Glengarry Glen Ross: Using BDI for Sales Game Dialogues

Tijmen Joppe Muller[†] and **Annerieke Heuvelink**[†] and **Karel van den Bosch**[†] and **Ivo Swartjes**[‡]

[†] TNO, Social and Behavioral Sciences

[‡] Ranj Serious Games

{tijmen.muller, annerieke.heuvelink, karel.vandenbosch}@tno.nl, ivo@ranj.nl

Abstract

Serious games offer an opportunity for players to learn communication skills by practicing conversations with non-playing characters (NPCs). To realize this potential, the player needs freedom of play to discover the relationships between its actions and their effects on the partner and the conversation. Scripting is currently the common approach to design in-game dialogue. Although scripting is a robust technique, the approach tends to produce deterministic conversations, allowing little control to the player. It is claimed that a Belief-Desire-Intention (BDI) approach to model the behavior of NPCs allows greater freedom to the player, and delivers better scalability and re-use of dialogues. This claim is evaluated by using BDI in the development of a sales-talk training game in the real-estate domain. It is concluded that BDI enables representative NPCs that respond appropriately and the game allows the player its freedom of choice to explore. The results also showed that BDI brings about new challenges to address, in order to further increase the quality of in-game dialogue.

Keywords: intelligent agents, behavior modeling, training, virtual characters, serious gaming, BDI

Introduction

Serious games are popular because they have the potential to provide a contextually rich and flexible training environment (e.g. Korteling et al., in press; Michael, 2006). They are considered to be effective for learning tasks that rely upon effective communication skills, like legal profession, project management, human resources (e.g. negotiating employment terms), sales, and many other professions (e.g. Core et al., 2006). Communication skills can be practiced in serious games by conducting conversations (e.g. interviews, sales conversations, negotiation talks) with one or more virtual characters.

To train communication skills effectively, a serious game should enable a player to discover the relationships between its actions and the consequences of these actions on the conversation partner. This puts high demands on the development of dialogue content. Ideally, in such games, the player can (a) practice with different types of conversational partners (some cooperative, others more intractable); (b) practice different conversational strategies to find out their merit (e.g., fact-driven versus emotion-driven); and (c) experience the effect of different styles of communication on the behavior of the conversational partner and the course of the conversation. It is key to create Non-Player Characters (NPCs) that are recognizable and react in a consistent manner to player actions, taking into account their personality and goals.

The common approach for designing dialogues in (serious) games is to make use of scripts. In scripted dialogue, the NPC's conversation options are tied to items from a predetermined action set of the player. This approach is often used because it is robust and allows complete control over the dialogue. Often authoring tools are available to game designers for producing dialogue content without programmer intervention. However, scripting does not fit all needs. If a game requires long dialogues, both the size and the complexity of the dialogue scripts grow exponentially. This makes adding new content laborious, reuse of content problematic and narrative consistency of the dialogue as a whole is hard to maintain. Another drawback is the fact that scripts prescribe the flow of a dialogue deterministically, which is especially undesirable in serious games designed to enable learning through exploration. Thus, to achieve the aforementioned goals for conversation-based training in serious games, alternative approaches to scripting are required.

One alternative to scripting dialogues is the use of the *belief-desire-intention* (BDI) model of human behavior, proposed by Michal Bratman (1987). It has been shown that with BDI it is possible to develop intelligent NPCs that behave autonomously and realistically in simulations and games (Shenandarkar et al., 2006; Van den Bosch et al., 2009). Within the BDI paradigm, the (re)actions of the NPC are not established beforehand, but are determined in real time by the reasoning process that takes the context

into account. The character may include many factors in its response selection, like its knowledge of the world, its intentions, its personality, but also, for example, the assumed intentions of its conversation partner. The BDI approach separates knowledge (i.e. dialogue content) from process (i.e. dialogue flow), allowing scalability and a non-deterministic flow of dialogue with mixed-initiative.

BDI has found many applications in Artificial Intelligence (AI), and recently also in Game AI, for example in the AAA game title F.E.A.R. (Orkin, 2005). Since then, a few commercial games as well as a multitude of scientific endeavors have adopted BDI as a means to create on-line flexible content in the form of dynamic story-telling or intelligent NPC behavior (e.g. Peinado et al., 2008). However, most of this work focuses on the (planning) capabilities that BDI supports for creating autonomous, mainly non-dialogue behavior. To our knowledge no commercial games exist that use BDI for creating flexible dialogues with NPCs, although research has shown its potential (e.g. Van Oijen et al., 2011).

This paper describes our exploration of advantages and drawbacks of using BDI for dialogues in serious games by means of a case study. Unique in our approach is a focus on feasibility within a commercial setting. The project is conducted in collaboration with a serious games studio specialized in narrative and dialogue-based serious games. We were mainly interested in (1) the nature of the development process when using the BDI approach in contrast to the scripted approach, in particular its suitability for creating a wide range and variety of behavior that can easily be extended; and (2) the gameplay characteristics enabled by a BDI approach.

To investigate these issues, we developed a sales training game, inspired by the movie *Glengarry Glen Ross* (1992). The player in our game assumes the role of real-estate salesman; a BDI-based NPC fulfils the role of sales lead (i.e. a potential buyer). In a sales conversation, the player must convince the NPC to visit one of the houses in his portfolio. The player can discover the NPC's wishes and influence its opinions by asking questions, emphasizing qualities of the house, and by providing anecdotic examples. The strength of the game is that the NPC does not behave in a predetermined fashion, but acts in accordance with its individual properties and is sensitive to the emerging flow and content of the conversation.

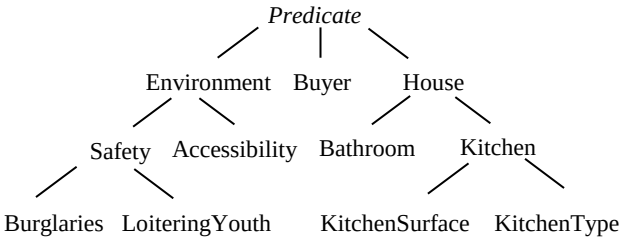

Figure 1. Part of the predicate ontology

The conversation takes place through text messages in natural language that the player selects by means of the house dossier. The house dossier dynamically represents the topics relevant for the conversation at that particular stage of the discussion (see Figure 3 for an impression).

The game uses three BDI agents: (1) an agent that represents the potential buyer (NPC, or buyer agent); (2) an agent that determines the player's possible communication options (the GUI agent); and (3) an agent that controls the scenario and the flow of the game (the scenario agent).

The following section explains how domain knowledge and dialogue concepts are represented. How these concepts are used to create gameplay is explained subsequently. Finally, the advantages and limitations of using BDI for developing human-agent dialogue are discussed.

Domain Knowledge Representation

In order to facilitate dialogue flexibility, extendibility and reusability, the message and information types as well as their content are formalized in ontologies. By using ontologies, the agents can reason over their knowledge and new knowledge can easily be added without the need to adapt the reasoning system.

Predicate ontology

The predicate ontology structures all topics relevant for the real-estate domain. The domain knowledge is structured as a predicate tree, giving the agent knowledge about the hierarchical organization of conversation topics (see Figure 1, with dummy top-level predicate *Predicate*). Main predicates are *House*, *Environment* and *Buyer*, which have various subpredicates (e.g. *Kitchen)*, which may have other subpredicates (e.g. *KitchenSurface*), and so on.

Building block ontology

The building blocks of the dialogue are defined in a building block ontology. It contains information about a predicate as defined in the predicate ontology. The building blocks types are explained below.

Facts

The *Fact* building block defines the ground truth about a given predicate. It is used to define the house-to-be-sold and can be used by both player and buyer to exchange information about the values of the predicates in the predicate ontology. For example, the fact that the house has a kitchen with a surface of 12 m^2 is denoted by:

```
Fact(KitchenSurface, 12)
```

Interpretation

The *Interpretation* building block defines a subjective value about a predicate. It can be used by both player and buyer to exchange subjective information about predicates

without giving the exact value. For example, the interpretation that the kitchen surface is large is given by:

```
Interpretation(KitchenSurface, large)
```

Using interpretations, the player can offer subjective information (e.g. "the kitchen is large"). The buyer agent may adopt the player's interpretation, but may also disagree. Interpretations allow the player to present topics in a positive although not fully truthful way.

Opinion

The *Opinion* building block represents the opinion about a certain predicate. The opinion is expressed by a number between 0.0 and 1.0, where 0.0 is as 'very bad', 1.0 is 'great' and 0.5 is 'reasonable'. For example, the opinion that the kitchen is good is expressed by:

```
Opinion(Kitchen, 0.85)
```

Wish

A *Wish* defines the value or values of a predicate for the ideal house and is only used by the buyer agent. A wish is expressed by one or more fact or interpretation values. For example, a buyer's wish regarding the kitchen surface is:

```
Wish(KitchenSurface, [11, 20])
```

In other words, the surface of the kitchen should be between 11 and 20 square meters. A wish can also be expressed in terms of subjective terms, like:

```
Wish(KitchenSurface, [average, large])
```

Importance

The *Importance* building block expresses how much importance the NPC attaches to the predicate (e.g. *KitchenSurface*) when judging the house. *Importance* thus heavily influences the buyer's interest to discuss the topic, and it also affects its decision to visit the house or not. Importance is expressed by a number between 0.0 (not important at all) and 1.0 (very important).

Argumentation

The *Argumentation* building block is used by the player to influence the buyer's opinion on a specific predicate. It allows the player to employ different playing strategies. An argumentation is linked to one predicate, but may increase or decrease the opinion about more predicates. For example, the argumentation that a nosy neighbor increases neighborhood safety may indeed affect the buyer's opinion on safety, but it may also affect its opinion of tranquility:

```
Argumentation(nosy neighbor,
        Safety, +0.1)
Argumentation(nosy neighbor,
     Tranquility, -0.2)
```

Illustration

An *Illustration* is an additional remark by the buyer agent that reveals its underlying motives, enabling the player to choose a strategy. For example, the following illustration is added to the buyer's reply when told that the house has an open kitchen, hinting that the buyer is a sociable person:

```
Illustration(KitchenType, open, "Great,
so I can keep chatting with my guests!")
```

Message ontology

A message ontology was designed to formalize communication between the agents. The communication types *Tell*, *Ask*, and *Acknowledge* were distinguished.

The *Tell* message is used by both the player and the buyer to share certain information on a specific predicate with the other. For example:

```
Tell(Fact(KitchenSurface, 12))
```

may be used by the player to inform the buyer about the (factual) size of the kitchen surface.

The *Ask* is used to inform after a specific predicate, e.g.:

```
Ask(Wish(KitchenSurface))
```

may be used by the player to ask the buyer after its wishes regarding the kitchen surface.

The buyer uses *Acknowledge* messages to tell the player it has received information. Acknowledgements may contain an implicit opinion. For example,

```
Acknowledge(argument_received_neg)
```

is used by the buyer to acknowledge it has received the player's argumentation, but was not convinced by it.

Game Architecture and Design

For the game, three agents have been developed in Jadex, a BDI reasoning engine (Pokahr et al, 2005).

Scenario agent

The scenario agent controls the game session. It initializes the game by starting the other agents and distributing the properties of the house of the selected scenario. Another responsibility is managing the *turn taking*. Both the player and the buyer agent need to have the turn in order to say something. They can get the turn by requesting it from the scenario agent. If the player requests a turn while the buyer agent is speaking, this may harm the interview atmosphere.

Buyer agent

The buyer agent is designed as a typical BDI model. It has a belief base, a set of goals, plans, and a reasoning system that makes use of several parameters.

Belief base

The belief base stores the world knowledge the buyer agent believes to be true and consists of Facts, Interpretations, Opinions and Wishes on the predicates from the ontology. The buyer agent also stores the dialogue history in its belief base , to make sure the agent does not repeat itself.

On game startup, the buyer is initialized with its individual parameters, wishes, predicate importances, rules to derive interpretations, prior knowledge about some facts about the house and illustrations. The belief base is continuously updated during the conversation. For example, when the buyer is told that the kitchen surface is $12m^2$, it will save this Fact in its belief base, derives an Interpretation that is also saved, and updates the Opinions of the predicate *KitchenSurface* and all its parents.

Goals and plans

The goals of the agent define its behavior; the plans describe the ways to achieve those goals. All goals are focused on assessing whether the house is worth visiting. This can be done either reactively – evaluating and responding to input from the player – or proactively, by asking questions and sharing wishes.

Reactive Behavior The reactive behavior consists of responding to the player. This can either be a question or some information given by the player. The player can ask after a Wish or an Opinion, which is answered by the buyer agent in either case. When the player tells a Fact, the buyer accepts this information. When the buyer is told an Interpretation, it adopt it as its own or reject it, based on its personality or because it does not agree with its own interpretation of the fact. The same holds for an Opinion. Finally, when the player gives an Argument and the buyer accepts it, the opinions of all the predicates associated with the argument are updated.

Proactive Behavior In proactive behavior, the agent can choose to ask specific questions about the house, express its opinions about what is told, or give information about its wishes. When asking a question about the house, the buyer selects a predicate based on the predefined predicate importances, its current knowledge, and the current conversation topic. The player can choose to answer with a fact, an interpretation, or an opinion. The buyer then evaluates the answer and may choose to accept it and update its beliefs, or may reject it and repeat the question.

When the buyer receives information, it typically acknowledges this with a neutral "Okay." or "Thank you.". The agent may add to be happy with the given information, or may provide an additional Illustration.

When giving information proactively, the buyer selects an undisclosed Fact, Wish, Opinion or Interpretation that may also be accompanied by an Illustration.

Parameters

To create variability in game play several parameters are defined for the buyer. These include thresholds for *Information Need*, *Willingness to Buy* and *Interview Atmosphere*, and the buyer's personality.

Willingness to Buy The agent's opinion on the house is a function of its wishes and knowledge about the house acquired during the conversation. The buyer concludes the conversation when the threshold is passed, meaning the house sufficiently fits the needs and wants to visit it.

Interview Atmosphere The NPC's opinion on the conversation itself is influenced by all actions the player takes; for example, interrupting a talking buyer affects the relationship negatively. If it drops below the set threshold, the buyer terminates the conversation, because it does not want to talk with the player anymore.

Information Need The buyer continuously checks the amount of information collected about the house. If it passes its information need, it will finish the conversation and, depending on its willingness to buy the house at that point, may or may not want to visit the house.

Personality

An aspect of the buyer agent that creates variety in game play is its personality, here defined in terms of extraversion and agreeableness. These personality traits were selected as they are known to affect the nature of discourse and are relatively easy to model and test (Brandenburgh, 2012).

Extraversion is the tendency to be assertive, and to direct one's interests outwards. In our game, an extravert agent is designed to take more initiative, prefers detailed predicates over general predicates (e.g. *KitchenSurface* over *Kitchen*), focuses more on the positive aspects of the house, is prone to tell information about itself, and includes illustrations and detailed information in answers. In

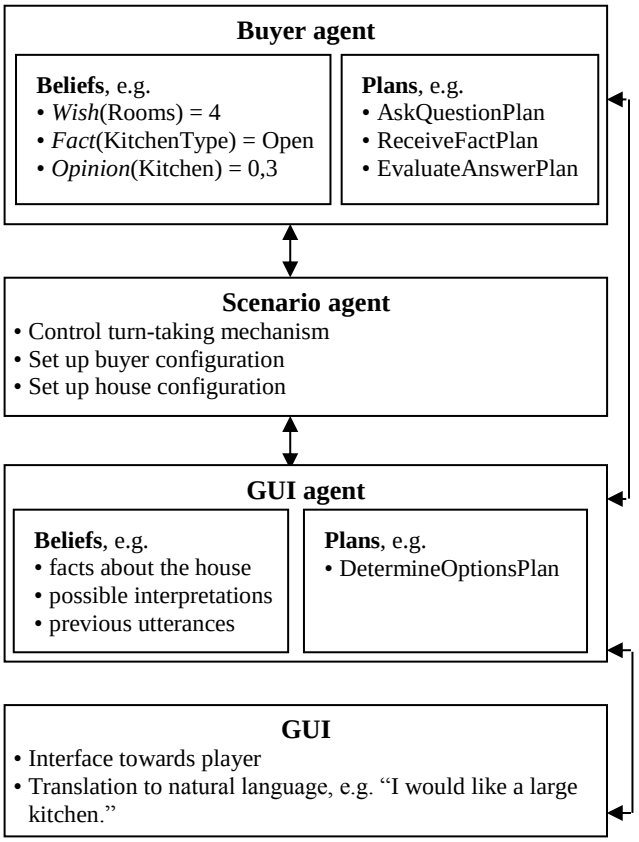

Figure 2. Agent architecture

contrast, an introvert shows little initiative, prefers general over detailed predicates, focuses on negative aspects of the house, tells no information about itself, and gives no illustrations or detailed information.

Agreeableness is the tendency to be pleasant and accommodating in social situations. An agreeable agent is designed to accept subjective information, takes argumentations into account and acknowledge the player's opinion. In contrast, a disagreeable agent does not accept subjective information, ignores argumentations and is not interested in the player's opinion.

The value for both traits are set independently and are represented by a value between 0.0 and 1.0. Depending on this value, the model acts more or less extravert or introvert (or agreeable or disagreeable). A human-subjects study showed that agents behaved in accordance with their personality profile (Van den Bosch et al., 2012)., thus providing a different game experience.

GUI and GUI agent

The GUI agent determines the dialogue options of the player, given the state of the conversation. When the player proactively takes the turn, the player can select multiple dialogue options (e.g. asking a question or telling a fact, interpretation, opinion, or argumentation) on any of the topics defined in the predicate ontology (see Figure 1). However, when the player has been asked a question, the player is obliged to answer. Then the dialogue options are limited to those relevant for answering this question.

Figure 3 shows the GUI designed for the sales game. The center of the screen shows the buyer agent; its utterances are presented in a text balloon below it. The turn taking mechanism is shown on the right . The blue sphere represents who has the turn. It is located at the orange middle when the turn is available – the player can request the turn by dragging the sphere towards him. In the upper right corner two indicators represent the current states of

Figure 3. The GUI of the real-estate sales game

'willingness to buy' and 'interview atmosphere' (note that the agent's thresholds are not visible). The GUI provides filters to support the player in selecting the dialogue option that fits the player's strategy among the many that are available. The orange files on the left allow the player to filter on possible topics (i.e. predicate ontology from Figure 1). The bottom left checkboxes enable the player to filter on communication types.

The GUI also takes care of translating messages and dialogue options into natural language. Dedicated sentence templates are used to translate the message types, building blocks and predicates into complete sentences. For example the buyer's utterance:

```
Ask(Fact(FeederRoads))
```

is presented using a sentence template as: "What is the position of the house with respect to feeder roads?". The player's dialogue option:

```
Tell(Interpretation(FeederRoads, near))
```

is shown as: "The house is close to feeder roads."

Discussion and Conclusion

Development Process

Developing a BDI model requires a greater up-front development effort when compared to scripts. Furthermore, it requires a much stricter collaboration between game designers, domain experts and programmers than with scripting approaches, as the NPCs behavior are not solely defined by the fixed scripts. However, once the model is designed and implemented, it has definite benefits in terms of scalability, variability and reusability. The non-deterministic property of behavior, characteristic for the BDI approach, allows adding variability to the model. This has been achieved in the present game, for example, by creating variability as a function of implemented personality traits. Different settings have been shown to produce different and recognizable NPC personalities and, as a result, a different game experience (Van den Bosch et al., 2012), improving replayability of the game.

The use of goals, plans, and a reasoning system in combination with a domain ontology supports scalability and reusability in a number of ways. First, it makes it easy to expand the number of conversation topics by adding new predicates to the domain ontology, as the reasoning system can immediately take them into account.

Second, it is relatively easy to add new 'scenarios' as new buyers and houses are designed independently and can be constructed from available content. Most of the effort of designing new content will go into creating coherent and challenging combinations of buyers and houses rather than scripting out large numbers of slightly varying paths depending on user choices. The need for authoring tools to

support game designers in developing content through ontologies is considered an important next step.

Finally, the agent mechanisms and building blocks are procedural and can therefore be reused in other conversational games. This contrasts to scripted dialogues , where reuse of conversation structure is not possible.

Gameplay Characteristics

We found the use of a BDI model to open up several game play possibilities that are beneficial for serious gaming, and that are much more difficult to achieve with a scripted approach. First, using a BDI model allows for mixed initiative in game dialogues. Since the dialogue flow is independent of the dialogue content, both the player and the NPC can take the turn to ask a question or tell information. This creates a new type of game dynamics, which was recognized by the game designers evaluating our game. In dialogue games, the choice to be reactive or proactive can be a big part of the player's game strategy and can clearly influence the outcome of the game. In our case, it is easy to think of scenarios where a player will never sell a house, because he is too reactive and not able to stress the positive points of the house for sale.

Second, using the BDI paradigm for a dialogue simulation model greatly increases the freedom of choice for the player, and the variability of the resulting dialogue. This, together with the mixed initiative, creates a type of game play that is completely different from scripting: one where the player is able to explore his options, experience their effects and decide on a strategy. This allows *explorative learning*, where developing a mental model of cause and effect is of critical importance for learning.

We discovered that BDI was not (yet) able to solve all issues relevant to conversational training. For example, dialogues were not immersive, due to the fact that expressive aspects were not included in the models. In its present form, the approach is better suited for learning conversational strategies than for learning conversational nuance, such as training bad news conversations or training to deal with social pressure.

The over-all findings warrant further research into the use of BDI for dialogue design. One issue, for example, is that the approach introduced a much higher freedom of play, allowing the player to try many options and strategies. However, it may be difficult for the player deciding how to use all opportunities to his benefit. Additional functions are needed for guiding the player in formulating a playing strategy. To give an indication, the current game offers the player often about 150 dialogue options to choose from. The mechanism to categorize options using filters that match with play strategies (e.g., show only questions) seemed to be insufficient for pilot-players. Another feature of this approach is that the player can test different strategies. This requires the player to think about and reflect on options. The current game made this hard as – due to the real-time nature of the game – the NPC grabbed the turn when the player took time to think. However, when issues like these are resolved by improving game play coordination, we conclude that BDI provides a valuable alternative for dialogue simulation in conversational games.

Acknowledgements

This research was partly funded by the GATE research programme.

References

Bosch, K. van den; Brandenburgh, A.; Muller, T.J.; Heuvelink, A. (in press). Characters with Personality!

Bosch, K. van den; Harbers, M.; Heuvelink, A.; & Doesburg, W. A. van (2009). Intelligent Agents for Training On-Board Fire fighting. In: Proceedings of Human-Computer Interaction Conference (pp. 463-472), Berlin: Springer-Verlag.

Brandenburgh, A. 2011. Influence of Personality on the Behavior of Conversational Agents. Master thesis, Vrije Universiteit, Amsterdam.

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

Core, M., Traum, D., Lane, H.C., Swartout, W., Gratch, J., Lent, M. van, & Marsella, S. (2006). Teaching Negotiation Skills through Practice and Reflection with Virtual Humans. Simulation. 82(11), pp685-701.

Korteling, J.E., Helsdingen, A.S., & Theunissen, N.C.M. (in press). Serious gaming @ work: learning job-related competencies using serious gaming. In: Derks, D., & Bakker, A. B. (2012). The Psychology of Digital Media @ work. London: Psychology Press.

Michael, D. (2006). Serious games: Games that educate, train, and inform. Boston, MA: Thomson Course Technology.

Oijen, J. van; Dignum, F.; and Doesburg, W. van. 2011. Goal-Based Communication Using BDI Agents as Virtual Humans in Training: An Ontology Driven Dialogue System. In F. Dignum, ed., *Agents for Games and Simulations II*, Lecture Notes in Computer Science, 38–52; Springer-Verlag.

Orkin, J. 2005, Agent Architecture Considerations for Real-Time Planning in Games. In *Proceedings of the Artificial Intelligence and Interactive Digital Entertainment Conference*: AAAI Press.

Peinado, F.; Cavazza, M.; and Pizzi, D. 2008. Revisiting Character-Based Affective Storytelling under a Narrative BDI Framework. *Interactive Storytelling*, 83-88: Springer.

Pokahr, A.; Braubach, L.; and Lamersdorf, W. 2005. Jadex: A BDI reasoning engine. In R. Bordini, et al. (Eds.), *Multi-agent Programming*, 149-174: Springer.

Shendarkar, A., Vasudevan, K., Lee, S., & Son, Y. (2006). Crowd Simulation for Emergency Response using BDI Agent Based on Virtual Reality. In: WSC 06. Proceedings of the Winter Volume, pp(545-553).