Narrative Planning: Balancing Plot and Character

Written by: Mark O. Riedl, R. Michael Young
Published: September 2010
Presented by: Tyler Ferry

Introduction

- Planning systems generally focus on the quickest way to achieve a set of goals
- However, planners don't consider the generation of a narrative. Specifically, they don't consider character believability
- Example: Romeo and Juliet
    - Given: Montagues and Capulets fighting
    - Want: Montagues and Capulets to not feud
    - Likely solution: Kill all characters in feud

Narrative and Planning

- **Narrative:** The recounting of a sequence of events that have a continuant subject and constitute a whole
- **Story:** A narrative that has a plot
- Narratologists break narrative down into two layers of interpretation:
  - **Fabula -** List of all the events that occur in the story world between the time the story begins and the time the story ends.
  - **Sjuzet -** A subset of the fabula that is presented via narration to the audience.

General Planners

- **POCL - <u>P</u>artial <u>O</u>rder <u>C</u>ausal <u>L</u>ink** planning.
  - STRIPS-like construction consisting of an operation with a precondition and an effect

- **Fabula Planning Problem:** Given a domain theory, find a sound and believable sequence of character actions that transforms an initial world state I into a world state in which goal propositions G hold.

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: The POCL algorithm.

IPOCL Planner

```
ACTION ::= ACTION-NAME (VARIABLE*)
           actors:  VARIABLE*
           happening:  BOOLEAN
           constraints:  LITERAL*
           precondition:  LITERAL*
           effect:  LITERAL*
LITERAL := PREDICATE ([VARIABLE | SYMBOL]*)
```

- **IPOCL - I**ntent-based **P**artial **O**rder **C**ausal **L**ink planning.

- An IPOCL planning problem is a tuple, <I, A, G,Λ>, s.t. I is the initial state, A is a set of symbols that refer to character agents, G is the goal situation, and Λ is a set of action schemata.

- Extension of traditional POCL planning and works by generating fabula plans in which characters act intentionally
- Tracks and differentiates between author goals (overall plot objectives) and character goals (what the character wants)

Frames of commitment

- **<u>Frames of commitment:</u>** structures representing a character's plan to achieve a goal
    - A tuple <*S'*, *P*, *a*, g_a, s_f> s.t. *S'* is a proper subset of plan steps in a plan, *P* is a plan, *a* is a symbolic reference to a character agent, g_a is a goal that agent a is pursuing, and $s_f \in S'$ is the final step and has g_a for one of its effects
- Balances causal coherence of events with character intentionality.
- Links actions to the character's internal motivations and creates intentional, believable characters.

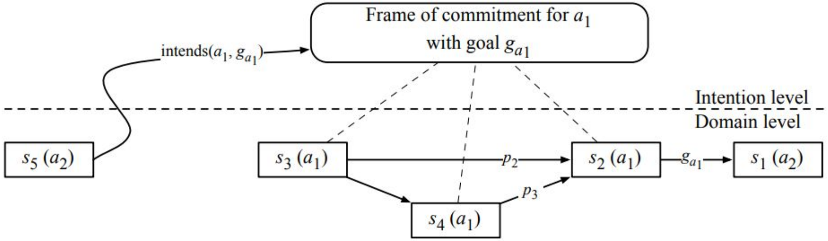


Figure 4: An IPOCL plan with a single frame of commitment and motivating step.

New Flaw Types

- **Open Motivation Flaw**:
  - A tuple, <c, p>, s.t. c is a frame of commitment in P and p is the sentence intends(a, g_a) such that a is the character of c and g_a is the internal character goal of c.
- **Intent Flaw**:
  - A tuple <s, c> where s is a step in P and c is a frame of commitment in P such that s $\xrightarrow{P}$ s_j is a causal link in the plan, s is not part of c, and s_j is a step in P, is part of c, and the character of s is the same as the character of s_j and c.
- **Intentional Threat Flaw**:
  - A tuple, <c_k, c_i>, such that frame of commitment c_k has an internal character goal that negates the internal character goal of another frame of commitment c_i

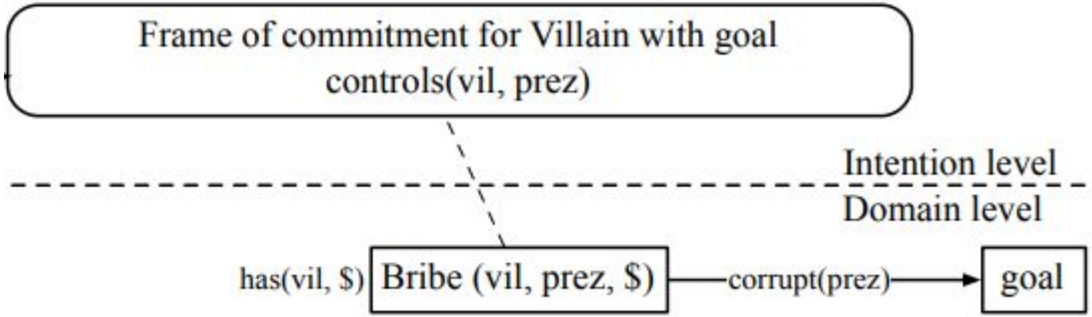
Example

- Initial:
  - *has(hero, $)*
  - *intends(vil, control(vil, prez)*
- Goal:
  - *corrupt(President)*

- Actions:
  - bribe(x, y, z)
    - Preconditions:
      - has(x, z)
    - Effects
      - corrupt(y)
      - controls(x,y)
      - has(y,z)
      - ¬has(x,z)

Example cont.

- New character called “villain/vill” created
- Bribe chosen because it has corrupt(president) effect, but requires explicit intention to do an action
    - We add open motivation flaw



Example cont.

- Add intention, removes motivation flaw.
- However, villain doesn’t have money which is prerequisite, opening a general flaw

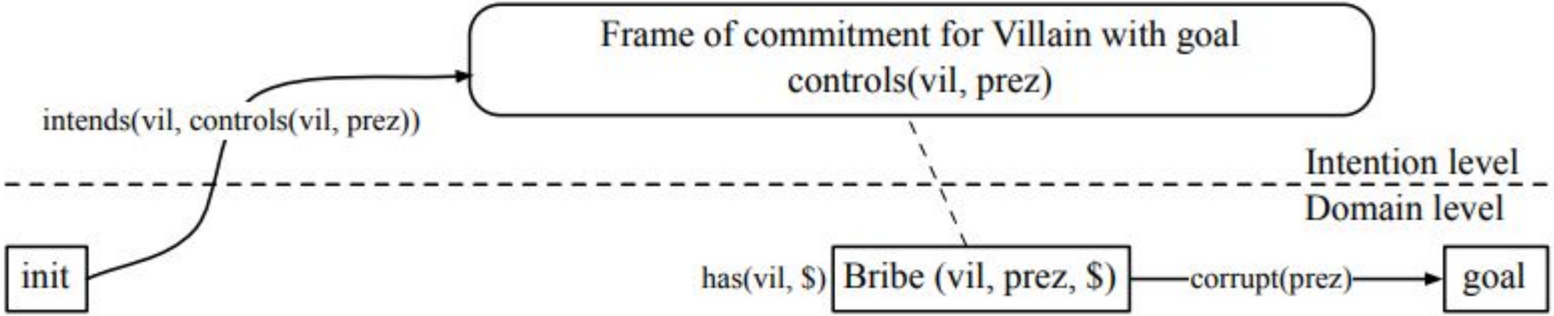


Figure 8: Example narrative plan after discovering the one action and corresponding frame of commitment.

Example cont.

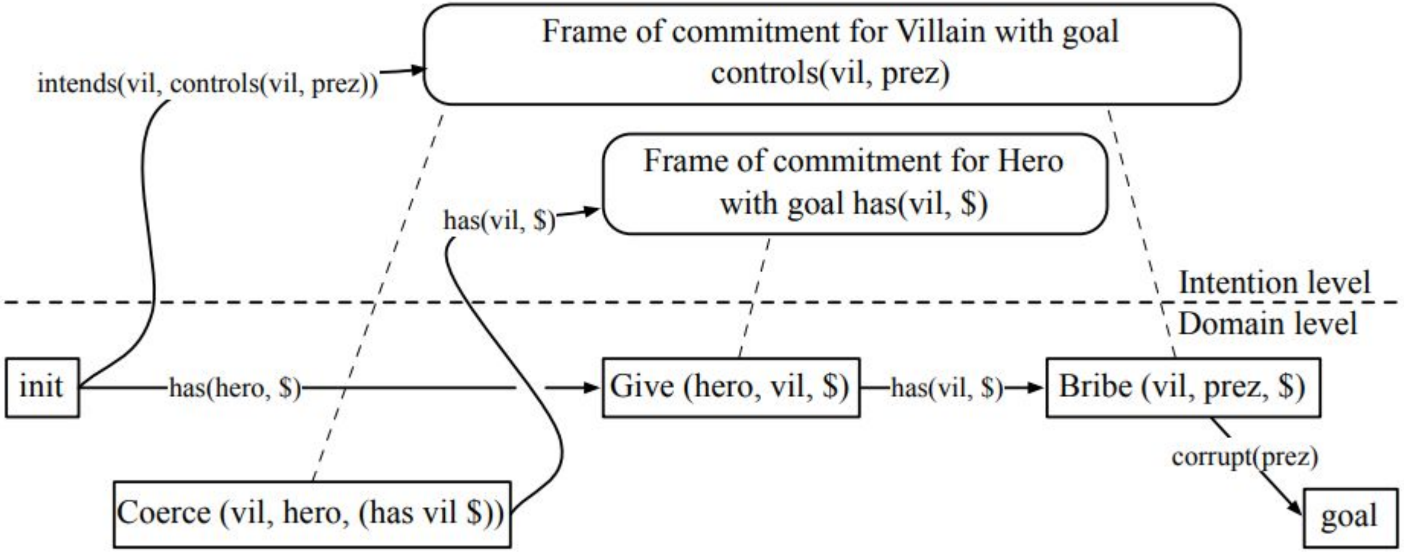


Figure 9: Solution IPOCL plan graph for the example narrative.

Complexity

- The computational complexity of the IPOCL algorithm is $\mathbf{O(c(b(e + 1)^a)^n)}$ s.t.
  - n is the depth of the search space
  - b is the number of ways that an action can be instantiated (e.g., the number of permutations of legal parameter bindings)
  - e is the number of effects of an instantiated action
  - a is the number of actors in an instantiated action

Conclusion

- Traditional planning algorithms are not sufficient for generating narratives with character intentionality with multiple agents who aren’t necessarily cooperating to achieve a singular goal state.
- IPOCL bridges gap between story causality and character believability which produces more coherent and believable stories.
- IPOCL isn’t perfect and could be extended with richer emotional models or interactive storytelling.

References

- Narrative Planning: Balancing Plot and Character https://cs.uky.edu/~sgware/reading/papers/riedl2010narrative.pdf
- Romeo and Juliet storyboard: https://www.bbc.co.uk/bitesize/guides/zxrjfrd/revision/1