented, the Author is prompted to either issue an `ok` reply, or, possibly after inspecting what effects it would have, to call for an alternative. An `ok` reply is followed by a return to the `planner` node.

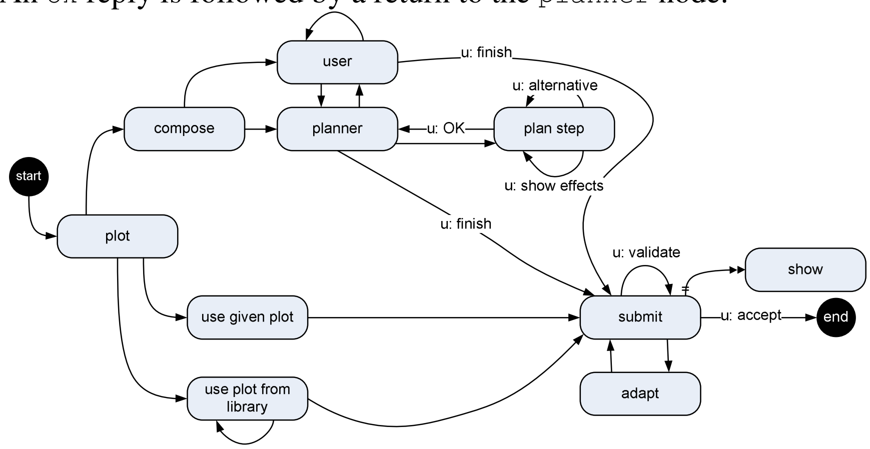

Fig. 3: flow of control of the **PlotBoard** prototype

The subsequence thus selected is then executed in a simulated mode (the initial state is saved, to be restored later), and the Author is asked whether the `plan step` iterations should continue, producing further subsequences to be appended to the plot so far obtained, or whether the composition process is finished for the time being (passing to the `submit` node), though still subject to possible adjustments.

If the Author is more inclined towards a closer arbitrary control than to the character autonomy policy described above, several options are available to determine the goals that the planning algorithm should try to achieve (cf. the the first 3 items of the menu for the `user` node below). Again the self-loops around the `user` node represent the possibility of alternative plot subsequences being offered to the Author's choice. These options permit step-wise composition, which can be entirely commanded from the `user` node, but can also alternate with the activation of goal-inference rules, by intercalating transfers of control to the `planner` node (item 7 of the `user` menu).

An additional purpose of the `user` node is to prepare and support the composition process, by allowing to pose queries about the database state at each step, to change the operation level, and to issue directives to alter the characters' beliefs and the value of user-controlled properties (items 4, 5, 6).

```
1: goal
2: operation
3: list of operations
4: query
5: operation level
6: directive
7: planner
8: finish
```

Whatever composition policy is preferred – autonomous, arbitrary, or mixed – the finished plot is passed to the `submit` node. At this point, the Author can either accept the plot, which terminates the process, or can go through one or more rounds of adaptatation, using the options offered at the `adapt` node below.

```
1: detail
2: summarize
3: change sequence
4: add operation
5: delete operation
6: replace operation
7: extend
8: queries or directives
9: insert motif
10: back to the submit options
11: stop
```

To help decide whether or not to accept the current plot or perform other adaptations, the `submit` menu gives the Author the option to validate the plot (again through the planning algorithm). This may be in order if the Author directly introduces specif changes (items 4, 5, 6 of the `adapt` menu), noting that in all other forms of adaptation the planning algorithm intervenes to prevent integrity violations. (But note in passing that validation warnings will be issued for plots whose composition required contextual changes forced by the Author's directives).

Another feature available at the `submit` node deserves special attention, since what it produces, together with the menu-based dialogues, constitutes the intended output of the **PlotBoard** tool. If selected, via the `show` option, it provides a visual display that can be repeated for the successive versions. For each operation in the current plot, the event it denotes appears as a rough drawing, side by side with a short template-driven natural language sentence.

We refer again to the diagram in fig. 3, to consider two ways to obtain a plot without requiring step-wise composition from scratch. In both cases, a full plot is used to start with, and in both cases the process converges afterwards to the `submit` node.

Branching into the `use given plot`, the Author can either enter the intended plot or retrieve a previously composed one. The planning algorithm is automatically called to inspect the plot, operation by operation, to check whether each of them can be applied in view of the pre- and post-conditions interplay. If an operation is found that can only be applied if a user-controlled property is tampered with, the possibility of changing the value of the property is indicated to the Author, who may or may not permit the execution of the necessary `vary` directive. If the Author denies permission, or if the offending property is not user-controlled, the plot is rejected.

In case the node `take plot from library` is chosen, the Library of Typical Plots (LTP) will be searched for items `(S,G,P)`, such that situation `S` currently holds, thereby propagating the instantiation of the parameter variables figuring in `S` to goal `G` and plot `P`. If more than one such item is found, the Author will have once more an opportunity to select the preferred `P` among the alternatives presented.

4.3. An example run

At the initial state, both Rama and Ravana are in their homes, respectively the royal palace and the city of Lanka, whereas Sita is alone in the forest. The two men love Sita, who only loves Rama. Starting to compose the plot, the Author invokes the planner in two stages, always selecting the detailed (level 2) alternatives. At this point the plot is, in natural language format:

Ravana rides from Lanka to forest. Ravana seizes Sita. Ravana carries Sita to Lanka. Rama rides from palace to Lanka. Rama defeats Ravana. Rama entreats Sita. Rama carries Sita to palace.

Wishing to try different versions, the Author looks at the `adapt` menu, shown in the previous section. The first change selected is the deletion (option 5) of the two events that close the narrative. The next step is to issue directives (8) to change the emotional attachments and certain of the characters' beliefs: now Sita loves Ravana and Rama believes this fact. This justifies adding (option 4) operation `entreat(Ravana,Sita)` as second event (after Ravana approaches the princess in the forest):

Ravana rides from Lanka to forest. Ravana entreats Sita. Ravana seizes Sita. Ravana carries Sita to Lanka. Rama rides from palace to Lanka. Rama defeats Ravana.

The plot now suggests the fake abduction pattern, wherein the villain seizes his pretended victim only to simulate a violent action. The Author wonders then if the same events could be arranged in some different sequence (3), and a dialogue ensues:

```
[f1:entreat(Ravana, Sita), f2:seize(Ravana, Sita)]
choose one of the fi tags: f1
[f1:seize(Ravana, Sita), f2:carry(Ravana, Sita, Lanka)]
choose one of the fi tags: f2
[f1:seize(Ravana, Sita), f2:ride(Rama, palace, Lanka)]
choose one of the fi tags: f1
```

Ravana rides from Lanka to forest. Ravana entreats Sita. Ravana carries Sita to Lanka. Ravana seizes Sita. Rama rides from palace to Lanka. Rama defeats Ravana.

The reordered plot might be interpreted in a radically different way. It now sounds as an overt elopement after which the seducer decides to restrict the woman's freedom. How would the narrative proceed from this point? Selecting the extend option (7) of the `adapt` menu, the Author proposes the goal: `current_place(Sita,palace)`, and the planner responds (figure 4) with: Rama captures Sita.

Is this a satisfactory way to end the narrative? The Author selects option 8 of the adapt menu and poses two queries, to learn what the characters think and how they feel:

```
1: queries
2: directives
options: 1
query: beliefs

   Rama believes that Sita loves Ravana
   Ravana does not believe that Sita loves Rama

more queries?(yes/no): yes
query: emotional_condition

   Sita is bored
   Rama is happy
   Ravana is bored
```

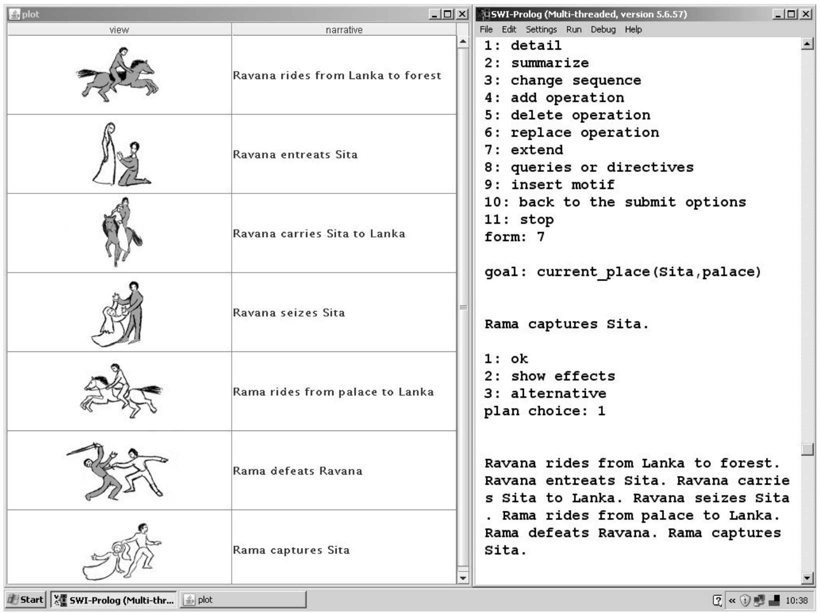

Fig. 4: a **PlotBoard** screen

Sympathizing with the princess, the Author decides to revert the situation. Perhaps her love for the hero could revive (as happened to Helen at the end of the Trojan war), and the last event is changed according to this expectation, replacing (6) the `capture(Rama,Sita)` operation by `rescue(Rama,Sita)`. How does the plot look now? Back at the `submit` menu, the Author asks to visualize the successive scenes, and accepts this result, a happy end for Sita as well as for the Author, who receives a grateful acknowledgement from the **PlotBoard** tool (figure 5).

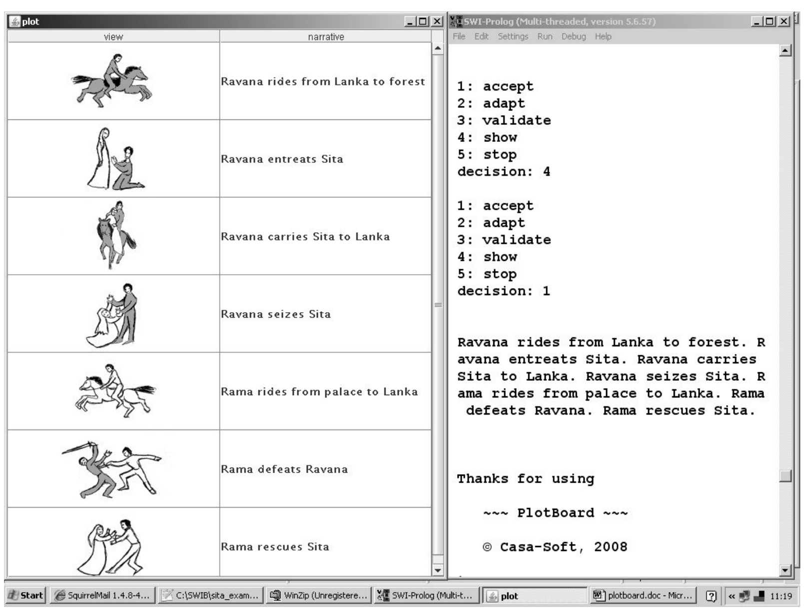

Fig. 5: the accepted plot

But much remains to be done. The *deus ex machina* directives, should be replaced eventually by something internal to the narrative. Also, how to explain that Rama knew without being told that Sita had become Ravana's prisoner? To gather suggestions, to be possibly (re)used after due modifications appropriate to the genre, the Author might have inspected (figure 6) the applicable motifs, before issuing the final accept response, in which case the *life token*, the *love potion* (twice) and the *ordeal* motifs would be indicated at one or more positions in the plot wherein the respective motivating situation holds.

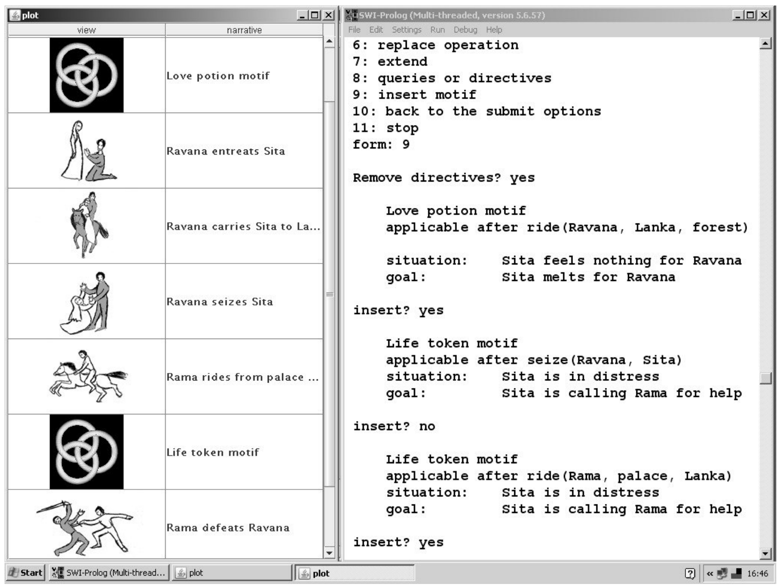

Fig. 6: insertion of motifs

5. Concluding remarks

Although the process of plot composition could surely be enriched far beyond what was presented here, the suggested four-sided approach seems to provide a sound initial basis. 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 major tropes. Other concepts may be adduced to extend the model. 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* [FT,CBBF] immediately comes to mind, as the technique or artisanship of conciliating the pending conflicts, which often requires a great deal of creativity.

The facilities associated with the four relations are adequate for other tasks, besides storyboarding, under suitable user interfaces. In interactive storytelling systems designed for entertainment, as well as in games, they might prove instrumental to support the production of coherent and diversified stories with more dramatic power, increasing their ability to cause surprise. Some alternatives for generating and adapting plots can even be adjusted to users' satisfaction models, so that there would be no longer a need to explicitly intervene to obtain varied and interesting outcomes.

Finally, let us recall that we have addressed the *fabula* 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 – the *story* level, where the concern is *how* to tell the events – is to find an adequate justification for the contextual disruptions, in our case introduced *ex machina* via user interaction. Such elaborations may be immediately plausible, like Helen's gradual disillusion with the martial virtues of Paris [Ho], or may appeal to knowledge made popular by the media, like the Stockholm syndrome to explain the victim's conversion to the terrorists' ideology [HM], or even be fanciful, like the expiration of the love potion in the *Tristan* romance [Ber]. Moreover a plurality of narrative objectives must be considered at this level. The description of the **Minstrel** system [Tu], for instance, postulates thematic, consistency, drama and presentation goals. At the third and last level – the *text* level – the narrative is represented in some medium, not necessarily printed pages. We have done some initial work [FC1] on the generation of natural language texts from plots of log-registered transactions, noting that the result should still be enhanced through the application of methods pertaining to computational linguistics [Mk]. As shown in [CPFF,CCFPF], computer graphic animation provides a particularly attractive way for displaying narrative plots, both in the realm of literary genres and in different domains of business information systems.

References

[Ar] Aristotle. "Poetics". In *Classical Literary Criticism*. P. Murray et al (trans.). London: Penguin, 2000.

[AT] A. Aarne, S. Thompson. *The Types of the Folktale: A Classification and Bibliography*. Helsinki: Suomalainen Tiedeakatemia, 1987.

[BBFC] K. K. Breitman, S. D. J. Barbosa, M. A. Casanova, A. L. Furtado. "Using analogy to promote conceptual modeling reuse". In Proc. of *Workshop on Leveraging Applications of Formal Methods, Verification and Validation* (ISOLA 2007), Poitiers, 2007.

[BCN] C. Batini, S. Ceri, S. Navathe. *Conceptual Design – an Entity-Relationship Approach*. Benjamin Cummings, 1992.

[BCT] K. K. Breitman, M. A. Casanova, W. Truszkowski. S*emantic Web: Concepts, Technologies and Applications*. London: Sringer, 2007.

[Ber] Béroul. *The Romance of Tristan*. A. S. Fedrick (trans.). London: Penguin, 1970.

[BM] L. Barros, S. Musse. "Planning algorithms for interactive storytelling". In *ACM Computers in Entertainment*, ACM, New York, NY, USA, v. 5, n. 1, 2007.

[Bo] W. Booth. *A rhetoric of Tropes*. Chicago: University of Chicago Press, 1974.

[Bu] K. Burke. *A Grammar of Motives*. Los Angeles: University of California Press, 1969.

[CBBF] M. A. Casanova, S. J. Barbosa, K. K. Breitman, A. L. Furtado. "Generalization and blending in the generation of entity-relationship schemas by analogy". In Poc. of the *Tenth International Conference on Enterprise Information Systems* (ICEIS). Barcelona, 2008.

[CCFPF] M. Camanho, A. E. M. Ciarlini, A. L. Furtado, C. T. Pozzer, B. Feijó. "Conciliating coherence and high responsiveness in interactive storytelling". In Proc. of the *3rd. ACM International Conference on Digital Interactive Media in Entertainment and Arts (DIMEA 2008)*, Athens, Greece, 2008.

[CCM] M. Cavazza, F. Charles, S. Mead. "Character-based interactive storytelling". *IEEE Intelligent Systems,* special issue on AI in Interactive Entertainment, 17(4):17-24, 2002.

[Cha] D. Chandler. *Semiotics: The Basics*. London: Routledge, 2007.

[Chr] Chrétien de Troyes. *Le Chevalier de la Charrete*. M. Rocques (ed.). Paris: Honoré Champion, 1983.

[CL] P. R. Cohen, H. J. Levesque. "Intention is Choice with Commitment". *Artificial Intelligence* 42, 2-3 213-261, 1990.

[CPFF] A. E. M. Ciarlini, C. T. Pozzer, A. L. Furtado, B. Feijó. "A logic-based tool for interactive generation and dramatization of stories". In Proc. of *Advances in Computer Entertainment Technology*, Valencia, Spain, 2005.

[Cu] J. Culler. *On Deconstruction: Theory and Criticism after Structuralism*. Ithaca: Cornell University Press, 1983.

[FC1] A. L. Furtado, A. E. M. Ciarlini. "Generating Narratives from Plots using Schema Information". In Proc. of the *5th International Conference on Applications of Natural Language to Information Systems*, Versailles, France, 2000.

[FC2] A. L. Furtado, A. E. M. Ciarlini. "Constructing Libraries of Typical Plans". In Proc. of the *13th International Conference on Advanced Information Systems Engineering*, 124 - 139, 2001.

[FCBB] A. L. Furtado, M. A. Casanova, S. D. J. Barbora, K. K. Breitman. "Analysis and Reuse of Plots using Similarity and Analogy". In *Proc. of the 27th International Conference on Conceptual Modeling (ER)*. Barcelona, 2008.

[FN] R. E. Fikes, N. J. Nilsson. "STRIPS: A new approach to the application of theorem proving to problem solving". *Artificial Intelligence* , 2(3-4), 1971.

[FT] G. Fauconnier, M. Turner. *The Way We Think*. New York: Basic Books, 2002.

[FV] A. L. Furtado, P. A. S. Veloso. "Folklore and Myth in *The Knight of the Cart*". In *Arthuriana*, vol 6, n. 2, pages 28-43, 1996.

[GB] D. Grasbon, N. Braun. "A morphological approach to interactive storytelling".In Proc. *CAST01, Living in Mixed Realities*. Special issue of *Netzspannung.org/journal, the Magazine for Media Production and Inter-media Research*, p. 337-340, Sankt Augustin, Germany, 2001.

[HM] P. C. Hearst, A. Moscow. *Patty Hearst: her own Story*. New York: Avon, 1988.

[Ho] Homer. *The Iliad*. H. Hammond (trans.). London: Penguin, 1987.

[Ka] H. A. Kauz. "A Formal Theory of Plan Recognition and its Implementation". In *Reasoning about Plans*. J. F. Allen et al (eds.). Morgan-Kaufmann, 1991.

[Li] Titus Livius. *History of Rome*, vol I. B. O. Fster (trans.). Cambridge: Harvard University Press. Book I, Chapter I: 13 1919.

[LJ] G. Lakoff, M. Johnson. *Metaphors We Live By*. University of Chicago Press, 1980.

[Lo] W. Lloyd. *Foundations of Logic Programming*. Berlin: Springer-Verlag 1987.

[Ma] C. Marchello-Nizia (org.). *Tristan et Yseut*. Paris: Gallimard, 1995.

[Mg] M. McGarry (ed.). "The Story of Deirdre". In *Great Folk Tales of Ireland*. London: Frederick Muller, 1979.

[Mk] K. R. McKeown. *Text Generation: Using Dscourse Strategies and Focus Constraints to Generate Natural Language Text*. Cambridge: Cambridge University Press, 1992.

[MS] M. Mateas, A. Stern. "Towards integrating plot and character for interactive drama". In *Socially Intelligent Agents: the Human in the Loop,* AAAI Fall Symposium, technical report, p. 113-118, Menlo Park, CA. AAAI Press, 2000.

[Or] A. Ortony (ed.). *Metaphor and Thought*. Cambridge: Cambridge University Press, 1996.

[Ov] Ovid. *Heroides and Amores*. G. Showerman (trans.). Cambridge: Harvard University Press, 1986.

[Pr] V. Propp. *Morphology of the Folktale*. S. Laurence (trans.). Austin: University of Texas Press, 1968.

[RG] A. S. Rao, M. P. Georgeff. "Modeling rational agents within a BDI-architecture". In Proc. of the *International Conference on Principles of Knowledge Representation and Reasoning*, 473-484, 1991.

[RY] M. Riedl, R. M. Young. An intent-driven planner for multi-agent story generation. In the *Proc. of the 3rd International Conference on Autonomous Agents and Multi Agent Systems*, 2004.

[Sa] F. de Saussure. *Cours de Linguistique Générale*. C. Bally, A. Sechehaye, A. Riedlinger (eds.). Paris: Payot, 2006.

[Sg] N. M. Sgouros. "Dynamic generation, managing and resolution of interactive plots". *Artificial Intelligence*, 107, pp. 29-62, 1999.

[THA] K. N. Truong, G. R. Haye, G. D. Abowd. "Storyboarding: An Empirical Determination of Best Practices and Effective Guidelines". In Proc. of the *6th conference on Designing Interactive systems*, University Park, 2006.

[Tu] S. R. Turner. "Minstrel: A computer model of creativity and storytelling". Technical Report UCLA-AI-92-04, Computer Science Department, 1992.

[Va] Valmiki. *Le Ramayana*. Philippe Benoît et al (trans.). Paris: Gallimard, 1999.

[Vi] Virgil. *Eclogues, Georgics, Aeneid*. H. R. Fairclough (trans.). Cambridge: Harvard Univ. Press, 1994.

[Wa] D. H. D. Warren. *A System for Generating Plans*. Edinburgh: University of Edimburgh, Department of Computational Logic, memo 76, June 1974.

[Wh] H. White. *Metahistory: The Historical Imagination in Nineteenth-Century Europe*. Baltimore: The Johns Hopkins University Press, 1973.

Appendix A: Conceptual schemas for the example

A.1. The static schema

```
entity(character,name).
attribute(character,gender).
entity(place,pname).
relationship(loves,[character,character]).
relationship(home,[character,place]).
relationship(current_place,[character,place]).
relationship(held_by,[character,character]).
relationship(consents_with,[character,character]).
user_controlled(loves(_,_)).
```

A.2. The dynamic schema

A.2.1. Level-1 operators

```
operation(abduct(M2,W),1).
deleted(current_place(W,Pc),abduct(M2,W)) :-
  home(M2,P2), not (Pc == P2).
/deleted(current_place(M2,Pc),abduct(M2,W)) :-
  home(M2,P2), not (Pc == P2).
added(current_place(W,P2),abduct(M2,W)) :- home(M2,P2).
/added(current_place(M2,P2),abduct(M2,W)) :- home(M2,P2).
added(held_by(W,M2),abduct(M2,W)).
precond(abduct(M2,W),current_place(W,Pc)) :-
   loves(M2,W),
   gender(W,female),
   home(M2,P2),
   home(W,P1),
   not (P1 == P2),
   place(Pc),
   not (Pc == P1),
   not (Pc == P2).

operation(elope(M2,W),1).
deleted(current_place(W,Pc),elope(M2,W)) :-
  home(M2,P2), not (Pc == P2).
/deleted(current_place(M2,Pc),elope(M2,W)) :-
  home(M2,P2), not (Pc == P2).
added(current_place(W,P2),elope(M2,W)) :- home(M2,P2).
/added(current_place(M2,P2),elope(M2,W)) :- home(M2,P2).
added(consents_with(W,M2),elope(M2,W)).
precond(elope(M2,W),current_place(W,Pc)) :-
   loves(M2,W),
   loves(W,M2),
   gender(W,female),
   home(M2,P2),
   home(W,P1),
   not (P1 == P2),
   place(Pc),
   not (W == M2),
   not (Pc == P1),
   not (Pc == P2).
```

```
operation(rescue(M1,W),1).
deleted(current_place(W,P2),rescue(M1,W)) :-
  home(W,P1), not (P2 == P1).
/deleted(current_place(M1,P2),rescue(M1,W)) :-
  home(W,P1), not (P2 == P1).
/deleted(held_by(W,M2),rescue(M1,W)) :-
  home(W,P1), home(M2,P2), not (M2 == M1), not (P2 == P1).
added(current_place(W,P1),rescue(M1,W)) :- home(W,P1).
/added(current_place(M1,P1),rescue(M1,W)) :- home(W,P1).
added(consents_with(W,M1),rescue(M1,W)).
precond(rescue(M1,W),(current_place(W,P2), /(not held_by(W,M1)))) :-
   gender(W,female),
   home(W,P1),
   home(M2,P2),
   not loves(W,M2),
   character(M1),
   not (M1 == W),
   not (M1 == M2),
   not (P1 == P2).

operation(capture(M1,W),1).
deleted(current_place(W,P2),capture(M1,W)) :-
  home(W,P1), not (P2 == P1).
/deleted(current_place(M1,P2),capture(M1,W)) :-
  home(W,P1), not (P2 == P1).
/deleted(held_by(W,M2),capture(M1,W)) :-
  home(W,P1), home(M2,P2), not (P2 == P1).
added(held_by(W,M1),capture(M1,W)).
added(current_place(W,P1),capture(M1,W)) :- home(W,P1).
/added(current_place(M1,P1),capture(M1,W)) :- home(W,P1).
precond(capture(M1,W),current_place(W,P2)) :-
   gender(W,female),
   home(W,P1),
   home(M2,P2),
   character(M1),
   not (M1 == W),
   not (M1 == M2),
   not (P1 == P2).
```

A.2.2. Level-2 operators

```
operation(ride(C,P1,P2),2).
deleted(current_place(C,P1),ride(C,P1,P2)) :- not (P1 == P2).
added(current_place(C,P2),ride(C,P1,P2)) :- not (P1 == P2).
precond(ride(C,P1,P2),current_place(W,P2)) :-
  loves(C,W),
  gender(C,male),
  gender(W,female),
  place(P1),
  place(P2),
  not (P1 == P2).
```

```
operation(seize(M,W),2).
added(held_by(W,M),seize(M,W)).
precond(seize(M,W),
  (/current_place(W,P),
   current_place(M,P),
   not held_by(W,M2))) :-
  gender(M,male),
  gender(W,female),
  gender(M2,male),
  not (M == M2),
  place(P),
  not home(W,P),
  not home(M,P).
precond(seize(M,W),
  (/current_place(W,P),
   /current_place(M,P),
   /consents_with(W,M),
   /(not held_by(W,M2)))) :-
  gender(M,male),
  gender(W,female),
  gender(M2,male),
  not (M == M2),
  place(P),
  not home(W,P),
  home(M,P).

operation(entreat(M,W),2).
added(consents_with(W,M),entreat(M,W)).
precond(entreat(M,W),
  (/current_place(W,P),
   current_place(M,P),
   /(not held_by(W,M)),
   not held_by(W,M2))) :-
  loves(M,W),
  loves(W,M),
  gender(M,male),
  gender(W,female),
  gender(M2,male),
  not (M == M2),
  place(P),
  not home(M,P),
  not home(W,P).

operation(carry(M,W,P2),2).
deleted(current_place(M,P1),carry(M,W,P2)) :-
  home(M,P2), not (P1 == P2).
deleted(current_place(W,P1),carry(M,W,P2)) :-
  home(M,P2), not (P1 == P2).
added(current_place(M,P2),carry(M,W,P2)) :-
  home(M,P2).
added(current_place(W,P2),carry(M,W,P2)) :-
  home(M,P2).
precond(carry(M,W,P2),
  (consents_with(W,M),
  /current_place(W,P1),
  /current_place(M,P1))) :-
  gender(W,female),
```

```
  gender(M,male),
  loves(M,W),
  loves(W,M),
  home(M,P2).
precond(carry(M,W,P2),
  (held_by(W,M),
  /current_place(W,P1),
  /current_place(M,P1))) :-
  gender(W,female),
  gender(M,male),
  loves(M,W),
  home(M,P2).

operation(defeat(M1,M2),2).
deleted(held_by(W,M2),defeat(M1,M2)).
precond(defeat(M1,M2),
  (/current_place(M2,P),
   /current_place(W,P),
   /held_by(W,M2),
   /current_place(M1,P))) :-
   gender(M1,male),
   gender(M2,male),
   not (M1 == M2),
   loves(M1,W),
   place(P).
```

A.3. The behavioural schema

A.3.1. Goal-inference rules

```
% For the ravisher
sit_obj(M2,
  (current_place(W,P3),not current_place(M1,P3)),
  (current_place(W,P2))) :-
  gender(M1,male),
  gender(M2,male),
  gender(W,female),
  not (M1 == M2),
  place(P3),
  not home(P3,_),
  home(W,P1),
  home(M2,P2),
  not (P1 == P2).

% For the protector
sit_obj(M1,
  (current_place(W,P2), believes(M1,not loves(W,M2))),
  (current_place(W,P1), not held_by(W,M1))) :-
  gender(M1,male),
  gender(M2,male),
  gender(W,female),
  not (M1 == M2),
  home(M1,P1),
  home(W,P1),
  home(M2,P2),
  not (P1 == P2).
```

```
% For the protector
sit_obj(M1,
   (current_place(W,P2), believes(M1,loves(W,M2))),
   (current_place(W,P1), held_by(W,M1))) :-
  gender(M1,male),
  gender(M2,male),
  gender(W,female),
  not (M1 == M2),
  home(M1,P1),
  home(W,P1),
  home(M2,P2),
  not (P1 == P2).
```

A.3.2. Default beliefs (only for the protector)

```
believes(C,F,_) :-
  believes(C,F).
believes(C,F,S) :-
  belief(C,F,S),
  not added_belief(C,F).

added_belief(C,F) :-
   (F = (not F1),Fe = F1,!;
    Fe = F),
   (believes(C,Fe);
    believes(C,not Fe)).

belief(M1,loves(W,M2),S) :-
  gender(M1,male),
  gender(M2,male),
  not (M1 == M2),
  gender(W,female),
  home(M2,P2),
   (once((current_place(W,P2),
          current_place(M2,P2),
          not current_place(M1,P2),
         not held_by(W,M2)),S),!;
  holds(held_by(W,M1),S)).

belief(M1,not loves(W,M2),S) :-
  gender(M1,male),
  gender(M2,male),
  not (M1 == M2),
  gender(W,female),
  home(M2,P2),
  once((current_place(W,P2),held_by(W,M2)),S),
  not holds(held_by(W,M1),S).

beliefs :-
  log(L),
  forall(believes(A,F,L),describe(believes(A,F,L))).

user_controlled(believes(_,_)).
```

A.3.3. Feelings at an indicated state

```
emotional_condition :-
  log(L),
  forall(character(C),
    (once(emotional_condition(C,S,L)),
     describe(emotional_condition(C,S,L)))).

emotional_condition(C,S) :-
  emotional_condition(C,S,start).

emotional_condition(C1,absolutely_happy,S) :-
  gender(C1,female),
  emotional_condition(C1,happy,S),
  not (state(Si,S), holds(held_by(C1,_),Si)).

emotional_condition(C1,happy,S) :-
  character(C1),
  loves(C1,C2),
  not (character(C3),
       not (C3==C2),
       holds(held_by(C1,C3),S)),
  once((holds(current_place(C1,P),S),
        holds(current_place(C2,P),S))).

emotional_condition(C,bored,S) :-
  character(C),
  not emotional_condition(C,happy,S).
```

A.4. Example initial state

A.4.1. Fixed properties

```
place(forest).
place(palace).
place('Lanka').
character('Sita').
gender('Sita',female).
home('Sita',palace).
character('Rama').
gender('Rama',male).
home('Rama',palace).
character('Ravana').
gender('Ravana',male).
home('Ravana','Lanka').
```

A.4.2. Varying properties

```
current_place('Sita',forest).
current_place('Rama',palace).
current_place('Ravana','Lanka').
loves('Rama','Sita').
loves('Ravana','Sita').
loves('Sita','Rama').
```

A.5. Library of typical plots

```
ind(M,
    (gender(M,male),loves(M,W),loves(W,M),home(M,Pm),
     current_place(W,Pw), not (Pw==Pm)),
    (current_place(W,Pm)),
    start=>ride(M,Pm,Pw)=>entreat(M,W)=>seize(M,W)=>carry(M,W,Pm) ).

ind(M,
    (gender(M,male),loves(M,W),loves(W,M),home(M,Pm),home(W,Pm),
     current_place(W,Pw), not (Pw==Pm)),
    (current_place(W,Pm)),
    start=>ride(M,Pm,Pw)=>seize(M,W)=>carry(M,W,Pm) ).
```

A.6. Motifs

```
motif(life_token,
      (C1,C2,C3),
      in_distress(C1),
     (held_by(C1,C3),
      current_place(C1,P3),
      current_place(C3,P3),
      loves(C2,C1)),
     calling_for_help(C1,C2),
     (current_place(C2,P3), not current_place(C2,P1))) :-
     character(C1), character(C2), character(C3),
     home(C1,P1),
     home(C2,P1),
     place(P3),
     not (P3 == P1).

motif(love_potion,
      (C1,C2,C3),
      indifferent_to(C1,C2),
      (current_place(C1,P),
       current_place(C2,P),
       not loves(C1,C2)),
      melts_for(C1,C2),
      (loves(C1,C2), not loves(C1,C3))) :-
      character(C1),
     gender(C1,G1),
      character(C2),
     gender(C2,G2),
     gender(C3,G2),
     not (C3 == C2),
     not (G1 == G2),
      place(P),
      not home(C2,P).
```

```
motif(ordeal,
       (C1,C2,C3),
       under_suspicion(C1),
      (loves(C2,C1),
       current_place(C1,P1),
       current_place(C2,P1),
       not believes(C2,loves(C1,C2)),
       not (C3 == C2)),
      vindicated(C1),
        (believes(C2,not loves(C1,C3)),
         believes(C2,loves(C1,C2)))) :-
      character(C1), character(C2), character(C3),
      home(C1,P1),
      home(C2,P1),
      home(C3,P3),
      not (P3 == P1).
```

Appendix B: Main menus of PlotBoard

B.1. plot menu

```
1: compose
2: use given plot
3: take plot from library
4: stop
```

B.2. compose menu

```
1: user
2: planner
3: stop
```

B.3. user menu

```
1: goal
2: operation'), nl,
3: list of operations
4: query
5: operation level
6: directive
7: planner
8: finish
```

B.4. planner menu

```
1: continue
2: user
3: finish
```

B.5. plan step menu

```
1: ok
2: show effects
3: alternative
```

B.6. use given plot menu

```
1: plan from program
2: plan manually formulated
3: queries or directives
4: stop
```

B.7. use plot from library menu

```
character: -
for objective: -
may use plan: -
1: ok
2: show effects
3: alternative
```

B.8. submit menu

```
1: accept
2: adapt
3: validate
4: show
5: stop
```

B.9. adapt menu

```
1: detail
2: summarize
3: change sequence
4: add operation
5: delete operation
6: replace operation
7: extend
8: queries or directives
9: insert motif
10: back to the submit options
11: stop
```