A Belief-Desire-Intention Model for Narrative Generation

Theo Wadsley
School of Computer Science and Engineering
University of New South Wales, Australia

Malcolm Ryan
School of Computer Science and Engineering
University of New South Wales, Australia

Abstract

Narrative AI needs to model more than just action. In this paper, we investigate the Belief-Desire-Intention (BDI) agent architecture to allow plots to be modelled in terms of character motivation. This allows authors to focus on elements of the character model which are highly relevant to plot. We describe an extended implementation of the ConGolog agent programming language which includes BDI syntax and semantics. Using this language, we provide an example of how plot could be advantageously modelled in terms of character motivation.

Introduction

In the search for artificial intelligence (AI) methods for modelling narrative, planning techniques have been a popular source of inspiration. As Fludernik (2012) expressed, "narratives are based on cause-and-effect relationships that are applied to sequences of events." It is natural, therefore, to focus on those AI methods which model cause-and-effect relationships between actions and events. Several milestone systems have been based on this approach (e.g. Cavazza, Charles, and Mead 2002; Riedl and Young 2003; Porteous and Cavazza 2009). While causal consistency is a necessary component of narrative, it is not a sufficient one. To create narratives that are not just consistent but also coherent and engaging, planning-based narrative generation systems must consider the multiple models underlying a complete explanation of narrative (Ryan, Hannah, and Lobb 2007):

- The *action model*, which describes the world in the fundamentals terms of temporal, spatial and causal consistency,
- The *character model*, which describes characters' psychologies to address the coherency of their actions in terms of emotions and motivations, and
- The *plot model*, which describes the dramatic purpose of characters and events in relation to the overall story.

Most existing systems do attempt to model all three levels, however their plot models are limited. Plot structure must be expressed in terms of abstract 'drama values' (Weyhrauch 1997; Mateas and Stern 2005; O'Neill and Riedl 2011) or trajectory constraints over the primitives of action and state (Riedl and Young 2003; Porteous and Cavazza 2009; Riedl and Young 2010). Both approaches fail to directly address the roles of characters' internal emotion and motivation in the plot. As Fludernik further expresses, "it is the experience of these protagonists that narratives focus on, allowing readers to immerse themselves in a different world and in the life of the protagonists."

We proposes a planning-based narrative system that uses character motivation as a first class object. Plot is expressed in terms of trajectory constraints over action, state, desire and intention. The model extends the ConGolog language (De Giacomo, Lespérance, and Levesque 2000) with a Belief-Desire-Intention (BDI) architecture (Bratman 1987). The paper will first cover relevant ideas in both narrative generation and artificial intelligence. The subsequent sections will detail our model and give an example of usage.

Background

Narrative Generation

Planning-based narrative generation systems have clear advantages over purely reactive ones. Reactive narrative generation systems (Meehan 1977; Bates, Loyall, and Reilly 1994; McCoy, Mateas, and Wardrip-Fruin 2009) are inevitably dependent on emergent plot. This presupposes that the character model is sufficient to produce engaging narrative. Combining reactive scenes into an overall plan (Weyhrauch 1997; Theune et al. 2004) only partially mitigates this problem. Complex plot modelling requires control over actions and characters at any level of detail. Only planning-based approaches provide this level of control.

An effective character model must consider internal state in terms of motivation and emotion. Creating a general model of character is likely intractable, so responses must be tailored to some degree. Psychological theories have provided insight into modelling character emotion (Theune et al. 2004; Sarlej and Ryan 2012), but the main focus of this paper is on motivation. Character models of motivation are frequently goal-driven and plan-based (Cavazza, Charles, and Mead 2002; Charles et al. 2003; Riedl and Young 2003). Character desire is defined by the set of goals they wish to achieve. Plans are the authored programs a character uses in response to achieve their goals. This approach is an effective compromise between character complexity and generality.

The plot model ties the action and character models into a coherent narrative. Significant research has been devoted to enhancing plot models in narrative generation. The most basic planning approach is to produce a narrative that achieves some goal state (Cavazza, Charles, and Mead 2002; Charles et al. 2003; Riedl and Young 2010). This plot model is clearly unsuitable for longer stories. Optimising abstract 'drama values' has seen some use in plot modelling (Barber and Kudenko 2009; Li and Riedl 2011; O'Neill and Riedl 2011). Nevertheless, producing a complete computational model of drama is likely intractable. Plot models defining state trajectory constraints (Roberts et al. 2006; Porteous and Cavazza 2009) provide the greatest narrative control, however this approach remains grounded in the action model. Sarlej and Ryan's (2012) representation of morals in terms of emotional trajectories is exemplary of plot modelling at the character level. We believe this approach can be applied to models of character motivation.

Golog

Golog (Levesque et al. 1997) is a high level logic programming language suitable defining planning structures. Golog programs define a non-deterministic grammar over states and action with a regular expression-like syntax. Plans are highly structured but allow multiple execution paths. This approach provides control and flexibility useful to automated narrative generation. ConGolog (De Giacomo, Lespérance, and Levesque 2000) extends Golog to allow interleaved concurrent execution. ConGolog can be used to define flexible character scripts for generated narratives. However it still only allows us to describe plans in terms of world primitives of action and state. We need the ability to write programs in terms of character desires and intentions.

Belief-Desire-Intention

The Belief-Desire-Intention (BDI) program architecture models reactive agents based on Bratman's (1987) theory of human reasoning. Agent internal state is completely defined in terms of beliefs, desires and intentions. *Belief* models the agents subjective knowledge of the world. *Desire* models an agent's abstract goals, while *intention* models how those goals are translated into action. BDI is a commonly used agent model in AI and serves as a basic model of character motivation. AgentSpeak(L) (Rao 1996) is a computable BDI language that provides the framework for our BDI character model. AgentSpeak(L) provides a computational model suitable for representing character action in terms of motivation. Nevertheless, AgentSpeak(L) presents a few limitations for narrative generation purposes: it is purely reactive, and plan bodies are simple linear procedures. Implementing an AgentSpeak(L)-like system in ConGolog overcomes both of these limitations.

Belief-Desire-Intention in ConGolog

The BDI model is an effective method to discuss character desires and intentions as first class objects. However as a purely reactive architecture it is unsuitable for plot modelling. By adding BDI syntax and semantics to the planning-based ConGolog language this limitation can be overcome. The system presented models each agent with a separate program. The following definitions are ground to a given agent a, and each agent program will be executed in parallel. The BDI extensions are defined in terms of the $Final$ and $Trans$ predicates underlying ConGolog semantics. $Trans$ indicates how a program state transitions in a single execution step, while $Final$ indicates that a program state is terminal.

Plans

Plans are modelled off those in AgentSpeak(L), associating a desire with a plan context and body. A program is a block statement containing a sequence of plan and procedure definitions, followed by a **respond** command for each distinct belief event $\pm b$ concurrently executed with the main program body δ_0.

$$\begin{gathered}\mathbf{plan}\, e_1 : \phi_1 \leftarrow \delta_1\, \mathbf{end}\, ; \ldots ; \mathbf{plan}\, e_n : \phi_n \leftarrow \delta_n\, \mathbf{end}\, ; \\ \mathbf{proc}\, p_1(\vec{u_1})\gamma_1\, \mathbf{end}\, ; \ldots ; \mathbf{proc}\, p_m(\vec{u_m})\gamma_m\, \mathbf{end}\, ; \\ \mathbf{respond}(\pm b_1) \parallel \ldots \parallel \mathbf{respond}(\pm b_k) \parallel \delta_0.\end{gathered}$$

Beliefs

The current implementation assumes an omniscient belief model. The implementation of a more advanced belief model will be the subject of future work.

$$Believes(a, \phi, s) \equiv Holds(a, \phi, s).$$

Desires

An event e is encapsulated by the **desire** command until formed into an intention. An agent desires e if $\mathbf{desire}(e)$ is an accessible command of the current program.

$$\begin{aligned}&Final(a, \mathbf{desire}(e), s) \equiv False.\\ &Trans(a, \mathbf{desire}(e), s, \mathbf{intention}(e, \delta), s) \equiv \\ &\exists p, \theta \,.\, p \in P \wedge p = \mathbf{plan}\, e' : \phi \leftarrow \delta\, \mathbf{end} \wedge \\ &\qquad e\theta = e'\theta \wedge Believes(a, \phi\theta, s).\end{aligned}$$

Intentions

The **intention** command associates an event with its response. If the main program body is empty or contains only goal statements then every action will be encapsulated by an **intention** command. Hence every action an agent performs can be associated with the intention it fulfils.

$$\begin{aligned}&Final(a, \mathbf{intention}(e, \delta), s) \equiv Final(a, \delta, s).\\ &Trans(a, \mathbf{intention}(e, \delta), s,\\ &\qquad \mathbf{intention}(e, \delta'), s') \equiv Trans(a, \delta, s, \delta', s').\end{aligned}$$

Achievement Goals

Achievement goals are the simplest type of events. When declared by command they transition straight into a desire.

$$Final(a, !g, s) \equiv False.$$
$$Trans(a, !g, s, \mathbf{desire}(+!g), s) \equiv True.$$

Test Goals

Test goals are implemented as a semantic extension of ConGolog's test actions. If the condition ϕ is satisfiable they retain the original ConGolog semantics and transition in no time. In the alternative case they form a desire.

$$Final(a, \phi?, s) \equiv False.$$
$$Trans(a, \phi?, s, \mathbf{desire}(+\phi?), s) \equiv \neg Believes(a, \phi?, s).$$
$$Trans(a, \phi?, s, nil, s) \equiv Believes(a, \phi?, s).$$

Belief Events

Events responding to changing beliefs are implemented through the **respond** command. A response to $+b$ or $-b$ may form into a desire whenever the belief b holds or does not hold respectively. Each **respond** command admits only a single desire or intention at a given time. An agent may only respond to the same belief event again when its original intended response is complete.

$$Final(a, \mathbf{respond}(\pm b), s) \equiv True.$$
$$Trans(a, \mathbf{respond}(+b), s, \mathbf{desire}(+b); \mathbf{respond}(+b), s) \equiv Believes(a, b, s).$$
$$Trans(a, \mathbf{respond}(-b), s, \mathbf{desire}(-b); \mathbf{respond}(-b), s) \equiv \neg Believes(a, b, s).$$

Example

The proposed system is implemented using Answer Set Programming (ASP) (Lifschitz 2002), and solved using the Potsdam Answer Set Solving Collection (Gebser et al. 2011). The language extends a ConGolog implementation in the Gringo language, the details of which will be described in a future publication. To demonstrate the use of this system, consider the following world.

The world has four mutually accessible locations: a *house*, a *burrow*, a *farm*, a *shed* and some *bushes*. There exist two characters: a *farmer* in the house and a *rabbit* in the burrow. The rabbit is *safe* in the burrow. The farmer and rabbit are both *hungry*. There are three potential sources of food: *vegetables* on the farm and *berries* in the bushes that both characters can eat, and the farmer can eat the rabbit.aCharacters can *go to* a location, *pick up* a co-located object, *drop* a carried object, or *eat* carried food.

If an agent is hungry and knows the location of some food, make a goal to get the food then eat it if successful. If an agent wants to get a thing: if it is an inanimate object, go to where it is and pick it up; if it is a target character that wants to get some food, go to where the food is and wait for the target character to come by. Then chase them down. If an agent wants to chase some target character: if the target is safe, give up; if the target is right next to the agent, pick them up; if the target is elseswhere, follow them. If another character intends to chase the agent, flee the location of the pursuer. If an agent wants to flee a location: if they're not at that location, don't do anything; if they're at that location, go to any other location.

Note that given the current definition a narratively uninteresting outcome is quite possible. The farmer, being hungry, could go to the farm and eat the vegetables. The rabbit, being hungry, could go to the bushes and eat the berries. Neither character having any more desires, the story would end. For example, a better story would involve the rabbit trying to acquire food but failing. This could be achieved in the character model by making the farmer only able to eat rabbit. However this solution is unnecessarily forced as there is no reason that the farmer could not eat vegetables or berries if necessary. Instead, this constraint should be imposed by the plot model. With this model our system generates the following story using the plot constraint that the rabbit begins with the intention to acquire food, and finishes hungry and without the intention to acquire food. The current system lacks a text generation capability. However using fairly simple language the generated narrative could be translated into the following story:

> The rabbit is hungry so it decides to get the vegetables. The farmer is hungry so she decides to get the rabbit. To get the rabbit the farmer goes to the farm. To get the vegetables the rabbit goes to the farm. The farmer wants to chase the rabbit, so the rabbit decides to flee. To flee the farm the rabbit goes to the shed. The rabbit gives up trying to get the vegetables. To chase the rabbit the farmer goes to the shed. To flee the shed the rabbit goes to the burrow. The farmer gives up chasing the rabbit. The rabbit is still hungry.

This example demonstrates how character motivation can be used to structure plot and enhance the story text.

Conclusion

AI planning techniques have proved well suited to narrative generation. Plot modelling has seen significant improvement using planning-based approaches. However plot models remain limited by their lack of focus on the character model. Combining AI agent models with planning approaches has the potential to bridge this gap.

This paper proposed a BDI extension to the ConGolog language for narrative generation. Its intention was to treat character motivation as a first class object when modelling plot. It is hoped that creating trajectory constraints over character motivations will prove to be a more natural plot modelling method. It is the intention of future work to extend this system with models of belief, emotion, and plan failure, and to test how authors evaluate this system compared to alternative plot modelling approaches.

References

Barber, H., and Kudenko, D. 2009. Generation of Adaptive Dilemma-Based Interactive Narratives. *IEEE Transactions on Computational Intelligence and AI in Games* 1(4):309–326.

Bates, J.; Loyall, A. B.; and Reilly, W. S. 1994. An Architecture for Action, Emotion, and Social Behavior. In Castelfranchi, C., and Werner, E., eds., *Artificial Social Systems*. Berlin, Heidelberg: Springer Berlin Heidelberg. 55–68.

Bratman, M. 1987. *Intention, Plans, and Practical Reason*. Cambridge, MA: Harvard University Press.

Cavazza, M.; Charles, F.; and Mead, S. J. 2002. Interacting with Virtual Characters in Interactive Storytelling. In *Proceedings of the First International Joint Conference on Autonomous Agents & Multiagent Systems*, 318–325. Bologna, Italy: ACM.

Charles, F.; Lozano, M.; Mead, S. J.; Bisquerra, A. F.; and Cavazza, M. 2003. Planning Formalisms and Authoring in Interactive Storytelling. In *Proceedings of the 1st International Conference on Technologies for Interactive Digital Storytelling and Entertainment*, 216–225.

De Giacomo, G.; Lespérance, Y.; and Levesque, H. J. 2000. ConGolog, a concurrent programming language based on the situation calculus. *Artificial Intelligence* 121(1–2):109–169.

Fludernik, M. 2012. *An Introduction to Narratology*. Hoboken, NJ: Taylor and Francis, 1st edition.

Gebser, M.; Kaufmann, B.; Kaminski, R.; Ostrowski, M.; Schaub, T.; and Schneider, M. 2011. Potassco: The Potsdam Answer Set Solving Collection. *AI Communications* 24(2):107–124.

Levesque, H. J.; Reiter, R.; Lespérance, Y.; Lin, F.; and Scherl, R. B. 1997. GOLOG: A logic programming language for dynamic domains. *Journal of Logic Programming* 31(1–3):59–83.

Li, B., and Riedl, M. O. 2011. Creating Customized Game Experiences by Leveraging Human Creative Effort: A Planning Approach. In Dignum, F., ed., *Agents for Games and Simulations II*. Berlin, Heidelberg: Springer Berlin Heidelberg. 99–116.

Lifschitz, V. 2002. Answer Set Programming and Plan Generation. *Artificial Intelligence* 138(1–2):39–54.

Mateas, M., and Stern, A. 2005. Structuring content in the Façade interactive drama architecture. In *Proceedings of the First Artificial Intelligence and Interactive Digital Entertainment Conference*, 93–98.

McCoy, J.; Mateas, M.; and Wardrip-Fruin, N. 2009. Comme il Faut: A System for Simulating Social Games Between Autonomous Characters. In *Proceedings of the 8th Digital Art and Culture Conference*.

Meehan, J. R. 1977. TALE-SPIN, an interactive program that writes stories. In *Proceedings of the 5th International Joint Conference on Artificial Intelligence*, 91–98. Cambridge, MA: Morgan Kaufmann Publishers Inc.

O'Neill, B., and Riedl, M. 2011. Toward a Computational Framework of Suspense and Dramatic Arc. In D'Mello, S.; Graesser, A.; Schuller, B.; and Martin, J.-C., eds., *Affective Computing and Intelligent Interaction*. Berlin, Heidelberg: Springer Berlin Heidelberg. 246–255.

Porteous, J., and Cavazza, M. 2009. Controlling Narrative Generation with Planning Trajectories: The Role of Constraints. In Iurgel, I. A.; Zagalo, N.; and Petta, P., eds., *Interactive Storytelling*. Berlin, Heidelberg: Springer Berlin Heidelberg. 234–245.

Rao, A. S. 1996. AgentSpeak(L): BDI Agents Speak Out in a Logical Computable Language. In *Proceedings of the 7th European Workshop on Modelling Autonomous Agents in a Multi-Agent World*, 42–55. Eindhoven, The Netherlands: Springer-Verlag.

Riedl, M. O., and Young, R. M. 2003. Character-Focused Narrative Generation for Execution in Virtual Worlds. In Balet, O.; Subsol, G.; and Torguet, P., eds., *Virtual Storytelling*. Berlin, Heidelberg: Springer Berlin Heidelberg. 47–56.

Riedl, M. O., and Young, R. M. 2010. Narrative Planning: Balancing Plot and Character. *Journal of Artificial Intelligence Research* 39:217–268.

Roberts, D. L.; Nelson, M. J.; Isbell, C. L.; Mateas, M.; and Littman, M. L. 2006. Targeting Specific Distributions of Trajectories in MDPs. In *Proceedings of the 21st National Conference on Artificial Intelligence*, 1213–1218. Boston, Massachusetts: AAAI Press.

Ryan, M.; Hannah, N.; and Lobb, J. 2007. The Tale of Peter Rabbit: A Case-Study in Story-Sense Reasoning. In Magerko, B. S., and Riedl, M. O., eds., *AAAI 2007 Fall Symposium on Intelligent Narrative Technologies*, 135–138. Arlington, VA: Menlo Park, California.

Sarlej, M. K., and Ryan, M. 2012. Representing Morals in Terms of Emotion. In *Proceedings of the Eighth Annual AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*.

Theune, M.; Rensen, S.; Akker, R.; Heylen, D.; and Nijholt, A. 2004. Emotional Characters for Automatic Plot Creation. In Göbel, S.; Spierling, U.; Hoffmann, A.; Iurgel, I.; Schneider, O.; Dechau, J.; and Feix, A., eds., *Technologies for Interactive Digital Storytelling and Entertainment*. Berlin, Heidelberg: Springer Berlin Heidelberg. 95–100.

Weyhrauch, P. W. 1997. *Guiding Interactive Drama*. Ph.D. Dissertation, Carnegie Mellon University.