Proceedings, The Eighth AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment

# Plan-Based Character Diversity

**Alexandra Coman, Héctor Muñoz-Avila**
Department of Computer Science and Engineering, Lehigh University, Bethlehem, PA 18015
alc308@lehigh.edu | hem4@lehigh.edu

**Abstract**

Non-player character diversity enriches game environments increasing their replay value. We propose a method for obtaining character behavior diversity based on the diversity of plans enacted by characters, and demonstrate this method in a scenario in which characters have multiple choices. Using case-based planning techniques, we reuse plans for varied character behavior, which simulate different personality traits.

## Introduction

Non-player character (NPC) diversity renders games more compelling and engaging, and increases their replay value. Large game environments populated by numerous computer-controlled characters become more realistic if these characters exhibit varied behavior.

Diversity can be obtained by modeling characters in terms of personalities, morals, emotional states, and social affinities. Such character modeling has been explored extensively (Cavazza, Charles, and Mead, 2002; Cavazza and Charles, 2005; Strong *et al*., 2007; McCoy *et al*., 2010; Cavazza *et al*., 2009; Thue *et al*., 2010).

We propose an alternative method for creating diverse characters: behavior diversity, where character behavior is represented as plans, i.e., sequences of individual actions. The knowledge requirements specific to this method are (1) a distance metric measuring the dissimilarity between behavior plans and (2) the means to generate such plans. Ours is a bottom-up approach to attaining character diversity: while character modeling endows characters with traits which determine their behavior (a top-down approach), we create diversity in terms of behavior and, as different types of behavior reflect different personality traits, this simulates personality variation. As behavior is represented in terms of plans, our approach to character diversity is based on *plan diversity*. The basic premise of our work is the availability of a plan repository (e.g. collected from logs of players' actions) and of a user-defined plan comparison criterion (a *plan distance metric*). We test our ideas in a scenario in which characters have multiple choices. We demonstrate how our techniques (exemplified herein using case-based planning) create varied character behavior typical of different personality traits.

## Character Diversity

We propose to obtain character diversity based on the diversity of plans acted out by characters, where plans[1] are represented as sequences of actions, such as <*move to location 1*><*attack enemy soldier*><*collect treasure*>. Such plans, made up of unitary actions that a character can execute, can be created easily based on players' gameplay traces. This in contrast to knowledge engineering requirements of techniques such as HTN planning (Paul *et al*., 2010) or goal-oriented action planning (Orkin, 2003), both of which require knowledge about the semantics of individual actions a character can execute (usually described in terms of the actions' preconditions and effects). HTN planning also requires defining task decomposition knowledge.

A character's behavior, represented as plans, reflects the character's personality traits, hence plan diversity can be the basis for exhibiting character diversity.

We exemplify our approach to creating character diversity in a scenario in which characters are endowed with abilities such as attack strength, technical skill, endurance, etc., but they do not have individual personalities modeled in terms of temperament, morals, etc. With plan diversity, we can, to some extent, simulate character personality variety, which is a crucial component of many game genres, including role-playing games (RPGs).

Assume we are generating plans to be acted out by characters in an RPG game scenario. In terms of unit category, each character can be a *peasant*, *soldier*, *archer* or *mage* (unit category determines *abilities*, but not *personality* and actual behavior). The map (Figure 1) contains buildings and units belonging to a friendly side and an enemy side as



[1] The described methods can also be used for characters operated by finite-state machines, as finite state machines also specify low-level actions which the character should execute.

well as neutral ones, including: a treasure, an enemy town hall, an abandoned guard tower, and a friendly peasant who is trapped in an enclosed space protected by a strong enemy orc. With regard to his/her attitude towards collecting the treasure, a character can be *greedy* or *not greedy*; with regard to his/her attitude toward the friend in need, the character can be *helpful* or *indifferent*. Furthermore, a helpful character can behave in a *reckless* or an *intelligent* manner in approaching their friend's rescue.

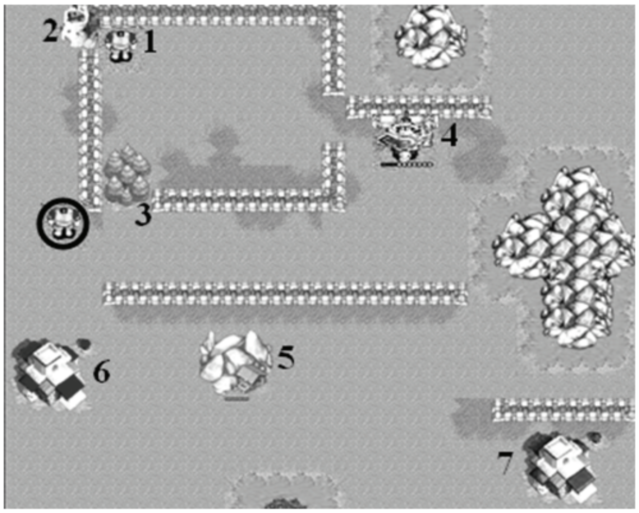


Fig. 1. Example Scenario. Our character is highlighted. Other map elements of interest are: (1) friend in need, (2) old guard tower, (3) trees blocking the escape path, (4) ruthless enemy orc, (5) treasure, (6) our town hall, (7) enemy town hall.

A character is *greedy* if s/he collects the treasure or loots the poorly-guarded enemy town hall (the designation of this behavior as "greedy" is a choice which can change from story to story: one might instead choose to consider collecting the treasure as indicative of diligence or good citizenship); *helpful* if s/he attempts to save the captive friend. A helpful character might be considered *reckless* if s/he attacks the unbeatable enemy orc, and *intelligent* if s/he frees the friend by either cutting down trees or demolishing the old guard tower to make a path for the friend to escape by. Any other actions carried out in the plan are considered irrelevant for the purposes of this classification (e.g. it does not matter if our character takes a shortcut or the long way to the treasure). *Greed*, *bravery* and *intelligence* are, therefore, the traits along which we will create plan variation. A player exploring the game world populated by such character types would now encounter NPCs which, while being part of the same unit category (e.g. *peasants*), exhibit varied behavior simulating personality differences.

## Generating Diverse Plans

In case-based planning (Spalazzi, 2001), solution plans for a new problem are generated by retrieving already-existing plans from a repository called a case base, and adapting the retrieved plans so as to match the new problem. The case base consists of a set of cases, i.e. (problem, solution plan) pairs. Retrieval is usually conducted based on a *similarity criterion*: the retrieved cases are the ones whose problems are assessed as most similar to the new problem. The underlying assumption is that similar cases will yield similar solutions through the adaptation process. Hence, if the source cases encode certain behavior trait, the adapted solutions can be reasonably expected to exhibit that same behavior trait.

In order to obtain sets of diverse plans, the *problem similarity* criterion needs to be supplemented with a *solution plan diversity* criterion. The problem of balancing the two potentially contradictory retrieval criteria has been extensively explored in case-based reasoning (Smyth and McClave, 2001; Shimazu, 2001; McSherry, 2002; McGinty and Smyth, 2003).

The pseudocode below shows the algorithm we use for diverse plan generation (Coman and Munoz-Avila, 2011b), which retrieves solution plans based on a composite criterion addressing both problem similarity (***Sim(*****c,Prob*****)*** in Step 5 of the algorithm) and solution plan diversity (***RelDiv*****(c.π,C.Π)**, Formula 1, in Step 5 of the algorithm).

**DiversePlanGeneration**(*Prob,k,CB, D*)
// *Prob* – problem; *k* – number of diverse plans to be //generated; *CB* – case base; *D* – distance metric;
// *c*.π – plan of case c; *C*.Π – set of plans in set of cases *C*;
// *Sim* – meassure of the similarity between case *c* and //problem *Prob*; *RelDiv* – Formula 1; *α* – parameter which //allows varying the weights assigned to the problem //similarity and plan diversity criteria.

1. C ← {}
2. c ← case in CB which maximizes ***Sim*****(c, Prob)**
3. **Add** c to C
4. **Repeat**
5. c ← case in CB which maximizes ***αSim(*****c, Prob*****)+ (1- α)RelDiv(*****c.π,C.Π*****)***
6. **Add** c to C
7. **Until** |Π| = k plans have been generated
8. Π' ← adaptations of plans in C.Π
9. **Return** Π'

We are given a planning problem, a case base, and a distance metric *D* measuring how diverse plans differ from one another.

Diverse case-based planning is conducted by first selecting the case in the case base whose problem is most similar to the new problem (Step 2), then repeatedly selecting a case which maximizes the retrieval criterion incorporating

*RelDiv* (relative diversity) between the plan in the candidate case and the plans in the previously retrieved cases (Step 5). This criterion is defined in Formula 1 (Smyth and McClave, 2001; Coman and Muñoz-Avila 2011a and b), where π is a plan, Π is a non-empty plan-set, and $D$: $\Pi \times \Pi \rightarrow [0,1]$ is a plan distance metric (a measure of the dissimilarity between two plans, Coman and Muñoz-Avila, 2011a and b). Any ties are broken by random selection. The solution plans of the retrieved cases are then adapted.

$$RelDiv(\pi, \Pi) = \frac{\sum_{\pi' \in \Pi} D(\pi,\pi')}{|\Pi|} \quad (1)$$

*RelDiv* (Formula 1) can be used with any distance metric, including *quantitative* distance metrics, which are computed using domain-independent plan features, such as the percentage of actions occurring in either plan but not in the other (Srivastava *et al*., 2007); and *qualitative* distance metrics (Coman and Muñoz-Avila 2011a and b), which are computed based on domain-specific information (e.g. for a real-time strategy game planning domain, one might consider distance *d* to describe the dissimilarity between a *mage* attack and a *soldier* attack, and distance *d*' to describe the dissimilarity between a *mage* attack and an *archer* attack, specifying that *d*' < *d*, since *mages* and *archers* in real-time strategy games are ranged units, while soldiers are melee units; the qualitative distance metric defined for this domain would incorporate *d* and *d*').

Character diversity can be obtained through qualitative distance metrics that encode indicators of various character personality traits such as greed and devotion to friends, as described in the previous section.

## Example

To generate diverse plans for the example scenario introduced in the previous section, assume the distance metric in Formula 2, where π and π' are plans, and *CharType*(π) is the *type of character* (not to be confused with the unit category: unit categories are *archer*, *soldier*, etc.) reflected in plan π, while *d* is a degree of distance between possible character types. The six character types used for exemplification are: (1) *greedy* and *indifferent*, (2) *greedy* and *intelligent*, (3) *greedy* and *reckless*, (4) *not greedy* and *indifferent*, (5) *not greedy* and *intelligent*, (6) *not greedy* and *reckless*. Formula 2 is extensible to include additional character types.

$$D_{Char}(\pi, \pi') = \begin{cases} 0, if\ CharType(\pi) = CharType(\pi') \\ d, 0 < d \leq 1, if\ CharType(\pi) \neq CharType(\pi') \end{cases} \quad (2)$$

The problem description consists simply of the *unit category* (this information is included in the initial state), with the 4 categories being: *peasant*, *soldier*, *archer*, and *mage*. Assume all 4 unit categories are represented in the case base. In addition to the problem description specifying a unit category, each case contains a plan that the unit is able to execute (e.g. for a *peasant* unit, the plan might specify moving to a certain location on the map and cutting down trees, then moving to another location and collecting a treasure). There are differences between units in terms of the ways in which they are able to manifest certain character traits (e.g. since no units but *peasants* are able to harvest treasure, *soldiers*, *archers* and *mages* manifest their greed by attacking the enemy town hall).

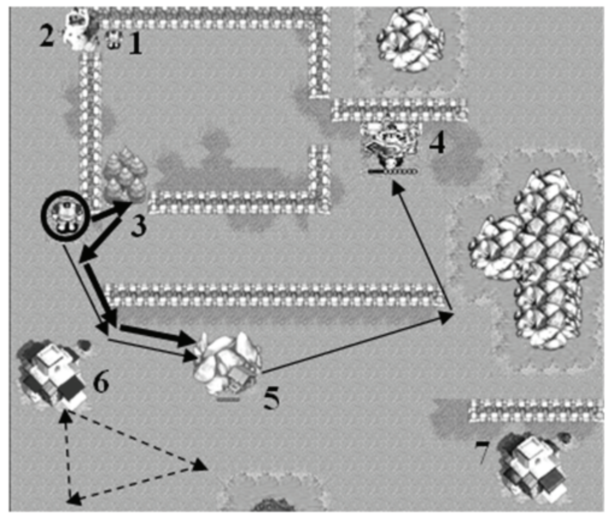


Fig. 2. Potential plans that a character can execute

Assume we want to generate 3 diverse behavior plans for a *peasant* unit (Figure 2). First, we select a case from the case base using solely the similarity criterion, i.e. the new problem description should match the case problem description, therefore the acting unit in the case should be a *peasant*[2]. Out of the retrieved *peasant* cases, we pick a case $c_1$ randomly, since our retrieval criterion is based on unit category only, and ties are broken by random selection. Assume that $c_1.\pi$, the plan of the picked case, specifies that our character should *collect* the nearby *treasure*, then *attack* the enemy *orc*, in an attempt to save their friend (the plan shown by the thin arrows in Figure 2). According to our description, this plan reflects *greedy* and *reckless* behavior.

[2] A more complex similarity criterion might specify degrees of similarity between units, e.g. *archers* being more similar to *mages* than they are to *soldiers*.

We now select a case $c_2$ based on the composite criterion combining problem similarity and plan diversity with respect to $c_1.\pi$. Based on the definition of $d$ in Formula 2, the selected plan, $c_2.\pi$, is a *not greedy* and *indifferent* character's plan: our peasant spends his/her time roaming the countryside, attacking irrelevant map locations or visiting the town hall, ignoring both the friend's predicament and the tempting treasure (see dashed arrows in Figure 2).

Finally, we select a case $c_3$, another *peasant* case, the plan of which should be as dissimilar as possible from both $c_1.\pi$ and $c_2.\pi$. Again, any ties are broken by random selection. The selected plan could be a *not greedy* and *intelligent* one, or a *greedy* and *intelligent* one (see thick arrows in Figure 2 for the latter). We now have a set of three plans reflecting meaningfully different character behavior. Of course, such a successful selection is conditioned by the availability of plans corresponding to all these character types in the case base, and by the adequacy of the plan distance criterion when it comes to capturing meaningful behavior-related differences between plans. In an actual gameplay session, on an extended map, the three plans above might be acted out by three *peasant* characters with different personalities.

## Experimental Evaluation

### Experimental Setup

We showcase character diversity based on diverse plans using the Wargus real-time strategy game engine, which we use for the purposes of demonstrating character trait diversity (Wargus does not normally exhibits it). Our scenario is carefully crafted towards testing these character traits, instead of the typical "harvest, build, destroy" gameplay generally characteristic of real-time strategy games.

To retrieve diverse plans, we use the algorithm described in the *Generating Diverse Plans* section (referred to as DivCBP) with the distance metric $D_{\text{Char}}$ (Formula 2) and $\alpha$=0.5 (in Formula 1). Retrieved plans are ran in the game as they are, as adaptation is not necessary. The casebase contains 100 cases for the 4 unit categories (*peasant*, *soldier*, *archer*, and *mage* cases). While certain case base plans are significantly different in terms of character behavior, a lot of them vary in ways which do not make for different character types (e.g. taking different paths to the same target map locations).

The possible *game outcomes* which are recorded are: the amount of gold collected, the amount of gold not collected; if the character's friend is free, if the character's friend is captive and the character is alive, and if the character is dead.

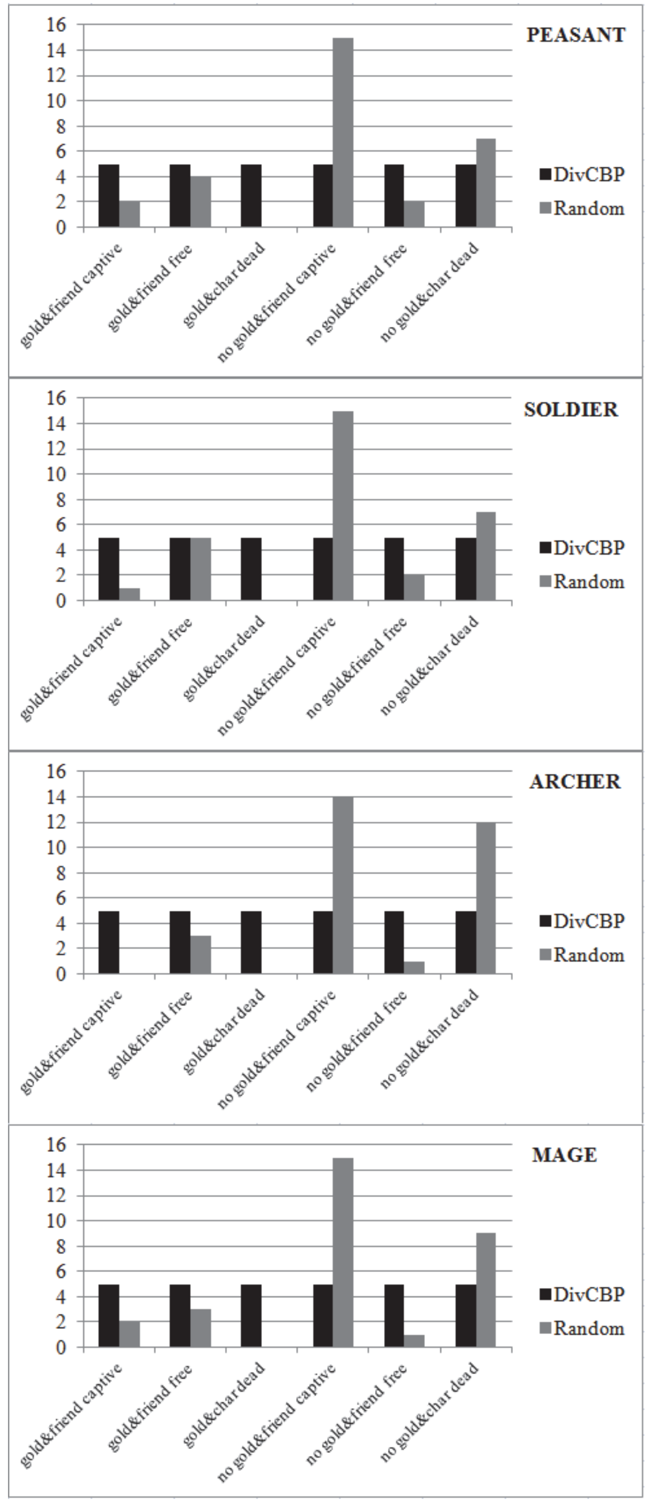


Fig 3. Number of outcomes of each type for each of the four unit categories

An alternative way of attempting to create diverse plans, without any knowledge engineering at all, is to select plans from the case base randomly. However, as we will show, this does not guarantee significant diversity, as reflected in game outcomes. By recording and comparing these outcomes, we can determine whether the diverse plans we generate actually lead to noticeably different gameplay.

For each unit category, we generate and run in the game 5 sets of plans (of 6 plans each). There are 30 gameplay sessions per unit, in all; and 30 corresponding in-game outcomes recorded.

### Experimental Results

In Figure 3, the bars represent the number of outcomes of each type for DivCBP and the baseline Random selection. The closer the bars corresponding to one method are to each other in terms of height, the more spread out the results are over the possible outcomes, indicating greater gameplay variety. As can be seen, DivCBP results are consistently better spread out over the possible outcomes than Random results.

The entropy of DivCBP results is consistently greater than that of Random results (1,1,1,1 vs. 0.73; 0.71; 0.59; 0.69), indicating that, when using DivCBP, the uncertainty regarding the character type that will be chosen is greater; with random selection, certain character types tend to be favored over others.

This indicates that selection based on DivCBP more reliably produces diverse character types (the diversity of which is reflected in different in-game outcomes) than Random selection.

## Related Work

Iszita, Ponsen , and Spronck. (2009) study diversity in the context of a dynamic scripting framework in which macros are learned to generate adaptive game AI. Macros are described through rules from a manually-created rule base. The fitness function (which determines the suitability of macros) incorporates criteria about which rules were used in other learned macros. The aim is for the new macro to use rules not used in other learned macros and, in this way, attain diversity. In contrast, we attain diversity by using metrics that capture meaningful character differences as defined by the user. In addition, we do not assume the availability of knowledge about the means to generate the sequence of actions (i.e., a rule base).

Goal-oriented action planning is a paradigm that advocates modeling character behavior by defining characters' simple activities using AI planning actions (Orkin, 2003). An AI planner is used to generate plans corresponding to types of behavior that combine individual activities. While we consider plans to be given (e.g. by observing players' logs), it is conceivable that the same principles presented here could be used in a goal-oriented action planning setting as illustrated in Coman and Munoz-Avila (2011b).

Complex character modeling has been explored to such a great extent that, rather than attempt exhaustiveness, we will cite several illustrative examples. NPCs have been modeled in terms of personality traits (McCoy *et al*., 2010), emotional states (Cavazza and Charles, 2005; Strong *et al*., 2007; Cavazza *et al*., 2009), relationships (Cavazza and Charles, 2005; McCoy *et al*., 2010; Thue *et al*., 2010), and needs (McCoy *et al*., 2010; Paul *et al*., 2010), among others. In the related field of behavioral robotics, emotions are modeled for Sony's AIBO robots Arkin *et al*. (2003). Substantial work deals with the complexities of generating interesting and realistic language/dialogue that is both character- and context-dependent (Cavazza and Charles, 2005; Strong *et al*., 2007; Lin and Walker, 2011). Another significant area of focus is NPC behavior in the context of storyline consistency and variety (Cavazza, Charles, and Mead, 2002; Porteous, Cavazza, Charles, 2010; McCoy *et al*., 2010; Thue *et al*., 2010). Several of these approaches to character modeling use HTN planning (Cavazza, Charles, and Mead, 2002; Cavazza and Charles, 2005; Paul *et al*., 2010) or other types of task-decomposition-based planning (Porteous, Cavazza, and Charles, 2010) to control storyline development and character behavior.

Learning character models from human or human-like behavior has been used as an approach to overcoming the knowledge acquisition difficulties (Lin and Walker, 2011; Chang *et al*., 2011). In contrast to this previous work, we take a bottom-up approach, creating NPC behavior diversity that simulates personality diversity, rather than modeling NPC personality which is then reflected in varied behavior. An interesting research direction is to combine these top-down character design approaches with our bottom-up approach.

Domain-independent diverse plan generation using various planning techniques has been conducted in the context of HTN planning (Myers and Lee, 1999), first principles planning with quantitative distance metrics (Srivastava *et al*., 2007 and Nguyen *et al*., 2011), first-principles planning with qualitative distance metrics (Coman and Muñoz-Avila, 2011a), and case-based planning (Coman and Munoz-Avila, 2011b). Eiter *et al*. (2012) address an answer-set programming formulation of the problem.

Case-based planning for game environments has been demonstrated, among others, by Fairclough (2004) and Ontañón *et al*. (2010). Real-time planning has been demonstrated in a game environment by Bartheye and Jacopin (2009). While we do not address real-time planning herein, the presented character diversity techniques could conceivably be applied in that context as well.

## Conclusions

We presented a bottom-up method for creating character diversity. Instead of requiring complex character modeling, our method is based on plan diversity, where plans describe character behavior. The knowledge requirements are a plan repository and user-defined plan diversity criteria. We generate diverse character behavior plans using distance metrics reflecting character traits as manifested in character behavior. We demonstrate the method in a game engine showing how our techniques create varied character behavior simulating such traits.

In future work, we intend to create game character diversity subject to various restrictions. Specifically, we are interesting in modeling situations in which traits of characters inhabiting the same game map region may tend to vary only within the boundaries of general traits specific to the culture of the region in question.

## Acknowledgements

This work was supported in part by NSF grant 0642882 and NRL. We thank the anonymous reviewers for their comments.

## References

Arkin, R.C.; Fujita, M.; Takagi, T.; and Hasegawa, R. 2003. An Ethological and Emotional Basis for Human-Robot Interaction. *Robotics and Autonomous Systems* 42(3-4): 191-201.

Bartheye, O., and Jacopin, E. 2009. A Real-Time PDDL-based Planning Component for Video Games. In *Proc.of AIIDE 2009*, 130-135. AAAI Press.

Cavazza, M.; Charles, F; and Mead, S.J. 2002. Sex, Lies, and Video Games: an Interactive Storytelling Prototype. *AAAI Technical Report SS-02-01*, 13-17. AAAI Press.

Cavazza, M., and Charles, F. 2005. Dialogue Generation in Character-based Interactive Storytelling. In *Proc. of AIIDE 2005*, 21-16. AAAI Press.

Cavazza, M.; Pizzi, D.; Charles, F.; Vogt, T.; and André, E. 2009. Emotional Input for Character-based Interactive Storytelling. In *Proc. of AAMAS 2009*, 313-320. IFAAMAS.

Chang, Y-H.; Maheswaran, R.T.; Levinboim, T.; and Rajan, V. 2011. Learning and Evaluating Human-Like NPC Behaviors in Dynamic Games. In *Proc. of AIIDE 2011*, 8-13. AAAI Press.

Coman, A., and Muñoz-Avila, H. 2011a. Generating Diverse Plans Using Quantitative and Qualitative Plan Distance Metrics. In *Proc. of AAAI 2011*, 946-951. AAAI Press.

Coman, A., and Muñoz-Avila, H. 2011b. Qualitative vs. Quantitative Plan Diversity in Case-Based Planning. In *Proc. of ICCBR 2011*, 32-46. Springer.

Eiter, T.; Erdem, E.; Erdogan, H.; and Fink, M. 2012. Finding Similar/Diverse Solutions in Answer Set Programming. *arXiv:1108.3260v1*. To appear in *Theory and Practice of Logic Programming*.

Fairclough, C. 2004. Story Games and the OPIATE System: Using Case-Based Planning for Structuring Plots with an Expert Story Director Agent and Enacting them in a Socially Simulated Game World. PhD Dissertation, University of Dublin, Trinity College.

Iszita, I.; Ponsen, M.; and Spronck, P. 2009. Effective and Diverse Adaptive Game AI. *IEEE Transactions on Computa-tional Intelligence and AI in Games* 1(1):16-27.

Lin, G.I., and Walker, M.A. 2011. All the World's a Stage: Learning Character Models from Film. In *Proc. of AIIDE 2011*, 46-52. AAAI Press.

McCoy, J.; Treanor, M.; Samuel, B.; Tearse, B.; Mateas, M.; and Wardrip-Fruin, N. 2010. The Prom: An Example of Socially-Oriented Gameplay. In *Proc. of AIIDE 2010*, 221-222. AAAI Press.

McGinty, L., and Smyth, B. 2003. On the Role of Diversity in Conversational Recommender Systems. In *Proc. of ICCBR 2003*, 276–290. Springer.

McSherry, D. 2002. Diversity-Conscious Retrieval. In *Proc. of ECCBR 2002*, 219–233. Springer.

Myers, K. L., and Lee, T. J. 1999. Generating Qualitatively Different Plans through Metatheoretic Biases. In *Proc. of AAAI 1999*, 570–576. AAAI Press.

Nguyen, T.; Do, M.; Gerevini, A.; Serina, I.; Srivastava, B.; and Kambhampati, S. 2011. Generating Diverse Plans to Handle Unknown and Partially Known User Preferences. *Artificial Intelligence* 190:1-31.

Ontañón, S.; Mishra, K.; Sugandh, N.; and Ram, A. 2010. On-line Case-Based Planning. *Computational Intelligence Journal* 26(1): 84-119.

Orkin, J. 2003. Applying Goal-Oriented Action Planning to Games. In: *AI Game Programming Wisdom 2*. Charles River Media.

Paul, R.; Charles, D.; McNeill, M.; and McSherry, D. 2010. MIST: An Interactive Storytelling System with Variable Character Behavior. In *Proc. of ICIDS 2010*, 4-15. Springer.

Porteous, J.; Cavazza, M.; and Charles, F. 2010. Narrative Generation through Characters' Point of View. In *Proc. of AAMAS 2010*, 1297-1304. IFAAMAS.

Shimazu, H. 2001. ExpertClerk : Navigating Shoppers' Buying Process with the Combination of Asking and Proposing. In *Proc. of IJCAI 2001*, 1443–1448. Morgan Kaufmann.

Smyth B., and McClave, P. 2001. Similarity vs. Diversity. In *Proc. of ICCBR 2001*, 347–361. Springer.

Spalazzi, L. 2001. A Survey on Case-Based Planning. *Artificial Intelligence Review* 16(1):3–36. Springer Netherlands.

Srivastava, B.; Kambhampati, S; Nguyen, T.; Do, M.; Gerevini, A.; and Serina, I. 2007. Domain Independent Approaches for Finding Diverse Plans. In *Proc. of IJCAI 2007*, 2016-2022. AAAI Press/IJCAI.

Strong, C.R.; Mehta, M.; Mishra, K.; Jones, A.; and Ram, A. 2007. Emotionally Driven Natural Language Generation for Personality Rich Characters in Interactive Games. In *Proc. of AIIDE 2007*, 98-100. AAAI Press.

Thue, D.; Bulitko, V.; Spetch, M.; and Webb, M. 2010. Socially Consistent Characters in Player-Specific Stories. In *Proc. of AIIDE 2010*, 198-203. AAAI Press.