

Date of acceptance

Grade

Instructor

Planning Based Computational Narrative With an Emphasis on Surprise Arousal

Amin Sorkhei

Helsinki March 15, 2015

UNIVERSITY OF HELSINKI

Department of Computer Science

Tiedekunta — Fakultet — Faculty		Laitos — Institution — Department	
Faculty of Science		Department of Computer Science	
Tekijä — Författare — Author			
Amin Sorkhei			
Työn nimi — Arbetets titel — Title			
Planning Based Computational Narrative With an Emphasis on Surprise Arousal			
Oppiaine — Läroämne — Subject			
Computer Science			
Työn laji — Arbetets art — Level		Aika — Datum — Month and year	Sivumäärä — Sidoantal — Number of pages
		March 15, 2015	10 pages + 0 appendices
Tiivistelmä — Referat — Abstract			
<p>At the galloping rate that computational creativity is flourishing, narrative systems has an important role to play and different computer scientists have tried to come up with methods and approaches in order to create intelligent writers for educational and entraining purposes [RY10]. Through the reach history of this newly emerged field, different well known methods like planning based computational narratives have been proposed while this category has been belittled to goal-driven problem-solving aesthetics [ZO10]. This short report has as its purpose to propose some general guidelines about computational narratives which are the building blocks of successful systems like MEXICA.</p>			
Avainsanat — Nyckelord — Keywords			
Säilytyspaikka — Förvaringsställe — Where deposited			
Muita tietoja — Övriga uppgifter — Additional information			

Contents

1	Introduction	1
2	Narrative and Planning	1
2.1	Introduction to Narrative	1
2.2	Quality of a Narrative	2
2.3	Introduction to planning	3
2.3.1	Customized planning versus classical planning	4
3	Tuned algorithms for narrative generation	5
3.1	POCL planners	5
3.2	IPOCL Planners: First encounter	7
4	MEXICA: A planning based narrative system	8
5	Discussion	8
6	Conclusion	9
	References	9

1 Introduction

As the first step, in order to have systems capable of narrating in a plausible way, some requirements must be satisfied by the system. To put this in other words the methodology and constraints followed by human writers, must be codded to the system. These requirements and constraints are called '*Cognitive Account*'. Additionally, the whole process of writing in human writers can be considered as a cycle of *engagement* and *reflection*.

2 Narrative and Planning

This section provides the reader with some background about both narrative and planning as the foundations of computational narrative. This section embarks on with a quick overview on narrative and what it is composed of.

2.1 Introduction to Narrative

Narrative and story-telling are old terms that has an specific role in human history. Since the dawn of human beings, people have tried to transfer their beliefs and ideas and even experiences to the next generation using narratives. On the other hand, psychologists believe that narratives have specific role to play in in pedagogical purposes. While the role of narrative and story-telling is quite appreciated in history, scientists believe that narrative is not well defined. One definition for narrative, proposed in [RY10], is:

Narrative: *A narrative is the recounting of a sequence of events that have a continuant subject and constitute a whole*

While this is a general definition regarding narrative and storytelling, we will need a more detailed investigation about what narratives are composed of. Narratologists break the narratigve into two layers of interpretation: *fabula* and *sjuzet*.

Fabula is the general content of the narrative, loosely speaking, fabula is what the story is going to talk about. On the other hand *sjuzet* is the way the story is going to be presented. To put this in order words, sjuzet decides the order of presentation of events in fabula.To put this in more understandable words, fabula is the raw material of the story while sjuzet determines the way a story is organized. By the

way of example, if a narrative is going to talk about the bibliography of a scientist, fabula contains the bibliography of the scientist, while sjuzet specifies the order the story is presented. It can be organized from the time the scientist dies and goes back to his birthday or the other way around. Regardless of which of these sjuzets are preferred, the fabula stays the same.

It is noteworthy to mention that the study focuses mainly on narrative generation in fabula level which is sometimes called '*Fabula Planning Problem*'. Officially Fabula Planning problem is defined as [RY10]:

Fabula Planning Problem: *Given a domain theory, find a sound and believable sequence of actions that transfer the initial state I to a predefined goal state G where some conditions are satisfied*

In this definition the domain theory, are a set of actions that change the story world. For example, committing a crime and rain fall are two simple examples of those actions.

2.2 Quality of a Narrative

So far we know what a Narrative and a 'Fabula Planning Problem' is and now we face an important question '*What is a good narrative?*'.

Inherently, audience of a story are inferring machines who try to infer and guess the final result of a story based on the actions performed by characters so far. In other words, human audience constantly, checks how the actions performed by a character are *believable* and *intentional*. More formally the audience pays attention to character intentionality and believability. Thus, if the audience find the actions committed by a character believable and intentional, the story can be considered as a good one. More importantly, believable and intentional characters bring about a story where human audience can guess the outcome of the story. Definitely, this can not be considered as a good story for people who are fan on surprise in narratives! As the final word, it is quite acknowledged that believable and intentional actions needs some goal to follow. In other words, believability and intentionality of characters are clearly based on character goals. The important fact which needs to be notices is that, character goals in a story are not necessarily are in the same direction with final goal of the narrative as a whole constitute. To put this in other words, character may have adversarial goals as compared to each other. As an example, a character may want to kill the other character, while the other one wants to strive and live.

More importantly, character goals may get resolved and formed during a narrative. As an illustration to this, in some stages of a story one character may prefer to avenge another character, while in future stage they may cultivate friendships due to some sacrifices.

2.3 Introduction to planning

This section provides an overview on planning and presents detailed reasons whether the classic form of planning can be used as a building block for computational narrative or not and finalizes with a brief introduction to customized planning algorithm for narrative generation.

Planning—in general terms— can be considered as one of the well established and old topics in Artificial Intelligence(AI). In the simplest form, through the planning process, intelligent agent tries to come up with a sequence of actions in order to accomplish a predefined goal. Formally planning can be considered using a state-system transition as a 4-tuple $\Sigma = (S, A, E, \gamma)$

$S = \{s_1, s_2, \dots\}$ is a set of states used to describe the world in a given moment

$A = \{a_1, a_2, \dots\}$ is a set of actions which can be performed by the agent

$E = \{e_1, e_2, \dots\}$ is a set of events which can be performed independent of the agent like having an accident.

$\gamma = S \times (A \cup E) \longrightarrow 2^S$ is a transition function which transits the current state of the world based on performed action or observed event.

Trivially for each planning problem, it is required to have a starting state and a goal state where the final goal on the planning can be described as getting started at starting point, executing the planned sequence of actions and reaching the final goal. Solving 8-tile puzzle can be considered as a simple example of planning problem where actions are moving tiles to the blank area and states are one of the $9!$ possible permutations of tiles. It is noteworthy that in this example, the event set is an empty set. As the final remark, planning can be divided to partial-order and total order planning. In partial-order some actions can be switched with each other while in the latter form – total-order planning – switching actions results in total failure. After this brief introduction to planning, a pivotal question can be whether this general form of planning can be considered as a medium for narrative generation or not. The answer to this question will be investigated in detail through the upcoming sections.

2.3.1 Customized planning versus classical planning

As we are familiar with basics of planning, one might consider this off-the-shelf form of planning as the building block of narrative generation. To put this in more details, based on this methodology, each character in the story can be considered as an agent who tries to final accomplish the final goal in the story by doing some actions. This naive idea gives some fundamental ideas about narrative generation while it is too simple to be deployed to generate narratives. The most important defect regards this methodology is the fact that, the final goal of the story may be totally different from the goal of each agent (planner). To put this in other words, believable stories are built upon believable characters which in return requires each character to have unique intentions –which may or may not be the same as the final goal of the main plot– and must perform believable actions justifying the unique goal regarding the character [RY10]. As an illustration to this, consider the following plot of a story which is obtained from [RY10]:

There are three characters in the world, a king, a princess and a knight. All these characters live in castle where there is a tower where each character can be locked up. The final goal of the story is that the princess is locked up in the tower and the king is dead.'

By using the naive idea of using off-the-shelf classical planning algorithm, the following story may be generated

- 1. The princess kills the king*
- 2. The princess locks herself in the tower*

which is not believable to human audience at all, while the plan is totally valid as it starts from the starting position and transforms the world to the goal state where the king is dead and the princess is locked in the tower. From the logical and causal point of view the story does not ring any bell in the human audience to persuade him or her about the intention of the characters. Why does the princess kill the king and why dose she lock herself in the tower. More importantly, is it a believable action for a character to lock him or herself in a place? . Since there is no rationale answer to these questions, the believability and the intentionality of characters is demolished. The following example depicts a narrative which totally satisfies all the requirements of intentionality and believability of characters as well as reaching the predefined goals.

1. *The king locks the princess in the tower*
2. *The knight kills the king*

In this case, the human reader can easily reason the intentionality of the characters. The princess has done some thing which should have agonized the king and the king, in return, locked the princess in the tower and the knight kills the king as the revenge.

As the conclusion, for narrative generating purposes, the planning algorithm needs to satisfy the following conditions:

1. The planning algorithm needs to have specific structure for satisfy character intentionality and believability. To put this in other words, it is required to have some events which motivates a character to pursue a specific goal and perform an action.
2. The actions in the domain theory must come in a structure where the participants of an action are determined and precondition and postcondition of an action is specified. In this way a chronological order can be depicted among the actions where the precondition of an action is unified with the post condition of another one.

The upcoming sections talk about *POCL* and *IPOCL* as two specific planning algorithms tuned for narrative generation purposes.

3 Tuned algorithms for narrative generation

In this section the provided material and figures are based on [RY10], unless it is specified explicitly. So far we have got some minimum requirements for algorithms required to perform planning based narrative generation. In this section we get the chance to go through the detail of Partial Order Causal Link (POCL) planner and the refined form, Intent-Driven Partial Order Causal Link (IPOCL) planner.

3.1 POCL planners

As mentioned before as one of the requirements for the tuned algorithms, the actions must come in a from where the participants, precondition and effects of actions are specified and stored. In order to do so the actions(operators) are in STRIPS constructs. In STRIPS construct each operator has a name, a precondition – the condition required for an action to be performed – and a postcondition which is the effect of the action. Each operator may have variables, which can bound with

ground variables to determine the characters engaged in the action. Formally a POCL Plan is defined as:

POCL Plan: A POCL plan is a tuple $\langle S, B, O, L \rangle$ such that S is a set of plan steps(actions), B is a set of bindings constraints on the parameters of the steps in S , O is a set of temporal ordering of the form $s_1 < s_2$ where $s_1, s_2 \in S$ and L is a set of causal links of the form $\langle s_1, p, q, s_2 \rangle$ where $s_1, s_2 \in S$ and p is the effect (postcondition) of s_1 and q is the precondition of s_2 and p unifies with q

There are some points need to be elaborated on through the above definition: 1.As it was mentioned above the terms step, action and operator may be used interchangeably and convey the same meaning.

2. A causal link connects two steps with a condition p and is written as $s_1 \xrightarrow{p} s_2$ when s_1 results in a condition p which is required as a precondition for s_2 to execute.

POCL planners try to plan steps in an iterative manner by resolving some flaws at each step. In general terms, flaws are whatever that preclude an answer to be a valid one. More specifically, POCL planners try to resolve two main form of flaws: 1. *Open condition* flaw: An open condition flaw occurs when a the goal state in the plan is not satisfied, or there is an step like s_2 in the plan where its precondition is not satisfied by neither the initial step nor a preceding step like s_1 . The POCL planners tries to resolve open condition flaw by non-deterministically instantiating a new step or use the existing steps in order to satisfy the required precondition. It means that the planner tries to instantiate a new step or use the existing steps to come up with an action which has a postcondition that unifies with the required precondition or the goal state. In this case causal links of the form can be well used to record the satisfaction of an open flaw and store the dependencies between actions.

2. *Causal threat* flaw: This threat occurs where the effect of a step may undo the effect of another step in the plan. In this case the planner tries to resolve the flaw by explicitly ordering the conflicting flaws [WYHR14]. The following figure contains information about how iteratively POCL planner tries to resolve the flaws. So far we got the chance to get familiar with a powerful variation of planning algorithms where the causal link between the actions are totally guaranteed. In other words, The result of the POCL algorithm is guaranteed to be coherent and actions appear in a logical manner. Here comes up an important question, is POCL the best algorithms that we can deploy?

POCL ($\langle S, B, O, L \rangle, F, \Lambda$)

The first parameter is a plan. On the initial call to POCL, there are only two steps in S – the dummy initial step whose effect is the initial state and the final step. F is a set of flaws. On the initial call, F contains an open condition flaw for each goal literal in the goal situation. $B = L = \emptyset$. Λ is the set of action schemata. Output is a complete plan or *fail*.

I. Termination. If O or B is inconsistent, fail. Otherwise, if F is empty, return $\langle S, B, O, L \rangle$.

II. Plan Refinement.

1. **Goal selection.** Select an open condition flaw $f = \langle s_{\text{need}}, p \rangle$ from F . Let $F' = F - \{f\}$.
2. **Operator selection.** Let s_{add} be a step that adds an effect e that can be unified with p (to create s_{add} , non-deterministically choose a step s_{old} already in S or instantiate an action schema in Λ). If no such step exists, backtrack. Otherwise, let $S' = S \cup \{s_{\text{add}}\}$, $O' = O \cup \{s_{\text{add}} < s_{\text{need}}\}$, $B' = B \cup B_{\text{new}}$ where B_{new} are bindings (e.g., assignments of ground symbols to variables) needed to make s_{add} add e , including the bindings of s_{add} itself, and $L' = L \cup \{\langle s_{\text{add}}, e, p, s_{\text{need}} \rangle\}$. If $s_{\text{add}} \neq s_{\text{old}}$, add new open condition flaws to F' for every precondition of s_{add} .
3. **Threat resolution.** A step s_{threat} threatens a causal link $\langle s_j, e, p, s_k \rangle$ when it occurs between s_j and s_k and it asserts $\neg e$. For every used step s_{threat} that might threaten a causal link $\langle s_j, e, p, s_k \rangle \in L'$, non-deterministically do one of the following.
 - **Promotion.** If s_k possibly precedes s_{threat} , let $O' = O' \cup \{s_k < s_{\text{threat}}\}$.
 - **Demotion.** If s_{threat} possibly precedes s_j , let $O' = O' \cup \{s_{\text{threat}} < s_j\}$.
 - **Separation.** Let $O' = O' \cup \{s_j < s_{\text{threat}}, s_{\text{threat}} < s_k\}$ and let $B' = B' \cup$ the set of variable constraints needed to ensure that s_{threat} won't assert $\neg e$.

III. Recursive invocation. Call POCL ($\langle S', B', O', L' \rangle, F', \Lambda$).

Figure 1: POCL algorithm

The answer to this question is negative as POCL provides no way to address the intentionality and believability of characters. To put this in other words, POCL does not provide us with a mechanism to make the story world characters appear intentional. In this case, we need some modification to POCL in order to include some other steps – called motivating steps – to make the characters appear intentional and as the result, believable. More importantly, we may need to add some modifications to the construct that is used to present operators. Through the upcoming sections, detail elaboration of the modifications will be presented.

3.2 IPOCL Planners: First encounter

As a quick recap, we are dealing with problems where the outcome should be a plan where a wide range of actions are assigned to story world characters. We need each character to be intentional, which means that we need them to pursue individual goals, where individual goal may be adversarial and even different from the final outcome (final goal) of the narrative. The intention and goal of the characters may form, change and even terminate during the story and not necessarily at the end of the narrative. At the mean time, There must be a total or temporal order between

the actions. It seems crystal clear that we need a modified representation for actions in the new algorithm in order to consider the character intentionality through the planning. From an algorithmic point of view, the new algorithm searches two spaces at the same time, the space of the character intentions and the space of the plans.

4 MEXICA: A planning based narrative system

Based on the cognitive account, including dictated constraints and requirements, the writer probes the long term memory (LTM) and tries to write some novel and interesting ideas as part of the story. This step is called engagement. It may happen that probing the LTM fails. In this case, the writer needs to bring the topic into conscious memory, review his text and others' story and finally select some new ideas. This phase is called reflection [PyPS01]. In order to follow this paradigm, it is necessary to redefine the reflection and engagement process in computers. through the engagement process, MEXICA comes up with some text that is driven by constraints. In reflection state, MEXICA, breaks impasses and evaluates the novelty of the story. As a rule of thumb, MEXICA is required to be fed with some initial data which is nothing but some written stories stored in LTM and called *Previous Stories*. All these stories are composed of two components, explicit elements which are clearly mentioned in the story, and tacit elements that are implied by the story. These tacit and explicit elements are stored in a structure called *story-world* and are further used as the guideline to probe the LTM. It is quite noteworthy to discuss the constraints obtained from explicit and tacit elements in more details, as they are the key elements in driving new and novel text. MEXICA tries to proceed the story based on valid actions. In order to define valid actions, MEXICA requires to define some post conditions that are the results of previous actions. these post conditions are all stored as constraints. As an illustration to this, consider this action *The princess cure the eagle knight's injuries in the palace*. This simple action brings about lots of latent and explicit postconditions, which are all required to be updated and saved.

5 Discussion

At this stage, the main question that crosses mind is that how the classical planning algorithms can be configured, in order to satisfy the intentionality and believability

of characters. In more details, in addition to a tuned version of planning, a sort of languages proves to be necessary in order to record the pre and post-conditions of actions. Thus in this case the new planning algorithm can come up with plans where the believability and intentionality of characters are preserved. Finally, there a thought provoking idea regarding the centralization of the algorithm. To put this in other words, are planning algorithms supposed to be centralized or one can consider each character as an agent cooperating with other to accomplish some goals.[BY09][WYHR14]

6 Conclusion

In this seminar, the paper tries to come ups with methods about analogy based-story generation and approaches that keep the story coherent and interesting. Further details about how computational narratives try to avoid boringness by evaluating the quality of the generated story will be presented.

References

- BY09 Bae, B.-C. and Young, R., Evaluation of a computational model of surprise arousal in narratives. *Affective Computing and Intelligent Interaction and Workshops, 2009. ACII 2009. 3rd International Conference on*, Sept 2009, pages 1–6.
- PyPS01 Pérez y Pérez, R. and Sharples, M., MEXICA: A computer model of a cognitive account of creative writing. *Journal of Experimental & Theoretical Artificial Intelligence*, 13,2(2001), pages 119–139.
- RY10 Riedl, M. O. and Young, R. M., Narrative planning: Balancing plot and character. *J. Artif. Int. Res.*, 39,1(2010), pages 217–268. URL <http://dl.acm.org/citation.cfm?id=1946417.1946422>.
- WYHR14 Ware, S., Young, R., Harrison, B. and Roberts, D., A computational model of plan-based narrative conflict at the fabula level. *Computational Intelligence and AI in Games, IEEE Transactions on*, 6,3(2014), pages 271–288.

- ZO10 Zhu, J. and Ontanon, S., Towards analogy-based story generation. *ICCC*, 2010, pages 75–84, URL <http://www.iiia.csic.es/files/pdfs/analogy-story.pdf>.