A Bayesian Network Framework for the Construction of Virtual Agents with Human-like Behaviour

Olav Bangsø, Nicolaj Søndberg-Madsen and Finn Verner Jensen
{bangsy, nicolai, fvj}@cs.aau.dk
Department of Computer Science
Aalborg University, Denmark

Abstract

It seems tempting to equip virtual agents, for example in a computer game, with graphical models for decision making. That is, if a virtual agent possesses an influence diagram of its relevant world, then it will be able to take rational decisions and appear very intelligent. However, this approach misses the point that agents are supposed to act human-like regardless how alien they may appear. What is missing is that emotions influence the choice of action, and rather than trying to incorporate emotions in the utilities it is preferable to make explicit the dynamics of emotions as well as their influence on the choice. The Twilight project performed at Aalborg University is based on the so called EMS, which is a system for simulating human emotions. According to this theory, emotions are affected when events influence goals or resources. The Comparator analyses how an event may influence goals and resources. The Evaluator determines the change of emotional profile (a vector containing degree of anger, fear, joy, and sorrow), and the Action Proposer picks the action mediating possibilities and emotional profile. This paper is inspired by the EMS. We focus on the Comparator and outline how the developed Bayesian network models also can be utilized for the Evaluator and the Action Proposer.

1 Introduction

Virtual agents acting credibly as human beings would be of interest in computer games. The class of games that would benefit the most from virtual agents acting credibly as humans is *Role-Playing Games* (RPGs). RPGs usually give the player more freedom than other classes of games. With this freedom the player can be expected to interact with any object in the game, including any virtual agent. Games are getting more and more realistic in all aspects however in many cases little effort is put into the virtual agents. This in spite of computer agents acting credibly as humans is among the oldest investigated topics in AI (Turing, 1950). We present a method to produce agents with human-like behaviour which is well suited for virtual agents in RPGs.

Some popular games, which have very simple virtual agents, are Diablo II (Blizzard Entertainment, 2000) and Neverwinter Nights (Bioware, 2002). In Diablo II each agent exhibit the same behaviour except from certain scripted events. In Neverwinter Nights the agents do not have independent behaviour; they wait only for the player to come and interact with them. Usually, rational virtual agents attempt to maximize their expected utility. This solution, however, results in predictable behaviour which usually is unwanted in computer games.

One of the aims of the Twilight project is to develop a system for simulating human emotions, the system is called an *Emotional and Motivational System* (EMS). Other aims include a system for dynamic narration and modeling of the players preferences based on the players action. Nikolaj Hyldig from the *Department of Communication and Psychology, Aalborg University* has proposed to base the system on theories on the human emotional and motivational system (i.e. psychology) which guides human actions (Hyldig, 2005).

The EMS is similar to the OCC model (Ortony, Clore, and Collins, 1988). In (Bartneck, 2002) some practical problems with implementing the OCC model for facial expressions are pointed out, along with solutions for some of them. Most notable are a reduction of the number of emotions and keeping a history of events. Expressing several emotions at once remains a problem along with modeling different personality types.

In this work we focus on actions and utilize *Bayesian Networks* (BNs) to simulate emotional responses. The driving force in the EMS is *Quality of life* (QoL) and it is used to determine both emotional state and choice of action. The choice of action is influenced by the value of all emotions, allowing several emotions to be expressed through the agents' actions. Using the resources available to the agent to model the world state for the agent, we eliminate the need for the history function proposed by (Bartneck, 2002). By allowing predefined preferences to influence both the change in emotion and choice of action it is possible to model different personality types

Each agent will get a certain amount of QoL by performing activities. Each activity requires a number of resources in order to be carried out and if a resource is lost it affects which activities the agent can carry out. The QoL received from a certain activity is determined not only by how effective the activity is but also by how motivated the agent is and by his general preference for that activity. We use QoL to determine the *Emotional Profile* ($\overrightarrow{EP}$) of the virtual agents.

We work with four kinds of emotions: *Joy*, *Sorrow*, *Anger* and *Fear*. The exact emotional response to an event depends on the change in the level of expected QoL in the situation after the event. *Anger* occurs when the agent estimates that it is difficult to retain the level of QoL it had before the event while *Sorrow* occurs when it has become unachievable. *Fear* occurs when the agent estimates a significant decrease in future expected QoL while *Joy* occurs when the expected QoL rises. The intensity of the emotion is determined by the change in expected QoL.

The EMS is decomposed into five steps. The *Comparator* compares how likely it is that the agent can retain or maintain its QoL before and after an event has occured. The *Evaluator* determines the emotional reaction to the new situation. It also determines the intensity of the emotional response. The *Action Proposer* proposes the action the agent will take in order to deal with the event. The action chosen should be influenced by the expected QoL and the agent's $\overrightarrow{EP}$. The *Physiological Change Generator* handles changes in the agent's appearance in the game to exhibit the emotions i.e. textures, animations etc. The *Actor* carries out the chosen action, if any.

In this paper we will concentrate on the Comparator and describe how the problem can be solved using Bayesian networks. The next section will describe the general structure of the models through an example. Section 3 will show how the models can support the Comparator. The last two sections outlines future work, Section 4 will outline how the developed Bayesian network models can be utilized for the Evaluator and the Action Proposer, while Section 5 describes what must be specified for our framework to be utilized.

2 Virtual Agents

The goal of this work is to have virtual agents performing close to how humans would. To this end we use QoL as the basic concept driving the agents. Activities and ownership are the only ways agents can receive QoL and the QoL an agent receives from a specific activity is influenced by how motivated the agent is for doing the activity. The motivation for the different activities may be influenced by different factors, called resources, and the motivation for each of the activities is influenced by the agent's preference for the specific activity. A rational agent would try to maximize the expected QoL, but in this work the $\overrightarrow{EP}$ of the agent also impacts the choice of activity.

The methodology for deciding which activity to perform can be described by the following steps:

1. An event changes the resources of the agent.

2. The expected QoL may change as an effect of the change in resources.

3. A change in expected QoL triggers an emotional response.

4. The expected QoL given the changed resources and the new $\overrightarrow{EP}$ dictate the activity to perform.

We will introduce the framework through an example. We use Object Oriented Bayesian Networks (Bangsø and Wuillemin, 2000) (OOBNs) in the example for ease of exposition, as the irrelevant details are hidden in the objects, we also believe that some classes can be reused, so some gain in specification time should be achieved. However the ideas in this paper are equally applicable to BNs. Some of the notation of OOBNs are illustrated in Figure 1. The dashed nodes are input nodes representing some node specified outside the scope of the class having an influence on nodes inside instances of the class. The boxes represent instances of classes, where the nodes on the upper half of the box are input nodes to the instance, and the nodes at the bottom are output nodes; nodes inside the instance that can be parents of nodes in the surrounding class.

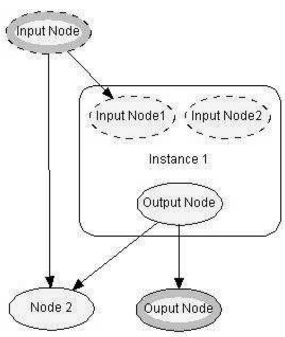

Figure 1: An OOBN containing an instance with two input nodes and an output node. The OOBN class itself has one input node and one output node.

As this work is in the context of computer games, the main focus is on having the agents exhibiting human-like behaviour, observable by the player. The agents in the example will have three types of activities that yield QoL, in a commercial game there will need to be more:

- Farming activities has to do with farming the land.
- Home activities are the activities we want the agents to be able to perform within the confines of their home.
- Spare-time activities are any activities we wish the agent to be doing when it is not working or at home.

The QoL received from each activity type is influenced by the agent's type, e.g. a soldier will not get much QoL from doing farming activities whereas it will be the meaning of existence for a farmer.

2.1 Farming Activities

The motivation for all farming activities are influenced by the season and the time of day. The farming activities are:

- Ploughing with a plough and a horse. The most efficient way of ploughing. Requires a field, a plough, and a horse.
- Ploughing with a plough. If no horse is available, ploughing with a plough is the most efficient. Requires a field and a plough.
- Ploughing with a shovel. The least efficient way of ploughing. Requires a field and a shovel.

The QoL received from each farming activity is influenced by the agent's energy level, e.g. if the agent has a low energy level, it is not expedient to plough with a shovel.

2.2 Home Activities

The home activities are:

- Cleaning. The motivation for cleaning is affected by the time of day and the season.

- Playing with children. Requires at least one child. The motivation is influenced by the agent's energy level. The QoL received is influenced by the time of day and the agent's charisma.
- Playing with spouse. Requires a spouse. The motivation is influenced by the season, the agent's energy level and the spouses charisma. The QoL is influenced by the time of day and the agent's charisma.

2.3 Spare-time Activities

The spare-time activities are:

- Pub visit. The motivation for going to the pub is influenced by the agent's marital status and energy level. The QoL is influenced by the time of day, whether the agent has company, and how much money the agent has.
- Dating. Requires a date. The motivation for dating is influenced by the agent's marital status and energy level. The QoL is influenced by the time of day, the marital status, the agent's charisma, and how much money the agent has.
- Playing chess. Requires a chess partner. The motivation is influenced by the time of day. The QoL is influenced by the agent's intelligence.

In Figure 2 an OOBN for the spare-time activity type can be seen. The general structure for activity type OOBNs can be extracted from this example. The parameters for the models are elicited from the game designer, as they tend to have an urge to have control over how characters in their games behave. Furthermore there will in general be no data available for the behaviour of game characters, the desired behaviour of them need not be the same as that of any population in existence.

There is an instance for each activity outputting a measure of the QoL of that activity. Each activity also has an instance outputting the agent's motivation for the specific activity. The preference for the activity type is a parent of the utility node. The input nodes are the resources, the decision node is a dummy variable and is used for reading the expected QoL of each activity.

2.4 Supporting the EMS

The developed OOBN models support the EMS in three ways; The OOBNs are models of expected QoL; we can enter events to the models to get the resultant change of expected QoL (part of the task of the Comparator of the EMS). The EMS should output an updated $\overrightarrow{EP}$, so it needs to determine the strength of the emotional response to the event (part of the task of the Evaluator); part of this task will include the size of the change in expected QoL. The OOBN models can provide a ranked list of candidate activities which is used in conjunction with $\overrightarrow{EP}$ to choose the activity to perform (the task of the Action Proposer).

In the following sections we describe in more detail how the models support the EMS.

3 Using the Models for the Comparator

Part of the EMS is the Comparator that compares the agent's expected QoL before and after an event has occured. In this section we will show how the OOBN models can be used to this end.

Any event can be modeled by how it influences the agent's resources, e.g. a theft event results in the stolen resources not being available for the agent. Once an event is modeled it can be inserted into the activity type OOBNs and the QoL yielded by the best activity given $\overrightarrow{EP}$ can be compared to the QoL yielded before the event was introduced. This gives the change in expected QoL for the event. Note that each resource also has a QoL attached for ownership, i.e. the agent gets QoL for having the resource, and if the resource is lost this will also affect the change in expected QoL. The emotional response then depends on this change as outlined in Section 1.

In Figure 3 an OOBN for the event that the agent's money is stolen can be seen. The output node *Money* should contain the expected

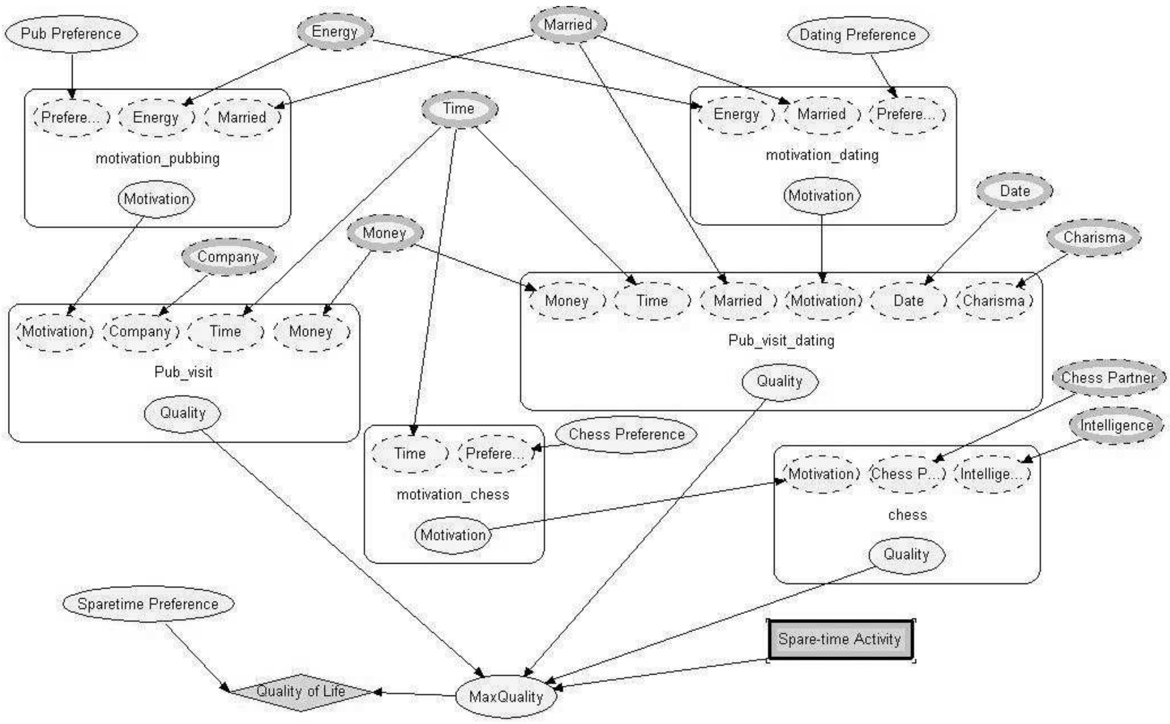

Figure 2: An object oriented BN for the spare-time activities.

Figure 3: A model for a theft event (theft of the agent's money). The *Theft* node models whether the event has occured or if it is a threat.

value for the agent's resource money, given the current value and the probability of the event occurring. The input node *Money* should be associated with the agents' *Money* node (from the activity models). The node *Theft* contains the probability that the event will occur, i.e. the probability that the threat will be carried out. If this *Theft* node is in the state false, the output node *Money* will have the same distribution as the input node *Money*, i.e. no theft has occurred, so the agent's money has not been affected. If it is in the state true, the event has occured so the agent's money will be affected by the event, and the output *Money* will then contain the new value for the agent's money. This model is inserted in the activity type OOBNs between the Money input node and the activities where Money is an input node. This is shown in Figure 4.

4 Outline of the Evaluator and the Action Proposer

In this section we outline ideas concerning the Evaluator and the Action Proposer, these are still preliminary, and should be seen as topics for future research.

The Evaluator determines to which extent the four emotions are influenced by the event. Determining the emotional influence is done according to some simple rules using the change in expected QoL as input:

1. Positive change in expected QoL results in joy.

2. Negative change in expected QoL, the change cannot easily be rectified (e.g. the murder of a loved one), results in sorrow.

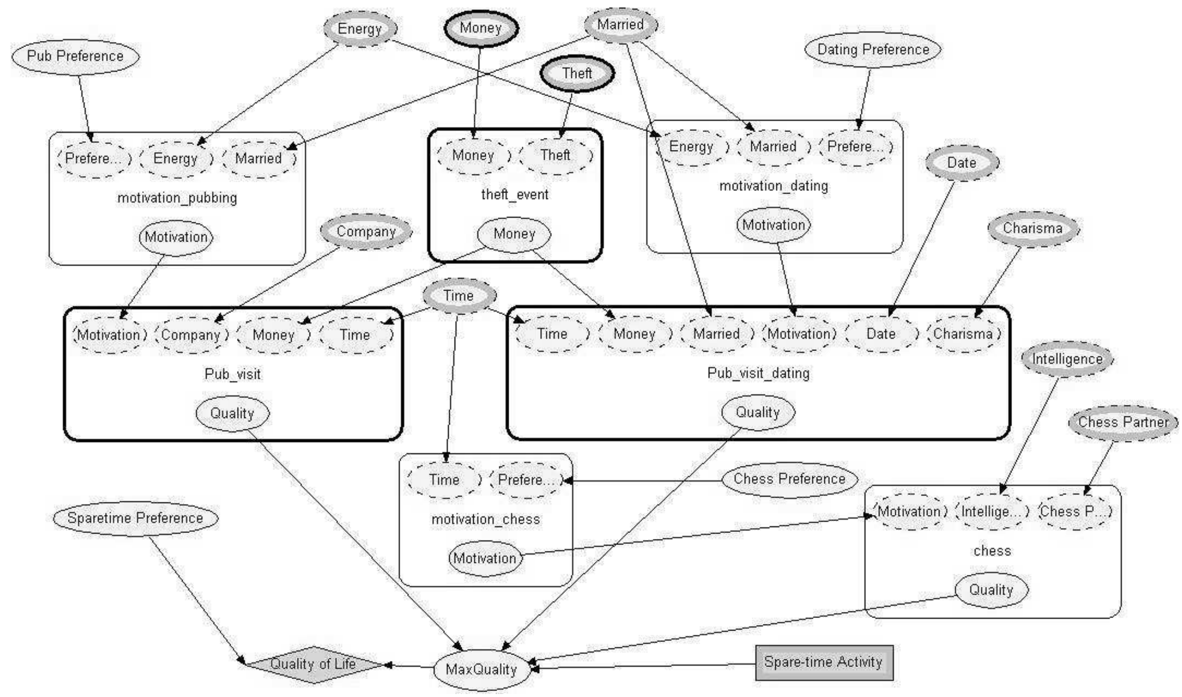

Figure 4: The Spare-time activity type OOBN where a model for a theft event (theft of the agent's money) has been inserted.

3. Negative change in expected QoL, the change can be rectified without a lot of effort, results in anger.

4. Negative change in expected QoL as a result of a threat, the antagonist can be defeated, results in anger.

5. Negative change in expected QoL as a result of a threat, the antagonist is next to impossible to defeat, results in fear.

Distinguishing between situations where 2. or 3. is fired is not a trivial matter. An approximation can be achieved by making BN models of the difficulty of getting resources, and then determining how difficult it is to bring the expected QoL back to at least what it was before the event. These models will resemble the models already described. The differences will be that activities are plans for acquiring resources, the activity types are different plans for acquiring a resource, and instead of expected QoL the utilities will be a measure of the expected inconvenience. The plans will also include an instance of a class outputting the probability of success of the plan, used to determine the expected effect on the resources of attempting the plan and for assessing the difficulty.

This approximation means that we risk ignoring the murder of a wife, unless the QoL for owning one is set high enough, as the highest expected QoL may not include a wife. Distinguishing between 4. and 5. can be done by comparing the resources of the agent and the resources of the antagonist.

Determining the strength of the influence on the emotions is a mixture of the difficulty of overcoming the problem and how serious the expected QoL is affected, i.e. how large the change in expected QoL is. Finding the strength is done during the determination of the emotion affected, the seriousness is the output of the Comparator.

The Action Proposer should output the next thing for the agent to do. It gets as input an $\overrightarrow{EP}$ of the agent and a list of activities, ranked by their expected QoL. Different agents will react differently to an emotional change, e.g. as a bully type agent is angered it is more likely to choose aggressive activities, while a pacifist agent will react to higher anger by being more likely to choose activities where the agent is alone. We envision a system where each agent has an *Emotional Factor vector*($\overrightarrow{EF}$) with an entrance for each emotion associated with each activity. The Action Proposer will then for each activity, multiply the dot-product of $\overrightarrow{EP}$ and $\overrightarrow{EF}$ with the expected QoL yielding the *Emotional Value*(EV):

$$EV = QoL * \overrightarrow{EP} \cdot \overrightarrow{EF}$$

To avoid predictability we will pick an activity at random using these values as weights.

Using only this will result in the agents always performing some activity, thus never trying to better their situation by trying to acquire more resources. As we already need models to assess the difficulty of (re)acquiring resources for the Evaluator, we might as well use these models for determining whether the agents should try to acquire resources. The expected inconvenience of each plan can be subtracted from the expected QoL of the situation where the plan has been attempted. This value is then used for the calculation of the EV of the plan. Plans for acquiring resources can then be used by the Action Proposer in the same way as activities. Note that this approach is myopic, but can easily be expanded to include any number of subsequent plans. This, however, comes at the cost of an exponential growth in the number of possible combination of plans to evaluate. Alternatively a greedy approach may be attempted, where the first step is to find the plan, or plans, with the highest expected QoL (after subtracting the expected inconvenience of the plan), then the expected situation after carrying out the plan is used to find the best plan to execute next, and so on, until no gain in expected QoL is possible. The greedy approach will have the advantage of giving a better QoL estimate, but will take longer to calculate, for each step in the sequence, each possible next step needs to be evaluated. Furthermore this approach will include only one choice of resource acquisition[1]; the best sequence of plans, so once the EV is calculated, we cannot be sure that this was in fact the best sequence.

To avoid the value of emotions being constantly growing, we will maintain a list of the contributions to $\overrightarrow{EP}$, and let the values of each contribution fade over time. Some events may, however, be so powerful that they are never completely removed from $\overrightarrow{EP}$ (traumas), how this occurs in humans is a matter of psychology, and it is being investigated in the Twilight project. Current events may also trigger the recollection of previous events, giving them renewed strength in $\overrightarrow{EP}$, this is, once again, a thing that we hope to look further into how to include.

5 Specification Tasks

In this section we will go through all the things that need to be specified for the outlined approach to be implemented completely. We aim at developing a system for automatically generating OOBN models from the specifications.

5.1 Resources

The resources that might be available for the agents and their possible values. Some resources are global values, e.g. the season, others are attributes of the agents, e.g. charisma, while others are objects or abstractions that can be found in the game-world, e.g. a plough is an object while money might be an abstraction. No matter what they are, we need to know were to get the values from the game. The QoL of each resource being in a specific state must also be specified, along with a standard preference for the resource. It may also be expedient to include the standard range an average agent will have of each resource, where it makes sense, e.g. it does not make sense for a resource like season, but it does for intelligence.

[1] It is possible to include all the sequences that has been evaluated, so there may be a few of the possible sequences available.

5.2 Activity Types, Activities, and Plans

The activity types along with all the activities in each activity type. For each activity, the resources that influence the motivation for the activity, which direction, and the strength of the influence along with the same for influence on the QoL of the activity. It is also a good idea to include a standard preference for both the activity types and the activities. Each activity should also be given a factor for each emotion, used for calculating the EV, by the Action Proposer. It may be good idea to let all $\overrightarrow{EF}$ for an agent sum to the same value, so that fine-tuning the behavior of agents can be done solely on the preferences.

Plans for acquiring resources should include the same as activities, with inconvenience replacing QoL. Furthermore the specification should include resources that influence the probability of the plan being successful, how they influence the probability, how resources are affected by attempting the plan, and how resources are affected by the plan being carried out successfully.

5.3 Agent Types and Agents

For each type of Agent the preferences of all of the above, where the agent type differs from an average agent, needs to be specified. If a preference is unspecified the standard preference will be used. The agent type can also include a standard range for the resources where an agent of that type will differ from the average agent. Agent types may also have $\overrightarrow{EF}$s that differ from the average agent, e.g. the bully type mentioned. It still makes sense to let $\overrightarrow{EF}$s sum to the same value, e.g. the bully type agent's inclination to beating people up will drop more than the average person's, when joy rises.

Each agent needs a type and a specification of the preferences where it differs from a standard agent of that type, note that preferences for activity/plan types are fixed inside an agent type. If some resource needs to be fixed at a specific value, or inside a range that differs from the average agent of that type, it must be specified. It is usually a good idea to make sure that the resources given to an agent fall within what is possible for that agent type to have. It is also possible to overwrite $\overrightarrow{EF}$s in the agent specification.

Acknowledgments

The authors would in particular like to thank Nikolaj Hyldig for valuable discussions on his ideas on the EMS which we have given a normative angle. The authors would like to thank Hugin for providing a tool for specification of OOBNs. We would also like to thank the anonymous reviewers for valuable comments.

References

A. Turing, 1950, *Computing machinery and intelligence*, Mind 59:433-460.

Blizzard Entertainment, 2000, *Diablo II.*

Bioware, 2002, *Neverwinter Nights.*

N. Hyldig, 2005, *EMS-20051228-2*, Twilight Project work document.

A. Ortony, G Clore, and A. Collins, 1988, *The Cognitive Structure of Emotions*, Cambridge University Press.

C. Bartneck, 2002, *Integrating the OCC Model of Emotions in Embodied Characters*, In Proceedings of the Workshop on Virtual Conversational Characters: Applications, Methods, and Research Challenges, Melbourne.

O. Bangsø, P.H. Wuillemin, 2000, *Top-down Construction and Repetitive Structures Representation in Bayesian Networks* In Proceedings of the Thirteenth International Florida Artificial Intelligence Society Conference, pages 282-286.