CB-POCL: A Choice-Based Algorithm for Character Personality in Planning-based Narrative Generation*

Julio César Bahamón and R. Michael Young

Liquid Narrative Research Group
North Carolina State University
Raleigh, NC, USA
{jcbahamo,young}@csc.ncsu.edu

Abstract

The quality and believability of a story can be significantly enhanced by the presence of compelling characters. Characters can be made more compelling by the portrayal of a distinguishable personality. This paper presents an algorithm that formalizes an approach previously described for the incorporation of character personality in narrative that is automatically generated. The approach is based on a computational model that operationalizes personality as behavior that results from the choices made by characters in the course of a story. This operationalization is based on the Big Five personality structure and results from behavioral psychology studies that link behavior to personality traits.

1998 ACM Subject Classification I.2.8 Artificial Intelligence

Keywords and phrases Artificial Intelligence, Planning, Narrative Generation

Digital Object Identifier 10.4230/OASIcs.CMN.2013.4

1 Introduction

The presence of interesting and compelling characters is an essential element of narrative composition [14, 6]. Effective characters enable the audience to form a clear mental model of their beliefs, desires, intentions, and morality. This understanding of the characters can lead to a better understanding of the entire story and to a more effective delivery of its content.

Well-defined characters add to the complexity of a story, enhance its discourse, and are vital for the realization of crucial story elements, such as events and dialog [22, 14, 6]. Among the factors that contribute to the effective definition of a character we include: physical attributes, talents, emotions, beliefs, and personality. Characters should portray these factors in an interesting and believable manner. In particular, the depiction of a distinguishable personality is one of the key features that makes a character interesting and compelling.

Characters also play an essential narrative role as agents of change. Change can result both from a character's actions and from its reaction to the actions of others [6, 22]. Therefore, actions are one of the main techniques used by creative writers to define and describe fictional characters [14, 5, 22]. Further consideration of narrative structure, specifically plot points where branching occurs [4], indicates that choices made by characters can have a direct impact in determining the actions they perform. Additionally, choices may be linked to

* Support for this work was provided in part by NSF award 0941421.

Workshop on Computational Models of Narrative 2013.
Editors: Mark A. Finlayson, Bernhard Fisseni, Benedikt Löwe, and Jan Christoph Meister; pp. 4–23
OpenAccess Series in Informatics
OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

specific personality traits, an idea supported by research in behavioral psychology that has found correlation between people's actions and their personality [21, 10].

We posit that the link between choice and personality can be used in narrative to enable the perception of specific personality traits. An audience that is made aware of the existence of multiple choices that are available to a character will form an opinion of such character's personality based on the available choices, the choice selection, and the events that provide a context to the choice. Our approach is thus based on a computational model that operationalizes personality as behavior that results from the choices made by characters in the course of a story.

This paper presents a precise formulation of an idea previously described in [3]. We provide a formalization of the elements used to incorporate choice in planning-based narrative generation. Additionally, we give a detailed description of an algorithm that considerably improves upon, expands, and refines the approach previously proposed. Finally, we discuss a scenario that illustrates in detail the operation of the algorithm during story generation.

2 Background and Related Work

Previous approaches to introduce character personality in automatically-generated narrative have focused primarily on a character's immediate reaction to events [18, 26]. In contrast, this research focuses on the story as a whole and in particular the role of observable actions and their effect on the mental model that the audience forms when experiencing a story.

2.1 AI Planning

Planning is an artificial intelligence technique used to solve problems by finding a sequence of actions to achieve a goal state from a given initial state [36, 32]. A classical planning problem is represented by three inputs: an initial world state, a desired goal state, and a set of available actions that enable transitioning between world states. One of the applications of AI planning is the automatic generation of narrative.

An approach used in planning is to search through the space of plans. Search tree nodes represent partially ordered plans and edges represent plan refinements [36]. The planning algorithm maintains a partially ordered sequence of actions and a list of causal links [33] that indicate when the effect of an action is required to establish the precondition of another. The planning process starts with a null plan, where the *start* action has no preconditions and its effects are the literals in the initial state. The *goal* action has as its preconditions the literals in the goal state and does not have any effects. The algorithm non-deterministically selects an open precondition and chooses an existing or new action that establishes it. When a new action is added to the plan, ordering constraints and causal links are updated if necessary. The planning process succeeds when all preconditions have been satisfied and all the threats have been resolved and fails when the plan structure or bindings become inconsistent.

2.2 Planning-based Narrative Generation

A significant body of work has been dedicated to the development of AI systems for the automatic generation of narrative. The use of AI planners to automatically generate stories was first introduced in systems such as TALE-SPIN [20]. Considerable effort has been dedicated since then to the development and improvement of techniques, algorithms, and architectures to enable the application of the problem solving capabilities of AI planners to the automatic generation of narrative that is both interesting and coherent [31, 29]. It is important to note that the work presented in this paper follows an approach that is distinct from that used by Riedl and Young in the IPOCL planning algorithm [31]. IPOCL focuses on character intentionality by identifying goals that explain a character's actions, which is done without considering the character's personality. In contrast, the approach described in this paper focuses on the use of specific actions to enable the portrayal of specific personality traits. It is envisioned that both approaches can be complementary as part of a solution aimed at producing more coherent and interesting narratives.

The work of Lebowitz on the UNIVERSE system [15] focused on the generation of stories using a plan-based approach combined with predefined character models. The system uses author goals to control the story-generation mechanism and character goals to ensure that their actions reflect their personality and backgrounds. The use of character goals is dependent on the definition of detailed character models based on the concept of stereotypes.

Some of the approaches for the automatic generation of narrative have focused on the implementation of systems that direct the interaction among characters. Work by Assanie on the extension of synthetic characters based on the Soar QuakeBot environment [13] dealt primarily with providing agents with the ability to adjust to changing goals provided by an external narrative manager [2]. One of the design objectives was the implementation of characters able to exhibit behavior that would resemble improvisational actors. Work by Riedl and Stern on drama managers focused on the implementation of semi-autonomous agents that have the ability to *fail believably* [28]. The system uses various techniques to ensure that agents avoid situations that are in conflict with the goals set by the drama manager and also to behave in a manner that justifies agent failure due to conflict with a goal set by the drama manager.

2.3 Emotion Expressed Through Facial and Physical Gestures

Research by Loyall [17] focused on the creation of believable agents, defined as autonomous agents with a rich personality and properties similar to those of characters in the traditional arts (e.g., film). The *Hap* agent architecture was developed to provide a language that enables authors to describe agent personality in terms of goals and behaviors.

André et al. worked on the development of lifelike characters as a means to improve interaction between humans and virtual characters [1]. Their approach centers on the use of the Five Factor Model model of personality and the OCC [23] theory of emotion to control the affective state of a virtual agent. The affective state determines the behavior, physical gestures, and dialog used by the agent when it communicates with humans.

The work of Doce et al. applied the OCC theory of emotion [23] and the Five Factor Model [11] to create distinguishable personalities in virtual agents [8]. The authors developed a model of personality that applies traits described in the Five Factor model (e.g., extroversion) to affect specific cognitive and behavioral processes, such as coping mechanisms and bodily expressions. The OCC theory of emotion is used to generate emotional states that influence the agent's planning mechanism and physical expressions [24].

Recent work in the area of virtual agents has focused on a specific subset of character actions: utterances in dialog [18, 26]. Of particular interest to this research is the work of Mairesse and Walker on PERSONAGE, a natural language generator that can be configured to generate dialog to meet a predefined set of personality requirements [18]. PERSONAGE is built on an architecture that uses the Five Factor Model [11] to create a mapping between personality traits and dialog utterances.

Table 1 Big-Five Factors Mapped to Observable Behavior.

Factor	Likely Behaviors	Unlikely Behaviors
Agreeableness	Honesty	Aggression
	Responsibility	Confrontation
	Ambition	Sabotage
	Empathy	Irritability
	Generosity	Selfishness
Conscientiousness	Organization	Impulsiveness
	Dutifulness	Lack of ambition
	Achievement	Mischief
	Reliability	Anti-social
	Risk avoidance	Criminal

3 Computational Model of Character Personality

The goal of this research is to facilitate the inclusion of compelling characters in narrative that is automatically generated. We do so by producing character behaviors that adjust: to authorial goals, to story events, and to user interaction. We aim to produce characters that can be distinguished by the visible manifestations of their personality.

We use a behavior model based on the Big Five Personality structure defined by Goldberg [11], which provides the following factors for the classification of personality types:

1. Extroversion
2. Agreeableness
3. Dependability
4. Emotional Stability
5. Culture (or Openness).

Each factor is linked to personality traits that can be mapped to behavioral manifestations, according to results obtained by social psychologists Mehl et al. [21] and Funder and Sneed [10, 21, 10, 35, 19]. A sample mapping is shown in table 1.

Our model uses a solution based on a declarative approach in which character properties are used to dynamically choose the actions they perform. Additionally, the effects of actions are evaluated to gauge whether these show behavior consistent with a character's personality traits. Finally, actions are considered in conjunction with a contrasting set of alternatives.

4 Definitions

In this section we provide a set of definitions necessary in the description of our algorithm for choice-based character personality. We work within the context of a planning-based narrative generation system, where a plan data structure is used to represent the events of a story and the temporal and causal relationships between them [37, 31].

4.1 Planning-based Narrative Generation

These definitions focus on the use of a partial order planner to generate stories, they are based on the work of Young and Riedl [37, 31].

▶ **Definition 1** (State). A state is a conjunction of literals used to describe what is true and what is false in a story world.

Any literals not explicitly described in the initial state of the story world are assumed to be false (closed-world assumption [27]).

▶ **Definition 2** (Character Name). A character name is a constant symbol that represents a story agent. Character names are unique.

▶ **Definition 3** (Action Schema). An action schema is a template for an action that is possible in the story world. An action schema is a tuple $\langle \mathsf{ActionType}, P, E, V, \mathsf{MainCharacter} \rangle$ where ActionType is a unique identifier for the action, P is a set of literals that must be true prior to the execution of the action (preconditions), E is a set of literals established by the execution of the action (effects), V is the list of free variables used in the template, and MainCharacter is a special variable used in the action schema to designate the story character primarily responsible for the execution of the instantiated action. The value of MainCharacter can be null ($\emptyset$).

We use a STRIPS-like representation [9] to describe the set of actions that are available in the story world. Preconditions are represented as a conjunction of all-positive literals, whereas effects may be a conjunct of both positive and negative literals.

For example, the plan step used to represent the story event "the knight kills the evil wizard" is an instance of the following action schema:

```
(action
   :action-type kill
   :variables ?main ?char
   :main-character ?main
   :preconditions (has sword ?main) (alive ?char)
   :effects (dead ?char))
```

▶ **Definition 4** (Planning Domain). A planning domain Λ is the set of all action schemas available in the story world.

▶ **Definition 5** (Planning Problem). A planning problem is a tuple $\langle \Lambda, S_0, S_G, C \rangle$ where Λ is a planning domain, S_0 is a set of literals that specify an initial state of the story world, S_G is a set of literals that specify a goal state, and C is a set of character names available in the story world. Each $c \in C$ is a unique identifier.

▶ **Definition 6** (Binding Constraint). A binding constraint on a pair of free variables or constants (u, v) indicates that u and v must unify in any well-formed formula. A negated pair $\neg(u, v)$ indicates that u and v cannot unify in any well-formed formula.

For example, the following binding constraints could be used to represent two characters that take part in a story event:

$\beta = \{$`(?main-character, the-knight), (?char, the-evil-wizard)})`

▶ **Definition 7** (Step). A step describes an instance of an action schema in a plan. A step is a tuple $\langle \mathsf{ActionType}, \mathsf{StepID}, \mathsf{Pre}, \mathsf{Eff}, \beta \rangle$, where ActionType is the unique identifier of an action schema, StepID is an identifier for a step that is unique within the plan in which the step occurs, Pre is the set of step preconditions, Eff is the set of step effects, and β is a set of binding constraints on free variables in Pre.

▶ **Definition 8** (Ordering Constraints). An ordering constraint over two steps s_i and s_j is denoted $s_i < s_j$, where s_i and s_j are steps in a plan and s_i must execute before s_j does.

In this representation, plan steps are partially ordered with respect to time [33].

▶ **Definition 9** (Causal Link). A causal link is denoted $s_i \xrightarrow{p} s_j$, where s_i is a plan step with an effect p and s_j is plan step with a precondition p. Causal links are used to keep track of dependencies between actions that exist when one action establishes a literal that is a precondition for another.

▶ **Definition 10** (Plan). A plan is a tuple $\langle S, B, O, L\rangle$ where S is a set of unique step identifiers, B is a set of binding constraints on free variables in S, O is a set of ordering constraints, and L is a set of causal links. A plan P is complete if and only if the following conditions are true:

- Every precondition of every step in P=$\langle S, B, O, L\rangle$ is satisfied, i.e., $\forall s_i \in S, s_i$=$\langle \mathsf{Pre}, \mathsf{Eff}, B\rangle$, $\forall p \in \mathsf{Pre}$ there is a causal link $s_i \xrightarrow{p} s_j \in L$.
- All threats have been resolved, i.e., for all causal links in $s_i \xrightarrow{p} s_j \in L$, there is no step $s_k \in S$ such that $s_i < s_k < s_j$ is a valid ordering under O and that has an effect $\neg q$, where q unifies with p.

4.2 Character Choice

These definitions are based on an analysis of creative writing principles and narrative structure, which was introduced in previous work [3]. The analysis focused on the essential role that characters play in the composition of a story and in particular their importance for the realization of key story elements such as events and dialog. We extend the plan-based representation of a story in a manner that enables the implementation of an intelligent action selection mechanism used to model the choices for action made by characters during the course of a story.

▶ **Definition 11** (Viable Alternative). A viable alternative over literal p is a tuple $\langle P, \mathsf{Prob}, s_{\text{need}}, p, a_i\rangle$, where $P = \langle S, B, O, L\rangle$, $\mathsf{Prob} = \langle \Lambda, S_0, S_G, C\rangle$ is a planning problem, $s_{\text{need}} \in S$, p is a precondition of s_{need}, and a_i is an action schema in Λ. We say that schema a_i is a viable alternative for p in plan $P = \langle S, B, O, L\rangle$ just when $\exists e_i \in \mathsf{Eff}_{a_i}$ such that p and e_i unify and there is no effect in a_i that is the negation of a literal in the preconditions of s_{need} in the context of B.

▶ **Definition 12** (Unintended Consequence). An unintended consequence with respect to a viable alternative a_i is a tuple $\langle P, \mathsf{Prob}, s_{\text{need}}, p, a_i, x_i\rangle$, where P =$\langle S, B, O, L\rangle$, Prob = $\langle \Lambda, S_0, S_G, C\rangle$ is a planning problem, $s_{\text{need}} \in S$, p is a precondition of s_{need}, a_i is a viable alternative over p, and x_i is an effect of a_i. We say that x_i is an unintended consequence of a_i just when $x_i \neq p$, and $\neg\exists q \in \mathsf{Pre}_{s_{\text{need}}}$ such that x_i and q unify in the context of B.

▶ **Definition 13** (Branching Point). A branching point is a tuple $\langle P, s_b, s_i, p\rangle$, where P = $\langle S, B, O, L\rangle$, $s_b, s_i \in S$, and $p \in \mathsf{Pre}_{s_j}$. We say that s_b is a branching point in plan P just when the following conditions are true:

- Step s_b can be ordered immediately before s_i, i.e., $s_b < s_i$ is a valid ordering under O and $\neg\exists s_k \in S$ such that $s_b < s_k < s_i$ is a valid ordering.
- Step s_i is the first step in a causal chain of events s_i, s_{i+1}, ..., s_{i+n}
- There exists an effect $e_b \in \mathsf{Eff}_{s_b}$ that unifies with p in the context of B and p is a precondition for s_i
- Step s_b can be instantiated from a specific action schema that has been selected from a set that contains two or more viable alternatives over p.

A branching point represents a narrative event that advances the story by requiring progress toward the story's goal state through one of at least two possible story paths. This definition is based on the concept of story kernels proposed by Barthes [4, 6].

▶ **Definition 14** (Choice). A choice is a tuple $\langle \mathsf{Prob}, \mathsf{BranchingPoint}, A, p\rangle$, where $\mathsf{Prob} = \langle \Lambda, S_0, S_G, C\rangle$ is a planning problem, $\mathsf{BranchingPoint} = \langle P, s_b, s_i, p\rangle$, and A is a set of viable alternatives over p, such that any action schema $a_i \in A$ may be instantiated to create step s_b.

A choice occurs in the story generation process when a branching point exists. Thus, a single step in the plan can be the source of zero or many choices, for example: $\{\{s_b, \emptyset\}\}$ represents a step without choices and $\{\{s_b, a_1\}, \{s_b, a_2\}, \{s_b, a_3\}\}$ represents a step with multiple choices (where a_1, a_2, a_3 are viable alternatives for an effect p established by step s_b).

5 The CB-POCL Algorithm

A preliminary version of an algorithm to incorporate choice in a planning-based story generation process was previously introduced in [3]. However, that version was limited because it did not take into account changes needed to support choices in the plan structure. In particular, it did not provide a detailed process for evaluating viable alternatives and establishing their preconditions. The previous version also did not present details on the method used to select specific action schemata for the instantiation of plan steps; details were preliminary both in terms of action selection and in the evaluation of their differences. Further, the algorithm's definition of the components of a choice within the context of a plan, as well as the elements necessary to support it, was not precise enough to disambiguate the non-deterministic selection of actions in a conventional POCL algorithm from the choice-based selection introduced in the paper. Finally, the previous version of the algorithm presented a method for ranking actions that did not include much detail about its integration with a behavior-based personality model.

This paper presents CB-POCL, an expanded and improved version that addresses in detail the changes to the planning process (see Algorithm 1). The current version is based on the extension of a partial-order causal link planning (POCL) algorithm, such as UCPOP [36, 25], to ensure that choice is treated as a first-class object during narrative generation, i.e., the story structure and contents promote the existence of choices and make their existence evident to the audience. This approach builds on prior research by Young and his colleagues [37, 29, 31, 12]. CB-POCL addresses the limitations of the previous work because it provides a detailed and formalized description of the mechanism used to incorporate choices in stories generated by a POCL process. Further, the definition of the elements and plan structure components needed to represent a choice in a planning context presented here precisely disambiguates the differences between the process for action selection used by conventional POCL algorithms and the algorithm we are developing.

5.1 Algorithm Description

In conventional POCL algorithms, actions are chosen nondeterministically to address open preconditions (flaws) until a complete plan is constructed or the process fails [36]. In our approach, specific actions may be added to the plan to facilitate the depiction of personality traits assigned to a character, e.g., honesty. Furthermore, the plan is built to support the ability to replace such actions with alternatives that depict contrasting personality traits.

The initial invocation of the algorithm is similar to that of UCPOP [36, 25]. The values in $\langle S, B, O, L\rangle$ are initialized to represent the null plan for the planning problem, and the value of agenda is set to the list of conjuncts in the goal state. Additionally, Choices is initialized to $\{\{s_0, \emptyset\}, \{s_G, \emptyset\}\}$, to indicate the lack of choices before the initial step or after the final step. On subsequent invocations of the algorithm, if the agenda is empty, the process succeeds. When flaws remain, one is selected, and the planner attempts to find an action schema or existing plan step whose effects establish the necessary precondition.

In the CB-POCL algorithm the selection is informed by an evaluation of the effects of all valid actions and in particular the unintended consequences that may result from the execution of an action. This is intended to represent a character's choice to perform a specific action from a set of possible alternatives. Two key factors are used when considering actions: (1) the action is viable and (2) the action can be used to portray a character's personality. Action evaluation is assisted by the use of an oracle function, focusing on the effects of the action upon other story participants. The function gauges whether the effects of a given action show behavior consistent with specific personality traits. This information is then used to determine if the action is part of a group of viable alternatives with enough contrast to clearly demonstrate a choice. For example, a set of alternatives for obtaining money that includes {Work, Borrow, Beg} does not provide the same contrast as a set that includes {SellFarm, Steal, Earn}. When only one alternative is viable or not enough contrast exists, the planner proceeds with a standard POCL process.

A modified process is used when multiple alternatives are viable, and a choice is added to the plan. The purpose of the modifications is twofold: (1) ensure that the viability of all the alternatives in the set is maintained and (2) preserve the contrast provided by the plan step during subsequent invocations of the planning algorithm. The viability of all the alternatives is ensured in order to create plans where choices exist, i.e., any of the viable alternatives in a set can be replaced with a member of the same set and the plan remains complete. To ensure viability, the union of the preconditions of all the alternatives is added to the agenda and these are treated as any other flaw. To preserve contrast, we flag the causal link drawn from the effect of the added step and the step whose flaw is being resolved (s_{need}). The flag is used to prevent the use of the same effect to solve other flaws, which would lessen the significance and clarity of the choice. The step is flagged to prevent its use in the plan unless it is in circumstances where it may enhance the portrayal of personality characteristics (e.g., honesty, aggression, or responsibility).

One of the potential problems that may result from the addition of choices to the plan is the existence of orphan steps, i.e., plan steps whose only purpose is to establish a precondition for one of the viable alternatives introduced by a choice. The following procedure is used when dealing with choice-related flaws in order to prevent the instantiation of orphan steps:

1. The use of steps that already exist in the plan is favored over the addition of new steps.
2. If using an existing step is not possible, the Initial State Revision (ISR) algorithm developed by Riedl and Young [30] is applied. ISR partitions the initial state of a planning problem into three sets: atomic ground sentences known to be true ($\mathcal{T}$), atomic ground sentences known to be false ($\mathcal{F}$), and atomic sentences whose truth value is undetermined ($\mathcal{U}$). The $\mathcal{U}$ set represents knowledge about the story world for which the author does not have a preference. ISR eliminates the flaw by committing to the value of a sentence in $\mathcal{U}$, enabling a causal link from the choice to the initial step (s_0).
3. If revision of the initial state is not applicable, a new step is added to the plan.

Other aspects of the POCL process work as expected, including: bindings update, causal link protection, failure detection, and the recursive invocation of the algorithm. Details of the functions that implement modified processes are provided in the following sections.

Algorithm 1 CB-POCL($\langle S, B, O, L\rangle$, Agenda, Λ, Choices)

```
if agenda is empty then return ⟨S, B, O, L⟩                ▷ Success, return the plan
end if
Select a flaw from the agenda. Each flaw in the agenda is represented by a pair ⟨p, s_need⟩, where
s_need ∈ S and p is a precondition of s_need.
S_add = GetViableAlternatives(p, s_need, ⟨S, B, O, L⟩), where each item in S_add is either a brand
new step instantiated from one of the schemas in Λ or an existing plan step that can be
consistently ordered prior to s_need.
if S_add = ∅ then
    return failure          ▷ No viable alternatives exist, a plan cannot be constructed.
end if
if Count(S_add) > 1 then                                    ▷ Update the plan structure
    Choice = true              ▷ Annotate the step to indicate that it is part of a choice
    s_need:choice = true
end if
                       ▷ Non-deterministically choose one of the viable alternatives
s_c, Contrast-set, Consistent-bindings = ChooseAction(p, s_need, S_add, ⟨S, B, O, L⟩, Rank)
                       ▷ Update causal links, bindings, and ordering constraints
L' = L ∪ {s_c →(p) s_need}
if Choice = true then
    Annotate the causal link, s_c →(p) s_need, to indicate that it is part of a choice
end if
B' = B ∪ {(u, v)|(u, v) ∈ MGU(q, p, B) and q is an effect of s_c}
O' = O ∪ {s_c < s_need}
if Choice = true then          ▷ Update the list of available choices, if applicable
    Choices' = Choices.
    for each viable alternative s_i in S_add do
        Let Choices' = Choices' ∪ {s_need, s_i}
    end for
end if
S' = S and agenda' = agenda                    ▷ Update the plan steps and the agenda
for each alternative s_i in (s_c ∪ Contrast-set) do
    if s_i ∉ S' then
        Add ⟨preconditions(s_i)\MGU(q, p, B), s_i⟩ to agenda'
        Add preconditions(s_i) to the preconditions list of s_c
    end if
end for
B' = B' ∪ Consistent-bindings
Add s_c to S'
for each causal link l = s_i →(p) s_j in L do                  ▷ Causal link protection
    for each plan step s_t that threatens l do         ▷ Nondeterministically choose one
        Promotion: If consistent, let O' = O' ∪ {s_j < s_t}
        Demotion: If consistent, let O' = O' ∪ {s_t < s_i}
        if neither Promotion nor Demotion can be chosen then return failure
        end if
    end for
end for
if B' is inconsistent then                                     ▷ Recursive invocation
    return failure
else
    Call CB-POCL(⟨S', B', O', L'⟩, agenda', Λ, Choices')
end if
```

5.2 The GetViableAlternatives Function

The GetViableAlternatives($p, s_{\text{need}}, \langle S, B, O, L\rangle$) function returns a set of actions newly instantiated from schemas in the planning domain library (Λ) or existing steps from the current plan that have p in their effects. An action (a_i) selected as a viable alternative must also be performed by a character whose personality the author wants to portray. Viable alternatives are evaluated to gauge whether their execution results in unintended consequences that show personality traits for a character. Choices are added to the plan only when the set of viable alternatives provides enough contrast to clearly portray a character's personality. When contrast does not exist, an action is chosen nondeterministically following the standard POCL process. For example, when all the viable alternatives result in unintended consequences consistent with honest behavior (a trait of agreeable personalities).

Algorithm 2 The GetViableAlternatives($p, s_{\text{need}}, \langle S, B, O, L\rangle$) function

$A = \emptyset$
if s_{need}:main-character = null **then** ▷ **Only consider steps by a main character**
  A = Nondeterministically choose any existing step $s_i \in S$
    OR a new action instantiated from Λ, such that the ordering constraint
    $s_i < s_{\text{need}}$ is consistent with O, s_i has an effect q that unifies with p given B,
    and s_i is not a step used in a Choice.
  return A
else ▷ **Prepare the initial list of viable alternatives**
  A = Select all existing steps $s_i \in S$ or new actions instantiated from Λ,
    such that the ordering constraint $s_i < s_{\text{need}}$ is consistent with O,
    s_i has an effect q that unifies with p given B, $s_i : \textit{choice} = \textit{false}$, and
    s_{need}:main-character unifies with s_i:main-character given B.
  ▷ **Evaluate the alternatives to check for contrast**
  C = The set of story characters such that s_{need}:main-character
    unifies with s_i:main-character given B
  A'=RankActions($A, C, p, s_{\text{need}}, \langle S, B, O, L\rangle$)
  if Count(A') ≥ 2 and Rank(A_{first}) $\neq$ Rank(A_{last}) **then**
    return A'
  else ▷ **Contrast does not exist**
    return (Nondeterministically choose an action from A')
  end if
end if

5.3 The RankActions Function:

This function evaluates a set of viable alternatives to gauge their compliance with the personality traits of a character. The output is a list of actions in descending order of compliance (see algorithm 3). An action's compliance with a personality trait is based on how closely the action is indicative of behavior that is typically associated with such trait, e.g., *avoidance of others* in the case of *introversion*.

5.4 The ChooseAction Function

The ChooseAction($p, s_{\text{need}}, S_{\text{add}}, \langle S, B, O, L\rangle$) function uses a selection mechanism informed by the personality traits of the character that performs the action and also by the context in

Algorithm 3 RankActions(Actions, Characters, p, s_{need}, $\langle S, B, O, L\rangle$)

for each action a_i in Actions **do**
  Rank$_i = 0$
  ▷ **Evaluate the unintended consequences of the action**
  for each character c_j in Characters **do**
    for each effect e_k established by a_i
      where $e_k \neq p$ and e_k does not unify with any other precondition of s_{need} **do**
      for each personality trait t_l assigned to c_j **do**
        Rank$_i$ = Rank$_i$ + EvaluateEffect(character, e_k, t_l, $\langle S, B, O, L\rangle$)
      end for
    end for
  end for
end for
A' = Sort Actions by their corresponding rank in descending order
return (A', Rank)

which such action takes place. The output of the function is a triple containing an action in S_{add} that represents the personality traits of a given character, a corresponding set of contrasting alternatives, and a set of bindings that ensure the consistency between the elements in the contrast set and s_{need}. These bindings are the minimal set of bindings that ensure that there are no conflicts within the set of all the preconditions of s_{need} and all the preconditions of the viable alternatives, thus guaranteeing their consistency with the plan. In this version of the algorithm, we use a behavior model that is a hand-crafted approximation of the Big Five structure [11].

Algorithm 4 The ChooseAction(p, s_{need}, S_{add}, $\langle S, B, O, L\rangle$, Rank) function

C = The set of story characters such that s_{need}:main-character
  unifies with s_i:main-character given B
▷ **Invoke the oracle function**
Choice = Nondeterministically choose an action from S_{add} that portrays the personality traits of s_{need}:main-character within the current story context.
▷ **For each alternative that is consistent with the character's personality, determine its contrasting set**
Contrast-set = Select the subset of actions from S_{add} that provides the most contrast with Choice.
Consistent-bindings = Consult B and then compute the set of bindings that ensure consistency between all the elements in Contrast-set and s_{need}.
return (Choice, Contrast-set, Consistent-bindings)

5.5 The *Oracle* Function for Trait Evaluation

An action a_i is said to demonstrate a personality trait t_k if its unintended consequences result in a state of the world that is consistent with behavior typical of people who score high on such trait. An action a_i is said NOT to demonstrate personality trait t_k if its unintended consequences result in a state of the world that is consistent with behavior that is typical of people who score low on the same trait. In a similar manner, the preconditions

Table 2 Rule for Generous Behavior – Effect.

Rule Name	GiveUpItems
Rule Type	Effect
Applies To	Agreeableness: generosity
Parameters	An action's unintended consequence $\Phi(x_1, x_2, \ldots, x_n)$, where $\Phi()$ is a predicate and $x_1, x_2, \ldots, x_n$ are variables. The current plan, $\langle S, B, O, L\rangle$. A character, c. OBJECT() is a predicate, such that OBJECT(x_i) = *True* if x_i represents a physical object in the story world, e.g., car, plane, sword, a bag of gold.

Description	▷ **If any of the variables is bound to an entity that is of type OBJECT**
	▷ **check whether the action's unintended consequence is the negation of a predicate**
	▷ **that is currently true. Then check the character's personality.**
	affectsObjects = False
	for each variable x_i in the action's unintended consequence **do**
	    if OBJECT(x_i) = True
	        affectsObjects = True
	    end if
	end for
	if affectsObjects = True AND
	   One of the variables in $(x_1, x_2, \ldots, x_n)$ represents character c AND
	   None of the variables in $(x_1, x_2, \ldots, x_n)$ represents characters other than c AND
	   $\Phi(x_1, x_2, \ldots, x_n)$ is the negation of a predicate that is currently true **then**
	    if c is Highly-Agreeable **then**
	        return 1
	    else
	        return -1
	    end if
	else
	    return 0
	end if

that must exist prior to the execution of an action can demonstrate behavior consistent with specific personality traits. An action a_i is said to demonstrate a personality trait t_k if the preconditions needed for its execution indicate a state of the world that is the result of behavior typical of people who score high on such trait. An action a_i is said NOT to demonstrate personality trait t_k if the preconditions needed for its execution indicate a state of the world that is the result of behavior typical of people who score low on such trait.

Initially we focus on the basic case in which given one personality trait, for a specific character, the planner evaluates the viable alternatives and selects the one most consistent with the trait. Note that effects and preconditions common to all the actions in a set of viable alternatives are ignored because they do not provide information to help differentiate the alternatives from each other.

5.5.1 Effects Evaluation:

The purpose of the EvaluateEffect(Character, e_j, t_k, $\langle S, B, O, L\rangle$) function is to gauge whether an action's effect demonstrates behavior that is consistent with a personality trait. We do this by considering the resulting state of the story if such an effect is established. For example, an action that results in the loss of treasure would be consistent with the behavior of a highly agreeable character, who is expected to act in a generous manner as indicated in table 1. In contrast, a non-agreeable character would be less willing to give up treasure and instead be more inclined to make choices that do not result in a state in which possessions are lost. This function applies all the rules of the appropriate type to the effect and character, in the context of the current partial plan, and returns an average of the score computed after applying all rules.

To enable the evaluation of effects, we use a declarative mapping between personality traits and behaviors. This mapping is implemented as an extensible set of behavioral rules that when applied can be used to evaluate individual effects. The rules are placed in a domain-independent knowledge base that can be maintained by the user. An example of a behavioral rule is shown in table 2.

6 Sample Scenario: A Hero's Quest

6.1 Story Argument

Princess Kayla has been kidnapped by the evil dragon Gomez. Joe, a local townsman, has been asked to rescue her. If Joe is to succeed in his quest, he must first obtain a sword that will enable him to slay Gomez and perform the rescue.

In this scenario, we consider Joe to be highly agreeable. Behavior that can be expected of him includes: honesty, responsibility, interaction, empathy, and generosity. Likewise, Joe is unlikely to exhibit behaviors such as aggression, sabotage, irritability, and selfishness. The story plan will be constructed to include choices that show behavior consistent with Joe's personality.

6.2 Scenario Alternatives

Consider a scenario where in order to obtain the sword Joe faces a choice from three alternatives: (1) he can buy the sword, (2) he can steal the sword through treachery, or (3) he can attack someone to take away the sword. Each one of these alternatives has a specific set of effects and preconditions that when added to the plan create a different version of the story. The contrast between the story produced by the chosen alternative and those that would result from the selection of the other alternatives is used to represent the act of making a choice – Joe's choice in this context. Furthermore, the selection of a specific alternative is determined by the personality traits that have been previously assigned to Joe. The plan is also constructed to support the alternate story paths that would result from the contrasting alternatives.

6.3 Action Schemata

The following action schemata are used to implement the alternatives in the scenario. Note that not all the actions used in the sample plan are included, just those with particular relevance to this discussion.

```
(action
        :action-type Buy
        :variables ?item ?buyer ?seller
        :main-character ?buyer
        :preconditions (has ?item ?seller)
                       (has gold ?buyer)
                       (knows ?buyer ?seller)
        :effects (has gold ?seller)
                 (not (has gold ?buyer))
                 (has ?item ?buyer))

(action
        :action-type Steal
        :variables ?item ?char ?victim
        :main-character ?char
        :preconditions (has ?item ?victim)
                       (confused ?victim)
        :effects (upset ?victim)
                 (has ?item ?char)
                 (not (has ?item ?victim)))


(action
        :action-type Fight
        :variables ?attacker ?victim
        :main-character ?attacker
        :preconditions (knows ?victim ?attacker)
                       (has sword ?victim)
        :effects (hurt ?victim)
                 (has sword ?attacker)
                 (not (has sword ?victim)))

(action
        :action-type Polish
        :variables ?char ?item ?location
        :main-character ?char
        :preconditions (at village ?char )
        :effects (has gold ?char))
```

6.4 Story Generation Walkthrough

Consider the partial story plan shown in figure 1, which shows the *Start* and *End* steps of the plan. At this point only the initial invocation of the algorithm has taken place and no steps have been added to address any open preconditions. After a few recursive invocations of CB-POCL, we may have the partial plan shown in figure 2. Note that the preconditions of the *End* step have been addressed by the addition of two new plan steps: *Slay* and *Rescue*. However, the addition of the new steps results in new preconditions that must also be addressed, in particular `(has sword Joe)` for the *Slay* step.

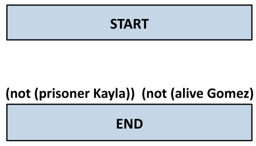

Figure 1 Sample Scenario – Initial Story Plan. In this figure, the rectangular boxes indicate steps in the plan. Time flows roughly horizontally downward. Initially, the start and end steps are the only steps in the plan.

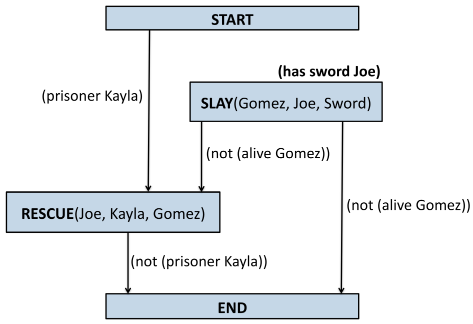

Figure 2 Sample Scenario – Partial Story Plan. In this figure, the rectangular boxes indicate steps in the plan. Time flows roughly horizontally downward. Solid arcs from one step to another indicate a causal link between the source step's effects and the destination step's preconditions.

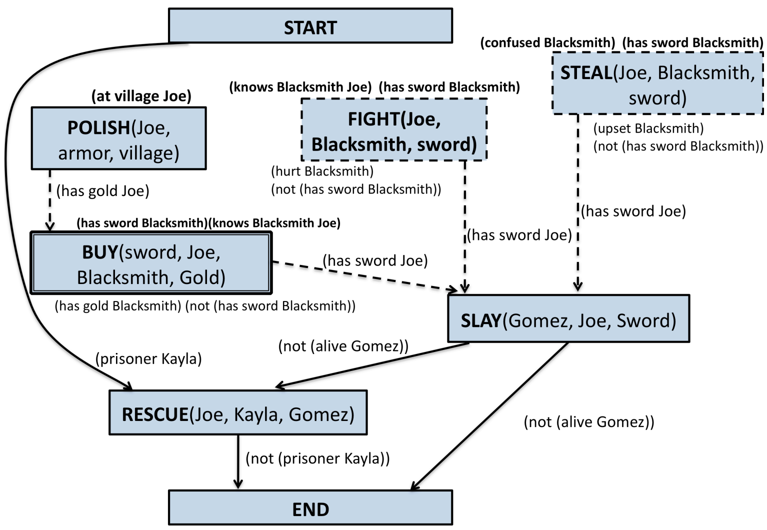

Figure 3 Partial Story Plan with Viable Alternatives Under Consideration. In this figure, steps drawn with dashed lines indicate viable alternatives that do not actually appear in the final plan used to generate the story. These are flagged as being part of a choice, which makes it possible to use them during discourse generation to help convey the existence of such choice. Similarly, dashed arcs indicate causal links that could potentially provide the needed condition from a viable alternative to a specific step; however, these links do not appear in the final plan. A box with a double border indicates a choice.

On a later invocation of the algorithm, we could have the situation shown in figure 3. The graph includes a set of viable alternatives being considered by CB-POCL to address the `(has Sword Joe)` precondition of the *Slay* step. Note that the addition of any of the alternatives – *Buy, Steal, or Fight* – establishes the necessary precondition to enable the execution of the *Slay* step. However, each of the alternatives has different effects not needed for the execution of the *Slay* step (unintended consequences). Evaluation of the unintended consequences by the oracle function would determine that the *Buy* alternative is more representative of a character with a highly agreeable personality and that *Steal* and *Fight* are appropriate contrasting options. Additionally, note that the oracle function also considers the `(has gold Joe)` precondition of the *Buy* option, which can be established by the *Polish* action further depicting behavior in accordance with that expected of a highly agreeable character. A highly agreeable character will choose the option of buying the sword, even though the precondition of having gold requires the extra effort of polishing armor and the buy action represents a loss of personal treasure. In contrast, a non-agreeable character will choose one of the remaining options because neither performing extra work to obtain gold nor giving it up in exchange for the sword is consistent with a non-agreeable personality.

Figure 4 shows the plan after the `(has Sword Joe)` precondition of the *Slay* step has been addressed by the addition of the *Buy* step. Steps to enable the execution of the *Buy* step have also been added. Alternate story paths based on the other two viable alternatives (*Steal, Fight*) are shown with a dashed border to illustrate that they could be part of the plan, yet they are not. Steps shown with a dashed border are not included in the final plan used to generate the story; however, since these are flagged as being part of a choice it may be possible to use them during discourse generation to help convey the existence of such choice. Also, the ordering of the *Travel* actions is important in this scenario because the main character must have visited all three locations (village, tavern, forest), which cannot be done simultaneously hence proper sequencing of these steps is required. Note that CB-POCL will ensure that the necessary supporting steps can be added to the completed plan by also addressing the corresponding preconditions: `(confused Blacksmith)` and `(knows Blacksmith Joe)`. ISR is applied to set the precondition that indicates who has the sword, which is needed for any of the alternatives being considered.

7 Discussion and Future Work

7.1 Algorithm Limitations

The CB-POCL algorithm currently does not guarantee that all the steps added to the plan are consistent with a character's personality traits. For example, in the scenario previously described the *Use Potion* step needed to enable the *Steal* viable alternative is not consistent with the agreeable personality of the character (Joe). A more robust version of the process used to construct the plan should take into account the entire chain of events that result from the addition of a viable alternative and not just the immediate steps. Additionally, plans produced by the algorithm may contain truncated action sequences that have been added only to establish preconditions for a viable alternative. Even though CB-POCL is designed to reduce the addition of isolated actions by favoring the use existing plan steps or applying Initial State Revision, there is no guarantee that truncated action sequences will never be part of a plan. Future work on this research will include exploring methods to further mitigate and ideally eliminate the occurrence of such sequences. Finally, this work in its current form does not address the limitations that stem from the inability to determine the completeness of a plan until the planning process has stopped. Working with a complete plan would be the ideal environment to identify the optimal structure and contents

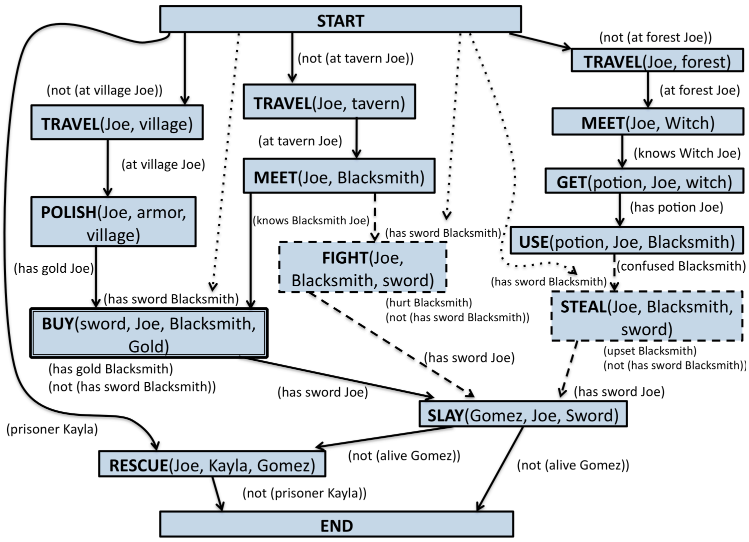

Figure 4 Partial Plan Showing Alternate Story Paths Based on Choice. In this figure, dotted arcs indicate causal links that are established from the initial state via initial state revision. In this figure, steps drawn with dashed lines indicate viable alternatives that do not actually appear in the final plan used to generate the story. These are flagged as being part of a choice, which makes it possible to use them during discourse generation to help convey the existence of such choice. Similarly, dashed arcs indicate causal links that could potentially provide the needed condition from a viable alternative to a specific step; however, these links do not appear in the final plan. A box with a double border indicates a choice.

to portray specific personality traits. However, since this is not feasible, future versions of CB-POCL should incorporate techniques to obtain an approximation of the knowledge needed to improve the plan structure while plan refinement is still in progress.

7.2 Planned Algorithm Enhancements

The current version of the algorithm does not consider action preconditions during the initial evaluation of viable alternatives. Preconditions are necessary for a personality-based action selection process when we reflect on their relevance to the state of the story world prior to the execution of an action. The need to establish a particular precondition may also result in the addition of other actions to the plan, thus providing more opportunities to portray a character's behavior. In addition to the elements considered for individual choices, it is necessary to consider changes to the process used to construct the plan structure. The algorithm should enable operations such as: changing the ordering of actions currently in the plan, increasing or reducing action decomposition, changing or introducing a causal chain of events, and dynamically introducing behavior-related constraints. These modifications would facilitate the construction of plans that treat choice as a first-class object. Finally, the action selection process presented in this paper uses a step-wise approach when reasoning about choices during story construction. An improved version of the algorithm should extend the model to enable reasoning about sub-plans and action decomposition.

7.3 Behavioral Model

The *oracle* function mentioned in this document must be replaced by a robust behavioral model that is less dependent on knowledge engineering and thus more flexible in its use.

Such a model will implement an operationalization of principles identified by behavioral psychology research, such as that of Mehl et al. and Funder and Sneed [21, 10].

One of the possibilities being considered for this component is the implementation of a model based on the goal-based hierarchical taxonomy proposed by Chulef, Reed, and Walsh [7]. The taxonomy includes 135 achievement goals identified through experimentation. Additionally, the authors propose conceptual relationships between personality traits and specific goals in the hierarchy.

Applying the concept of goal hierarchies, personality could be implemented as a set of goals of varying importance. For example, being likable, following social norm, and safety. In this model, the importance of specific goals can be determined by the personality traits they are associated with. Furthermore, characters could have goals of two types: communal and individual [34]. Another factor to consider is that goals require stimuli that activates them and resources that enable their achievement [34]. These affect the likelihood of enacting behavior. Goals can also be seen as incentives or deterrents for behavior. Finally, conflict may also have a role in the expression of personality. For example, when a character's goals are frustrated this could result in aggressive behavior. Highly social-goal oriented individuals are less likely to behave in an uncivil manner when they experience frustration [16].

8 Conclusion

This paper presents the next step toward the development of an intelligent mechanism that enables the automatic generation of narrative that elicits the perception of distinct character personalities without the need of a labor-intensive process. The solution is based on a declarative approach, in which character properties and the story context are used to model the choices that determine the set of actions that they perform in the course of the story.

The approach described extends a conventional POCL algorithm to ensure that choice is treated as a first-class object. The planner's data structures and supporting processes are modified to this effect. Additionally, the nondeterministic selection of actions is replaced by a mechanism that incorporates choice as a key component of the story generation process. Furthermore, personality is operationalized in terms of the behavior that results from the choices made by characters as they perform their role in the narrative.

Finally, an essential part of our future work will be to validate whether narratives generated using the CB-POCL algorithm result in characters whose personality traits are distinguishable to human audiences. We plan to conduct user studies designed to measure whether character behavior generated by the algorithm elicits in the audience the perception of corresponding personality traits.

References

1 Elisabeth André, Martin Klesen, Patrick Gebhard, Steve Allen, and Thomas Rist. Integrating models of personality and emotions into lifelike characters. In Ana Paiva, editor, *Affective Interactions: Toward a New Generation of Computer Interfaces?*, number 1814 in Lecture Notes in Computer Science, pages 150–165. Springer, 2000.

2 Mazin Assanie. Directable synthetic characters. In Ken Forbus and Magy Seif El-Nasr, editors, *Proceedings of the AAAI Spring Symposium on Artificial Intelligence and Interactive Entertainment*, number SS-02-01 in AAAI Technical Reports, pages 1–7, 2002.

3 Julio César Bahamón and R. Michael Young. A choice-based model of character personality in narrative. In Mark Alan Finlayson, editor, *Proceedings of Computational Models of Narrative 2012*, pages 164–168, İstanbul, 2012.

4 Roland Barthes and Lionel Duisit. An introduction to the structural analysis of narrative. *New Literary History*, 6(2):237–272, 1975.

5 Colin Bulman. *Creative Writing: a Guide and Glossary to Fiction Writing.* Polity, Cambridge, UK, 2007.

6 Seymour Benjamin Chatman. *Story and Discourse: Narrative Structure in Fiction and Film.* Cornell University Press, Ithaca, NY, 1978.

7 Ada S. Chulef, Stephen J. Read, and David A. Walsh. A hierarchical taxonomy of human goals. *Motivation and Emotion*, 25(3):191–232, 2001.

8 Tiago Doce, João Dias, Rui Prada, and Ana Paiva. Creating individual agents through personality traits. In Jan Allbeck, Norman Badler, Timothy Bickmore, Catherine Pelachaud, and Alla Safonova, editors, *Intelligent Virtual Agents, 10th International Conference, IVA 2010, Philadelphia, PA, USA, September 20-22, 2010, Proceedings*, number 6356 in Lecture Notes in Computer Science, pages 257–264. Springer, 2010.

9 Richard E. Fikes and Nils J. Nilsson. STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 2(3-4):189–208, 1971.

10 David C. Funder and Carl D. Sneed. Behavioral manifestations of personality: An ecological approach to judgmental accuracy. *Journal of Personality and Social Psychology*, 64(3):479–490, 1993.

11 Lewis R. Goldberg. An alternative "description of personality": The big-five factor structure. *Journal of Personality and Social Psychology*, 59(6):1216, 1990.

12 Justin Harris and R. Michael Young. Proactive mediation in plan-based narrative environments. In Themis Panayiotopoulos, Jonathan Gratch, Ruth Aylett, Daniel Ballin, Patrick Olivier, and Thomas Rist, editors, *International Conference on Intelligent Virtual Agents*, number 3661 in Lecture Notes in Computer Science, pages 292–304. Springer, 2005.

13 John E. Laird and Randolph M. Jones. Building advanced autonomous AI systems for large scale real time simulations. Presented at the Computer Games Development Conference, Long Beach, California, 1998.

14 Alice LaPlante. *The Making of Story: a Norton Guide to Creative Writing.* W. W. Norton, New York, 2007.

15 Michael Lebowitz. Creating characters in a story-telling universe. *Poetics*, 13(3):171–194, 1984.

16 Wu Liu, Shu-Cheng Steve Chi, Ray Friedman, and Ming-Hong Tsai. Explaining incivility in the workplace: The effects of personality and culture. *Negotiation and Conflict Management Research*, 2(2):164–184, 2009.

17 A. Bryan Loyall. *Believable agents: building interactive personalities.* PhD thesis, Carnegie Mellon University, 1997.

18 François Mairesse and Marilyn Walker. PERSONAGE: Personality generation for dialogue. In Annie Zaenen and Antal van den Bosch, editors, *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 496–503, Prague, Czech Republic, 2007. Association for Computational Linguistics.

19 Robert R. McCrae and Oliver P. John. An introduction to the five-factor model and its applications. *Journal of Personality*, 60(2):175–215, 1992.

20 James R. Meehan. TALE-SPIN, an interactive program that writes stories. In Dabbala Rajagopal Reddy, editor, *Proceedings of the 5th International Joint Conference on Artificial Intelligence. Cambridge, MA, August 1977*, pages 91–98. William Kaufmann, 1977.

21 Matthias R. Mehl, Samuel D. Gosling, and James W. Pennebaker. Personality in its natural habitat: Manifestations and implicit folk theories of personality in daily life. *Journal of Personality and Social Psychology*, 90(5):862–877, 2006.

22 Matt Morrison. *Key concepts in creative writing.* Palgrave Macmillan, Houndmills, Basingstoke, Hampshire; New York, 2010.

23 Andrew Ortony, Gerald L. Clore, and Allan Collins. *The cognitive structure of emotions*. Cambridge University Press, 1990.

24 Catherine Pelachaud. Multimodal expressive embodied conversational agents. In Hongjiang Zhang, Tat-Seng Chua, Ralf Steinmetz, Mohan S. Kankanhalli, and Lynn Wilcox, editors, *Proceedings of the 13th Annual ACM International Conference on Multimedia*, pages 683–689. Association for Computing Machinery, 2005.

25 J. Scott Penberthy and Daniel S. Weld. UCPOP: A sound, complete, partial order planner for ADL. In Bernhard Nebel, Charles Rich, and William R. Swartout, editors, *Proceedings of the 3rd International Conference on Principles of Knowledge Representation and Reasoning*, pages 103–114. Morgan Kaufmann, 1992.

26 Aaron A. Reed, Ben Samuel, Anne Sullivan, Ricky Grant, April Grow, Justin Lazaro, Jennifer Mahal, Sri Kurniawan, Marilyn Walker, and Noah Wardrip-Fruin. A step towards the future of role-playing games: The SpyFeet mobile RPG project. In Vadim Bulitko and Mark O. Riedl, editors, *Proceedings of the Seventh AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*. AAAI Press, 2011.

27 R. Reiter. A logic for default reasoning. *Artificial Intelligence*, 13(1–2):81–132, 1980.

28 Mark O. Riedl and A. Stern. Failing believably: Toward drama management with autonomous actors in interactive narratives. In Stefan Göbel, Rainer Malkewitz, and Ido Iurgel, editors, *Technologies for Interactive Digital Storytelling and Entertainment, Third International Conference, TIDSE 2006, Darmstadt, Germany, December 4–6, 2006*, number 4326 in Lecture Notes in Computer Science, pages 195–206. Springer, 2006.

29 Mark O. Riedl and R. Michael Young. Character-focused narrative generation for execution in virtual worlds. In Olivier Balet, Gérard Subsol, and Patrice Torguet, editors, *Virtual Storytelling: Using Virtual Reality Technologies for Storytelling, International Conference, ICVS 2001, Avignon, France, September 27–28, 2001, Proceedings*, number 2897 in Lecture Notes in Computer Science, pages 47–56. Springer, 2003.

30 Mark O. Riedl and R Michael Young. Open-world planning for story generation. In Leslie Pack Kaelbling and Alessandro Saffiotti, editors, *IJCAI-05, Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence, Edinburgh, Scotland, UK, July 30-August 5, 2005*, pages 1719–1720. Professional Book Center, 2005.

31 Mark O. Riedl and R. Michael Young. Narrative planning: Balancing plot and character. *Journal of Artificial Intelligence Research*, 39:164–167, 2010.

32 Stuart J. Russell and Peter Norvig. *Artificial intelligence: a modern approach*. Prentice Hall/Pearson Education, Upper Saddle River, N. J., 3rd edition, 2010.

33 Earl D. Sacerdoti. The nonlinear nature of plans. In *Advance Papers of the Fourth International Joint Conference on Artificial Intelligence, Tbilisi, Georgia, USSR, 3–8 September 1975*, pages 206–214, 1975.

34 Mei Si, Stacy C. Marsella, and David V. Pynadath. Thespian: Using multi-agent fitting to craft interactive drama. In Michal Pechoucek, Donald Steiner, and Simon Thompson, editors, *Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 21–28. Association for Computing Machinery, 2005.

35 Jason Vandenberghe. The five domains of play: Mapping the five-factor model in psychology to game design. *Game Developer Magazine*, 19(5):44–46, 2012.

36 Daniel S. Weld. An introduction to least commitment planning. *AI Magazine*, 15(4):27–61, 1994.

37 R. Michael Young. Notes on the use of plan structures in the creation of interactive plot. In M. Mateas and P. Sengers, editors, *Narrative Intelligence: Papers from the AAAI Fall Symposium*, number FS-99-01 in AAAI Technical Reports, pages 164–167, 1999.