Extending Believable Agent Frameworks with Predicate Logic Dialogue Generation

Kaylen Wheeler

September 28, 2012

Abstract

Despite advances in graphics, physics, and AI, modern video games are still lacking in believable social simulation. Story, dialogue, and character behaviour are more often scripted than allowed to emerge dynamically. The more complex and interactive stories in modern games may allow the player to experience different paths in dialogue trees, but such trees are still required to be manually created by authors. Recently, there has been research on methods of creating emergent believable behaviour [1, 5, 8], but these are lacking true dialogue construction. Because the mapping of natural language sentences to meaningful computational representations (logical forms) is a difficult problem [16], it may be best to represent inter-character dialogue as logical expressions. The proposed thesis will extend an existing believable agent framework with a predicate logic-based dialogue system that will allow for true construction and interpretation of dialogue by non-player characters.

1 Introduction

Throughout the history of video gaming, the majority of efforts to increase believability have focused on advancement in graphics and physics, powered largely by advances in computer hardware. Modern games often have near-photorealistic graphics and highly believable physical simulation. However, advances in AI, although present, have not added much to believabilty. Rather, developments in game AI are focused on providing challenging and interesting gameplay by creating agents that can either cooperate or compete with the player. Such agents, although effective in creating entertaining gameplay, fail to create believable characters, specifically with regards to social behaviour.

In addition to social behaviour, the ability to generate dialogue dynamically is also lacking. Modern games which are praised for their complex dialogue systems, such as the *Mass Effect* series [2] do not in fact have their characters generate dialogue, but rather select from pre-written (and voiced) dialogue based on certain conditions. Even experiments in interactive drama such as Facade [8] rely on complex ways of selecting from a set of existing dialogue acts. Facade is a highly variable game experience taking only about 20 minutes to play through and consisting of a very limited environment (a single room with one player character and two NPC's), and yet several hundred thousand lines of code were written of handle all possible dialogue acts [7]. Clearly, such an approach is not scalable to open world, multi-agent environments.

Believable agent frameworks are already in place which simulate emotional and social interactions. These systems allow for autonomous agents to form goals based on their current emotional state and social relations, and perform actions to satisfy those goals.

The proposed thesis will involve the implementation of an extension dialogue module to FAtiMA [5], an existing believable agent framework. Currently, the agents in FAtiMA are limited to selecting from an existing finite set of actions to perform based on their goals and current state. Using the same goals and current state, the new dialogue module would allow agents to perform not only action selection, but action construction, creating arbitrary logical expressions to communicate with other agents. Additionally, the module will allow for the interpretation of such logical expressions to affect the current state of an agent.

2 Background and Previous Work

This thesis will involve the synthesis of both Believable Agent Frameworks (abbreviated here as BAF) and Natural Language Processing (NLP). This section will provide background and previous research from each of these areas.

2.1 Believable Agent Frameworks

A BAF is any framework that simulates the social and emotional behaviours of humans in a believable way. The term "believable" is difficult to define because it is subjective by nature. However, it is often suggested that believability is not truly an attempt to fool the user into actually believing what is presented; it is rather an attempt to allow the user to willingly suspend disbelief. [1]

A well-known attempt to create more believable narrative and gameplay is the Facade project [7]. Facade is a one-act interactive drama with a natural written language interface. The player takes on the role of a character visiting a married couple, and is able to converse in real-time with the other two characters by typing natural language text.

Although Facade was created to select appropriate responses to many situations, each of these responses was hand-authored. Additionally, natural language input from the user was not intrepreted to its fullest possible extent by the program. Rather, surface text processing was used which searched user input for a number of key words and phrases and mapped it to one of a finite, pre-defined set of discourse acts. [6] In some ways, Facade can be considered similar to dialogue trees in games such as *Mass Effect*, but with an interface that gives the illusion of natural language interaction.

Some of the creators of Facade were also involved in the creation of the "Comme il Faut"(CiF) [11], a social AI system. The intention of CiF was to provide a "system for authoring playable social models" [11]. Rather than authoring dialogue trees, which the authors called "burdensome" and "highly constrained", the Social AI System would allow the authoring of a "space of possible stories". Through the definition of social and cultural rules that can be applied to a given social state, believable social behaviour is allowed to emerge.

CiF was used to implement the game *Prom Week* [9]. *Prom Week* pioneered what the creators termed "Social Physics" [10] – intuitive rules of social interaction that could guide the user's gameplay toward accomplishing the game's goals, similar to how physics are used in many puzzle games. However, like Facade, and despite its advanced modelling of social and emotional phenomena, the dialogue is still composed of a finite number of templates that are selected in the appropriate situations.

A more easily accessible (in terms of source code) and more modular BAF is the FearNot Affective Mind Architecture (FAtiMA) [5]. FAtiMA was developed initially as an engine for *FearNot!* [3], a serious game aimed at teaching school children between ages 8 and 12 how to deal with being bullied.

FAtiMA has a number of modules, each based around modelling aspects of believable agents. These include modules for modelling Theory of Mind [4], memory, emotional intelligence, dialogue, and others. It also has the ability to be easily extended by adding new modules. This feature allows different models of the same phenomena to be implemented and tested for validity within the same framework (e.g. evaluating different theories of emotion appraisal for their believability).

In terms of dialogue, FAtiMA currently has a module that maps Theory of Mind (ToM) - related goals to pre-defined discourse actions. Theory of Mind goals differ from other goals in that they seek to affect the state of another agent's beliefs rather than the state of the world itself. These dialogue acts, however, are not creted dynamically and must be manually authored and explicitly mapped to corresponding agent goals.

2.2 Natural Language Processing

Natural language processing is still an ongoing area of research. Recent developments have lead to more advanced search engines and data mining techniques, as well as computer-aided natural language translation. However, the problem of mapping natural language sentences to meaningful forms that can be effectively interpreted by a computer is still a very open problem.

Zettlemoyer [16] has conducted some interesting research on the topic of mapping natural language sentences to λ -calculus expressions, which can be easily represented as sentences in first-order logic. His research makes use of combinatory categorical grammars (CCG) [14] generated by advanced machine learning techniques in order to parse sentences of various natural languages.

An example of a mapping of natural language to λ -calculus is demonstrated below. Here, the sentence "What states border Texas?" is mapped to an equivalent λ expression. The expression takes as a parameter a single unbound variable x, which must satisfy the condition of being a state and bordering texas. When the expression is used as a query to a knowledge base, x unifies with all values that satisfy both conditions.

"What states border Texas?"

$$\lambda x.(state(x) \wedge border(x, Texas))$$

$$x = \{NewMexico, Oklahoma, Arkansas, Louisiana\}$$

Although the specifics of the theory and implementation are beyond the scope of – and likely of little relevance to – this proposal, the important realization is that natural language sentences can be transformed into these forms, as well as the fact that software exists that can accomplish this transformation [15]. Using software to translate logical expressions into natural language could allow the creation of language-independent dynamic dialogue systems that can be easily localized into different languages.

3 Problem Description and Proposed Solution

Although many of the Believable Agent Frameworks researched so far have had extensive flexibility in the use of authored dialogue, the fact still remains that the dialogue must be authored. For each dialogue act, its representation (i.e. words used in the dialogue) as well as each of its effects on the world or other agents must be explicitly defined. Additionally, explicit mappings between agents' states or goals to dialogue actions must be provided (e.g. the goal to make another character laugh must explicitly relate to the "tell joke" action).

The proposed solution is to create a system that alleviates the need to explicitly author dialogue acts. Rather than providing a set of mappings between domain-specific dialogue acts and goals, a domain-independent module will be created that constructs dialogue acts to match goals. The system will be implemented as an extension to FAtiMA's existing dialogue system.

FAtiMA uses knowledge bases to represent both the state of the world and the beliefs of individual agents. These knowledge bases have capabilities similar to first-order logic systems. Rather than generating natural language dialogue, first-order logic expressions will be created to accomplish the goals that have been specified by the agents.

If possible, the solution may also include natural language "rendering" and "de-rendering" modules. These would allow logical expressions to be converted to and from natural language sentences. This is not necessary in the implementation, but may be included if time allows in order to demonstrate how natural language could further improve believability.

3.1 Illustrative Example

To understand the differences between authoring and generating dialogue actions, an example will be provided. This represents a hypothetical scenario involving two characters – Good Guy and Bad Guy, who compliment and insult one another. Three main components are defined for each character – goals, action tendenceies, and actions. These are defined in pseudocode, visible in Figure 1. The pseudocode represents first-order logic expressions as conjunctions ($\wedge$) of conditions. Conditions are defined as predicates (i.e. $good(x)$ means "x is good"). For clarification, unbound variables are introduced with λ.

Goals represent states that the agents are striving to achieve. In the case of Good Guy, his goal is that *target*(i.e. Bad Guy) should be happy. Bad Guy's goal, on the other hand, is that *target*(i.e. Good Guy) should not be happy.

Action tendencies represent how agents act when faced with certain situations. The particular situations illustrated here involve knowledge, and are defined with the *knows* predicate. (At the moment, FAtiMA supports action tendencies in response to events. To support this dialogue system, it will need to be extended to analyze changes in knowledge bases and generate events based on them) For Good Guy's action tendencies, unbound variables x, r, and p are introduced. The variable x unifies with some predicate that satisfies the *relative* predicate, while p unifies with some predicate which is *good* or *bad*. The results (after the $\mapsto$ symbol) are either happiness or uhnappiness. Thus, the action tendencies for Good Guy can be described as "If I know something good about a relative, I am happy." and "If I know something bad about a relative, I am not happy.". Similarly, Bad Guy's action tendency is "If I know something good about me, I am happy.".

A finite set of actions is defined, representing potential effects that agents may have on the world. A continuous partial-order planner [12, 13] is used to construct plans composed of sequences of actions that achieve the agent's goals. However, the actions in the sequence may only be selected from a finite, predefined set of actions. The goal of this thesis will be to allow for actions to be constructed from the goal they are trying to achieve.

3.1.1 Example Plan

An example plan is shown in Figure 2. The example shows how the proposed method of action construction can be used to generate dialogue dynamically. In the example scenario, Bad Guy is attempting to fulfill the goal of making some target (i.e. Good Guy) feel unhappy.

Using FAtiMA's current action selection method, any potential actions must be pre-defined by the author. For example, Figure 1 has some actions defined, such as $insultMother$ and $compliment$. If FAtiMA's current method was being used, each character would have only one option for dialogue. (Naturally, real scenarios would have many more action definitions, but this is just for illustration purposes.)

The important difference in Figure 2 is that Bad Guy's set of actions are

Good Guy

Goals:

$$isHappy(target)$$

Action Tendencies:

$$\lambda x.\lambda r.\lambda p.(knows(self, relative(r) \wedge r(x, self) \wedge good(p) \wedge p(x)) \mapsto isHappy(self))$$

$$\lambda x.\lambda r.\lambda p.(knows(self, relative(r) \wedge r(x, self) \wedge bad(p) \wedge p(x)) \mapsto \neg isHappy(self))$$

Actions:

$$compliment(target) \mapsto knows(target, isHandsome(target))$$

Bad Guy

Goals:

$$\neg isHappy(target)$$

Action Tendencies:

$$\lambda p.(knows(self, good(p) \wedge p(self)) \mapsto isHappy(self))$$

Actions:

$$\lambda x.(insultMother(target) \mapsto knows(target, mother(x, target) \wedge isFat(x)))$$

General Knowledge

$$good(isHandsome), good(isThin)$$

$$bad(isUgly), bad(isFat)$$

$$relative(mother), relative(father), relative(brother), relative(sister)$$

Figure 1: An example of goal, action tendency, and action specification for two agents. The left side of the $\mapsto$ symbol represents the preconditions in the case of action tendencies and action definitions in the case of actions. The right side represents the effect. The variables *self* and *target* are references to the agent's self and another in the environment. Unbound variables are introduced with λ-notation.

not analyzed for a solution. Instead, actions are *constructed*. Bad Guy is able to construct logical statements that will change Good Guy's knowledge state. In this case, the posisble statements involve bad things said about Good Guy's relatives. Using an expression-construction algorithm (which is yet to be determined), Bad Guy is able to construct a variety of expressions which have the desired effect. In this case, the expressions constructed equate to "Your relative is fat." and "Your relative is ugly." for each relative. In the current system, the fat and ugly insults would have had to be separately authored, as well as different insults for each family member. The proposed system will enable dynamic and varied dialogue construction for any knowledge-related goals. Even in this simple example, the automated dialogue generation increases the variability of dialogue eight times. With more complex knowledge bases and goals, it is possible that variability can increse exponentially, greatly increasing game dynamicity and reducing the burden on the authors.

4 Design

4.1 Overview of FAtiMA

FAtiMA is an Agent architecture with planning capabilities designed to use emotions and personality to influence the agent's behaviour [5]. FAtiMA consists of a core system (FAtiMA.Core) that allows external modules to be used to implement more complex behaviour. However, this section will focus on the core components.

In order to better understand descriptions of the system, the core concepts and terminology will be introduced here.

Agent Agents are the primary actors in FAtiMA. They represent entities which have emotions and goals, and may take action to affect their surroundings and other agents.

World In order for FAtiMA agents to function, they must exist in a world. A world contains a set of objects and other agents that exist together and can affect one another.

Scenario A scenario consists of information that is necessary to describe the world. In FAtiMA, scenarios are specified in XML files listing the agents and objects in the world.

Appraisal The process of determining how an agent should process an event, and how that event should affect their emotional state.

Coping The process of selecting, based on the agent's current emotional state, which actions to take.

Reactive Level For both appraisal and coping, this level determines instinctive reactions that are executed immediately (e.g. flinching when getting

Action Construction

Goal:

$$\neg isHappy(target)$$

Knowledge about target:

$\lambda x.\lambda r.\lambda p.(knows(target, relative(r) \wedge r(x, target) \wedge good(p) \wedge p(x)) \mapsto isHappy(target))$

$\lambda x.\lambda r.\lambda p.(knows(target, relative(r) \wedge r(x, target) \wedge bad(p) \wedge p(x)) \mapsto \neg isHappy(target))$

Planner infers that the effect:

$$\neg isHappy(target)$$

Can be accomplished with:

$\lambda x.\lambda r.\lambda p.(knows(target, relative(r) \wedge r(x, target) \wedge bad(p) \wedge p(x)) \mapsto \neg isHappy(target))$

This becomes the new Goal.
There are no actions for the planner to select from. Instead it generates speech actions.

$talkAction_1(me, you) \mapsto communicate(me, you, \lambda x.(mother(x, you) \wedge isFat(x)))$

$talkAction_2(me, you) \mapsto communicate(me, you, \lambda x.(mother(x, you) \wedge isUgly(x)))$

$talkAction_3(me, you) \mapsto communicate(me, you, \lambda x.(father(x, you) \wedge isFat(x)))$

$talkAction_4(me, you) \mapsto communicate(me, you, \lambda x.(father(x, you) \wedge isUgly(x)))$

$talkAction_5(me, you) \mapsto communicate(me, you, \lambda x.(brother(x, you) \wedge isFat(x)))$

$talkAction_6(me, you) \mapsto communicate(me, you, \lambda x.(brother(x, you) \wedge isUgly(x)))$

$talkAction_7(me, you) \mapsto communicate(me, you, \lambda x.(sister(x, you) \wedge isFat(x)))$

$talkAction_8(me, you) \mapsto communicate(me, you, \lambda x.(sister(x, you) \wedge isUgly(x)))$

Because p can unify with anything that is *bad*, it can either unify with *isUgly* or *isFat*. Additionally, r can unify with $mother$, $father$, $brother$, or $sister$. This allows many possible speech actions from a few predicates and goals.

Figure 2: Bad Guy's plan for insulting Good Guy's mother with action construction.

hit). These correspond to an agent's *action tendencies*, and represent reactions that the agent has no control over.

Deliberative Level This level enables the agents to construct long-term complex plans for achieving their goals.

Memory FAtiMA supports both episodic and semantic memory. Episodic memory represents short-term memory relating to events that have taken place, while semantic memory is represented as a knowledge base.

Knowledge Base (KB) Knowledge bases are the primary means of long-term information storage for an agent. A KB represents the agent's perceived state of the world. Information is represented through use of predicates representing facts or relations between objects in the world. The predicates used here are similar to those used in logic programming systems such as Prolog.

Goal Goals represent states of the world or of a knowledge base that agents want to achieve. In order to achieve goals, plans are created from actions.

Action An action represents a single effect that the agent may perform on the world. Each action will affect a change on the state of the world or a knowledge base. Currently, actions are pre-defined, but the aim here is to enable construction of those actions which transmit information (i.e. affect a knowledge base).

Plan Plans bring all the previously mentioned components together and make them work. Through planning, agents can determine which sequences of actions (if any) can be used to accomplish their goals.

Sensors Refers to the means by which agents detect events in the world. At the moment, these are triggered by actions taken by other agents or objects. For the purpose of the new system, they will have to monitor changes to knowledge bases.

Effectors Processes through which the agent can affect the world (i.e. actions).

FAtiMA's current interaction model is shown in Figure 3. Appraisal and Coping are executed in a cycle. Actions are appraised, which affects the emotional state, world state, and memory. Coping analyzes these states and affects the world yet again, and these actions are re-appraised.

4.2 Extensions

Figure 4 shows the proposed extensions to the interaction model. Effectors are split into two categories: Action Selection and Dialogue Construction. In addition, a new linguistic interpretation component is added that causes changes in the state of the knowledge base.

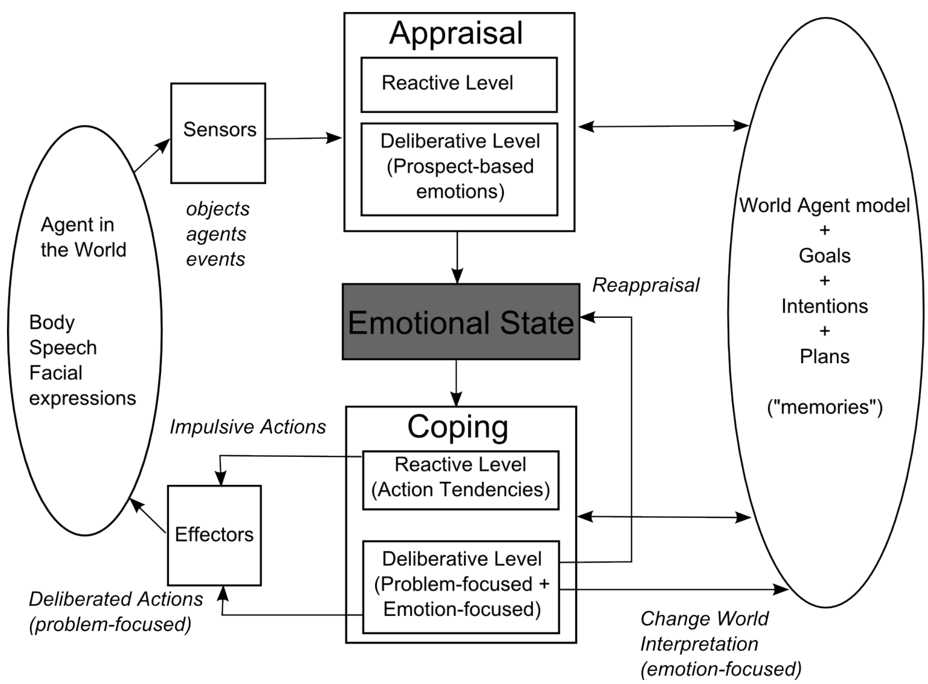


Figure 3: An overview of the current FAtiMA architecture [12].

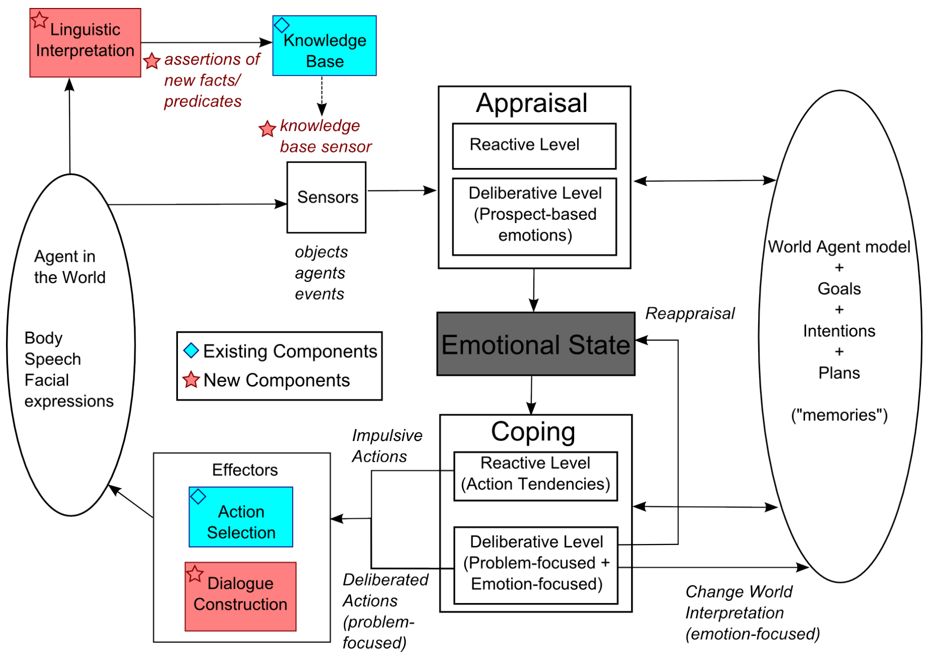


Figure 4: An overview of the proposed changes to the FAtiMA architecture. Objects with diamonds (blue) represent existing components that were not displayed in the previous diagram. Objects with stars (red) represent new components.

The dialogue construction component will require a search algorithm that is able to take the current knowledge base state and goal as input, and produce an optimal assertion that causes the knowledge base to satisfy the goal. It is expected that this search algorithm will be the most difficult aspect of the implementation, as it will likely involve changing the implementation of the knowledge base system itself rather than treating it as a "black box".

The linguistic interpretation module will be simpler, and will most likely be implemented with only small modifications to the existing knowledge base system.

4.2.1 Language Renderer and Derenderer

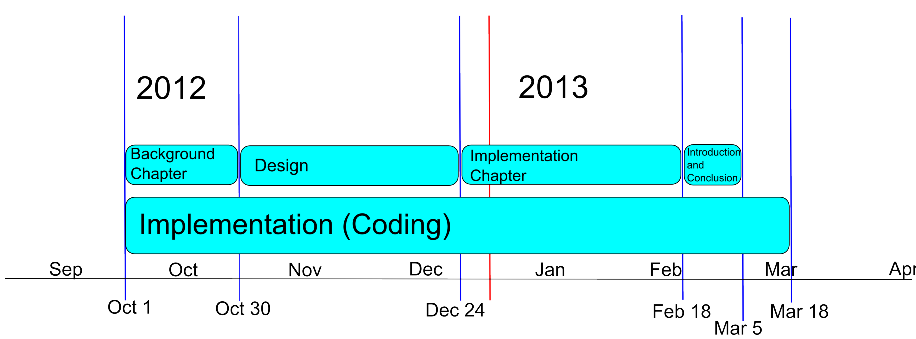
Optional components are the language renderer and derenderer. These, if implemented, would be able to transform natural language text to logical forms and vice-versa. This will only be possible if time and other constraints allow, but will most likely be accomplished through use of OpenCCG [15] software.

5 Conclusion

This thesis will serve as a proof of concept to demonstrate the potential for dynamically generated speech among game characters. Direct generation of dialogue as logical forms has not yet been attempted to my knowledge, and the implementation of this thesis may lead to new posisbilities for interaction with intelligent game characters. Natural language rendering could also greatly improve believability. As games grow in complexity, such technology may become indispensable for the creation of truly dynamic game experiences while alleviating the burden on game authors.

6 Timeline

The following figure shows a proposed timeline for thesis completion. Writing of the thesis and implementation of the system will be done concurrently. The planned completion date is March 21, 2013.



References

[1] Gavan Acton. Playing The Role : Towards An Action Selection Architecture For Believable Behaviour In Non Player Characters and Interactive Agents (Spine Title : Action Selection Architecture For Non Player Characters) (Thesis Format : Monograph) by The thesis by. 2009.

[2] BioWare. *Mass Effect*. Microsoft Game Studios, 2007.

[3] Ecircus. *FearNot!*

[4] Stacy C Marsell, David V Pynadath, and Stephen J Read. PsychSim : Agent-based modeling of social interactions and influence.

[5] Samuel Mascarenhas and Ana Paiva. FAtiMA Modular : Towards an Agent Architecture with a Generic Appraisal Framework.

[6] Michael Mateas and Andrew Stern. Natural Language Understanding in Façade : Surface-text Processing 2 Introduction to Surface Text Rules. pages 1–12.

[7] Michael Mateas and Andrew Stern. Writing Façade: A Case Study in Procedural Authorship Michael Mateas and Andrew Stern 1. I.

[8] Michael Mateas, Andrew Stern, and Georgia Tech. Façade : An Experiment in Building a Fully-Realized Interactive Drama. 2.

[9] Michael Mateas and Noah Wardrip-Fruin. *Prom Week*. Expressive Intelligence Studio, 2011.

[10] Josh Mccoy, Mike Treanor, Ben Samuel, Michael Mateas, and Noah Wardrip-fruin. Prom Week : Social Physics as Gameplay. pages 1–4, 2011.

[11] Josh Mccoy, Mike Treanor, Ben Samuel, Noah Wardrip-fruin, and Michael Mateas. Comme il Faut : A System for Authoring Playable Social Models. 2010.

[12] Ana Paiva. Feeling and Reasoning : A Computational Model for Emotional Characters. pages 127–140, 2005.

[13] Stuart J Russell and Peter Norvig. Artificial Intelligence, A Modern Approach - 2nd Edition.pdf, 2003.

[14] Mark Steedman and Jason Baldridge. Mark Steedman and Jason Baldridge 1 I. (Steedman 1985):1–62, 2003.

[15] Michael White and Jason Baldridge. OpenCCG, 2011.

[16] Luke S Zettlemoyer and Michael Collins. Learning to Map Sentences to Logical Form : Structured Classification with Probabilistic Categorial Grammars. (x), 2004.