

An Embodied Conversational Agent with Autistic Behaviour

by

Wessel Johannes Venter

*Thesis presented in partial fulfilment of the requirements for the degree
of Master of Computer Science at Stellenbosch University*



Department of Mathematical Sciences (Computer Science)
Stellenbosch University
Private Bag X1, 7602 Matieland, South Africa

Supervisors:

Prof. Lynette van Zijl Ms Jana Forrester

January 2012

Declaration

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the owner of the copyright thereof (unless to the extent explicitly otherwise stated) and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Date:

Copyright © 2012 Stellenbosch University
All rights reserved.

Abstract

An Embodied Conversational Agent with Autistic Behaviour

W. J. Venter

*Department of Mathematical Sciences (Computer Science)
Stellenbosch University
Private Bag X1, 7602 Matieland, South Africa*

Thesis: M.Sc. (Computer Science)

January 2012

In this thesis we describe the creation of an embodied conversational agent which exhibits the behavioural traits of a child who has Asperger Syndrome. The agent is rule-based, rather than artificially intelligent, for which we give justification. We then describe the design and implementation of the agent, and pay particular attention to the interaction between emotion, personality and social context. A 3D demonstration program shows the typical output to conform to Asperger-like answers, with corresponding emotional responses.

Uittreksel

'n Gestaltegespreksagent met Outistiese Gedrag

W. J. Venter

Departement van Wiskundige Wetenskappe (Rekenaarwetenskap)

Universiteit Stellenbosch

Privaatsak X1, 7602 Matieland, Swid-Afrika

Tesis: M.Sc. (Rekenaarwetenskap)

Januarie 2012

In hierdie tesis beskryf ons die ontwerp en implementasie van 'n gestaltegespreksagent wat die gedrag van 'n kind met Asperger se sindroom uitbeeld. Ons regverdig die besluit dat die agent reël-gebaseerd is, eerder as 'n ware skynintelligensie implementasie. Volgende beskryf ons die wisselwerking tussen emosies, persoonlikheid en sosiale konteks en hoe dit inskakel by die ontwerp en implementasie van die agent. 'n 3D demonstrasieprogram toon tipiese ooreenstemmende Asperger-agtige antwoorde op vrae, met gepaardgaande emosionele reaksies.

Acknowledgements

Now people were bringing little children to him for him to touch, but the disciples scolded those who brought them. But when Jesus saw this, he was indignant and said to them, “Let the little children come to me and do not try to stop them, for the kingdom of God belongs to such as these. I tell you the truth, whoever does not receive the kingdom of God like a child will never enter it.” After he took the children in his arms, he placed his hands on them and blessed them.

Mark 10:13–16 (NET¹)

I would like to express my sincere gratitude to the following:

- Prof. Lynette van Zijl for her guidance and boundless patience.
- Jana Forrester for her expertise of autism and ever willingness to assist.
- Morné Chamberlain for the countless hours which he unswervingly sacrificed to lend me assistance with technical matters.
- My fellow post-graduate students and friends across faculties with whom I shared this distinct undertaking.

¹Scripture and/or notes quoted by permission. Quotations designated (NET) are from the NET Bible® copyright © 1996–2006 by Biblical Studies Press, L.L.C. <http://bible.org> All rights reserved.

ACKNOWLEDGEMENTS

v

- My parents for the various kinds of support which they have provided over all these years.
- The friends, leadership and members of St. Paul's church in Stellenbosch for their fellowship and love.

Dedications

Soli Deo gloria.

Contents

| | |
|--|------------|
| Declaration | i |
| Abstract | ii |
| Uittreksel | iii |
| Acknowledgements | iv |
| Dedications | vi |
| Contents | vii |
| List of Figures | ix |
| List of Tables | x |
| Nomenclature | xi |
| 1 Introduction | 1 |
| 1.1 Background | 2 |
| 1.2 Goal Statement | 3 |
| 1.3 Motivation and Application | 4 |
| 1.4 Project Name | 5 |
| 1.5 Terminology | 5 |
| 1.6 Thesis Outline | 5 |

| | |
|--|-----------|
| <i>CONTENTS</i> | viii |
| 2 Literature Overview | 7 |
| 2.1 Embodied Conversational Agents | 7 |
| 2.2 Other Work | 28 |
| 3 Design and Implementation | 34 |
| 3.1 Design Specifications | 35 |
| 3.2 Design and Implementation Issues | 38 |
| 3.3 The Thandiwe Engine | 40 |
| 3.4 The Demonstration Program | 50 |
| 4 Evaluation | 57 |
| 4.1 Evaluation Intent | 57 |
| 4.2 Configuration | 58 |
| 4.3 Internal Evaluation | 60 |
| 4.4 Expert Opinion | 64 |
| 4.5 Limitations | 70 |
| 5 Conclusions and Future Work | 72 |
| 5.1 Review of Evaluation Results | 72 |
| 5.2 Future Work | 73 |
| 5.3 Conclusion | 76 |
| Appendices | 78 |
| A Thandiwe Reaction Files | 79 |
| A.1 Document Type Definition (DTD) | 80 |
| A.2 Example | 82 |
| A.3 Discussion | 84 |
| B Character Profile | 86 |
| B.1 Introduction | 86 |
| B.2 Profile | 87 |
| B.3 OCEAN Derivation | 89 |
| C What to say (and not to say) about autism | 92 |
| Bibliography | 97 |

List of Figures

| | | |
|-----|---|----|
| 3.1 | Core System of the Thandiwe Engine | 42 |
| 3.2 | Screenshot 1: Tutor | 55 |
| 3.3 | Screenshot 2: Agent | 55 |
| 3.4 | Screenshot 3: Interactive Environment | 56 |
| 4.1 | PAD Values for First Run | 61 |
| 4.2 | PAD Values for Second Run | 62 |

List of Tables

| | | |
|-----|------------------------------------|----|
| 4.1 | Mood space progression | 59 |
| 4.2 | Typical Results Analysis | 63 |

Nomenclature

Abbreviations

| | |
|------|---|
| AI | Artificial Intelligence |
| AIML | Artificial Intelligence Markup Language |
| ALMA | A Layered Model of Affect |
| API | Application Programming Interface |
| AS | Asperger Syndrome |
| ASC | Autism Spectrum Condition |
| BDI | Belief-Desire-Intention |
| ECA | Embodied Conversational Agent |
| GUI | Graphical User Interface |
| IRC | Internet Relay Chat |
| JVM | Java Virtual Machine |
| NLTK | Natural Language Toolkit |

OCEAN Openness, Conscientiousness, Extraversion, Agreeableness, Neuroticism

OCC Ortony, Clore and Collins

PAD Pleasure-Arousal-Dominance

RDBS Relational Database System

SQL Structured Query Language

TRF Thandiwe Reaction File

VE Virtual Environment

VM Virtual Machine

XML Extensible Markup Language

Chapter 1

Introduction

In recent years, the study of embodied conversational agents (ECAs) has received considerable research attention in the field of computer science. It is envisaged that ECAs will play a host of roles in future human-computer interaction scenarios: from casual informative [13, 81] to pedagogical [45, 72] roles. The success of today's 3D massively multiplayer online role-playing games (MMORPGs) [55] is a clear indication of how comfortable humans are to socially interact with the avatars of other people. A logical next step is to create artificial intelligence (AI) personalities and attach these to avatars to achieve a variety of different roles in the game [98]. Such AI agents could be used as marketing tools (examples in [13, 81]), instructional or pedagogical agents [45, 72], or even simple, high-functioning companions [83]. There are thus many interesting and challenging opportunities in this field of study.

1.1 Background

1.1.1 ECAs as Representations of Human Beings

ECAs are usually designed to resemble human beings [20]. On a superficial level, they have anthropomorphic appearances. On a deeper level, ECAs resemble human beings through their behaviour. Even endowing agents with only the ability to communicate using language makes those agents more human than an animal, as only humans use language to communicate. Believable agents thus mimic human behaviour and traits.

One can consider human behaviour to be driven by two main traits: cognition and affections. Cognition refers to the processes and interplay of perception, learning and reasoning. It refers to the “logical”, “thinking” functions of the brain. Cognition can be simulated in ECAs by making use, for example, of sensors (to facilitate perception), databases (to act as a memory) and various algorithms (to make sense of what is perceived and already in memory, and to formulate a response or reaction to a particular situation).

In addition to cognition, humans also have emotions. While the human being is a thinking, calculating and logical (cognitive) organism, it is also a feeling and impassioned (emotive) organism. Historically, cognition and emotion have been viewed as separate parts of human nature. Modern thinking, however, is that the two are actually tightly integrated (see, for example, [90]). This new way of thinking about emotions allows one, to some degree, to quantify emotions and understand their causes and effects. Closely linked to emotions are also the concepts of personality and moods. These three concepts—emotions, moods and personality—taken together is what we shall refer to as affections. Once it is possible to quantify affections—even if only approximately—it is possible to endow an ECA with affections.

1.1.2 Non-neurotypical ECAs

The two aforementioned traits of human beings have been well studied for application to ECAs [18, 21, 45] and some impressive ECAs have resulted from this work [83]. These studies have all endeavoured to create ECAs which display typical human behaviour, and more specifically, *neurotypical* behaviour. A person who is neurotyp-

ical is one who has had typical neurological development. What interests us, is how to develop an ECA which resembles someone who has not had typical neurological development. Specifically, we are interested in how to develop an ECA which depicts a child that has autism and Asperger Syndrome (AS). This depiction will showcase autistic behaviour, especially as it manifests itself in verbal communication.

1.1.3 Autism

Autism is a brain development condition which impairs social interaction and communication, and causes restricted and repetitive behaviour [102]. It is a spectrum condition—commonly referred to as Autism Spectrum Condition (ASC)—meaning that people who have autism are affected to different degrees. A person who has autism is considered high-functioning if that person can adapt well to society with comparatively little therapy. AS identifies a specific subset of ASC, often at the high-functioning end of the scale [3, 50].

Because the neurological development of a person with ASC is different from that of a neurotypical person, the traits mentioned in Section 1.1.1 are still present, but function differently. For example, it is easier for a person with ASC to experience sensory over-stimulation (as a result of their *cognition*), which can lead to socially inappropriate behaviour as a result of anxiety (which is a result of their *affections*).

Note that in this thesis we shall be using the terms “Asperger Syndrome” and “autism” interchangeably. The reason for this is that the American Psychiatric Association (APA) is planning to remove AS as a diagnosis separate from autism in the next revision of the *Diagnostic and Statistical Manual of Mental Disorders* (DSM-5) [36].

1.2 Goal Statement

Our goal is to develop a prototype system which allows for the construction of rule-based conversational agents which exhibit autistic behaviour [100, 101]. Our premise for believing that we can effectively implement an agent using rule-based systems (as opposed to more sophisticated AI methods), is that even basic conversational systems can be convincing if designed for a specific purpose and placed in a specific context [63, 76]. This is important because, as mentioned in the previous section,

a person with ASC has the same core traits as a person who is neurotypical. We therefore want to convince an audience of this just as we would want to convince them about any other agent. The difference lies in how we wish to present these traits. Rule-based systems will allow for the necessary configuration of the traits mentioned in the previous sections.

To demonstrate the prototype system, we further propose to develop a prototype ECA with autistic behaviour. This will be an enhancement to our mainly conversational system which adds a visual component, “embodying” our agent in some virtual environment (VE). ECAs allow for non-verbal interaction by being able to communicate with the user through the use of body language (see for example [23]). Having an embodied agent is important for our purposes, as people with ASC often exhibit unexpected body language. Body language may not be completely lacking in an individual with ASC, but could be restricted or appear clumsy. The same also applies for facial expressions [50, p. 38]. To be able to demonstrate these differences in body language would help us to portray autistic behaviour. Being only a prototype, however, our ECA will be primitive.

We make no assumption as to who the intended audience for our system can be: we believe that a broad range of applications will be possible, and that both neurotypical and people with ASC could benefit from them.

Due to the purview of our proposed system, we intend to combine existing technologies in order to create a versatile system which would allow for the creation of believable agents with ASC.

1.3 Motivation and Application

While we stated in the previous section that we do not make any assumptions as to our audience, we do have in mind some possible applications for agents created using our system. One is that the agents can be used to educate neurotypical people about AS by exposing them to typical AS behaviour. Another application could be as an educational tool for children with ASC by providing them with a tutor or companion with which they can relate. See Section 5.2.2 for what we think are viable possibilities for future research using our system.

1.4 Project Name

“Thandiwe” is a gender-neutral Nguni name. This name means “loved one” and reflects the hope that the work presented in this thesis will in some way contribute to a better general understanding of ASC, which in turn will lead to greater acceptance by the public of people who are on the autism spectrum.

The name Thandiwe is associated with the main project implementation—the *Thandiwe Engine*—as well as with various lesser components of the project, such as the *Thandiwe Response File* (see Appendix A).

1.5 Terminology

In this thesis, the following conventions will be followed regarding terminology. *Agents* will refer to autonomous computer characters. *Avatars* will refer to computer characters who are not autonomous, but rather are directly controlled by a human being.

The term Asperger Syndrome will be used *in lieu* of Asperger Disorder, as the latter term is often considered offensive. It also poorly characterises the perception some people with AS and their families have of AS. Further, the terminology of [50] and the recommendation of the British National Autistic Society (see Appendix C) will be followed. Accordingly, there will be referred to people *with* AS, rather than “Asperger people”.

Neurotypical people are those who have had “normal” neurological development, as opposed to people with autism (and AS) who have had atypical neurological development.

1.6 Thesis Outline

The rest of this thesis is structured as follows:

Chapter 2 surveys the relevant literature on ECAs. The designs proposed in the literature are discussed, as well as user feedback on the specific ECAs. A sample of

selected literature which is indirectly related to ECAs is also considered for pertinence.

Chapter 3 looks at the details of our actual implementation and the design behind it. The various components which comprise the system are discussed, as well as how the components function as a whole. The chapter also details a prototypical test implementation built on top of the framework and explains the design choices that were made in this test implementation.

Chapter 4 describes the evaluations which were done for this project and reports the results which were obtained. The test implementation is used to evaluate the usefulness of the underlying framework. The evaluation was done by analysing the inner dynamics of the system, as well as by presenting the system to an expert in autism and considering their reported feedback.

Finally, in Chapter 5, the implementation of the Thandiwe system is evaluated and a summary is given of our findings. The direction of future work is also discussed.

Chapter 2

Literature Overview

This chapter presents an overview of literature concerning the field of embodied conversational agents, especially those related to autism. The first section of this chapter focuses on ECAs in general. The second section considers literature indirectly related to ECAs.

To our knowledge, no previous attempt has been made to create an ECA with a non-neurotypical personality.

2.1 Embodied Conversational Agents

We make the following definitions with respect to ECAs [23]:

Definition. *A conversational agent is a computer program which accepts a possibly restricted natural language input from a user and then provides a possibly restricted natural language response.*

Definition. *An embodied conversational agent (ECA) is a conversational agent which has a digital graphical (2D or 3D, usually anthropomorphic) representation.*

Definition. *A virtual environment is a space (usually 3D) represented in a computer program which is navigable by an ECA or user. It can contain various kinds of objects. The ECA or user can often interact with some of these objects, while others allow no interaction. A virtual environment can represent a real world location or environment.*

Following these definitions, there exists a profusion of literature on ECAs. We therefore chose to limit ourselves primarily to only those areas of research which are relevant to our research.

2.1.1 ECA Design Considerations

There are many facets to ECA design and implementation. At one level there are internal considerations, such as how the agent will reason and whether it will have mechanisms to drive affections or not. One must also think about how the agent will communicate with the user. At another level, there are also external considerations, for example: how the agent will be animated, how the agent should look, and how the agent will be able to interact with its environment. This section will review literature which concerns such design and implementation facets.

2.1.1.1 Cognition

“Cognition” is the term for the processes and interplay of perception, learning and reasoning [11, pp. 15–17] [31, p. 9]. It refers to the “logical”, “thinking” functions of the brain. For an ECA to function in any way, it needs at least some mechanism whereby to reason. Naturally, the efficacy of an ECA increases if the abilities to perceive and learn are added. Computers cannot yet reason fully about the world as human beings can. For this reason computers need to be programmed to reason according to theories of cognition, such as the Belief-Desire-Intention model, which is discussed in Section 2.2.2.

Bringsjord et al. [18] contend that, at some point, mere behaviourism (designing agents to fake intelligence through their behaviour or reactions—see for example [63]) always breaks down at some point. They thus shun rule-based systems and conventional AI practices and attempt to create a truly “aware” and reasoning computer

program. Specifically, they aim to create an agent which can reason about the state of mind of another entity—whether it is a human or another agent. In other words, the authors aim to endow a computer program with a *Theory of Mind*. Possessing a Theory of Mind is to have the ability to reason about the thoughts, emotions and reasoning of other people [7]. In order to do so, the authors are working towards mathematically codifying principles of “common-sense psychology” (CSP). When they have such mathematical constructs, they can use an automated theorem prover to reason about more complex abstract concepts, such as false belief.

The result of this work is Eddie, an autonomous agent which lives in Second Life¹. Eddie represents someone who does not have a Theory of Mind, similar to either a neurotypical child of less than approximately four years of age, or a child with autism, as some have suggested [8]. This is demonstrated by Eddie failing the Sally-Anne test². Eddie is the first step in building an agent which will be able to attribute false beliefs to others, and thus attain a certain sense of intelligence. A parallel endeavour by the authors is to implement natural language processing (NLP) inside an MMO (Second Life). This is done by transforming text to an unambiguous subset of English and then applying a multi-sorted logic (MSL) [84] to the text. The MSL not only allows meaning to be extracted from the English text, but to reason about that meaning as well.

The authors admit that their research is aimed at the entertainment industries rather than for pure scientific pursuit. They also admit that they are not concerned with computational efficiency, but rather in creating a program which can successfully apply CSP in certain scenarios. Finally there is the admission of an impasse: it is necessary for them to define abstract concepts such as belief, desire, hope and fear in such a way that it can be represented computationally. This is still an open problem.

While this project is ambitious and impressive, it is of little use to us. The authors are neither interested in rule-based solutions, nor in having a system which is widely

¹Second Life (<http://secondlife.com/>) is a massively multiplayer online game (MMO) where users inhabit virtual worlds embodied in avatars. It is primarily used for socialising, but it has recreational and commercial uses as well.

²The Sally-Anne test measures the ability of someone to attribute false beliefs to someone (that is, whether someone demonstrates a Theory of Mind). The test administrator, the subject and an observer are together in a room with two empty baskets (A and B) and a toy. The administrator places the toy into box A and then the observer leaves the room. While the observer is away, the administrator moves the toy from box A to B. The observer returns and the administrator asks the subject in which box the observer will say the toy is. People who can attribute false beliefs to others will say box A; those who cannot will say box B.

accessible. The Thandiwe system should be simple and not bound to expensive hardware.

2.1.1.2 Affections

As mentioned in Section 1.1.1, modern theories of human thinking emphasise the interconnection of logical and calculated thinking, and the emotional side of humans. A person who is more *logical* looks at a situation from a more neutral and detached standpoint. A person who is more *feeling* might invest themselves more emotionally in a situation and empathises with the parties involved. That said, these two extremes are not absolute and any human being (having both the ability to reason and feel) lies on a spectrum which stretches from extreme thinking to extreme feeling [87].

As with people who are neurotypical, people with AS are also both thinking and feeling. It would therefore be in our interest to incorporate affections into our framework if we wish to accurately portray people with AS. It is interesting to note that, where neurotypical people and people with AS do differ, is that people with AS are typically more varied in their emotions and moods. At times they may seem very calculated and even cold. At other times they can be overwhelmed by emotions. While this is true even for neurotypical people, the timespan between these two extremes can be exceptionally short for people with AS [3, 50]. It would therefore be interesting to model these dynamics in our framework.

In recent years, research has somewhat moved away from trying to build purely reasoning systems. Rather, researchers have come to recognise the importance of affections in intelligent systems. We now consider models of affect from the literature.

Models of Affect. The most popular models of affect in the literature are the OCEAN model of personality [66] (also known as the “Big Five” factors or the “Five Factor Model”) and the OCC model of emotions (named after the authors Ortony, Clore and Collins [90]). The former model claims that the personalities of all people can be described by five dimensions: *Openness*, *Conscientiousness*, *Extraversion*, *Agreeableness* and *Neuroticism*. As part of the fundamental character of a human being, the OCEAN values of a person are usually considered to be constant.

The OCC model defines 22 fundamental emotions for any human being. According

to the model, these emotions can always be seen as being present, although one normally dominates, eclipsing the others. The model also describes the effects which events, objects and agents (usually other people) can have on these emotions. The stimulus from an event, object or agent can change the balance of the emotions, perhaps making another one the dominant emotion.

Another model of affect is the PAD model of moods [79]. According to this model, the mood of a person can be defined in a 3D space. The three axes are *Pleasure-Displeasure*, *Arousal-Nonarousal* and *Dominance-Submissiveness*. This provides for nine basic moods: *Exuberant*, *Dependent*, *Relaxed*, *Docile*, *Bored*, *Disdainful*, *Anxious*, *Hostile* and *Neutral*.

ALMA. Gebhard [45] developed *A Layered Model of Affect* (ALMA). While other models of affect usually only focus on personality and emotion, ALMA additionally includes moods as an intermediary level between personality and emotion. ALMA uses OCEAN and OCC for personality and emotion, respectively. For moods, the system implements the PAD model of moods. Personality is viewed as being constant, but emotions can change quickly. Moods are thus medium-term. At any point in time, an agent has a *current* mood, a *default* mood and a mood towards which the current mood *tends*. The current mood is the mood which an agent is in at the time in question. The mood tendency is the mood towards which the agent is being moved due to the emotions being experienced. These emotions are caused by external factors, such as events, objects and agents. The default mood is defined by the personality. If there is no external stimulus, the mood tends towards the default mood.

Per-agent data files determine how moods and emotions change according to personality over time and with different kinds of external stimuli. External stimuli are represented by appraisal tags, which can roughly be likened to the PARs discussed in Section 2.1.1.7. The authors also explore the effects of emotions on the believability of ECAs [46, 47, 48]. Their findings show that people respond better to agents which implement the ALMA system than to those which do not have a system to provide affections.

ALMA is, to our knowledge, the most comprehensive model of affections which has been used in ECAs of which we are aware. For that reason, we selected ALMA to handle affections within the Thandiwe system.

Kshirsagar. Kshirsagar [69] designed a system similar to that of Gebhard. The system, like ALMA, also makes use of the OCEAN and OCC models for representing personality and emotions, respectively. It differs in its implementation of moods. Rather than using the PAD model, this system uses a more primitive model. Whereas the PAD model operates in three dimensions, Kshirsagar only uses one (*good-neutral-bad*). Additionally, this system is not rule-based, as is the one of Gebhard, but rather employs a Bayesian Belief Network (BBN) [108]. This allows for some degree of randomisation between moods and emotions, something which the author claims is more realistic than the rigour of a rule-based system. The system has a dialogue capability, based on a variation of the AIML standard³. The standard AIML was adapted so that the internal emotional state of the agent determines the given outputs. The emotional effect on the agent of a user response is also encoded into the AIML. Yet another modification made was to the emotional model. The author chose to closely link emotions to facial expressions; to quote from Section 2.2 of the article: “By emotion, we understand a particular state of mind that is reflected visually by way of facial expressions”. The 22 OCC emotions—along with two additional emotions *surprise* and *disgust*—were thus mapped to the six fundamental facial expressions defined by Ekman [39]. These six categories of emotions—rather than the emotions themselves—are used to construct the transition matrices for the BBN.

For our purposes, we prefer the more sophisticated mood model of ALMA over the one presented by Kshirsagar, as it would provide a more comprehensive way of modelling affections. We believe that this will contribute to a richer modelling of autistic behaviour. However, we shall follow the practice of adapting AIML to accommodate affective actions and reactions, but do so indirectly, rather than directly (but see Section 5.2.1).

2.1.1.3 Verbal Communication

One of the most striking characteristics of people with autism is their struggle to communicate, both verbally and non-verbally [50, pp. 34–38]. In this section we consider verbal and non-verbal communications in ECAs.

³AIML is an open, XML-based chatterbot standard. For more information on AIML and how it is used in the Thandiwe system, see Section 3.3.2.

Cassell has done prominent work concerning verbal [12, 13] and non-verbal [21, 22, 24, 25] human-ECA interaction. We first consider verbal communication. In the section following hereafter, non-verbal communication is considered.

Bickmore and Cassell [12] emphasise the importance of small talk and conversational storytelling between ECAs and users. The reason for this emphasis is that both of these types of verbal communication are important in human-to-human verbal communication and users tend to assign human qualities to ECAs. Humans therefore have similar expectations of verbal communication with ECAs.

Small talk is a “light” exchange between parties on neutral topics. It is perhaps most important when there is a degree of unfamiliarity between the parties involved in a conversation. In general it serves a number of roles. From the perspective of an ECA, these roles are

- putting the user at ease and building solidarity,
- transiting away from awkward moments which can arise during a conversation,
- establishing certain facts about the user, and
- asserting its expertise in a particular domain of knowledge.

Related to small talk is the notion of “face” in a conversation. “Positive face” is the desire by a person to be liked by others. “Negative face” is the desire to assert authority over others. Small talk is a means of establishing face for oneself, as well as to fend off face “threats” by others⁴.

Bickmore and Cassell assert that *conversational storytelling*—that is, storytelling which is considered to be legal historic narratives rather than fictitious and embellished narratives [89]—is another important part of human-ECA discourse. Mauldin [77] has already noted that storytelling has been used to achieve greater realism in chatterbots. As with small talk, conversational storytelling can serve various roles:

- **Phatic:** Storytelling can be part of small talk.

⁴The reader may note that people with autism find small talk difficult, awkward and discomfiting [3, Ch. 7]. People with autism converse factually and take comments at face value and literally. If we were to incorporate small talk into an ECA with AS, it would have to be more fact-driven and relevant to what concerns the agent.

- **Establishing expertise:** To assert authoritative knowledge over some knowledge domain by relating relevant stories.
- **Encouraging self-disclosure:** One party can elicit personal details from another party by first disclosing some personal information through a story.
- **Persuasion/Problem solving:** Sketching scenarios—possibly related to the current one—to illustrate some point.
- **Providing requested information:** Stories can be used to answer questions.

Characteristics of conversational storytelling are that they are usually unprepared and that they are not monologues, but are part of dialogues and as such there is still interaction going on between the different parties. An effective conversational story should also be something which is relevant to the listener. It is also necessary to place a story correctly in a discourse.

The authors conclude their discussion by noting that an agent which is capable of effective small talk and conversational storytelling does not need to be AI complete (that is, a true, fully functional and fully reasoning AI system which has the intellectual and reasoning capabilities which match or surpass that of a neurotypical human being). The reason for this is that there are strict rules regarding small talk and conversational storytelling. These have been carefully studied in human-to-human interactions and can be emulated by a computer system. They stress that, for such a system to be effective in its purpose, it needs to be designed well.

2.1.1.4 Non-verbal Communication

Non-verbal interaction complements verbal interaction and indeed is essential in face-to-face communication between humans [57, Ch. 7]. To study non-verbal interaction with ECAs, Cassell and Thorisson [24] (using established psychological theories regarding inter-person communication) focus on two kinds of feedback which can be given through the use of body language: *emotional* and *envelope*. Emotional feedback describes an emotion to the hearer without the speaker necessarily having to feel that emotion. Envelope feedback concerns the process of a conversation. An example of envelope is the subtle facial movements which indicate that the speaker has finished speaking and is surrendering the floor to the hearer. Envelope feedback

involves facial expressions, gaze and hand gestures. The authors hypothesise that envelope feedback is more important to a human who is interacting with an ECA than emotional feedback is. Until the study in question, most research had focused on emotional feedback. To test their hypothesis, the authors used a system which could be finely tuned to give envelope display. Using this system they built three characters which all looked different, but functioned exactly the same internally. Users then had to interact with the characters. A character could provide content only; content and envelope; or content, envelope and emotional display. The users were assessed using a questionnaire and through analysing video recordings of the interactions. The results showed that non-verbal communication was important for the users who interacted with the ECAs and that of the two kinds of non-verbal communications, envelope was more important than emotional.

In Cassell et al. [22] the authors develop a rule-based system which links non-verbal communication and verbal communication. The system allows intonation, hand and facial motions, and hand gestures to complement what is spoken by the ECAs. (Such linking is called “synchrony”.) At the heart of this system lies a BDI dialogue planner (see Section 2.2.2). A speech synthesiser is used to produce audible verbal replies⁵. The decision to make use of a speech synthesiser is supported by the fact that the timing of gestures in relation to speech utterances is crucial. Gestures are generated when the semantic content of what is being said can be related to spatial displays (that is, what is being said can be either literal, metaphorical or spatialiseable). The coordination between the gestures and the utterances is done using *parallel transition networks* (PaT-Nets) [5]. PaT-Nets allow for coordination rules to be represented as automata which execute in parallel where transitions happen either conditionally or probabilistically. Various Pat-Nets are used to control gestures (body and hand) and gaze. A separate system, the *Facial Action Coding System* (FACS, a system which provides a taxonomy of human facial expressions) is used for generating facial displays.

See Section 2.1.2.5 for a result of these studies of verbal and non-verbal ECA communication.

Again, people (especially children) with AS have difficulty interpreting emotional and envelope feedback and often apply it incorrectly [3, Ch. 6] [50, Ch. 4]. They

⁵Due to the limitations of current speech synthesisers, many people who implement ECAs (including ourselves) opt to not make use of them. “Verbal” response is given either using pre-recorded speech, or (as in our case) to just give textual output on a screen.

are often described as having “poker faces” [3, p. 133] [50, p. 140]. The lack of subtle communication hints which are normally present in the face and body can unnerve neurotypical people who have never before dealt with children with AS. Problems with interpreting emotional and envelope feedback range from simply not understanding the significance of the nuances in the other person’s body language to simply missing it, because they were, for example, looking at the person’s mouth rather than at the eyes.

These facts have an impact on how we implement envelope in our system. It might seem that Cassell’s work is not directly applicable to our own. However, the work is in fact interesting in our case, in so far as it highlights the subtleties of communication. We need to decide whether the different subtleties are applicable to our purpose or not. An ECA with AS will still display unintentional envelope, for example, such as fidgeting when uncomfortable. But the agent probably will not engage in intentional gesturing to emphasise what is being said.

2.1.1.5 Animation

A puppet being controlled by a puppeteer is more life-like than one lying on the side, not being used. Adding movement to an otherwise still and rigid anthropomorphic representation can thus be important in making that representation seem believable. Similarly, ECA realism can be enhanced by animating the agent (that is, adding movement to an otherwise static 3D model). Indeed, if one wants to implement any non-verbal communication for an ECA (as discussed in the previous section), animation is essential. Animation has traditionally been done through keyframing or motion capture [33]. As such low-level animation is time-consuming, systems have been developed to specify animations on a higher level. One such system has been developed by Huang et al. [58, 59, 60]. Their system, called STEP, provides a high-level abstraction for animating 3D characters. STEP scripts typically consist of primitive actions, composite operators and high-level interaction operations. Huang et al. [61] describe how the STEP system was adapted to make use of XML.

Other similar systems exist, for example Jack (http://www.plm.automation.siemens.com/en_us/products/tecnomatix/assembly_planning/jack/index.shtml). However, from the point of view of our own implementation, these systems are either difficult to incorporate into new projects (such as Jack), or suffer

from the typical problems associated with non-commercial software (no support, outdated graphics standards, lack of documentation, *et cetera*). Hence, since the Thandiwe project requires little animation, we chose a more traditional solution (see Section 3.2 for more detail).

2.1.1.6 Realism and Consistency in Behaviour

One concern when designing an ECA is how the user will respond to the embodiment of the agent. The *Realism Maximisation Theory* suggests that the more realistic an agent appears, the better a person will respond to that agent [78]. The *Uncanny Valley Theory* states that humans respond better to a realistic (in both behaviour and appearance) agent only up to a certain point, after which added realism will disturb the user [86]. The *Consistency Theory* asserts that humans will respond best to an agent which has consistent behaviour, regardless of the realism of that level [88]. Groom et al. [51] explored these theories by testing user responses towards various types of agents. They found that humans responded best to agents which were not super-realistic and which exhibited slightly inconsistent behaviour. Their study thus provided support for the Uncanny Valley Theory. This provides a guideline along which efforts should be directed when designing an agent to which a user should respond best.

The interesting question in our case is of course how these theories apply in the case of an ECA with AS. For example, in educational software for children with AS, would they respond better to an ECA with AS, or not? And how exactly would one define “consistency” and “realism” in the case of an ECA with AS? The reader is reminded that this thesis simply sets out to develop the framework for the use of an ECA with AS, but these questions would definitely be interesting issues in future work.

2.1.1.7 Environment

How ECAs interact with their environment is not a trivial issue. Analogous with robotics, the more an ECAs has to interact with its (3D) environment, the more complex the underlying mechanisms become. We consider below one proposed solution to how ECAs can interact with an environment.

Allbeck and Badler [2] propose the concept of *Parametrised Action Representations* (PARs) as a standard for the manner in which agents interact with a virtual environment. PARs are essentially tags containing metadata which are added to world objects. When an agent interacts with an object, the tag of that object is read. The tag contains information such as, for example, what the object can be used for and how it can be used (such as grasping points if an agent wants to pick up an object). PARs are interpreted by a system called PARSYS. A further system, called EMOTE, combines PARSYS with Leban Movement Analysis (LMA) [85] and actual theories from psychology to represent personalities and emotions discreetly. For personalities, the OCEAN model is used. For emotions, the OCC model is used. EMOTE thus enables agents with different personalities and emotions (defined by a special agent PAR) to interact differently with objects based on personality and mood.

PARSYS is definitely worth considering when implementing an ECA which must interact with a VE in a complex way. But because our focus is on conversational (rather than interactive) agents, we decided to forgo including PARSYS in our system in order to narrow our scope. As for EMOTE, when using the OCC model, one already has a defined mechanism whereby emotions are influenced by objects (see Section 2.1.1.2).

2.1.2 ECA Prototypes

Having considered various aspects which need to be taken into account when designing an ECA, we now turn our attention to ECAs which have already been implemented. The ECAs which we consider in this section have mostly arisen from formal academic studies.

2.1.2.1 COSMO

Lester et al. [72] developed COSMO as an educational and instructional tool. The central idea of COSMO was to make computer-based learning engaging and effective through an ECA which made use of strong emotional displays, such as exaggerated gestures and facial expressions. The framework around which COSMO was built, was termed the *emotive-kinesthetic behaviour sequencing* framework, which reflects this central idea. The hypothesis was that an emotive and engaging ECA would

quickly build rapport with a user and that subsequently its energy and enthusiasm would be transferred to the user.

The implementation of COSMO consisted of a three phase process:

1. **Emotive pedagogical agent behaviour space design:** In this phase, the possible behaviours of the agent are defined, primarily through animation and sound.
2. **Speech act-based behaviour space structuring:** This phase primarily structures the behaviours defined in the previous phase. This is done mostly according to speech acts. It thus correlates the emotive states of the agent with what it is communicating at that time. In the case of COSMO, a one-to-one mapping of emotive states and communication exists for a restricted collection of acts.
3. **Full-body emotive behaviour sequencing:** This phase orchestrates the structures, defined in the previous phase, with user needs and accomplishments. Mechanisms exist to identify when a student is struggling with a problem. If such a situation is identified, a series of behaviours is strung together to assist the student in reaching a particular goal. If the student succeeds, a series of congratulatory or encouraging behaviours is displayed.

In considering COSMO, we recognised that our focus is fundamentally different from that of COSMO: we not interested in synchronising instructive communication, but in emulating AS behaviour. Furthermore, COSMO does not have internal states, but only predefined emotions. Clearly, the use of internal states will allow for more flexibility in many end applications. Finally, COSMO always exaggerates emotions, while our aim is to accurately portray emotions, whether normal, exaggerated or subdued (whatever the case may be for some particular situation). We do share design goals with COSMO by limiting (at least initially) our development to a one-agent-to-one-user scenario. Speech synthesis is outside the scope of this work, but our framework is to be designed in a modular fashion (see Section 3.1.3), so that the later addition of speech synthesis will be possible.

For an overview of pedagogical ECAs, see Johanson et al. [65].

2.1.2.2 Lokutor

Milde [80] implemented an ECA called Lokutor. The aim of Lokutor is to directly link sensory input and the deliberative process. Lokutor can “sense” the world by accepting sensory input. Sensory input comprises visual and tactile perception, which is achieved through detecting specially tagged objects in the virtual world.

The control architecture comprises three parts:

- a *deliberative system*, which breaks up complex actions into sequences of simpler actions;
- a *behaviour system*, which is embedded in the virtual world. The behaviour system comprises a collection of sensors and actuators and allows for basic actions on the part of the ECA; and
- a *motivational system*, which monitors the internal state of the behaviour system.

The deliberative and behaviour systems work in tandem to enable the agent to plan and execute actions. The former breaks a complex task up into simpler tasks, which the latter system can then execute.

Language processing was also important when Lokutor was developed. A domain specific grammar and lexicon were developed which were pre-compiled into the project to speed up speech recognition. The author claims that training the speech recognition component is not necessary and the user can speak natural and continual sentences. As long as the speech is in English and part of the knowledge domain of the agent, what the user says is translated into text, which is broken down into attribute-value pairs. This text-processing is done in two ways:

- a *finite state transducer* (FSTN) is used for shallow processing; and
- a phrase structure rule-based *chart parser* (CP) is used for deep processing.

Natural language instructions are broken down into action-value pairs by the CP. Action schemes then map high-level actions into basic actions. These are then mapped

to internal sensors, which facilitate communication between the deliberative system and the behaviour system.

Much of the focus of Lokutor was having the agent interact with the environment. As mentioned in Section 2.1.1.7, our primary goal is to build a conversational agent and not an interactive agent. We do note that the emotional state of an agent with AS can be influenced by external stimuli. However, we explain in Section 3.3.5 how we integrated these stimuli into the Thandiwe system.

2.1.2.3 Max

Kopp et al. [68] built Max, which served as a museum guide at a computer museum, in order to gauge how the ordinary public would interact with an ECA. This allowed the authors to observe Max without having the bias of users who know the mechanics behind the ECA, as can be found in the laboratory. Max accepts input from a keyboard and replies using a synthesised voice. It also uses its body during communication, making use of gestures and facial expressions. It can also track people with its eyes.

Internally, Max has an emotional component and a deliberative component, which are separate. The former affects how Max communicates with people, and also handles its displays (for example, its facial expressions). Max's emotions can, of course, also be affected by what people say. The deliberative component employs a BDI architecture (see Section 2.2.2, where we give a brief description of BDI models). The knowledge (beliefs) of the agent is separated into static (long-term) knowledge and dynamic knowledge. The former contains various persistent rules and axioms needed by the system to function, as well as a memory of past user interactions. The latter contains information about the immediate context, such as information about the person or persons it is currently engaged with and a history of exchanges with them. The internal states of the agent also allow it to take the initiative, rather than only being reactive. If the deliberative component fails to provide a plan for output, it defaults to simple rule-based reactions.

Max employs *decision-action* rules in its planning. Each rule can test either user input or internal states. One or more such rules can then be used to decide on some action. This rule system is based on AIML (see Section 3.3.2). A rule specifies a

communicative function, comprised of three parts to provide a dialogue act for the ECA to perform:

- **Performative:** the action which the dialogue act should achieve.
- **Reference level:** the level of the dialogue act, which can be “content”, “discourse” or “course of interaction”.
- **Content:** the information, such as a query from the user or some interactional signal from the user.

When the system receives input, it preprocesses the text and then tests it against the available condition-reaction rules. Such rules (which are usually atomic) can be used by the BDI-based JAM planner [62] (BDI is described in Section 2.2.2) to make up both long-term and short-term plans. Long-term plans are plans which can be accomplished over time, such as returning later to ask a user for their name if the user initially refuses to give it. Short-term plans are plans which must be fulfilled immediately, such as how to answer a question which has just been posed by a user. The planner finally selects a response, comprised of an utterance and/or display. If both an utterance and a display are part of the response, the two must be synchronised. To this end, actions were mapped to communicative functions. When a response is then given, all appropriate actions are selected. This selection is refined based on factors such as what would be better in this instance. Deciding factors can be the position of the hands (to transit from one behaviour to another), selecting the next action as the one which would provide for the most fluid transition. This final selection of actions is then used to accompany the utterance.

During trials, on the whole, the public responded positively towards Max, interacting with it and attributing social qualities to it. The authors acknowledge certain factors which may have hampered interaction from the users’ side, such as the fact that input had to be given via a keyboard. The authors also noted a need to enhance certain social behaviours of Max—such as flirting—as the members of the public often tried to engage Max in such behaviour.

While the end products are quite different, our system shares some ideas with Max, such as extending AIML, and that the internal affective state of the agent determines its response (see Section 3.3.5). Max is more advanced in its deliberation, however. We contend, however, that while a BDI architecture could conceivably have been

used in our solution, it would not necessarily have been practical for our purposes (see our discussion on BDI in Section 2.2.2). Also, the only affections which Max has are emotions, while we consider it important for our purposes to have the full range of affections (emotion, mood and personality).

2.1.2.4 Milo

Milo is a commercial ECA from Microsoft and Lionhead Studios, which was developed specifically for the Xbox 360 gaming console (see Molyneux [83]). Milo represents a boy with whom a player can interact using the Kinect motion detection system for the Xbox 360 gaming console. The user is able to interact with the virtual world within which Milo lives and with Milo himself. As the user interacts with Milo, its personality is shaped based on what the user does. The user can even speak to Milo. Rather than using standard speech recognition, spoken words are matched to a database of previously recorded words. The authors claim that this—along with making use of intonation—give much better results than trying to match the sounds which make up a word.

As Milo is as yet unreleased and proprietary software, almost nothing is known about the inner mechanics of the game. Molyneux admits, however, that “most of it is a trick, but it is a trick which actually works”. This implies a mixture of advanced (and proprietary) AI technologies and standard “tricks” to make ECAs more believable (see Section 3.1.3). Milo is also a learning agent, but does not merely learn from a single user. A part of Milo resides on a cloud (distributed online storage) and is contributed to and extended by all users. This means that Milo’s learning is a distributed, community effort and can constantly be refined.

Milo certainly appears to be a quite advanced ECA and holds great promise for the future of ECAs, especially if a software development kit (SDK) could be made available for academic use. At such a time, it could be interesting to adapt the distributed learning aspect of Milo to learn the behaviour of an agent with AS.

2.1.2.5 Rea

Rea is a real estate agent designed to showcase how an agent with a specialised knowledge domain can apply techniques borrowed from studied human-to-human

conversations to build rapport with a user to the end of better communicating its knowledge to that user [13, 20]. The agent operates in real-time and can interpret a user's voice and body movements (including gestures and gaze). Rea itself uses its body—along with speech—to communicate with the user. For discourse planning, the authors decided to move away from conventional static world discourse planners and to implement a so-called *activation network* planner. Their motivation was that an activation network is better at rapidly shifting between different types of discourse. This is necessary in the Rea agent, as storytelling and questioning, for example, can all be interleaved. A conventional planner could begin to yield poor results for the rapid pace at which goals can be invalidated during real-time face-to-face conversation. Each node in the activation network represents a *joint project*. A joint project is a singular goal which may take multiple dialogue turns to accomplish. The selection of a joint project by Rea is influenced by a number of factors:

- **Solidarity:** Certain dialogue courses require the agent to have a certain level of solidarity with the user. If the agent does not believe that this level of solidarity has been met, then it will pursue small talk to build solidarity with the user.
- **Topic:** Rea keeps track of current and previous topics. It is more likely to select a joint project which lies within the scope of the current topic than one which does not.
- **Relevance:** The planner will favour topics with which it thinks the user is familiar.
- **Task goals:** Joint projects are selected which will further the current internal goals of Rea.
- **Logical preconditions:** Certain utterances do not make sense until others have been spoken or some information about the user has been established. The planner takes such logical preconditions into account when deciding on a discourse response.

Note that in some cases it is necessary for Rea to make a “best effort” in selecting a dialogue option, as it must deliver a response to the user in a reasonable amount of time.

Experiments showed that users had a stronger preference for Rea when it used small talk while trying to sell the user a house than when it did not.

When Rea was created in 2000, it was necessary for her to run on a distributed system, requiring “several” computers. The system is definitely not intended for desktop usage. Rea is an advanced agent, and similar capabilities fall outside the scope of this thesis. However, this agent was interesting to us because its success was based on small talk, which is exactly what a user with AS would *not* like in agents.

2.1.2.6 SimHuman

SimHuman is a project created by Vosinakis and Panayiotopoulos [103] to develop a stand-alone system which strikes a balance between believability, accuracy and realism on the one hand, and real-time workability on the other hand. The system includes a VE, agents and objects which can be animated, and planning and reasoning mechanisms.

Keyframing is used for animation. SimHuman also has a library of pre-computed body postures (the so-called *animation library*) and can load data from geometry files.

The system implements a complete physics engine. It allows for collision detection and response. To ease computation for collision detection, bounding boxes (cylinders) are computed and placed around significant body parts.

Dynamic actions, such as path planning and obstacle avoidance, are achieved by providing an agent with a sensing mechanism. This is based on ray casting, which detects objects within the agent’s field of view. Hence, the agent can construct an internal map of the environment, as well as build a memory of its environment. More complicated dynamic actions are achieved through a process of procedural refinement.

A SimHuman agent has a behaviour controller. This controller works on the *sense-decide-act* principal. Rather than using a sequential sense-decide-act loop, each component runs in parallel with the other components. This allows the agent to continually sense the environment while acting. It is thus able to register changes in the environment and then alter the plan—and action—as necessary. An agent can thus be given goals to accomplish in a world which changes dynamically. The agent

also has a set of beliefs about the world which, at any given moment, can either be true or false. False beliefs can be cleared up by sensing the appropriate part of the environment again. When an agent senses objects, relations between objects are abstracted. Thus, instead of thinking in terms of coordinates, the agent understands concepts such as “next to” or “on top of”.

Both objects and agents (all world entities) are subject to the physics engine. Because pre-computed animation is used, a pass must be made through the physics engine before the 3D renderer is invoked. The two main components of SimHuman are the *motion controller* and the *behaviour controller*. These two controllers continually interact with each other. The motion controller manages geometric, skeletal and animation data, while the behaviour controller deals with goals, beliefs and action planning.

Vosinakis and Panayiotopoulos give an example which showcases rudimentary problem solving. It seems, however, that for any decent implementation, the module for abstracting sensed objects needs to be vast, which in turn may complicate the Prolog code. The agent needs to know (or be able to identify) every object in the world. SimHuman is a behavioural agent and communication is neglected in the paper. Our focus is different, as our stated goal is to create a conversational agent. The SimHuman project is thus not directly applicable to our work.

2.1.3 Applications of ECAs and VEs for People with Autism

ECAs and VEs previously have been developed specifically for children with autism, mostly for social skills training applications. We now consider such research.

2.1.3.1 Myoushu

Chamberlain [26] developed the *Myoushu* game engine specifically so that VEs could be implemented for therapy and educational tools for children with autism. While the focus of his work was primarily on developing the underlying game engine, he did implement small demonstration programs, which were well received when tested with primary school children [99].

2.1.3.2 Ehrlich

Ehrlich and Miller [38] worked on a project similar to than that of Chamberlain, although their implementation is less sophisticated. It lacks, for example, the networking and progress logging capabilities of the *Myoushu* system. Admittedly, the focus of the authors were rather on the end implementation of a game which can train a child with autism in social skills. To this end they seemed successful, as their VE is a rich and visually appealing world. There are several scenarios, each with decisions and choices which the user needs to make. An incorrect choice is identified as incorrect to the user and then the user is given another opportunity to attempt the correct answer. Internally, each scenario has a decision tree which the player must navigate. Scenarios are scripted. To assist a user, there is an *advice centre* which is an audio-visual aid for the user who must make a decision. This external, explicit advice centre is meant to train an internal advice centre within children who have autism.

2.1.3.3 VASI

Ho et al. [56] explored yet another mechanism for social skills training using VEs and autonomous agents. Rather than develop a VE, the authors enhanced the existing eCIRCUS (<http://www.e-circus.org/>) system. Their aim was to improve the cognition of the ECAs by giving them autobiographic memories (AMs). The result of this research was the *Virtual Agent Story Interface* (VASI). The system works by having the ECA act out a scenario for the user. VASI allows agents to record their internal states (emotions, goals and intentions) and snapshots of their environment over a period of time. After the scenario has finished, the user has the ability to “replay” the scenario from the perspective of the agents by accessing their AMs. The user can then see the effects which certain events within the game had on the agent. The user then has the ability to make suggestions to the agents for alternative behaviours. When replaying the scenario, the agents will incorporate the suggested behaviours. The VASI system aims to help build a Theory of Mind in the user, as the consequences of events and the actions of agents are linked to the emotions of all the agents involved.

2.1.3.4 Concerns of Escapism

There are concerns regarding the use of ECAs and VEs for social skills training in children who have autism. Perhaps the biggest concern is that children with ASC might become overly reliant on VEs, seeking to “escape” the real world and the difficulties which they face in it by “living” within the safety of VEs. There is also concern about how effective social skills training using VEs can be in transferring the learned skills into the real world, and whether VEs are taken seriously enough by children to be effective to this end. Parsons et al. [91] seek to address these concerns by taking a “person-centred approach”: conducting various interviews with two teenagers with ASC on their experiences with using ASCs for social skills training. The study found that there was a positive response to the use of VEs: the participants were able to recognise that the games were meant to simulate real world situations. Additionally, lessons learned (that is, regarding appropriate behaviour) through the VEs were transferred to the real world. The authors stress the importance of having a facilitator present for social skills training, even when VEs are used. They also note that it is important for a program to be sufficiently complex and realistic to be effective.

2.2 Other Work

Because of the broad scope of our topic, we thought it only fair to include a section in this literature survey where we briefly touch upon some important or interesting topics related to ECAs, but which are not necessarily concerned with them *per se*.

2.2.1 Avatars

Embodied characters which are not necessarily autonomous are also worthy of consideration. The concept of assigning a virtual representative to a computer user is nothing new. When a picture or 3D model specifically represents a human being, it is often referred to as an *avatar*. Fabri et al. [42] describe an experiment to look at the use of avatars in human-to-human dialogue in a problem solving context. The theory is that humans rely extensively on reading faces to make decisions in conversation. Because chat over the Internet usually is only text-based—like Internet Relay Chat (IRC) and instant messaging (IM)—the authors introduce avatars of

which users can change the facial expressions during the course of a conversation. The experiment showed that certain aspects of the conversational experience were enhanced by the use of avatars over that of ordinary text-based conversation, while others were not. The authors devised another experiment to see whether the use of avatars or ECAs could enhance the learning or therapeutic experiences of computer programs designed for children with autism. The results of the experiment showed strong evidence that participants (all young children or teenagers with varying degrees of autism living in the UK) could recognise the emotions of the virtual characters and that such virtual characters could therefore potentially be used in social skills training. Further research supported these findings [27, 28].

2.2.2 Belief-Desire-Intention Models

The *Belief-Desire-Intention* (BDI) model of reasoning has been used in agents (for example [68]). It has also been used (or adapted) to examine autistic behaviour [43]. BDI software architectures are based on the BDI psychology model by Bratman [15]. While there are models of reasoning other than BDI, it seems to be the most pervasive in the field of computer agents.

In this section, when we refer to a BDI *model* we mean a psychological model. When we refer to a BDI *architecture*, we specifically mean some software implementation of the psychological model.

2.2.2.1 BDI in Brief

As the name indicates, the BDI model breaks an agent's reasoning about the world [15] into three parts: it has certain *beliefs* about the state of the world (for example, known propositions), it has *desires* (that is, certain actions or outcomes it wants to bring about) and *intentions* (actions that the agent is committed to bring about)⁶. A BDI model is concerned with reasoning about how to achieve certain goals. Starting from an initial state (the beliefs, which can vary over time), the agent selects a plan from a set of plans to effect an outcome (the desires). To this end, it might be nec-

⁶Note that the distinction between desires and intentions is that, with the latter, there is already a commitment to bring about the desire. Generally speaking, a desire can be suppressed (if necessary), where there would be no intention (commitment) to seeing the desire fulfilled.

essary to solve several sub-plans which it would have to commit to first (intentions). BDI models can be described by formal logic.

A BDI architecture is sometimes implemented in the reasoning part of an agent. Over the years, different BDI architectures (all of which adhere to the basic definition of the BDI model) have been proposed. Many of these extend the basic BDI model in some way. Examples of BDI architectures (or extended architectures) are DESIRE [17], GOAL (<http://mmi.tudelft.nl/~koen/goal.php>), IRMA [16], JAM [62], PRS [107] and SPARK (<http://www.ai.sri.com/~spark/>). Discussing these architectures is beyond the scope of this thesis.

Perhaps the biggest limitation of the BDI model is that it does not support learning in itself: the agent cannot adapt for future behaviour based on past behaviour. Another concern is that the BDI architecture may have been superseded by more modern cognitive architectures, such as SOAR [70]. The debate on the relevance on BDI continues, but it would seem that, just as rule-based systems in the face of real AI solutions, the BDI model still has applications for which it is well-suited. For a discussion on the contemporary relevance of the BDI model of agency, see Georgeff et al. [49].

An agent which implements a BDI architecture is designed for problem solving; that is, reaching certain *goals*. The fact that many implementations are also based on formal logic [95] means that a BDI architecture is an AI construct, and not a rule-based one. Both of these facts make BDI unsuited for our purposes. But if one would wish to extend the scope of our project, a BDI architecture (or, at least a modified one, as described below) might be worthwhile to consider.

2.2.2.2 BDI and Autism

In [67, Ch. 18] the authors present an interesting discussion on dialogue and also an overview of the BDI model. One of the examples given in that chapter is how one (using the BDI model) can infer that the question

"Can you give me a list of flights from Atlanta to Boston?"

is not (in the context of telephony flight booking) questioning the ability of the hearer to provide such a list (this is assumed), but is in fact requesting a list of flights from

Atlanta to Boston. A question such as this will often enough be interpreted literally by a child who has AS. This could suggest that a child with AS may lack some of the “built-in” rules for plan inference which a neurotypical person has (which could also simply be context-blindness, see [3, Ch. 5]).

Galitsky [43] realised this and adapted the BDI model to create a platform from which children with autism could be taught some of these missing reasoning steps. He specifically refers to teaching children who have autism the basic axioms from which they can deduce appropriate responses and behaviours.

Galitsky (using as basis the Theory of Mind about children with autism, see [8]) alters the BDI model by merging desires and intentions into one, called *wants*. His reasoning is that in the mind of a child with autism, there is often no distinction between the desires and intentions, because if there is a desire, the child will act to fulfil it without restraint. To the model the author adds *knowledge*: things about the world which the child knows to be true and which cannot be false. This *want-know-belief* model can be used to teach small children with autism how to reason about what they *want* in relation to what they *know* and *believe*. It can also be used to explain concepts like “inform”, “deceive”, “explain” and “forgive”. The result of this work can be applied by psychiatrists directly to children with autism, or can be presented to children with autism through the medium of computer software.

This work was not concerned with the simulation of autistic behaviour in a virtual agent, but rather with the establishment of a “common ground” from which children with autism can be taught to reason about the world. It would seem plausible, however, that some BDI architecture (whether one developed from this model or another, more traditional one) could be used to simulate autistic behaviour by carefully selecting and omitting predicates of the BDI logic. However, such logical reasoning, especially about a large domain, is still problematic on contemporary home computers, and this makes it unsuitable for our purposes.

2.2.3 Robotics

Robots are sometimes closely linked with software agents, as they *embody* (although physically, rather than only visually) something which can be designed to exhibit some *behaviour* to meet some goal. We therefore thought it good to consider at least one interesting example of research concerning robots and affections and autism.

Shybot is a companion/educational robot created by Lee et al. [71] for children who have autism. The primary goal of Shybot is to act shy around strangers and friendly around those which it has learned to “know”. The purpose is twofold: firstly it *mimics* a typical behaviour of many children who have autism (shyness) and secondly it *teaches* the children about shyness. The latter is achieved when the children try to understand the behaviour of the robot and then apply what they learn to themselves. In this sense Shybot “holds up a mirror” to the behaviour of children with autism (regarding shyness) and the problems that it may cause people around them.

What interests us about Shybot is that it is a robot designed to embody one specific human behaviour with the aim of demonstrating and teaching the user that behaviour.

The authors used as a starting point the Uncanny Valley theory (discussed in Section 2.1.1.6). They recognised that a human will imbue a robot or ECA with human characteristics if it seems to exhibit some other human characteristics. They also noted that humans can be disturbed by something which looks very human but is not. Therefore, they designed Shybot to be pet-like in appearance, while it exhibited the human characteristic of shyness through its behaviour. Shybot would run away to exhibit shyness. It could also display a nervous “heartbeat” through LEDs visible underneath its translucent covering. Shybot also has a spinning antenna which would spin faster to indicate happiness. Through these basic mechanisms Shybot can emulate the basic behaviours associated with “being shy” and “being at ease” with someone.

2.2.4 Affective Computing

Affective computing concerns computers which can sense human emotions, or which can display emotions, or both [92]. The goal of affective computing is ultimately to improve human-computer interaction (HCI). As such, affective computing can be of interest to one planning to implement an ECA if the ECA either must have artificial emotions, or interpret the emotions of the user.

Picard has devoted much attention to HCI over the past decade. In [93] she describes, amongst others, an effort to build an unembodied pedagogical agent which can aid a user during normal desktop usage. Other attempts at such embodied pedagogical

agents have failed (see, for example, [96]). Picard's idea was to create an agent which is not intrusive, but responds when the user needs assistance. To determine when a user needs assistance, HCI technologies such as pressure sensitive mice were used. The agent is also completely unembodied, lacking visual and vocal components. It also does not use the pronoun "I". For the most part it tries to empathise with the user rather than make suggestions. Such empathy on its own can alleviate frustration. Field trials showed favourable responses from users to this type of pedagogical desktop agent.

The lesson from this study is that one should be careful when designing a conversational agent. Rather than simply asking, for example, "What can the agent do for the user?", one also needs to ask "How will the user react to the agent?". The latter is the most important question for our purpose, as a measure of success for us would be to have a person who is unfamiliar with AS react similarly to our agent as they would to a child who has AS.

As for affective computing where the computer senses the emotions of the user, the scope of our project precluded the utilisation of such mechanisms. In the context of an ECA which must resemble a child who has AS, one needs to ask whether such added complexity would be justified, as children with AS often have difficulty interpreting the emotions of others. And when they do, it is often learned behaviour and limited in scope.

Affective computing has also been applied to children with autism to help them understand the emotions of other people and *vice versa*. The interested reader is referred to El Kaliouby et al. [40] and Picard [94].

Chapter 3

Design and Implementation

In the previous chapter we discussed various aspects of ECA design and how these aspects were approached in the literature. In this chapter we discuss how these aspects relate to our project and how we approached them. We consider the available options and motivate our decisions.

We first present the requirements of our system and then frame it around the necessary limitations which we must accommodate. We then give a broad overview of the design rationale. The difficulties which were encountered with implementing the system are presented. Finally, the implementation details of the system are discussed, as well the details of a demonstration implementation which was built using the system.

3.1 Design Specifications

3.1.1 Requirements

To briefly reiterate our goal, we wish to build a system around which ECAs with non-neurotypical behaviours can be designed. In building an ECA, it is necessary to focus on two broad facets: embodiment (graphics) and conversation (reasoning).

Embodying an agent delivers more than aesthetic benefits. This is because humans communicate using their bodies as well as through speech. In simulating communication with someone who has AS, we want to demonstrate their body language (or lack thereof) as well. It is therefore necessary to provide some graphical platform on which to model and animate an agent created using our system.

In order to conduct a conversation, an agent needs to accept and recognise user input, decide on a response, and then deliver that response. Of these three processes, recognising the input and deciding on a response (which we shall collectively call *reasoning*) is arguably the most intricate. There is no standard way of implementing reasoning in an ECA and many different solutions to this problem exist (see Section 2.1.2). From the onset, however, we realised that our system requires several capabilities. Firstly, language processing is necessary to understand human input. Complex language processing is not required, as the goal is to only broadly demonstrate non-neurotypical behaviour. Secondly, some mechanism is necessary whereby a response can be generated based on the input. This mechanism needs to be logical, at least in part. It would also be desirable to tie emotions into this process, as human beings—including non-neurotypical human beings—are emotional creatures. Thirdly, because in this thesis the focus is specifically on autistic behavioural traits, it will be beneficial to implement some database to store data about special interests. This will allow for the simulation of the deep, but rigid, memory regarding special interests which is typical of children and adults with autism [50]. It is with these requirements—along with the limitations set out below—that we set out designing our system.

3.1.2 Scope and Limitations

In this section we present the limitations to the scope of our work.

From the onset we decided to use rule-based systems. A true AI system falls outside the scope and time frame for this work. We contend, however, that a rule-based system can be sufficient for our purposes in modelling autistic behaviour. We present our supporting arguments in the following section. Additionally, due to time constraints, we shall make use of third party libraries to provide the various different components of the intended system, rather than creating every component of the system ourselves. We do not wish to place any undue restrictions on where an implementation making use of our system will be deployed, however. To appeal to as broad an audience as possible, it should be possible to implement agents built on our system which can run on desktop computers as well as servers.

The restrictions above determined which literature and existing systems were considered. We tended towards literature which dealt with specific components of ECAs which were also rule-based. Those which provided stand-alone libraries for specific components were of particular interest.

Where possible, we opted to use *extensible markup language* (XML) to configure system components and specify rules. XML is a popular and open standard (see <http://www.w3.org/XML/>). This decision meant that we would have a mostly uniform way for configuring our system. Because different third-party libraries and components were used to build the system, it uses different XML flavours rather than one specific flavour. This has the advantage, however, that each component can be configured separately. It should be noted that, where XML configuration was not available (such as with NLTK—see Section 3.2.2.1), we did not specially create an XML standard. Doing so would have been time consuming and, in the end, not aid us in achieving the main focus of our project. The preference for XML was thus only a secondary specification or recommendation, rather than a requirement.

While we wanted our solution to be generic and system-independent, we also needed a platform on which to test it. We made the decision to use the Myoushu system mentioned in Section 2.1.3.1. Reasons for this decision include the fact that it was locally developed, and hence we had expertise available. Moreover, the interface to the system is sufficiently flexible to allow us to easily interface different systems.

Our choice of development environment and programming language was influenced by our above-mentioned design choices. We settled on Java as programming language, as Myoushu provides a Java interface, and the ALMA library is only available in Java. Note that it is possible to use a game engine other than Myoushu, as long

as a Java interface is available. We do note that Jython also allows us to use Python code.

3.1.3 Design Rationale

3.1.3.1 Arguments for Rule-based Systems

As mentioned, our goal is to implement a *rule-based* system, rather than a true artificial intelligence system. The reason for this is that rule-based systems are easier to implement and understand, and thus easier to configure, usually even by a person who is unfamiliar with the inner workings of the system. AI systems provide the ability to learn and adapt, which are invaluable characteristics of any system which simulates human behaviour. But implementing believable human-like true AI agents is still an unresolved problem, both theoretically and technologically [29]. A hybrid solution is another option: having the system being driven partly by rules and partly by AI. Time restrictions did not allow us to explore this possibility at this time, but it would certainly be interesting to investigate at a later time.

Another reason why we chose to use a rule-based system, is that we believe that such a system can simulate the rigid thinking of people with autism particularly well. We want to emphasise that the minds of people who have autism are as complex as those of neurotypical people. Our choice of using a rule-based system does not imply in any way that the thought processes of people with autism are simplistic, but we do think that the rigid, systematic *behaviour*—which characterises autism—can, to some degree, be modelled by rule-based systems.

3.1.3.2 Examples of Successful Rule-based Systems

We now present two examples of rule-based systems which have historically enjoyed much success and popularity [77]: the chatterbots Eliza [105] and Parry [30]. Eliza is a psychologist and Parry is a paranoid schizophrenic. While Eliza had quite a small ruleset, she was successful because she was portrayed as an Rogerian psychologist, always posing questions to the user to keep the initiative in the conversation. Parry had a much larger ruleset, could change certain internal states, and was also able to provide colourful discourse because of its personality.

The efficacy of these agents come from two qualities which they have: they have “personality” (or, at the very least, a believable character profile associated with them), and are able to respond according to what the user types. The former quality makes the agents believable by giving them a degree of depth. Consider, for example, how “real” a fictional character in a book or animated movie can be if it is vividly portrayed and revealed as having a “background” and a “personality”. An ECA is but one such medium for character portrayal. The latter quality feigns intelligence and perception. These illusions are remarkably effective: even a user knowing he or she was interacting with a simplistic computer program could, after a few lines, find himself or herself engrossed in a “conversation”. There is even evidence to suggest that simple chatterbots like Eliza and Parry can pass the Turing test (albeit informally; see [63]). It should be stressed, however, that our goal is not to design an agent which can necessarily pass the Turing test, but rather to create a rule-based system which convincingly portrays an autistic person in some social context by displaying authentic behaviour, even if the user is fully aware that he or she is working with a computer program. Once the rule system is in place, a ruleset can be created to fit the ECA to a particular role.

3.1.3.3 Modularity

It is desirable to create a system which is modular. This allows the system to be extended in the future. It would then be possible to, for example, write multiple variant modules (which we refer to as “tasks”) which all fulfil the same role, and then substitute the modules as necessary. Modular systems have been successfully used in many other similar systems (see, for example, [6, 68]).

3.2 Design and Implementation Issues

During the process of designing and implementing our system and demonstration program, we encountered several difficulties. This section briefly highlights some of the more significant challenges which we faced.

3.2.1 Issues Regarding Design

During the design process, no issues arose which were difficulties in themselves. It was necessary, however, to be careful, as bad design decisions could translate into severe implementation issues. It was important to be continually be aware of the project specifications, as outlined in Section 3.1. One particular aspect which needed careful attention was that different third-party libraries were going to be used and that these libraries would have to interface with each other. To this end it was necessary to make sure that our system could adapt the data as necessary at the various stages of the process flow.

3.2.2 Issues Regarding Implementation

Several problems arose during the implementation of our system, as well as during the development of the demonstration program which was built on top of the system.

3.2.2.1 System Implementation

It was originally envisioned that the graphical representation of the agent would be defined by our system. This would be difficult to realise fully, as every graphical system and game engine have different data formats to store and represent 3D data. One solution which seemed feasible, however, would be to at least define the animations associated with the agent programmatically using XSTEP (see Section 2.1.1.5). While XSTEP would still have required an adapter to work with the VE, in theory it would have provided a definition for animations which would be uniform across platforms using XSTEP. Unfortunately, XSTEP appeared to have been an abandoned project and was not at a stage suitable to our requirements. We subsequently forwent any plans to integrate the graphical representation of the agent with our system, as the restrictions on the project did not allow us time to investigate further solutions.

Some third party libraries also proved troublesome. We made slight modifications to the source code of the AIML interpreter which we chose to use (Program D—see Section 3.3.2) in order to eliminate redundant dependencies from our system. ALMA seems to contain a bug where its system visualisation cannot be disabled, even if the

API is instructed to do so. This can occasionally, under specific circumstances, lead to problems in the system back-end, but does not interfere with the mechanics of the system and does not affect the user experience.

Our goal was initially to code our system in C++ to achieve high execution and memory efficiency. ALMA is a Java library, however, with no equivalent in C++. Although the ALMA library is under a BSD license, the author was unwilling to release the source code. We therefore could not port the library to C++ or compile it natively. The decision was therefore made to develop the whole project in Java. This restriction also influenced our choice for other libraries, such as which AIML interpreter and database system to use.

Yet another library which we wanted to use was the Natural Language Toolkit (NLTK, see <http://www.nltk.org>), which is written in Python. We therefore chose to integrate Jython (<http://www.jython.org>) into our system kernel. Jython is a pure JVM interpreter for Python and can run on the same JVM which executes Java code. This eliminated the need to run another VM in addition to the JVM. An additional benefit of integrating Jython into our system is that our system can now support interactive scripting.

3.2.2.2 Demonstration Program Implementation

It was necessary to acquire a myriad of skills relating to 3D modelling to design and implement our demonstration program (see Section 3.4.2). While it was interesting and useful to gain these skills, it was rather time-consuming.

3.3 The Thandiwe Engine

This section describes the implementation details of our system. The reader may refer to Figure 3.1 for a brief overview.

The system consists of a kernel which manages a number of modules. Each module provides a specific function within the system. The kernel is minimalistic and modules are relied upon to extend the system. We make a distinction between core and non-core modules. Core modules are built into the system and are required for the system to function properly. These modules are loaded automatically when the

kernel is started. Non-core modules can be written by an application developer when implementing some application on top of our system. Such non-core modules may be required to extend the functionality of the engine (for example, to provide speech recognition). These modules need to be added manually to the kernel before the kernel is started.

The core modules are briefly described below:

- *CognitiveTask* accepts user input (text) and delegates it to other modules to process.
- *AIMLTask* interfaces with Program D, allowing AIML input to be processed.
- *TimerTask* implements a simple timer to regulate module execution time.
- *QueryBuilderTask* interfaces with the knowledge database and NLTK through a simple command pattern [44] to translate natural language questions into SQL questions.
- *MessagingTask* allows messages to be passed between modules and between threads.
- *AffectTask* reads affect-response files and interfaces with the ALMA library to determine a reaction.
- *OutputTask* formats the output from other modules to produce a coherent English response and passes it along to the VE.

The Thandiwe engine itself has been designed to run in a single thread. This minimises the number of active threads, as external libraries tend to make liberal use of threads, to the extent that system performance is influenced. We also cannot assume that these libraries themselves are thread-safe. We thus use only a single thread in which to run our kernel and modules, in order to maintain better control over thread interactions. To effect this, each module is assigned an execution interval and an iteration limit. The execution interval indicates how often a module should be executed. If a module has the opportunity to execute before a full execution interval has elapsed since its last execution, it passes up the opportunity. If a module misses an execution interval (due to other modules executing for too long), its execution can be repeated up until the iteration limit to “catch up”.

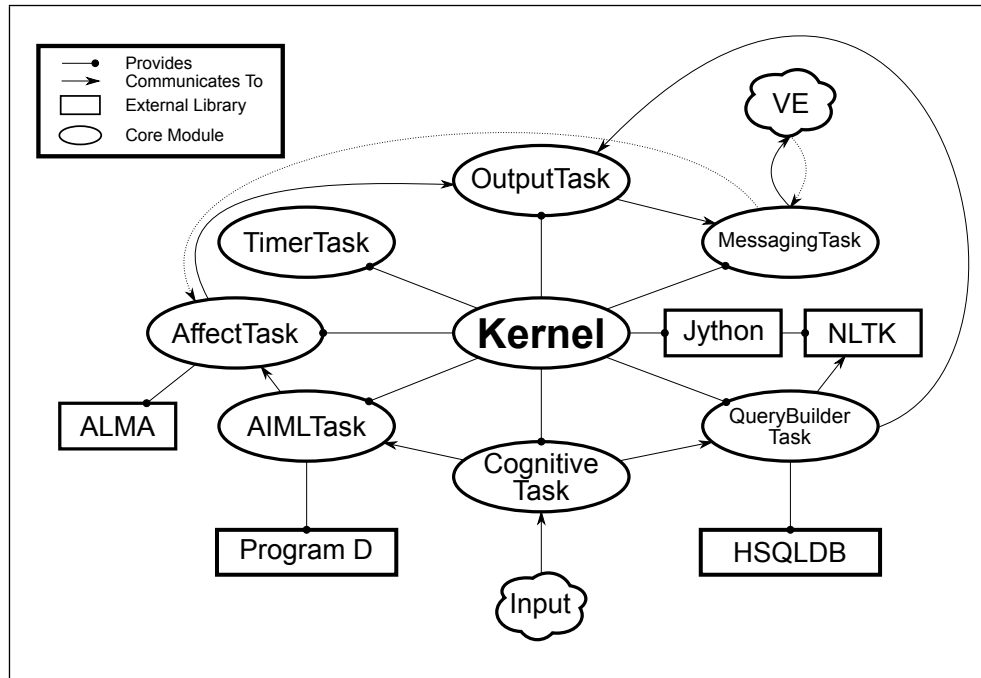


Figure 3.1: This figure briefly summarises the flow of data through the core modules.

In the rest of this section, we describe the roles of each of the listed core modules (with the exception of the elementary *TimerTask*) in more detail.

3.3.1 Input Delegation

We limited the type of input to text only. Other options would have been to convert speech to text and allowing for gesture recognition through the use of cameras and motion detectors, as in [13]. Converting speech to text has long been studied, but is still error prone, at least without providing extensive training data to the converter [34]. A setup for gesture recognition is complex [82] and would have taken a long time to implement. Due to the time restrictions on this project, it was decided to use only textual input. The reader may note that the modularity of the system design allows for any of these aspects to be included as non-core modules as future work.

Input can be directed to different parts of the engine by defining one or more filters for recognising the text input. The default is to send input to the task which implements the AIML interpreter. If certain types of input need to be directed to other parts of the engine (either the logical part or the knowledge-base), then such a filter needs

to be defined. To this end we provide an interface which allows a *destination* and a *threshold* to be specified, and which must implement a function returning some *probability* of whether an input text was matched by the filter or not. The destination is the part of the engine to which the input should be passed. The threshold is the minimum value of the match probability needed to match the input to the filter. The probability is an indication of how well the input matched the filter.

In our demonstration implementation, we implement two basic deterministic filters. These filters make use of simple keyword matching. Keywords are extracted from the relevant database. This method was deemed sufficient for our purposes, as the words extracted from the database are domain-specific words. The filters only ever return the probabilities 0.0 and 1.0 indicating, respectively, a mismatch and a match. But, due to the design of the interface, one could implement a sophisticated neural network to match inputs against specific topics.

3.3.2 Language Processing

In Chapter 1 we already mentioned the “primeval” chatterbots Eliza and Parry which proved engaging despite their relative simplicity¹. Encouraged by their successes in engaging users, we looked at some of the “descendants” of these computer programs as a starting point for our own project, as chatterbots are typically rule-based. To aid our search, we looked at entrants of the Loebner prize. The Loebner prize is an annual competition which provides a formal setting for Turing tests [41, 73]. The competition was designed to encourage people to develop a computer program which can pass the Turing test. Many of the systems which have been entered for the Loebner prize have, however, not been made public. In the end, we decided to use A.L.I.C.E., which was designed by Richard Wallace and which won the Loebner prize in 2000, 2001 and 2004 [73]. There were several reasons why we chose A.L.I.C.E. to be part of our system:

- A.L.I.C.E. has an open standard, called AIML [19].
- The AIML standard has useful features, such as capturing context (in a limited scope).

¹These programs are analogous to Conway’s Game of Life, which engrosses viewers in apparent chaotic—yet ordered—“living” universes, even though the entire system is driven by only four simple rules.

- There already exists numerous AIML interpreters, so it would not be necessary to create our own.
- In informal tests, we found that A.L.I.C.E. is as believable as other similar alternatives, such as Jabberwacky (see <http://www.jabberwacky.com/chat-jabberwacky>).

Based on the relationship with A.L.I.C.E., the core of our system was designed to interpret AIML files to process input. This would effectively take care of the language processing aspect of our project. We augmented the output mechanism, however, as we wanted to add additional dimensions to our agent beyond just being a standard chatterbot. Rather than returning a natural language response, a code is returned. This code is linked to a response in an XML file which can contain a natural language response and a non-verbal (animated) response, as well as send emotional information to the affect engine. We discuss this XML response mechanism in more detail in Section 3.3.5.

AIML processing was integrated into *AIMLTask*. We used Program D (http://www.aitools.org/Program_D) as our AIML interpreter, as it is stable, mature and developed in native Java. It was developed by Noel Bush, who collaborated on the AIML specification [19]. In addition, we also make use of PyAIML (<http://pyaiml.sourceforge.net>) for some lightweight testing during development.

3.3.3 Data Store

QueryBuilderTask adds a relational database system (RDBS) to the engine. We chose HSQLDB (<http://hsqldb.org>) as our RDBS for the following reasons:

- It has been implemented in pure Java. This conforms to our goal to minimise our dependency on native libraries. It can thus also be accessed from Jython, allowing us to write Python scripts which can directly access the same database which the underlying system uses without the need for ODBC/JDBC drivers.
- HSQLDB is fast (compared to other RDBSs which run on VMs) and stable.
- HSQLDB offers a variety of options to connect to a database: it can host a server on a network, it can connect to files directly, and even host a DB in-

memory. It is the ability to connect to files without the need to host a server which makes HSQLDB ideal for our purposes.

Having an RDBS provides us with the ability to endow our system with a persistent knowledge-base.

The actual role of the query builder task is to translate user questions regarding the special interests of the agent into SQL queries. To accomplish such a translation, we use NLTK and feature-based grammars, as described by Bird et al. [14, Sec. 10.1]. The reader may note that this is a relatively naive way of translating questions to SQL. There is some justification for this simplified approach. Firstly, the “proper” way of “understanding” natural text, as developed in full in [14], is sufficiently complex to fall outside the time limits of this project. For future work, the translation part of the query builder task can be extended to allow more generalised query translation. Secondly, we abstracted the method given in [14] slightly by making use of keyword substitution. Keywords are defined as starting with an @-sign and containing only uppercase letters and underscores (_). After the substitution, the replaced values are concatenated at the back of the string from first to last and separated by a hash (#). For example, if we have a list of dinosaur names such as

```
names = ['Triceratops', 'Stegosaurus', 'Allosaurus']
```

then a user input string such as

```
"Where did Stegosaurus live?"
```

would be substituted as

```
"Where did @DINO_NAME live?#Stegosaurus"
```

because “Stegosaurus” matched an entry in the list of dinosaur names. After the question (the part of the above string before the first hash) has been translated into SQL (ignoring case transformations and punctuation mark omissions here), a query such as the following is produced:

```
SELECT location FROM dino_species WHERE name='@DINO_NAME';
```

Finally, the keywords are replaced with the original words to obtain

```
SELECT location FROM dino_species WHERE name='Stegosaurus';
```

This process allows us to streamline the feature-based grammars needed by NLTK to translate questions to SQL queries, as we do not need to accommodate all the values already in the database, but only keywords which can be substituted for a value in a given SQL table column.

The reason for implementing this module is to provide a “rote” memory for our agent: the agent can “know” much about a certain topic, but cannot necessarily transfer that knowledge to another domain [50, pp. 32–33, 142–143], [3, Ch. 7]. This method of implementing a rote memory can be particularly effective if, for example, the agent’s special interest lies in bus or train timetables, as has been known to be the case [50, p. 32–33]. It is also not always necessary to have a complex SQL implementation to simulate such a rote memory, as, for example, it has been recorded that a child can go from the year of a historical event to the event, but not from the event to the year [50, p. 143].

3.3.4 Messaging

The *MessagingTask* is responsible for interthread and inter-module communication. Each thread or module registers a mailbox with the respective appropriate message handler. Depending on whether an interthread or inter-module message is to be sent, an instance of *SharedMessage* is sent to either *ThreadMessageHandler* or *TaskMessageHandler* directly.

Both types of message handlers are subscription-based at the class level. To be able to make use of a message handler (either module or thread), a class must be registered with that message handler. Registration (and sending and receiving of messages) makes use of class identifiers to identify a mailbox. The decision to use class identifiers as mailbox addresses was supported by the following arguments:

- It is elementary to address a class identifier, but much more complicated to address an instance of a class. The alternative would be to make use of factories [44, pp. 107–116] for the modules and threads, but this would complicate

matters in other areas (such as when writing custom modules and threads) which seems unnecessary when a module or thread is supposed to have a single, atomic role in the system.

- Modules and threads in the Thandiwe system are intended to be singletons. Therefore only one instance of the class can ever exist: the object becomes synonymous with the class.

The connections between modules and between threads are persistent and one subscribed module or thread can communicate with any other subscribed module or thread. Sending a message does not block a module or thread. Bi-directional communication is possible, but for inter-module communication messages are generally only passed “downstream”. As mentioned, messages are wrapped in a *SharedMessage* object and can thus be of variable type and size, as pass-by-reference is used.

Inter-Module Communication

Messages between modules within the Thandiwe engine are sent directly to the mailbox of the destination module (class): no processing of the message is done by the module itself. The module simply hosts the respective mailboxes. Where the messaging module does play a role, is when receiving messages from another thread via the thread message handler (see Section 3.3.4 below). Interthread messages are only passed between threads; threads do not access the messaging mechanism of Thandiwe modules directly. When an interthread message arrives at the thread which hosts the Thandiwe kernel, it is passed on to the messaging module. There the contents of the message is inspected and then the message is passed on to the relevant module. For example, user input would be passed on to *CognitiveTask*, while a game state—periodically sent out—which directly effects the affective state of the agent, is passed on to *AffectTask*.

Interthread Communication

As with messages sent between modules, messages sent between threads are directly addressed to thread classes. Unlike the module message handler, though, the thread message handler is not wrapped within a module or even a thread of its own. An instance of the thread message handler must be created and then attached to each

game thread. As mentioned above, a thread running the Thandiwe kernel can already handle messages sent via this mechanism. Any other threads need to implement their own way of handling messages. This is not difficult to do, however, as only *OutputTask* sends out messages at this time².

3.3.5 Affect Computations

The *AffectTask* module serves a dual role. Firstly, it interprets responses to user input (which has already been processed by the AIML interpreter). Secondly, it is responsible for adjusting the internal affective states of the agents. We now discuss each of these roles.

While the AIML interpreter is relied on to process input (see Section 3.3.2), it is less instrumental in providing output. Rather than directly providing a response, a code unique to the relevant AIML rule(s) is provided. This code links AIML rules to corresponding rules in a custom XML configuration format which we designed. We call this XML configuration format the *Thandiwe Response File* (TRF) format. The DTD definition of this XML flavour, as well as a discussion of the format, can be found in Appendix A. TRFs add two additional dimensions to the standard chatterbot output. The first is *conditions*: one can define logical conditions which would give different responses from the agent for the same user input based on the internal affect state of the agent. The second dimension is that the agent output is not limited to only a textual response: one can define animations and influences to the internal affect state of the agent as well. This is done by encapsulating so-called “sets”. A set defines the conditions and reactions related to a specific input. These sets are stored internally in a cache using a least recently used (LRU) replacement policy. The motivation for using a LRU cache policy lies in the nature of chatterbots: it is possible to loop responses from a chatterbot by repeating the inputs. This is because chatterbots are rule-based. The following conversation excerpt with Jabberwacky demonstrates how a chatterbot can be led to loop a response:

```
USER: Everyone has a darkside.
JABBERWACKY: Are you working for voldemort?
USER: I do not follow Harry Potter.
```

²A future feature of the Thandiwe engine could be fault reporting. If a fatal error occurs within the Thandiwe engine, it would then send a message to other threads to inform them that the Thandiwe engine thread has ceased operation.

```
JABBERWACKY: Harry potter is a dumb kid all he can do is magic.  
I can insta-teleport to other planets I'm much cooler.  
USER: Everyone has a dark side.  
JABBERWACKY: Are you working for voldemort?  
USER: I do not follow Harry Potter.  
JABBERWACKY: Should I grade you now?  
USER: Everyone has a dark side.  
JABBERWACKY: Are you working for voldemort?  
USER: I do not follow Harry Potter.  
JABBERWACKY: Who do you follow?  
USER: Everyone has a dark side.  
JABBERWACKY: Are you working for voldemort?
```

While it is mostly undesirable for our purposes to have looping responses, it might be possible that looping occurs if the user decides to keep talking about one specific topic. If the user moves on to another topic, then it is unlikely that some of the sets used in that topic will be repeated in another topic. Thus the further the conversation moves away from one topic to another, the less likely it becomes that sets needed by the first topic will be needed again [97, pp. 334–336], hence the choice of a LRU cache eviction policy.

The affect module also ties in with the ALMA (see Section 2.1.1.2) library. In our engine, ALMA handles all the internal affective states of our agent. The affect module makes all the current internal affective states of the agent available to the other modules. It also influences these internal states through *act* tags. These tags can come from different places. Usually it is associated with agent output. In this case, act tags are less of an outward response to what the user said, and more of an inward response. For example, if what the user said was pleasing to the agent, the agent response (as defined in a TRF) would pass the *nice_thing* act tag to the ALMA system. Tags can also come from the game engine (that is, not be related to a response to user input). An example can be found in our demonstration program, while the user is talking to the agent and the radio is playing in the background (see Section 3.4). Due to sensory integration problems of the agent, while the radio is playing, a *bad_event* act tag would periodically be sent to the agent, thus steadily deteriorating the demeanour³ of the agent over time.

³We use the word “demeanour” to refer to the combination of the variable affections (emotions and moods) of the agent.

3.3.6 Engine Output

The output module structures and formats the output. At present the module is rather simple. It combines multiple strings into one string, and formats the text. It is possible to extend the functionality of the module if needed. One scenario where the output module can play a more involved role, is if the text is to be tagged with phonetic symbols for speech synthesis (if there is a lack of a dedicated speech synthesis module).

Lastly, the output text is then sent to the virtual environment. There it is up to the specific implementation to decide on how the text is to be displayed.

3.4 The Demonstration Program

Having developed the separate components of our system, we wanted to integrate these by showing an envisioned example of its use. This section describes the implementation of the demonstration program. In short, the demonstration program allows the user to interact with an ECA built using our system in a restricted environment.

3.4.1 The Scenario

In our demonstration program, the user takes on the role of a person who is about to conduct a short and informal interview with a boy who has AS. As the demo starts, the user finds themselves in a waiting room outside a therapist's office. The user sees the child's "therapist", who effectively provides the user with a brief tutorial on the user interface and what to expect from the agent. When she is finished speaking, she leaves the waiting area. The user must now open the door which leads into the office and speak with the agent. The user can discuss different topics with the agent. Two of these topics will influence the demeanour³ of the agent in a positive way, two topics will influence it in a negative way and one topic is neutral, having no dramatic effect on the demeanour of the agent. When the user is finished with the conversation, they can exit the demo by leaving through the same door through which the therapist left.

To demonstrate one of the sensory integration problems which people with AS normally have, a couple of scenarios were built into the demo. The first one is a radio inside the office which plays baroque music. While the radio is on, the mood of the agent will deteriorate steadily. This demonstrates the fact that most, if not all, people with AS have to strain to distinguish between competing sensory (audio) sources [3] (in our example, the music and the conversation with the user). If the user switches the radio off, then the mood of the agent will not deteriorate in this manner. At some point, a dog will start barking in the background. This second scenario will have the same negative effect on the agent as the first one, but this time the user can do nothing to stop the source of the problem. If both the radio and dog are audible, then agent's mood will deteriorate rapidly.

3.4.2 Graphical Design

We first created our VE scene (the office described in the previous section) using Google SketchUp (<http://sketchup.google.com>), which allows for rapid 3D development, especially for architectural designs. We then imported this scene into Blender (<http://www.blender.org>), a free and open-source general-purpose 3D authoring tool. We made numerous iterations over the scene in Blender to reduce vertex, edge, face, and object count; to clean up defects which resulted from porting between SketchUp and Blender; and to make changes which were necessary to export the scene to the Ogre dotScene format, which the Myoushu engine requires. We also used Blender to create basic animations within the scene, such as doors opening and closing.

For creating our 3D character models, we considered the MakeHuman project (<http://www.makehuman.org>). While this looked promising, the project was not mature enough for our needs at the time when we needed the software. We thus decided to rather create our 3D character models in a program called Avatar Studio 2 (<http://www.web3d-fr.com/Avatars/AvatarStudio/AvatarStudio.php>). These character models had to be saved in .wrl format, which then had to be converted to .x3d via a program called Flux Studio (see <http://mediamachines.wordpress.com>), because the standard .wrl importer for Blender does not work correctly with the version of VRML which Avatar Studio 2 produces. We were then finally able to import the 3D models into Blender using a third-party .x3d importer. Here we tweaked the models slightly and animated them (see the next paragraph). We created three character

models in this way: one for the agent, one for the therapist (described in the previous section) and one for the user.

To animate our character models, we built a library of basic motion-captured movements using the publicly available database at the Carnegie Mellon University Graphics Lab (<http://mocap.cs.cmu.edu>). Using Blender we bound those motions to our models and so created animated characters. Note that because our system does not provide a tight integration with an animation system, this library can only be used with game engines built on top of Ogre, such as the Myoushu game engine [26], as the animation data is specifically interpreted by Ogre.

After all of this, we exported our scene and our character models to the Ogre dotScene and mesh formats, which can be interpreted by the Myoushu game engine.

3.4.3 Agent Design

Providing an agent with a background and a profile goes a long way towards making that agent believable. Eliza was a Rogerian psychologist and Parry was a paranoid-schizophrenic [52]. Even contemporary agents, such as those entered for the Loebner Prize, have personalities: A.L.I.C.E., for example, was designed with the personality of a sentient computer entity (rather than pretending to be human). For the two years (2005 and 2006) in which Jabberwacky won the Loebner prize, each winning entry was an agent based on the Jabberwacky system, but each bot had a different personality. These were *George* (<http://www.jabberwacky.com/chat-george>) [10, 74] and *Joan* (<http://www.jabberwacky.com/chat-joan>) [75], respectively.

To create an agent which has “depth”, we drew up a personality and behavioural profile for our demonstration agent. This profile can be found in Appendix B. Once we had this profile drawn up, we were able to clearly define agent “knowledge” regarding aspects such as family, school and special interests.

Another factor which generally determines the success of an agent, is the size of its conversation data set. Examining the source code of a reimplementaion of Eliza in Pascal⁴ shows that it only had 116 different responses⁵. Winners of the Loebner prize

⁴This reimplementaion of Eliza in Pascal can be found at <http://www.cs.cmu.edu/afs/cs/project/ai-repository/ai/areas/classics/eliza/pascal/>.

⁵Despite this relatively small number, Eliza was successful because of *how* it interacted with the user: it usually tried to ask a question to the user, thus taking initiative away from the user

have been known to make use of tens of thousands to over a million of conversation lines [106]. Compiling such a dataset is a very time consuming and arduous task. For this reason we have chosen, for our demonstration implementation, to make use of a restricted dataset which focuses on only a few specific topics. To assist the user in operating only within these topics, we limited the input that the user can give. Rather than free-form typing, we provide the user with a menu-driven system displaying the possible conversation options available at that time. This dialogue system is described in more detail in the following section. Note, however, that the input is still interpreted by our system as if it were a text string which was entered in a free-form style.

As stated in Section 3.4.1, we wanted the user to be able to steer the conversation in any of three directions: one which will affect the demeanour of the agent in a positive way, one in a negative way, and one which is fairly neutral. Because we intentionally want the user to steer the demeanour of the agent, we restricted the user's speech acts to deliberate perlocutionary acts (that is, speech acts which produce an effect on the *listener* [67, p. 728]). Most of these acts are also illocutionary (that is, speech acts which is performed by the *speaker* [67, p. 728]). The neutral topic consists of bland, factual questions of no particular interest. The positive conversation branches also consist of factual questions, but they elicit positive responses because they are concerned with our agent's special interests [50, pp. 65–66]. The negative conversation branches are personal, probing questions, dealing with abstract concepts such as feelings and Theory of Mind. Apart from the fact that these questions are of a personal nature, the abstract subjects of these questions are also meant to confuse and frustrate the agent [3].

3.4.4 Dialogue System

The dialogue system was scripted in Python. Because we wanted some of the dialogue options to repeat, a simple dialogue tree [1, Ch. 7] would not have suited our purposes. We therefore implemented our dialogue system as a directed, cyclical graph. A set of possible responses (transitions) is kept and transitions in the graph can add to or remove options from this set. Additionally, the graph nodes are grouped according to topic. If the user chooses to focus in on a particular topic,

so that he/she only responds, rather than risking having the user ask questions which the program cannot handle.

dialogue options not directly related to that topic are muted until the user indicates that they are done exploring that topic. A sink in the graph terminates the conversation with the agent (although it is possible to interrupt the conversation and resume it later, for example to switch off the radio).

3.4.5 User Interface

The interface was designed to be fairly simple to use. The user has a first-person perspective, as opposed to a third-person perspective. It is only necessary to use the mouse; keyboard controls are provided for people who are familiar with general first-person perspective computer games, but it is not necessary to use them. The user can look around by moving the mouse. They can move their avatar forward using the left mouse button and can interact with certain game objects using the right mouse button. Examples of objects which allow interaction are the doors which can be opened, the radio which can be switched on and off, and speaking with the agent. When the user can interact with an object, a small notification is displayed at the bottom to show the action which is possible.

Screenshots of the demonstration program are shown in Figures 3.2–3.4.



Figure 3.2: Fundiswa the tutor welcomes the user and provides a brief introduction to the program.

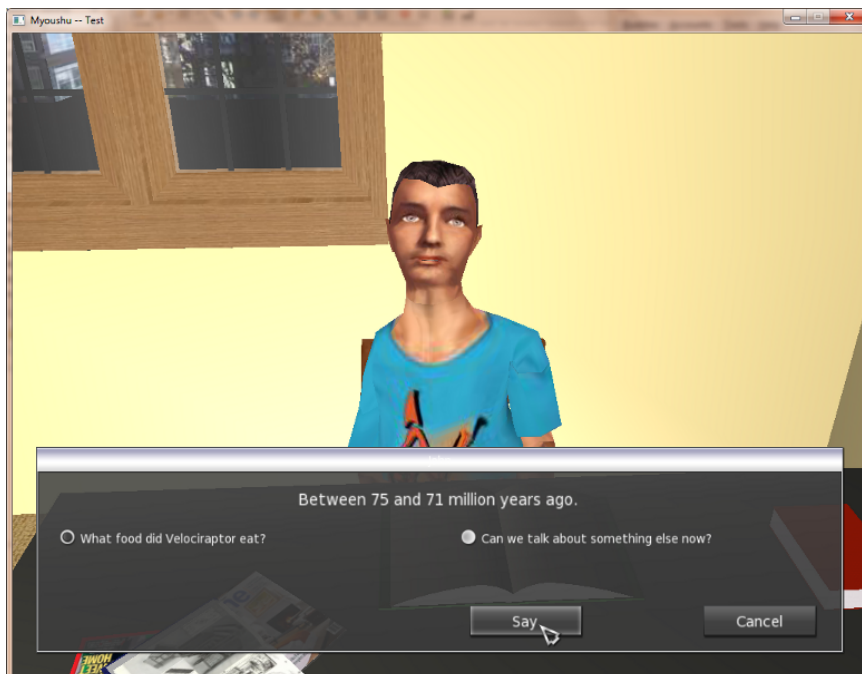


Figure 3.3: Our agent provides an answer to a previous question on one of his special interest topics. The answer was obtained by querying the SQL database.



Figure 3.4: The agent can be affected by the environment. In this case, the playing radio causes the agent sensory integration problems. If the user switches off the radio, the agent will be able to focus better and its demeanour will improve.

Chapter 4

Evaluation

This chapter describes efforts in evaluating the Thandiwe engine. A two-stage evaluation process was undertaken. The first stage consisted of analysing the inner dynamics of the Thandiwe engine. For the second stage of the evaluation, the system was presented to an expert in AS who assessed the credibility of the Thandiwe system from an AS point of view. In Chapter 5 we draw our conclusions from the analyses done in this chapter.

4.1 Evaluation Intent

Before presenting the evaluations which were done on the Thandiwe system, it is important to pause briefly and discuss what the intent was in analysing the system. The Thandiwe system is an engine which manages the internal aspects—affections and cognition—of an ECA. Our design was of such a nature that the engine would be especially applicable to ECAs intended to exhibit behaviour typical of someone who has AS. The system is rule-based, but the operation of the system as a whole is largely removed from the specific rules which are employed. Additionally, the system is fundamentally removed from any specific presentation layer, by which we mean

that which the end-user will ultimately see, which includes the virtual environment and the user interface.

For our analyses, it was necessary to create a demonstration program, which would provide a presentation layer and allow for human interaction with the underlying engine. This demonstration program is described in Section 3.4. The demonstration program is, at best, a crude prototype of what is envisioned to be built *on top* of the Thandiwe engine. It in itself was not a final goal.

The distinction between the underlying engine and the presentation layer is important to note. The reason is that it is not desirable for people to form an opinion of this project based only on the visual element of what has been done. The VE is, to the outside observer, certainly the most prominent aspect of the project. But for our purposes in doing this thesis, we had practical limitations as to the scope of the demonstration program and rules which were applied in the system. These limitations are outlined in Section 3.4.3.

The intent of the following analyses, therefore, is not to evaluate how well the agent in our demonstration program reacts to a *specific* situation, but to consider the mechanics of the engine and to assess whether it is sufficient to build a more complete agent which would react appropriately to various situations.

4.2 Configuration

As the Thandiwe engine is rule-based, it is necessary to first configure it in an appropriate manner. In this section we focus particularly on the configuration of affections (that is, the ALMA component) and how they relate to the responses which the agent can provide to the user.

To configure the response files, it was decided to make the responses of the agents conditional on the current mood of the agent. The reason for this is that emotions are considered unstable, while personality remains constant for the duration of the simulation. (It should be noted, however, that the reaction files can be configured to make use of emotions and personality values in addition to moods, should it be necessary for some implementation.) For our limited demonstration, we decided to further restrict ourselves to four of the eight fundamental moods. Note that, in reality, a fifth mood, *docility*, is present in every run of the demonstration program.

This mood is the default (see Section B.3). We never make explicit use of this mood for determining responses, however, so it suffices to consider only the four which are mentioned. The four moods which we do consider, *anxiety*, *hostility*, *relaxation* and *exuberance*, are opposites within the PAD mood space. Additionally, for a certain trajectory in the PAD space, *hostility* can be considered a “stronger negative” mood than *anxiety*, and *exuberance* a “stronger positive” mood than *relaxation*. This can be seen in Table 4.1, which shows the progression from the default mood to the two “extreme” moods.

| Mood | <i>P</i> | <i>A</i> | <i>D</i> | Associated Emotions |
|------------|----------|----------|----------|--|
| Exuberance | + | + | + | Gratification, Happy For, Joy, Love, Pride |
| Relaxation | + | - | + | Relief, Satisfaction |
| Docility | + | - | - | N/A |
| Anxiety | - | + | - | Anger, Disliking, Disappointment |
| Hostility | - | + | + | Fear, Remorse, Shame, Hate |

Table 4.1: The progression through the mood space necessary to achieve the desired moods.

To achieve these moods, it was necessary to configure events which would change the emotions of the agent, as emotions steer moods in the ALMA engine. As can be seen from Table 4.1, if we want the agent to move from the default mood to a positive mood, it is necessary for the *pleasure* dimension to remain positive, *arousal* must change gradually from negative to positive, and *dominance* must change rapidly from negative to positive. To move from the default mood to a negative mood, the *pleasure* dimension must change rapidly from positive to negative, *arousal* must change rapidly from negative to positive, and dominance must change gradually from negative to positive. Using this table, it was possible to deduce the emotional changes necessary to facilitate the desired mood changes.

Table 4.1 also shows which emotions are associated with the different moods, as defined in [45], which additionally lists how each emotion numerically affects the mood of each agent. Using inspection, it was found that the emotions *love* and *hate* would allow the desired progression to the “extreme” mood through the appropriate “intermediate” mood from the default mood as outlined earlier. Using the description of these emotions in [90], it was possible to correctly configure the reaction files (see Appendix A) with the appropriate appraisals to produce the correct emotional (in addition to verbal and visual) responses. For our limited demonstration program, it sufficed to make use only of the *GoodEvent* and *BadEvent* appraisals (each with an

intensity of 1.0, which is the maximum values for appraisal intensities) to affect the desired emotions after each input from the user (see [45, 90]).

4.3 Internal Evaluation

The first step in evaluating our work is to verify that the system as a whole functions as we expected. In addition to the system being large, it is built up from various individual components, most of which are third-party software (see Section 3.3). It is therefore only prudent to test the system “internally” to ensure that the system functions as expected—that is, that the various rule-based components interact with each other correctly. Investigating this is the purpose of the current section. The following section will investigate whether our expectations were valid and that the system, as we envisioned it, can indeed be used to create ECAs with AS behaviour which are believable.

To test the Thandiwe engine before presenting it to other people, it is necessary to run multiple simulations, each time trying to elicit different responses from the agent, and observing how the agent reacts and if it is different from the previous runs. To measure this, one can state the conditions under which the agent will give certain responses, and then measure how the affections of the agent changes over time.

To measure the “affections” of the agent, it is necessary to identify which values will be meaningful to measure. For the ALMA system, the options are to measure how personality, moods and emotions (see Section 2.1.1.2) vary over time. In our system, personality remains constant, so simply stating the OCEAN values is sufficient. These are given in Appendix B. Emotions are cumbersome to measure, as one either needs to measure the intensity of all 22 at a time, or only the dominant emotion. Only the dominant emotion is used to influence the rest of the system, but providing only the value of the dominant emotions says nothing about the relation to other emotions. So it would not be known whether the dominant emotion is strong or only marginal.

In the demonstration program, moods, rather than emotions, were used to determine reactions. The reason for this is that, in the ALMA system, emotions are unstable and change rapidly. Moods were introduced precisely to function as an intermediary

between rapidly changing emotions and constant personality, allowing one to be able to say something meaningful about the overall current state of the affections of an agent. Consequently, measuring mood changes was more sensible than measuring emotions. If reactions were based on emotions, it would have been better to measure them in some way.

4.3.1 Typical Results

We now consider some typical results. Note that these results were obtained before the calibration of the rules was completed.

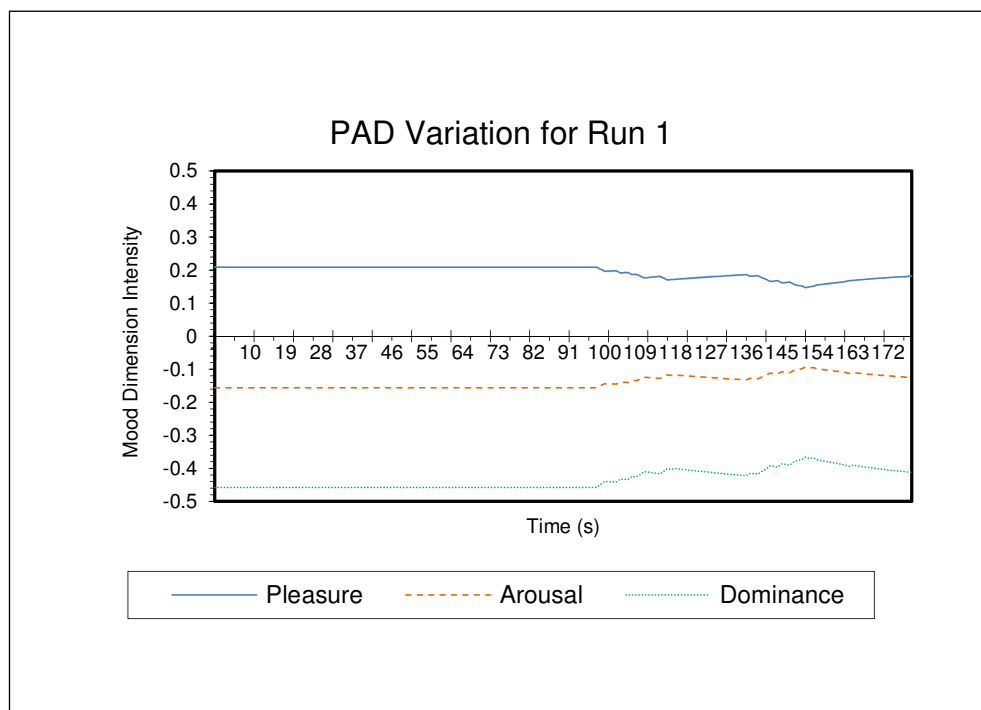


Figure 4.1: The PAD values for the first run.

Figure 4.1 shows a run which lasted nearly three minutes (178 seconds). At the beginning, *pleasure* is positive (0.209), while *arousal* and *dominance* are negative (-0.156 and -0.458 , respectively). This indicates the default mood of the agent, which is *docility*. These values remain constant for 90 seconds. It can be seen that, after 90 seconds, the mood dimensions start to fluctuate. *Pleasure* (which remains positive) shows a nett decline, with a minimum of 0.147. *Arousal* and

dominance (both remain negative) show a nett incline, peaking at -0.093 and -0.366 respectively.

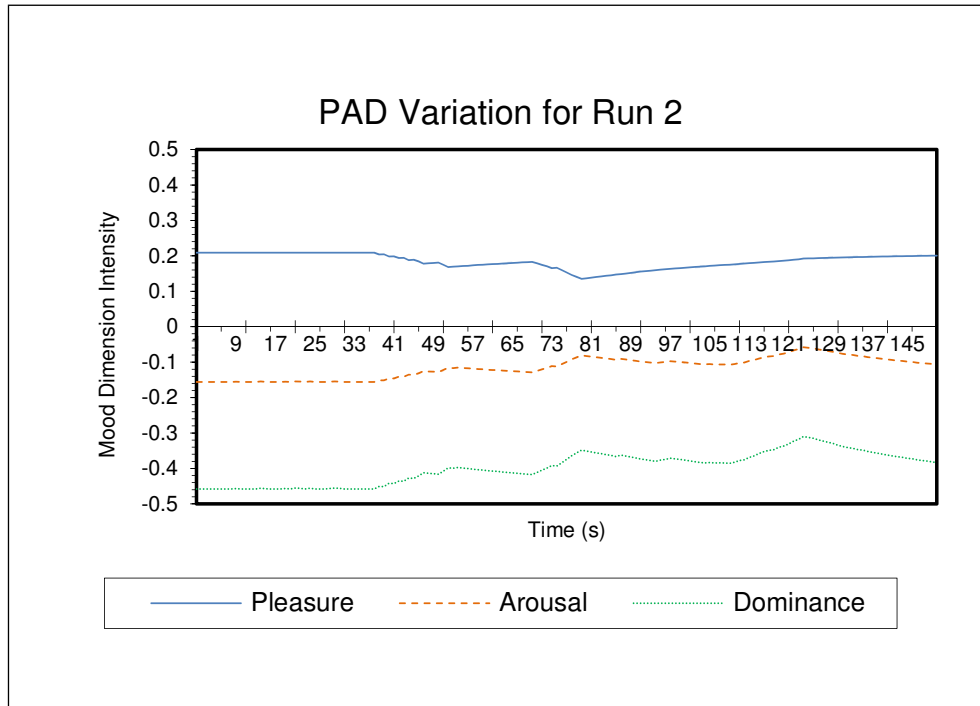


Figure 4.2: The PAD values for the second run.

Figure 4.2 shows a run which lasted just over two and a half minutes (151 seconds). It is similar to Figure 4.1, except that the mood changes already start in a minor way at 9 seconds and the changes in Figure 4.1 are being reflected from 38 seconds onwards. From about 100 seconds onwards, the trend continues beyond what was seen in Figure 4.1. While *arousal* and *dominance* reach global maxima beyond this point (-0.058 and -0.310 respectively at 124 seconds), *pleasure* does not, as it appears to decay to the original value.

It needs to be noted that, while the first run was longer than the second run, it was approached more slowly than the second run. With the second run, the agent was engaged in non-neutral conversation topics sooner than in the first run, and the exchanges between the user and the agent were quicker.

Figure 4.1 and Figure 4.2 both show how the mood dimensions can be influenced through conversations in the Thandiwe system. As these figures are similar in both how the conversation with the agent was conducted and the results obtained, we shall

focus on Figure 4.2 for the remainder of this discussion, as it is the more detailed of the two. Seven distinct parts of the conversation can be identified. These are detailed in Table 4.2.

| Part | Time (<i>s</i>) | Topic | PAD Progression |
|------|-------------------|-----------------------------|-----------------|
| 1 | 0–38 | Introduction and Small Talk | |
| 2 | 39–52 | School Topic | –P+A+D |
| 3 | 53–69 | Neutral Topic | |
| 4 | 70–79 | Family Topic | –P+A+D |
| 5 | 80–109 | Neutral Topic | |
| 6 | 110–124 | Dinosaur Topic | +P+A+D |
| 7 | 125–151 | Neutral Topic | |

Table 4.2: The seven distinct parts which can be identified from the results indicated by Figure 4.2.

Part one shows the time which elapsed during the introduction (the tutorial and navigating to the agent), as well as the initial exchanges in conversation, which were not stimulating to the agent. At this stage the mood space of the agent is in the default octant of *docility*. The first change, which is observed in part two, shows that the mood space of the agent begins to tend towards $-P+A+D$, which is *hostility*. This shows the effects of the negative emotions which the agent experiences while discussing the “school” topic. Part three shows a gradual return to the default mood during the neutral interlude which separates the provocative topics. Part four shows another tend towards hostility, which is expected when discussing the “family” topic. The change happens slightly more rapidly, indicating that the agent is experiencing the negative emotions more intensely. Part five, similar to part three, shows an interlude. The agent is not being stimulated at this point, allowing a gradual return to the default mood as it “cools down”. Part six again shows a move away from the default mood, but towards a different mood than what was seen in parts two and four. This change sees the mood space tend towards *exuberance* (only the *arousal* and *dominance* dimensions change direction, while the *pleasure* dimension continues in the same direction as before). At this point the “dinosaur” topic was being discussed, which was pleasing to the agent. Finally, part seven shows another return to the default mood, as the conversation was concluded in a neutral fashion and the player leaves the game.

Note that these runs excluded the “background distractions” of the radio and the dog barking, which would have amplified the negative affections and dulled the positive affections.

From these experiments we are satisfied that the system is functioning as expected. We were able to make predictions about how the agent should react for given inputs, and observed the appropriate reactions. We also verified that the various components, for example the affections and the knowledge-base, worked and interacted as expected by providing appropriate responses. At times, analysing the results was an intricate process. This was good to observe. While it means that creating a finely tuned system is, at present without authoring tools (see Section 5.2.1), an arduous task, the complexity lends itself to creating agents which, if done correctly, are likely to seem less superficial to a user.

4.4 Expert Opinion

We wanted our project to be evaluated by someone who was not involved in the technical aspects of its development. Because of the focus on AS, it was necessary to approach an expert in that field for our second stage evaluation. The reasons for this was that our field is Computer Science and we have not had any formal training to make credible assessments about autistic behaviour and realism. Even our personal interaction with children who have AS has been limited and with a small number of individuals. For these reasons we felt it necessary to seek out an expert who can make credible assessments, but who is also unbiased in the sense that they had been removed from any of the technical details of what we had done.

Due to the limited number of highly trained and accessible experts in autism in South Africa, it had to suffice to approach only a single expert. Our expert was Jana Forrester, an educational psychologist who specialises in AS. She was involved with the Vera School for Learners with Autism in Rondebosch, Cape Town, South Africa for many years. She is now a private consultant.

4.4.1 Demonstrations

The evaluation proceeded as follows: starting with only rudimentary knowledge of the project, the expert ran one iteration of the demonstration program. Afterwards, the expert's experience with the system was discussed and more information concerning the system was revealed. The demonstration program was then run again. This process continued for three iterations. At the end of the iterations, the expert was asked to assess the credibility of the system in terms of positive and negative experiences.

After the system was set up, it was explained to the expert what the intent of the evaluation was, as discussed in Section 4.1. It was also explained that the demeanour of the agent could be affected positively, negatively, or not at all through the use of five conversation topics (see Section 3.4.3), but it was not divulged what the topics were. The expert was left to her own devices to navigate and use the demonstration program. Only when it became apparent that she was struggling with something in particular was advice offered. The following observations were made during the first run-through of the demonstration program:

1. Navigating the game initially seemed to be a problem. This was not a major concern and was expected to a degree, as the navigation system can be expected to be intuitive only to people who have experience in playing typical first-person computer games. (By the end of the session the expert could comfortably navigate the VE, however, which is an indication that the navigation system can be mastered with little practice.)
2. It was noted that the expert intentionally avoided the conversation topic on spaceships.
3. In contrast, she immediately opted to explore the "school" topic. It was apparent that she wanted to agitate the agent to see how it would react. When that topic had concluded, she was satisfied, saying to herself "Now he is upset!" based on the reactions of the agent.
4. With the family topic the expert again showed interest and pursued it when it became available. Unfortunately, due to the nature of the rule-based system, the user inputs and agent responses were not coherent for this topic.

5. Finally, the expert touched on the dinosaur topic, but ended that course of dialogue almost immediately. From what was observed, this may have happened accidentally.

In summary, the AS expert chose to explore the two topics which were structured in such a way as to agitate the agent, while almost completely avoiding any topic which would affect the demeanour of the agent in a positive manner.

After the first run of the demonstration program, the expert was asked if she could identify the different topics and how they would affect the agent, which she did correctly. The first run of the demonstration program was then discussed in general. It was revealed that the expert felt quite restricted with the available dialogue options, as well as with the fact that some of the responses were illogical. When asked about why she avoided the special interest topics, the expert explained that she had hoped to see that the AS agent would stubbornly insist on speaking on the topic. The expert was reminded not to focus too much on the presentation layer, and further explanation was given as to the differences between Thandiwe and the VE (which was built on the Myoushu engine). She was also encouraged to explore the “positive” (special interest) topics. After this discussion, the expert was eager to run through the demonstration a second time. The following observations were made during this second run through the demonstration program:

1. The expert thoroughly explored all the conversation topics.
2. It was detected that the agent responded differently than in the previous run. This was due to the agent first being engaged in a topic which would affect its demeanour positively, as opposed to the previous run where all the topics were negative.
3. While exploring the dinosaur topic, the expert remarked that she knew a boy with AS who would be “crazy about” engaging the agent in that topic¹.

¹The special interest topics loop indefinitely, each time giving the user the option to ask the agent a question which it then answers from its knowledge base. The questions themselves are randomly generated. While this mechanism was used to make it possible to vigorously “excite” the agent (this counters other effects in the game which can affect the demeanour of the agent in a negative manner, such as the radio playing in the background), it seems that this minor addition to the demonstration program could find a greater use in a larger application intended for use by children with AS.

When the second run had concluded, the system was explained to the expert in more detail and she was shown some of the configuration files. It was explained that the system is rule-based, and that there were practical limitations to the number of rules with which the agent could be programmed; a fact which necessitated that the user be restricted in the demonstration program.

A third run of the demonstration program was then made, but nothing of significance was observed during this run.

4.4.2 Expert Evaluation

At the end of the session, the expert was asked to evaluate the system as a whole in terms of negative and positive points. These are listed below, as well as some of the recommendations which the expert made.

Negative Observations

1. The program was “static” in that it had little sound, no speech and few animations.
2. There was a feeling of being “hamstrung” by the available questions which could be asked at any given point.
3. There were only questions and no way to “transition” smoothly to other topics. The existing dialogue contained many sudden “jumps” which are not typical of a person who has AS (or of a neurotypical person who is talking to someone with AS).
4. Nothing in the demonstration program was verbal: it was all text-based. It would have been better to have speech recognition and speech synthesis or voice recordings of responses.

Positive Observations

1. The one agent animation that was present (which relates to the tutor turning around and walking out of the door) was liked for being “quirky”.

2. There was a good balance between personal and content questions from which to choose.
3. Anyone can “participate”, meaning that program can be accessible to someone regardless of their technical expertise or whether they have AS or not.
4. The demonstration program was deemed, on a whole, as a good departure point from which to build more sophisticated and realistic agents and VEs.

Recommendations

1. The first suggestion was to perhaps have an “execution plan” against which the responses of the agent in the demonstration program can be compared. The execution plan would broadly cover the various optional dialogue branches and enable someone to reproduce a particular reaction from the agent by following the execution plan.
2. A person who is about to play the demonstration should be given more context about the setting and purpose of the demonstration program. This could take the form, for example, of a succinct summary of Appendix B, or clarifying the “goal” of the game better in the tutorial which is already present at the beginning of the game.
3. Because the demonstration program lacks verbal or auditory feedback, it was suggested that the textual responses should be annotated with colours, symbols or icons to provide an indication of the state of the demeanour of the agent.

4.4.3 Responses to Criticisms

It is interesting to note that most of the criticism given relates to the presentation layer, while the mechanics of our engine were viewed favourably.

The following responses can be provided to the criticisms which the expert raised:

1. The program is “static” due to the limited scope within which it was constructed. Animations², speech and sound can be added, but these mostly

²While the Thandiwe system can theoretically integrate with an animation system, this feature was not fully implemented, as making animations for the agent would have been too time consuming.

concern the presentation layer and stands apart from the Thandiwe engine.

2. Due to the limited ruleset, it was necessary to restrict the user in what could be said to the agent. The agent only has restricted knowledge over a handful of knowledge domains. If the user was free to say anything to the agent, it would likely get stuck repeating a small number of default reactions, which would lead to confusion and frustration on the part of the user.
3. The fact that the user is mostly limited to only asking the agent questions is admittedly not natural nor ideal. The reason for only allowing the user to ask questions, is that it was felt that, given the limited number of rules, the user should not (be allowed to) lose the initiative in the conversation. It is an oft-used technique with chatterbots to take the initiative away from the user and to interrogate the user (see Section 3.1.3.2). For our purposes, however, we wanted the user to rather explore the AS of the agent and felt that asking a series of questions was the best way to do this. Additionally, if the user were to lose the initiative in the conversation, the structure of the rules and conversation options would change significantly, which is a scenario which simply could not be accommodated at this time. As for the “jumps” in a conversation, it is always a risk in rule-based systems. If a certain user input cannot be matched to a rule, a default reaction is returned. Or a rule might match which incorrectly identifies the topic or intent of the user, which could then yield an unexpected response. The likelihood of both of these situations can be reduced by adding more rules which, as has been stated before, was outside the scope of this thesis. Alternatively, the agent responses could have been designed to be deliberately vague. This technique was not desirable for our purposes, as we did not feel this would be appropriate for an agent with AS.
4. As stated in the first point, speech concerns the presentation layer and can be added to work with the Thandiwe engine. While everything is currently text-based, established technologies could be used to recognise user speech and turn it into text, as well as to synthesise the responses of the agent into speech.

Furthermore, the suggestions which were made by the expert were all welcomed as being sensible and practical. These would be strongly considered for future addition to the demonstration program.

At the conclusion of our demonstration session with the expert, we were satisfied that the mechanics of our agent became apparent during the different demonstration runs, that she had generally viewed our system in a positive light, and that she was excited by the potential of our system and what could possibly be done with it.

4.5 Limitations

At this point it is only right to acknowledge the limitations of the evaluations which have been done for the Thandiwe system.

4.5.1 More In-depth Internal Evaluations Needed

The efficacy of the “background distractions” mentioned in Section 4.3 must still be established for the VE which we created. As the mechanisms which allow the background distractions to influence the emotions of the agent are the same as that which allow conversations to influence the emotions, the assumption was made that it is not necessary to explicitly analyse it. It would, however, be better to validate this part of the system.

4.5.2 More Expert Evaluations Required

When polling individuals, especially when it is for opinions, it is always better to have a larger sample than a small one. In our case we only had a single expert which evaluated our agent. It would be best to find more experts to evaluate the agent.

4.5.3 Observations Need to be Sanitised

Observations made of the reactions of the expert was done by the author. Apart from the fact that reading emotions can often be subjective and flawed, the additional danger exists that the interpretation is biased. The credibility of the observations could potentially be enhanced by having third parties (who were not otherwise involved with the project) doing the demonstrations and observations. Or, at least, the observer should be accompanied by one or more persons so that observations

can be compared and discussed. Whatever strategy is employed, the strategy itself should not potentially compromise the results by, for example, making the subject nervous because they are being observed by a crowd. Videotaping might be the best solution, as minimal involvement is necessary and the video can be analysed at a later stage by anyone.

4.5.4 Evaluations in Future Work

Given these limitations, further evaluation of the Thandiwe system should be done. This should be taken into consideration especially for any further work which is to be done involving the Thandiwe system (see Section 5.2).

Chapter 5

Conclusions and Future Work

In the previous chapter we discussed results obtained when analysing how the game engine responds to user input during a conversation. We also explored how the system can be used to create believable agents with the behavioural traits of people with autism. In this chapter we first discuss what conclusions can be drawn from these analyses. Thereafter we describe future work which can enhance our system, as well as provide suggestions on how the project described in this thesis can be used in other projects.

5.1 Review of Evaluation Results

The “internal” evaluation was discussed in Section 4.3. In the analyses it was seen that the various components interacted with each other in an expected manner and that the system thus as a whole functions correctly. We could then proceed to refine all the necessary rules in order to fully simulate the agent and environment which we intended to.

The “external” evaluation was discussed in Section 4.4. Overall, the expert’s opinion of our system was a good one. A few important suggestions were made, how-

ever, mostly regarding the demonstration program and considerations to be made when doing an agent implementation with our engine. We are satisfied, however, that—given sufficient rules—a believable agent with autistic behavioural traits can be constructed with our system.

5.2 Future Work

In this section we discuss what still needs to be done with our project to enhance it, as well as areas of research to which our project can lead.

5.2.1 Technical Work

XML Flavour Consolidation

At present, the system makes use of two main XML flavours: AIML [19] and TRF. The former is used to process input and then connects the engine to the appropriate XML file in the latter flavour. This is inefficient and the system can be improved by merging the two standards. Merging the two standards would also mean merging an AIML interpreter (at present, the open source Program D) with our own interpreter. This holds several advantages in terms of simplifying and customising the AIML interpreter to better suit the needs of the Thandiwe engine, which was already necessary to do to a lesser extent, as mentioned in Section 3.2.2.1. On the other hand, merging these two standards will effectively mean that it would be necessary to fork from the current AIML standard. The forked standard would then have to be manually updated the merged with our XML dialect with any new modifications made to the original AIML standard when they occur, should it be desirable to do so.

Creation of Authoring Tools

At present it is cumbersome to create the configuration files. One has to carefully plan out which causes and effects are necessary to calibrate in order to achieve the desired emotional changes in the agent. While the ALMA library provides a GUI for creating character affect profiles, and there are various third party tools for writing AIML scripts, there currently exist no editors to create TRFs. It would

be advantageous to implement a unified graphical editor for authoring not only TRFs, but also AIML and ALMA files as well. A basic GUI program can then later be extended to provide advanced features, such as detecting inconsistencies in the dialogue authoring or suggesting parameters based on previously entered values.

Formalising an API

We placed high importance on following good software design and development practices. From the inception of this project, it was envisioned that it could one day be used in other systems, such as Second Life. At present, the application programming interface (API) does require some tidying up, however. It is still necessary to formalise the API and to provide interfaces to different languages. For example, in the case of Second Life, it would be necessary to provide an interface for the Linden Scripting Language.

5.2.2 Research Possibilities

Implementation as a Teaching Tool about Autism

By greatly expanding the reactions of an agent, including verbal and non-verbal, an ECA “with autism” of comparable efficacy (if not better, due to the niche context) to contemporary ECAs can be built. An agent which is believable in a broader domain than the one presented in this thesis, can be included in games and educational software, bringing autism to media which are dominated by neurotypical thinking. One outcome of this can be that neurotypical people, who typically have no or very limited knowledge of or exposure to people who have autism, can be taught about autism and the people who have it.

Implementation as a Teaching Aid for People with Autism

ECAs with AS can also be included in educational software for children who have AS, especially those concerning social skills training. It could be that a child who has AS will find it more appealing to engage with an ECA with AS than with an ECA which is intended to be NT. The idea is that the child and the ECA could progress

through social scenarios together, learning from each other. In one case, the ECA could provide encouragement, praise and correction in a way which appeals to the child. The agent would be able to counsel the child from an empathetic perspective. The roles can also be reversed: in such a case the ECA could be like Shybot [71] (see Section 2.2.3), demonstrating behaviour to the child in a context where the child questions the behaviour. The child can then encourage the ECA to rather act in a way which is more socially mainstream.

While it may not necessarily be the case that children with AS will find interacting with ECAs with AS more appealing than with ECAs which are intended to be NT, we believe that this is an interesting idea which should definitely be investigated.

Developing Reaction and Knowledge Libraries

The aforementioned research possibilities largely hinge on the condition that the agent has a large set of reactions to draw from which is specific to the various domains which it can encounter. Reactions are specific to individuals and certain personalities, which means that no definitive library of reactions could ever be compiled from which all agents could be constructed. However, it is possible to construct a library of generic reactions which could serve as a guide for constructing an agent with a specific personality. This library would be generic not across personalities, but across *ages, cultures, and degrees of autism*. In our demonstration program, we focussed specifically on an older boy who is high functioning from a western, middle class family. However, the agent might as well have been a girl, or a toddler or an adult. The agent could have been low functioning. Or the agent could have been from a different ethnic group or social standing, where different perceptions of autism may have influenced the world-view of the agent. For certain agents, non-verbal reactions will be more significant than the verbal reactions. This will typically vary with age and/or severity of autism. Other agents will provide different verbal reactions due to different ages or social context. The construction of such a library would be a fascinating undertaking in the study of reactions of people with autism from different walks of life to various situations.

It would also be beneficial to construct a knowledge library for special interests. Other than the reaction library, a knowledge library can be truly generic, as a special interest can be on any topic and at any depth. “Typical” special interests can be

identified which represent domains of knowledge which are highly prevalent amongst children with AS [9]. Examples are the knowledge of spaceships (which falls in the domain of folk physics), and of dinosaurs (which can be viewed as falling in the domain of taxonomy). But such a knowledge library would still only serve as to provide examples, as many people with autism have esoteric special interests, such as, for example, glue and paste, Parisian geography, organisational psychology, and Hebrew [50]. While the library itself can be generic, the application thereof to the Thandiwe engine can be facilitated by providing an appropriate SQL query builder for each topic (see Section 3.3.3).

5.3 Conclusion

In this thesis we considered the problem of building a prototype system in which to develop embodied conversational agents which exhibit autistic behaviour. The system is mostly designed around existing technologies and is rule-based to allow for flexible configuration. We also implemented a prototype agent to demonstrate the capabilities of our system.

In the design of our system we looked at the various aspects of ECA design as found in the literature. Specific ECA systems were considered, as well as related technologies and studies. Particular attention was paid to systems or technologies which has direct application towards people with autism, although no work comparable to our own was found. We then used what was learned from the literature to design our own system. We laid out the particular requirements around which we were designing the system, the scope within which we were working, the rationale of some of the design choices which were made, and some of the issues which we encountered. The system and its components were then detailed. After this the prototype agent—and the environment which it inhabits—were implemented. The agent was based on a character profile which was drawn up in an effort to make the agent more believable not merely as a computer character, but one which resembles a human being.

The prototype agent was then evaluated in two phases. The first phase analysed the “rule-flow” of the system and confirmed that careful analysis can predict the output of the system. In the second phase the prototype agent was presented to an expert in Asperger Syndrome who critically evaluated the believability of the agent. The overall response from the expert was favourable. Following the outcome of these

analyses, we are satisfied that our end product meets the goals and requirements which we set out for ourselves in the beginning of this thesis.

Finally, we presented suggestions of work which can still be done. The future work was divided into technical work which can enhance our system, and research possibilities which can flow from our work. We are confident that projects which build on our work, as well as the many new developments in the related fields, will result in exciting research which can benefit both people who are neurotypical and people who have autism.

Appendices

Appendix A

Thandiwe Reaction Files

Here we present the XML flavour which we developed to govern the reactions which our system produces for some given input, based on the internal affective state of the ECA. We chose to call this flavour *Thandiwe Reaction Files* (TRF). First we present the *definition* of the XML flavour, then an example of what such a reaction file looks like, and then a high-level discussion of what data the XML encodes. This chapter is not meant to be a formal definition of an XML standard. The reason for this is that we do not wish to create a “standard” at this time (see Section 5.2.1). Rather, we feel that it will suffice to simply give a high-level discussion at this time.

The definition is presented as a Document Type Definition (DTD), which is a standard for defining XML flavours. Unlike other standards for defining XML flavours, such as XML Schema (XSD) and Regular Language for XML Next Generation (RELAX NG), DTD is itself not XML. It is, however, reasonably simple to understand. It has certain technical limitations, but nothing which would hinder what we are trying to do. Based on this, we therefore thought that it would suffice to use DTD rather than XSD or RELAX NG.

A.1 Document Type Definition (DTD)

```

<?xml version="1.0" encoding="UTF-8"?>

<!ELEMENT decisions (set*)>
<!ELEMENT set (input,rule)>
<!ELEMENT input (#PCDATA)>
<!ELEMENT rule (condition*,output+)>
<!ELEMENT condition (affect|cognition)>

<!ELEMENT affect (emotion*,mood*,personality*)>

<!-- ===== -->

<!-- OCC -->
<!ELEMENT emotion (admiration?,anger?,disliking?,
  disappointment?,distress?,fear?,fears_confirmed?,
  gloating?,gratification?,gratitude?,happy_for?,
  hate?,hope?,joy?,liking?,love?,pity?,pride?,relief?,
  remorse?,reproach?,resentment?,satisfaction?,shame?)>
<!ELEMENT admiration (val)>
<!ELEMENT anger (val)>
<!ELEMENT disliking (val)>
<!ELEMENT disappointment (val)>
<!ELEMENT distress (val)>
<!ELEMENT fear (val)>
<!ELEMENT fears_confirmed (val)>
<!ELEMENT gloating (val)>
<!ELEMENT gratification (val)>
<!ELEMENT gratitude (val)>
<!ELEMENT happy_for (val)>
<!ELEMENT hate (val)>
<!ELEMENT hope (val)>
<!ELEMENT joy (val)>
<!ELEMENT liking (val)>
<!ELEMENT love (val)>
<!ELEMENT pity (val)>
<!ELEMENT pride (val)>
<!ELEMENT relief (val)>
<!ELEMENT remorse (val)>

```

```

<!ELEMENT reproach (val)>
<!ELEMENT resentment (val)>
<!ELEMENT satisfaction (val)>
<!ELEMENT shame (val)>
<!ELEMENT dominant (val)>

<!-- PAD -->
<!ELEMENT mood (pleasure?,arousal?,dominance?)>
<!ELEMENT pleasure (val)>
<!ELEMENT arousal (val)>
<!ELEMENT dominance (val)>

<!-- OCEAN -->
<!ELEMENT personality (openness?,conscientiousness?,
  extroversion?,agreeableness?,neuroticism?)>
<!ELEMENT openness (val)>
<!ELEMENT conscientiousness (val)>
<!ELEMENT extroversion (val)>
<!ELEMENT agreeableness (val)>
<!ELEMENT neuroticism (val)>

<!ELEMENT cognition (chk)>
<!ELEMENT chk (var,val)>
<!ELEMENT var (#PCDATA)>

<!ELEMENT val ((lt,gt?)|gt|eq|true|false)>
<!ELEMENT lt (#PCDATA)>
<!ELEMENT gt (#PCDATA)>
<!ELEMENT eq (#PCDATA)>
<!ELEMENT true EMPTY>
<!ELEMENT false EMPTY>

<!-- ===== -->

<!ELEMENT output (eec?,act?,action*,response)>

<!ELEMENT act (#PCDATA)>

<!ELEMENT eec (desirability?,praiseworthiness?,
  appealingness?)>
<!ELEMENT desirability (#PCDATA)>

```

```

<!ELEMENT praiseworthiness (#PCDATA)>
<!ELEMENT appealingness (#PCDATA)>

<!ELEMENT response (#PCDATA)>

<!ELEMENT action ((before|after|delay)?,animation)>
<!ELEMENT before EMPTY>
<!ELEMENT after EMPTY>
<!ELEMENT delay (#PCDATA)>
<!ELEMENT animation (#PCDATA)>

<!-- ===== -->

<!ATTLIST set
  id ID #IMPLIED
  cat CDATA #IMPLIED
>

```

Thandiwe Reaction File Specification (DTD)

A.2 Example

```

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE reaction PUBLIC '-//SUN//DTD Thandiwe Reaction XML
  1.0//EN' 'http://www.cs.sun.ac.za/~wventer/dtd/Reaction.dtd'
>

<reaction>
  <set xml:id="test_00001">
    <rule>
      <condition>
        <affect>
          <personality>
            <neuroticism>
              <val><gt;0</gt></val>
            </neuroticism>
          </personality>
        </affect>
      </condition>
    <output>

```

```

    <eec>bad_event</eec>
    <response>Rule 1, Response 1</response>
    <animation when="before" delay="100">jump01</animation>
    <animation when="during" delay="200">twitch04</
      animation>
  </output>
  <output>
    <response>Rule 1, Response 2</response>
  </output>
</rule>
<rule default="true">
  <output>
    <eec>bad_event</eec>
    <response>Rule 2, Response 1</response>
  </output>
  <output>
    <response>Rule 2, Response 2</response>
    <animation when="during" delay="200">twitch12</
      animation>
  </output>
</rule>
<rule>
  <condition>
    <affect>
      <personality>
        <openness>
          <val><gt>0</gt></val>
        </openness>
      </personality>
    </affect>
  </condition>
  <output>
    <eec>good_event</eec>
    <response>Rule 3 Response 1</response>
    <animation when="before" delay="100">jump02</animation>
  </output>
  <output>
    <response>Rule 3 Response 2</response>
  </output>
</rule>
</set>

```



```
</reaction>
```

Example Thandiwe Reaction File

A.3 Discussion

Sets

A TRF contains one or more *sets*. Sets are associated with unique identifiers (`xml:id`). These identifiers are produced by the AIML interpreter for some given input. Thus, a set of possible reactions are associated with some natural language input. For one such input, a set defines various checks (called “rules”) which can influence which reaction (called “outputs”) the ECA will have to the input. This makes a set a basic unit for dealing with input.

Rules

A set has at least one *rule*. Of all the rules a set can have, one must always be marked as “default”. A rule contains at least one *condition* and one *output*.

Conditions

Conditions are used to check the internal affective states of the ECA. These can be personality traits¹, mood states and emotional states. The personality traits are checked against the OCEAN values, moods are checked against PAD values and emotions against OCC values. Checks are in the form of expressions which evaluate to boolean *true* and *false* values. Specifically, we include checks for *equality*, *less than*, *greater than*, and boolean *true* and *false*. Atomic expressions cannot be compounded together explicitly using boolean *and* and *or* conditions, but this can be done implicitly. As mentioned, multiple rules are allowed. Internally, a boolean *or* divides these rules. It was also mentioned that multiple conditions are allowed within a single rule. These multiple conditions are internally combined with an

¹Personality traits are constant within our system, so in certain cases it can be redundant to check them. These checks can be used, however, if the same TRFs are used for different agents.

implicit boolean *and*. While this system can, admittedly, be cumbersome to use if the conditions applied to a set are complex, it serves its purpose.

Conditions can also accommodate logical values, called *cognition*. A cognition can apply a logical check on some non-affective internal value. For example, assume that the agent has a variable `isRaining` in its knowledge base. This variable reflects something about the game state which the agent (supposedly) knows. This value can be checked for truth in cognition to elicit a response. At present, the TRF specification supports such logical checks, but, as explained in Section 3.1.3, the engine itself does not support this functionality at this stage.

Outputs

Associated with each condition is one or more *output*. An output is what the TRF interpreter eventually passes on to the rest of the system and the VE. An output must have a *response*² and can optionally contain one *act* and multiple *actions*. A response is the “spoken words” with which the ECA reacts. Specific acts are defined within the ALMA configuration files and affects the internal affective state of the ECA. For example, shouting at the ECA might elicit the `bad_act_other` act, which can be defined to steer the ECA towards the *disliking* emotion.

Actions define animations which the ECA can perform. Animations can be played *before* the ECA starts to “speak”, while it is “speaking” or after it has finished “speaking”. *Delays*, measured in milliseconds, can be associated with each before/during/after timing. For example, the amount of time that must pass between the end of the animation and the start of the “speech”, the amount of time to wait after the “speech” has started before the animation starts, and the amount of time to wait after the “speech” has ended and before the animation starts, respectively. The names of available animations are defined internally in the VE and has to be made available to the person writing the TRF.

If multiple outputs are associated with a single condition which succeeds, then one output is chosen at random. If no rule exists for which all the conditions are met, an output is chosen from the default rule.

²A future revision of TRF may relax this restriction, as a reaction need not always be verbal. This condition was added owing to the chatterbot roots of the system.

Appendix B

Character Profile

B.1 Introduction

In this appendix, we present a character profile for our ECA, which features in our demonstration program. When designing a fictional character with such complex behaviour as someone who has Asperger Syndrome, or if a character simply has to be believable in a personal conversation, it is vital to have a thorough character profile in mind. Even if not all the detail in the profile is used in the end story or product, it helps to shape the character in the mind of its creator, helping him or her to better relate to the character and thus produce a more believable story [37]. Agents with “character” are also proposed by Dryer [35]. To design our character, we drew on inspiration from discussions by clinical experts about AS [3, 50, 102], personal accounts from children who have AS [54, 64] and fiction which accurately portrays AS in children [53]. Note, however, that this profile was never intended to be exhaustive.

B.2 Profile

B.2.1 Personality

John is an eleven-year old boy who has AS. He was diagnosed at the age of five when his parents noticed that he stopped speaking, did not make eye contact, and did not like being touched, held or cuddled. While John can now be quite vocal, he still does not naturally make eye contact and does not like to be touched unexpectedly and especially hates being crowded.

Since an early age, John has had a marked obsession with dinosaurs. He preferred dinosaur toys and collected children's magazines which featured dinosaurs. He memorised a large number of facts about a wide variety of dinosaurs and often astounded adults by the sheer amount of raw facts which he could recite about the animals. This obsession with dinosaurs has abated somewhat over the past couple of years, although John retains all of the details which he had memorised years earlier. Of late John has been showing an interest in the Star Trek franchise and seems to have developed an obsession with it. In particular, he engrosses himself with learning everything about the spaceships¹ which feature in that science fiction story. While he does most of this learning on websites which are dedicated to the franchise, he has also pestered his parents into buying all sorts of books and DVDs, including "factual" books such as technical manuals and The Star Trek Encyclopedia.

John has been receiving specialised care since his diagnosis, meaning the best part of his childhood. He understands his limitations in relation to neurotypical people, as do his parents and older brother. His younger sister, however, does not yet fully grasp the implications of her brother's AS. He enjoys swimming, which was originally proposed as exercise to help him with his motor control. Other than that, he shows no interest in sport.

Some of the daily difficulties which John faces include: fine motor control (which makes things like writing very unpleasant), coordination problems (which manifests as general clumsiness), sensitivity to noises, reactions to unexpected physical contact, and a difficulty in interpreting body language and facial expressions.

¹Our choice of the dinosaur and spaceship topics reflect typical areas of special interest in children who have autism, especially the latter, which lies in the domain of "folk physics" (see [9]).

B.2.2 School and Friends

He attends Groenebosch Primary School (which is fictional), which is a local mainstream school, after his parents and his educational psychologist decided that he would be able to make the adjustment from being home schooled. He has been in Groenebosch Primary since the beginning of last year. Initially he had some difficulty adjusting to the new environment, but after a few months he seems to be coping much better. Previously John had received home-schooling and attended the (fictional) Vela School for Learners with Autism for some specialised learning.

Interestingly, John gets only average grades for mathematics and does not seem to particularly enjoy the subject, but does very well in grammar. He does not do well in oral, essays and comprehension tests, however. He does not like history, science or geography, but he does like biology and computer studies.

All of the teachers at Groenebosch Primary are aware of the fact that John has AS. Some of his teachers therefore devote more time and patience to him. Despite these good intentions, only a very few of these teachers adequately understand AS. In contrast, other teachers try not to treat him differently from the other children, feeling that one of the challenges which he must overcome in a mainstream school is learning how to integrate into a larger social structure.

John has an interest in making friends, but has few of them. His efforts are often met with ridicule and bullying, but after more than a year at the school, most of the children in school are aware of his AS and either actively ignore him or act friendly towards him. He is often still confused by many social activities and conventions, but has learned to rely on a handful of close friends in such situations.

B.2.3 Family

The Petersons live in Groenebosch (which is a fictional suburb) near Cape Town, South Africa. They are a Caucasian, middle-class family.

John's father is Robert and he is aged 42; Robert works as a programmer for a small start-up company in the city centre. His mother—Alison, aged 40—works as a personal assistant, doing so half of the day from home where she can better look after the children—mostly only John, now that the youngest is growing up—after

school. Both Robert and Alison are devoted parents who work hard to provide for their children, but also try to be available to them as much as possible.

John has two siblings: an older brother Adam (12) and a younger sister Caroline (9). Both are neurotypical and both of them also attend Groenebosch Primary school. Adam is very protective of his younger brother and shows much patience and empathy towards him. Caroline, on the other hand, shows less patience with her brother and often demonstrates a lack of comprehension of her brother’s needs.

The Petersons have a yellow Labrador retriever called Denver. He has been part of the family for quite some years now and the children have grown up with him. Everyone in the family is fond of Denver, including John.

B.3 OCEAN Derivation

A character profile does not only allow us to write a dialogue for our agent, but also allows us to choose the OCEAN values of the agent’s personality. There is currently no evidence for an “autistic personality”, although certain OCEAN values seem typical of people with a high autistic quotient (AQ) [104]. We thus claim that people who have AS have just as diverse a range of different personalities as people who are neurotypical². A person with AS can have any combination of OCEAN values³. We now look at how we can derive John’s OCEAN values from the above profile⁴.

- While John’s ability to absorb and vigour for seeking out information about his special interests may be thought to be an indication of a high score for “openness”, it actually is not. A more careful examination will show that John actually greatly lacks in active imagination, has little regard for aesthetics and he has difficulty identifying his own inner feelings, let alone phrase them for

²While this is true in general, we were unable to find a formal study for OCEAN personalities in people who have AS, or even autism in general

³In this character design we tried to follow the common correlation identified by Austin [4] and Wakabayashi et al. [104]. They found that people with autism typically have low scores for Extraversion and high scores for Neuroticism. The former also identified low Agreeableness, while the latter identified low Conscientiousness as being typical of people with high AQ scores.

⁴Note that, in reality, OCEAN values are not “guessed” based on brief personality descriptions, but rather are obtained from reputable tests, such as the NEO Personality Inventory (NEO PI-R) or NEO Five-Factor Inventory (NEO-FFI) test [32].

other people. This is an indication of a medium-negative “openness” value. We select a value of -0.6 .

- Like many people who have AS, John is very diligent in his work (which he comprehends). But for him, working on something which he understands is more than a pleasant escape from a misunderstanding world. He feels pride and satisfaction in a job well done. This is an indication of a medium-positive “conscientiousness” value. We select a value of 0.5 .
- Although John likes having friends—sometimes actively seeking new friends out—and can be talkative at times, one might assume that John is fairly extraverted for someone with AS. The basic human need for social interaction is, however, virtually universal. Upon closer examination, one sees that John typically prefers to be alone for a few hours a day, although he is not totally recluse. He is also tired out by prolonged social interactions, even with people that he knows well, such as his family. This is an indication of a low to medium negative “extraversion” value. We select a value of -0.4 .
- John has difficulty to be accommodating of others. Yet he still recognises the necessity of being accommodating, but he still struggles with a natural tendency not to be so. This is an indication of a low-negative “agreeableness” value. We select a value of -0.2 .
- While John is often angry and feels isolated, this in itself is not necessarily an indication of neuroticism, but are side-effects of his AS: feeling unable to communicate his feelings in a “proper” way and having constant strain to understand and decipher a world built around people who are neurotypical are taxing on him. However, John is more susceptible to these negative feelings than the norm (regardless of the measure), but is resilient against long periods of anger and depression. This is an indication of a low-positive “neuroticism” value. We select a value of 0.3 .

Following the formula given in [45], we are now able to calculate John’s base mood traits as

$$\begin{aligned} \textit{Pleasure} &= 0.21 \times E + 0.59 \times A + 0.19 \times N \\ &= 0.09 \end{aligned}$$

$$\begin{aligned} \textit{Arousal} &= 0.15 \times O + 0.30 \times A - 0.57 \times N \\ &= -0.32 \end{aligned}$$

$$\begin{aligned} \textit{Dominance} &= 0.25 \times O + 0.17 \times C + 0.60 \times E - 0.23 \times A \\ &= -0.35 \end{aligned}$$

The score of +P–A–D indicates that John has a default mood of being *docile*, as described in [45].

Appendix C

What to say (and not to say) about autism

The following was published by the British National Autistic Society (NAS) as a guideline about how to use terminology surrounding autism. It cautions the reader as to what can be offensive and incorrect, and what is considered clinically and politically correct to say. The web page, <http://www.nas.org.uk/nas/jsp/polopoly.jsp?d=4293>, was accessed on the 18th of May 2009, but has subsequently disappeared. What follows is a verbatim reproduction of the text of that web page.

What to say (and not to say) about autism

The National Autistic Society exists to champion the interests of all people with autism and relies upon the media to report on issues relating to autism in a factual and inoffensive way.

We encourage journalists to use positive language about autism and to check that copy is in line with the following guidelines. Includes [sic] guidelines at the bottom

of the page on the correct pronunciation of ‘Asperger’.

| Don't say | Do say |
|--|--|
| an autistic/autist/autie/aspie <i>[Some individuals may refer to themselves in this way, however]</i> | person/baby/child with autism/ with an autism spectrum disorder (ASD); autism (including Asperger syndrome); he/she has autism |
| suffers from OR is a victim of autism | has autism/an autism spectrum dis- order (ASD) <i>[NB the term ASD is the most widely accepted across all audiences but many people prefer the term ‘autism spectrum condition’ because it avoids th negative connotations of ‘disabil- ity’ or ‘disorder’.]</i> |
| retarded/mentally handicapped/ backward | person with a disability |
| disease/illness/handicap | disability OR condition |
| Asperger syndrome is a mild/rare form of autism | Asperger syndrome is a form of autism <i>[Asperger is pronounced with a hard ‘g’]</i> |
| normally developing children | typically developing children |
| normal | neurotypical <i>[NB this term is only used within the autism community so may not be ap- plicable in, for example, the popular press]</i> |

| Myths | Facts |
|--|---|
| Autism (including Asperger syndrome) is a rare condition. | Autism (including Asperger syndrome) is no longer a rare condition and is thought to affect about 588,000 people in the UK today. |
| All people with autism have a [sic] extraordinary ability like the Dustin Hoffman character in the film <i>Rainman</i> . | People with autism who have an extraordinary talent are referred to as 'autistic savants'. Savants are rare: Between 2 and 3% of the UK population have some degree of learning disability, but only 0.06% of these were initially estimated to possess an unusually high level of specific ability. Savant ability is more frequently associated with those having some form of autism rather than other disabilities. Current thinking holds that at most 1 or 2 in 200 individuals with an autism spectrum disorder might have a genuine savant talent. However, there is no reliable frequency estimate as yet there is still no register of people with autism in the UK. |
| Asperger syndrome is a middle class malady made up by parents to excuse their badly behaved children. | Asperger syndrome is a very real and very disabling condition that has its own set of diagnostic criteria. It is often diagnosed slightly later than autism at around 11–13 years but its effects are just as real and can be devastating if people's needs are not met. |

| | |
|--|---|
| Only children have autism and they can get better or grow out of it. | Autism is a lifelong developmental disability with no cure. Children with autism grow up to be adults with autism. |
| Autism is the result of emotional deprivation or emotional stress. | Autism is a complex developmental disability involving a biological or organic defect in the functioning of the brain. |
| Autism is a new phenomenon. | The first detail description of a child we now know had autism was written in 1799 by Jean Itard in his account of the wild boy of Aveyron. |
| A person with autism cannot be educated. | With the right structured support within and outside of school, individuals with autism can be helped to reach their full potential. |
| People with autism wish to avoid social contact. | People with autism are often keen to make friends but, due to their disability, find this difficult. |
| Autism is due to parental rejection or cold, unemotional parents. | Autism has nothing whatsoever to do with the way parents bring up their children. |

How to pronounce ‘Asperger’

Asperger is a German name and should be pronounced with a hard ‘G’, as there is no soft ‘G’ in the German language.

Someone wrote to the NAS recently to say that that [sic] people might see negative connotations with the wrong pronunciation when a soft ‘G’ is used. She pointed out that ‘Aspergers’ with a soft ‘G’ (as in purge) is the name of a religious rite, a purging ceremony, concerned with the purging of sin. ‘Aspergere’ is Latin for ‘to wash or sprinkle’.

Related resources

Autism Helpline 0845 070 4004

The National Autistic Society, registered office: 393 City Road, London, EC1V 1NG, United Kingdom.

Tel: +44(0)20 78332299, Fax: +44 (0)20 7833 9666, Email: nas@nas.org.uk

VAT registration number: 653370050; registered as a charity in England and Wales (269425) and in Scotland (SC039427)

© The National Autistic Society 2008

Bibliography

- [1] Adams, E.: *Fundamentals of Game Design*. Voices that Matter. New Riders, 2nd edition, 2009. ISBN 9780321643377. (Cited on page 53.)
- [2] Allbeck, J. and Badler, N.: Toward Representing Agent Behaviors Modified by Personality and Emotion. In *Workshop Embodied Conversational Agents—Let’s Specify and Evaluate Them! at AAMAS 2002*. Bologna, Italy, 2002. (Cited on page 18.)
- [3] Attwood, T.: *The Complete Guide to Asperger’s Syndrome*. Jessica Kingsley Publishers, London, 2006. (Cited on pages 3, 10, 13, 15, 16, 31, 46, 51, 53, and 86.)
- [4] Austin, E. J.: Personality Correlates of the Broader Autism Phenotype as Assessed by the Autism Spectrum Quotient (AQ). *Personality and Individual Differences*, vol. 38, no. 2, pp. 451–460, 2005. ISSN 0191-8869. doi:DOI:10.1016/j.paid.2004.04.022.
URL <http://www.sciencedirect.com/science/article/B6V9F-4CK1VD1-8/2/c4cfe6316588a14ff632f636aa934045> (Cited on page 89.)
- [5] Badler, N. I.; Webber, B. L.; Becket, W.; Geib, C.; Moore, M.; Pelachaud, C.; Reich, B. and Stone, M.: Planning and Parallel Transition Networks: Animation’s New Frontiers. In Shin, S. Y. and Kunii, T. L., editors, *Computer Graphics and Applications: Proceedings of Pacific Graphics ’95*, pp. 101–117. World Scientific Publishing, River Edge, NJ, USA, 1995. (Cited on page 15.)
- [6] Balkenius, C.; Morén, J.; Johansson, B. and Johnsson, M.: Ikaros: Building Cognitive Models for Robots. *Advanced Engineering Informatics*, vol. 24, no. 1, pp. 40–48, 2009. doi:<http://dx.doi.org/10.1016/j.aei.2009.08.003>.
URL <http://www.luics.lu.se> (Cited on page 38.)

- [7] Baron-Cohen, S.: *Natural Theories of Mind: Evolution, Development and Simulation of Everyday Mindreading*, chapter Precursors to a Theory of Mind: Understanding Attention in Others, pp. 233–251. Basil Blackwell Publishing Ltd, Oxford, 1991. (Cited on page 9.)
- [8] Baron-Cohen, S.; Leslie, A. M. and Frith, U.: Does the Autistic Child have a ‘Theory of Mind’? *Cognition*, vol. 21, no. 1, pp. 37–46, 1985. (Cited on pages 9 and 31.)
- [9] Baron-Cohen, S. and Wheelwright, S.: ‘Obsessions’ in Children with Autism or Asperger Syndrome: a Content Analysis in Terms of Core Domains of Cognition. *British Journal of Psychiatry*, vol. 175, pp. 484–490, 1999. (Cited on pages 76 and 87.)
- [10] BBC News: Brit’s Bot Chats Way to AI Medal. Online, 2005. Accessed on 6 February 2011.
URL <http://news.bbc.co.uk/2/hi/technology/4263278.stm> (Cited on page 52.)
- [11] Best, J.: *Cognitive Psychology*. West Publishing Company, 5th edition, 1986. ISBN 9780314931542. (Cited on page 8.)
- [12] Bickmore, T. and Cassell, J.: Small Talk and Conversational Storytelling in Embodied Conversational Interface Agents. In *AAAI Fall Symposium on Narrative Intelligence*, pp. 87–92. 1999. (Cited on page 13.)
- [13] Bickmore, T. and Cassell, J.: “How about this Weather?”—Social Dialog with Embodied Conversational Agents. In *Proceedings of the American Association for Artificial Intelligence (AAAI) Fall Symposium on “Narrative Intelligence”*, pp. 4–8. Cape Cod, MA., USA, 2000. (Cited on pages 1, 13, 24, and 42.)
- [14] Bird, S.; Klein, E. and Loper, E.: *Natural Language Processing with Python*. O’Reilly Media, first edition, 2009. ISBN 978-0596516499. (Cited on page 45.)
- [15] Bratman, M. E.: *Intention, Plans, and Practical Reason*. Cambridge University Press, 1999. (Cited on page 29.)
- [16] Bratman, M. E.; Israel, D. J. and Pollack, M. E.: Plans and Resource-Bounded Practical Reasoning. *Computational Intelligence*, vol. 4, no. 3, pp. 349–355, 1988. doi: 10.1111/j.1467-8640.1988.tb00284.x
URL <http://dx.doi.org/10.1111/j.1467-8640.1988.tb00284.x> (Cited on page 30.)
- [17] Brazier, F.; Dunin-Keplicz, B.; Treur, J. and Verbrugge, R.: Beliefs, Intentions and DESIRE. In *Proceedings of the 10th Workshop on Knowledge Acquisition*. Online, 1996.
URL <http://ksi.cpsc.ucalgary.ca/KAW/KAW96/brazier/default.html> (Cited on page 30.)

- [18] Bringsjord, S.; Shilliday, A.; Taylor, J.; Werner, D.; Clark, M.; Charpentier, E. and Bringsjord, A.: Toward Logic-Based Cognitively Robust Synthetic Characters in Digital Environments. In *Proceeding of the 2008 conference on Artificial General Intelligence 2008*, pp. 87–98. IOS Press, Amsterdam, The Netherlands, The Netherlands, 2008. ISBN 978-1-58603-833-5. (Cited on pages 2 and 8.)
- [19] Bush, N. and Wallace, R.: Artificial Intelligence Markup Language (AIML) Version 1.0.1. *Working draft*, A.L.I.C.E. AI Foundation, 2005.
URL <http://www.alicebot.org/TR/2005/WD-aiml/> (Cited on pages 43, 44, and 73.)
- [20] Cassell, J.; Bickmore, T.; Vilhjálmsón, H. and Yan, H.: More than Just a Pretty Face: Affordances of Embodiment. In *IUI '00: Proceedings of the 5th International Conference on Intelligent User Interfaces*, pp. 52–59. ACM, New York, NY, USA, 2000. ISBN 1-58113-134-8.
URL <http://doi.acm.org/10.1145/325737.325781> (Cited on pages 2 and 24.)
- [21] Cassell, J.; Nakano, Y. I.; Bickmore, T. W.; Sidner, C. L. and Rich, C.: Non-Verbal Cues for Discourse Structure. In *Meeting of the Association for Computational Linguistics*, pp. 106–115. 2001. (Cited on pages 2 and 13.)
- [22] Cassell, J.; Pelachaud, C.; Badler, N.; Steedman, M.; Achorn, B.; Becket, T.; Douville, B.; Prevost, S. and Stone, M.: Animated Conversation: Rule-Based Generation of Facial Expression, Gesture & Spoken Intonation for Multiple Conversational Agents. In *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '94*, pp. 413–420. ACM, New York, 1994. ISBN 0-89791-667-0. (Cited on pages 13 and 15.)
- [23] Cassell, J.; Sullivan, J.; Prevost, S. and Churchill, E., editors: *Embodied Conversational Agents*. MIT Press, Cambridge, Massachusetts, 2000. (Cited on pages 4 and 7.)
- [24] Cassell, J. and Thorisson, K. R.: The Power of a Nod and a Glance: Envelope vs. Emotional Feedback in Animated Conversational Agents. *Applied Artificial Intelligence*, vol. 13, pp. 519–538, 1999.
URL <http://www.informaworld.com/10.1080/088395199117360> (Cited on pages 13 and 14.)
- [25] Cassell, J. and Torres, O.: Turn Taking vs. Discourse Structure: How Best to Model Multimodal Conversation. In Wilks, Y., editor, *Machine Conversations*, pp. 143–154. Kluwer, The Hague, 1998. (Cited on page 13.)
- [26] Chamberlain, M.: *A 3D Virtual Environment Development Platform for ASD Therapy Tools*. Master's thesis, Stellenbosch University, 2009. (Cited on pages 26, 27, and 52.)

- [27] Cheng, Y.; Moore, D.; McGrath, P. and Fan, Y.: Collaborative Virtual Environment Technology for People with Autism. In *Proceedings of the Fifth IEEE International Conference on Advanced Learning Technologies, ICALT '05*, pp. 247–248. IEEE Computer Society, Washington, DC, USA, 2005. ISBN 0-7695-2338-2. doi: <http://dx.doi.org/10.1109/ICALT.2005.85>.
URL <http://dx.doi.org/10.1109/ICALT.2005.85> (Cited on page 29.)
- [28] Cheng, Y. and Ye, J.: Exploring the Social Competence of Students with Autism Spectrum Conditions in a Collaborative Virtual Learning Environment—The Pilot Study. *Computers & Education*, vol. 54, pp. 1068–1077, 2010. ISSN 0360-1315. doi: <http://dx.doi.org/10.1016/j.compedu.2009.10.011>.
URL <http://dx.doi.org/10.1016/j.compedu.2009.10.011> (Cited on page 29.)
- [29] Clocksin, W. F.: Artificial Intelligence and the Future. *Philosophical Transactions: Mathematical, Physical and Engineering Sciences*, vol. 361, no. 1809, pp. 1721–1748, 2003. doi:10.1098/rsta.2003.1232. (Cited on page 37.)
- [30] Colby, K. M.: *Artificial Paranoia: A Computer Simulation of Paranoid Processes*. Elsevier Science Inc., New York, NY, USA, 1975. ISBN 0080181627. (Cited on page 37.)
- [31] Coren, S.; Ward, L. M. and Enns, J. T.: *Sensation and perception*. John Wiley & Sons, 6th edition, 2004. ISBN 9780471272557. (Cited on page 8.)
- [32] Costa, P. T. and McCrae, R. R.: *NEO-PI-R Professional Manual: Revised NEO Personality and NEO Five-Factor Inventory (NEO-FFI)*. Psychological Assessment Resources, Odessa, FL, 1992. (Cited on page 89.)
- [33] Dalmau, D. S.-C.: *Core Techniques and Algorithms in Game Programming*. New Riders Games. New Riders, 2004. ISBN 9780131020092. (Cited on page 16.)
- [34] Deng, L. and Huang, X.: Challenges in Adopting Speech Recognition. *Communications of the ACM*, vol. 47, pp. 69–75, 2004. ISSN 0001-0782. doi:<http://doi.acm.org/10.1145/962081.962108>. (Cited on page 42.)
- [35] Dryer, D. C.: Getting Personal with Computers: How to Design Personalities for Agents. *Applied Artificial Intelligence*, vol. 13, no. 3, pp. 273–295, 1999. (Cited on page 86.)
- [36] DSM-5 Task Force: Proposed Revision: 299.80 Asperger’s Disorder. <http://www.dsm5.org/ProposedRevisions/Pages/proposedrevision.aspx?rid=97>, 2009. (Cited on page 3.)
- [37] Egri, L.: *The Art Of Dramatic Writing: Its Basis in the Creative Interpretation of Human Motives*. Simon and Schuster, 1960. ISBN 9780671213329. (Cited on page 86.)

- [38] Ehrlich, J. A. and Miller, J. R.: A Virtual Environment for Teaching Social Skills: AViSSS. In Potel, M., editor, *IEEE Computer Society*, pp. 10–16. IEEE, 2009. (Cited on page 27.)
- [39] Ekman, P.: *Emotion in the Human Face*. Cambridge University Press, New York, 1982. (Cited on page 12.)
- [40] El Kaliouby, R.; Picard, R. W. and Baron-Cohen, S.: Affective Computing and Autism. *Annals of the New York Academy of Sciences*, vol. 1093, pp. 228–248, 2006. doi:10.1196/annals.1382.016. (Cited on page 33.)
- [41] Epstein, R.: The Quest for the Thinking Computer. *AI Magazine*, vol. 13, no. 2, pp. 81–95, 1992. ISSN 0738-4602. (Cited on page 43.)
- [42] Fabri, M.; Elzouki, S. and Moore, D.: Emotionally Expressive Avatars for Chatting, Learning and Therapeutic Intervention. In Jacko, J., editor, *Human-Computer Interaction. HCI Intelligent Multimodal Interaction Environments*, vol. 4552 of *Lecture Notes in Computer Science*, pp. 275–285. Springer Berlin / Heidelberg, 2007. URL http://dx.doi.org/10.1007/978-3-540-73110-8_29 (Cited on page 28.)
- [43] Galitsky, B.: Extending the BDI Model to Accelerate the Mental Development of Autistic Patients. In *Second International Conference on Development and Learning (ICDL'02)*, pp. 82–88. IEEE Computer Society, Los Alamitos, CA, USA, 2002. ISBN 0-7695-1459-6. doi:10.1109/DEVLRN.2002.1011803. (Cited on pages 29 and 31.)
- [44] Gamma, E.; Helm, R.; Johnson, R. and Vlissides, J.: *Design Patterns—Elements of Reusable Object-Oriented Software*. Addison-Wesley, 1995. (Cited on pages 41 and 46.)
- [45] Gebhard, P.: ALMA: A Layered Model of Affect. In *AAMAS '05: Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems*, pp. 29–36. ACM, New York, NY, USA, 2005. ISBN 1-59593-093-0. URL <http://doi.acm.org/10.1145/1082473.1082478> (Cited on pages 1, 2, 11, 12, 59, 60, 90, and 91.)
- [46] Gebhard, P. and Kipp, K. H.: Are Computer-Generated Emotions and Moods Plausible to Humans? In *Proceedings of the 6th International Conference on Intelligent Virtual Agents (IVA'06)*, pp. 343–356. Marina Del Rey, USA, 2006. (Cited on page 11.)
- [47] Gebhard, P.; Klesen, M. and Rist, T.: *Coloring Multi-Character Conversations through the Expression of Emotions*, vol. 3068, pp. 128–141. Springer, Berlin/Heidelberg, 2004. ISBN 978-3-540-22143-2. (Cited on page 11.)
- [48] Gebhard, P.; Klesen, M. and Rists, T.: Adding the Emotional Dimension to Scripting Character Dialogues. In *Proceedings of the 4th International Working Conference on Intelligent Virtual Agents*, pp. 48–56. 2003. (Cited on page 11.)

- [49] Georgeff, M.; Pell, B.; Pollack, M.; Tambe, M. and Wooldridge, M.: The Belief-Desire-Intention Model of Agency. In Müller, J.; Rao, A. and Singh, M., editors, *Intelligent Agents V: Agent Theories, Architectures, and Languages*, vol. 1555 of *Lecture Notes in Computer Science*, pp. 1–10. Springer, Berlin/Heidelberg, 1998. 5th International Workshop, ATAL'98, Paris, France.
URL <http://www.springerlink.com/content/v0xlyxfwky11j24c/> (Cited on page 30.)
- [50] Gillberg, C.: *A Guide to Asperger Syndrome*. Cambridge University Press, Cambridge, UK, 2002. ISBN 0-521-00183-8. (Cited on pages 3, 4, 5, 10, 12, 15, 16, 35, 46, 53, 76, and 86.)
- [51] Groom, V.; Nass, C.; Chen, T.; Nielsen, A.; Scarborough, J. K. and Robles, E.: Evaluating the Effects of Behavioral Realism in Embodied Agents. *International Journal of Human-Computer Studies*, vol. 67, pp. 842–849, 2009. doi:10.1016/j.ijhcs.2009.07.001. (Cited on page 17.)
- [52] Güzeldere, G. and Franchi, S.: Dialogues with Colorful Personalities of Early AI. *Stanford Humanities Review*, vol. 4, pp. 161–169, 1995. ISSN 1048-3721.
URL <http://www.stanford.edu/group/SHR/4-2/text/dialogues.html> (Cited on page 52.)
- [53] Haddon, M.: *The Curious Incident of the Dog in the Night-Time*. Vintage, London, UK, 2003. ISBN 978-1400032716. (Cited on page 86.)
- [54] Hall, K.: *Asperger Syndrome, the Universe and Everything*. Jessica Kingsley Publishers, 116 Pentonville Road, London N1 9JB, UK, 2001. (Cited on page 86.)
- [55] Harding-Rolls, P.: Subscription MMOGs: Life Beyond World of Warcraft. Online, 2009. Accessed on 11 December 2011.
URL <http://www.webcitation.org/5mHRYdUkd> (Cited on page 1.)
- [56] Ho, W. C.; Davis, M. and Dautenhahn, K.: Supporting Narrative Understanding of Children with Autism: A Story Interface with Autonomous Autobiographic Agents. In *IEEE International Conference on Rehabilitation Robotics 2009, ICORR2009*, pp. 905–911. 2009. ISSN 1945-7898. doi:10.1109/ICORR.2009.5209580. (Cited on page 27.)
- [57] Hogan, K. and Stubbs, R.: *Can't Get Through: Eight Barriers to Communication*. Pelican Publishing, 2003. ISBN 978-1589800755. (Cited on page 14.)
- [58] Huang, Z.; Eliëns, A. and Visser, C.: A Platform for Embodied Conversational Agents based on Distributed Logic Programming. In *Proceeding of the Workshop AAMAS*. 2002. (Cited on page 16.)

- [59] Huang, Z.; Eliëns, A. and Visser, C.: STEP: A Scripting Language for Embodied Agents. In *Proceedings of the Workshop on Lifelike Animated Agents*. 2002. (Cited on page 16.)
- [60] Huang, Z.; Eliëns, A. and Visser, C.: Implementation of a Scripting Language for VRML/X3D-based Embodied Agents. In *Web3D '03: Proceedings of the eighth international conference on 3D Web technology*, pp. 91–100. ACM, New York, NY, USA, 2003. ISBN 1-58113-644-7.
URL <http://doi.acm.org/10.1145/636593.636608> (Cited on page 16.)
- [61] Huang, Z.; Eliëns, A. and Visser, C.: XSTEP: A Markup Language for Embodied Agents. In *Proceedings of the 16th International Conference on Computer Animation and Social Agents (CASA 2003)*, pp. 105–110. IEEE Computer Society, Washington, DC, USA, 2003. ISBN 0-7695-1934-2. (Cited on page 16.)
- [62] Huber, M. J.: JAM: A BDI-theoretic Mobile Agent Architecture. In *Proceedings of the Third International Conference on Autonomous Agents (Agents'99)*, pp. 236–243. Seattle, WA, USA, 1999. (Cited on pages 22 and 30.)
- [63] Humphrys, M.: How my program passed the Turing test. Online, 2005.
URL <http://computing.dcu.ie/~humphrys/eliza.html> (Cited on pages 3, 8, and 38.)
- [64] Jackson, L.: *Freaks, Geeks & Asperger Syndrome*. Jessica Kingsley Publishers, 116 Pentonville Road, London N1 9JB, UK, 2002. (Cited on page 86.)
- [65] Johanson, W. L.; Rickel, J. W. and Lester, J. C.: Animated Pedagogical Agents: Face-to-Face Interaction in Interactive Learning Environments. *International Journal of Artificial Intelligence in Education*, vol. 11, pp. 47–78, 2000. (Cited on page 19.)
- [66] John, O. P. and Srivastava, S.: The Big-Five Trait Taxonomy: History, Measurement, and Theoretical Perspectives. In Pervin, L. A. and John, O. P., editors, *Handbook of Personality: Theory and Research*, pp. 102–138. Guilford Press, New York, second edition, 1999.
URL <http://www.uoregon.edu/~sanjay/pubs/bigfive.pdf> (Cited on page 10.)
- [67] Jurafsky, D. and Martin, J. H.: *Speech and Language Processing*. Prentice Hall, Upper Saddle River, New Jersey, USA, first edition, 2000. ISBN 0-13-095069-6. (Cited on pages 30 and 53.)
- [68] Kopp, S.; Gesellensetter, L.; Krämer, N. C. and Wachsmuth, I.: A Conversational Agent as Museum Guide—Design and Evaluation of a Real-World Application. In Panayiotopoulos, T.; Gratch, J.; Aylett, R.; Ballin, D.; Olivier, P. and Rist, T., editors, *Intelligent Virtual Agents*, vol. 3661 of *Lecture Notes in Computer Science*,

- pp. 329–343. Springer Berlin / Heidelberg, 2005.
URL http://dx.doi.org/10.1007/11550617_28 (Cited on pages 21, 29, and 38.)
- [69] Kshirsagar, S.: A Multilayer Personality Model. In *SMARTGRAPH '02: Proceedings of the 2nd International Symposium on Smart Graphics*, pp. 107–115. ACM, New York, NY, USA, 2002. ISBN 1-58113-555-6.
URL <http://doi.acm.org/10.1145/569005.569021> (Cited on page 12.)
- [70] Laird, J. E.; Newell, A. and Rosenbloom, P. S.: SOAR: An Architecture for General Intelligence. *Artificial Intelligence*, vol. 33, pp. 1–64, 1987. ISSN 0004-3702. doi:10.1016/0004-3702(87)90050-6.
URL <http://portal.acm.org/citation.cfm?id=27701.27702> (Cited on page 30.)
- [71] Lee, C.-H. J.; Kim, K.; Breazeal, C. and Picard, R.: Shybot: Friend-Stranger Interaction for Children Living with Autism. In *CHI '08 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '08, pp. 3375–3380. ACM, New York, NY, USA, 2008. (Cited on pages 32 and 75.)
- [72] Lester, J. C.; Towns, S. G. and Laboratory, P. J. F. M.: Achieving Affective Impact: Visual Emotive Communication in Lifelike Pedagogical Agents. *International Journal of Artificial Intelligence in Education*, vol. 10, pp. 278–291, 1999. doi:10.1.1.84.6318. (Cited on pages 1 and 18.)
- [73] Loebner Prize, The: Home Page of The Loebner Prize in Artificial Intelligence. On-line. Accessed 5 February 2011.
URL <http://loebner.net/Prizef/loebner-prize.html> (Cited on page 43.)
- [74] Loebner Prize, The: Loebner Prize 2005 Information. On-line, 2005. Accessed 14 January 2012.
URL http://www.loebner.net/Prizef/2005_Contest/loebner-prize-2005.html (Cited on page 52.)
- [75] Loebner Prize, The: Loebner Prize 2006 Information. On-line, 2006. Accessed 6 February 2011.
URL http://www.loebner.net/Prizef/2006_Contest/loebner-prize-2006.html (Cited on page 52.)
- [76] Mackenzie, I.: Warning over Sexy Instant Messaging called ‘Fembot’. Online, 2010. Accessed on 12 February 2010.
URL <http://www.bbc.co.uk/newsbeat/10057391> (Cited on page 3.)
- [77] Mauldin, M. L.: ChatterBots, TinyMuds, and the Turing Test: Entering the Loebner Prize Competition. In *AAAI '94: Proceedings of the twelfth national conference on Artificial intelligence (vol. 1)*, pp. 16–21. American Association for Artificial Intelligence, Menlo Park, CA, USA, 1994. ISBN 0-262-61102-3. (Cited on pages 13 and 37.)

- [78] McBreen, H. and Jack, M.: Evaluating Humanoid Synthetic Agents in e-Retail Applications. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, vol. 31, no. 5, pp. 394–405, 2001. doi:10.1109/3468.952714. (Cited on page 17.)
- [79] Mehrabian, A.: Pleasure-Arousal-Dominance: A General Framework for Describing and Measuring Individual Differences in Temperament. *Current Psychology*, vol. 14, no. 4, pp. 261–292, 1996.
URL <http://www.springerlink.com/content/g071r4n59240u537/> (Cited on page 11.)
- [80] Milde, J.-T.: Lokutor: Towards a Believable Communicative Agent. In Rickel, J.; Johnson, W. L. and Lester, J., editors, *Proceedings of the workshop on “Achieving Human-Like Behavior in Interactive Animated Agents”*. ACM, Fourth International Conference on Autonomous Agents, Barcelona, 2000. (Cited on page 20.)
- [81] Milde, J.-T.: The Instructable Agent Lokutor. In Zühlke, W. and Schukat-Talamazzini, E. G., editors, *KONVENS 2000 / Sprachkommunikation, Vorträge der gemeinsamen Veranstaltung 5. Konferenz zur Verarbeitung natürlicher Sprache (KONVENS), 6. ITG-Fachtagung “Sprachkommunikation”*, pp. 169–174. VDE-Verlag GmbH, Berlin, 2000. ISBN 3-8007-2564-9. (Cited on page 1.)
- [82] Mitra, S. and Acharya, T.: Gesture Recognition: A Survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 37, no. 3, pp. 311–324, 2007. ISSN 1094-6977. (Cited on page 42.)
- [83] Molyneux, P.: Peter Molyneux Demos Milo, the Virtual Boy. TED Talk, 2010. Recorded at TEDGlobal 2010 in Oxford, UK.
URL http://www.ted.com/talks/peter_molyneux_demos_milo_the_virtual_boy.html (Cited on pages 1, 2, and 23.)
- [84] Monk, J. D.: *Mathematical Logic*, vol. 37 of *Graduate Texts in Mathematics*, chapter 29, pp. 473–487. Springer, 1st edition, 1976. ISBN 9780387901701. (Cited on page 9.)
- [85] Moore, C.-L. and Yamamoto, K.: *Beyond Words: Movement Observation and Analysis (Instructor’s Guidebook)*. Routledge, 3rd edition, 1988. ISBN 9782881242526. (Cited on page 18.)
- [86] Mori, M.: Bukimi No Tani [The Uncanny Valley]. *Energy*, vol. 7, no. 4, pp. 33–35, 1970. (Cited on page 17.)
- [87] Myers, I. B. and Myers, P. B.: *Gifts Differing: Understanding Personality Type*. Davies-Black Publishing, Mountain View, CA, USA, 1995. ISBN 0-89106-074-X. (Cited on page 10.)

- [88] Nass, C.; Foehr, U.; Brave, S. and Somoza, M.: The Effects of Emotion of Voice in Synthesized and Recorded Speech. In *Proceedings of the AAAI Symposium Emotional and Intelligent II: The Tangled Know of Social Cognition*, pp. 91–96. AAAI Press, North Falmouth, MA, 2001. (Cited on page 17.)
- [89] Norrick, N. R.: Conversational Storytelling. In Herman, D., editor, *The Cambridge Companion to Narrative*. Cambridge University Press, 2007. (Cited on page 13.)
- [90] Ortony, A.; Clore, G. L. and Collins, A.: *The Cognitive Structure of Emotions*. Cambridge University Press, The Edinburgh Building, Cambridge CB2 2RU, UK, 1988. (Cited on pages xii, 2, 10, 59, and 60.)
- [91] Parsons, S.; Leonard, A. and Mitchell, P.: Virtual Environments for Social Skills Training: Comments from Two Adolescents with Autistic Spectrum Disorder. *Computers & Education*, vol. 47, no. 2, pp. 186–206, 2006. ISSN 0360-1315. doi: DOI:10.1016/j.compedu.2004.10.003.
URL <http://www.sciencedirect.com/science/article/B6VCJ-4DVVY4G-1/2/1627ad7044c1166013a523a00e1be924> (Cited on page 28.)
- [92] Picard, R. W.: Affective Computing. *Technical report*, M.I.T Media Laboratory, Cambridge, MA, USA, 1997. (Cited on page 32.)
- [93] Picard, R. W.: Affective Computing for HCI. In *Proceedings of HCI International (the 8th International Conference on Human-Computer Interaction) on Human-Computer Interaction: Ergonomics and User Interfaces-Volume I*, pp. 829–833. L. Erlbaum Associates Inc., Hillsdale, NJ, USA, 1999. ISBN 0-8058-3391-9. (Cited on page 32.)
- [94] Picard, R. W.: Future Affective Technology for Autism and Emotion Communication. *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 364, no. 1535, pp. 3575–3584, 2009. doi:10.1098/rstb.2009.0143.
URL <http://rstb.royalsocietypublishing.org/content/364/1535/3575.abstract> (Cited on page 33.)
- [95] Rao, A. S. and Georgeff, M. P.: BDI Agents: From Theory to Practice. In *Proceedings of the First International Conference on Multiagent Systems (ICMAS'95)*, pp. 312–319. 1995. (Cited on page 30.)
- [96] Richards, D. T.: Clippy: Misunderstood Animated Pedagogical Agent or Spawn of Satan? Online, 2009. Theory and Research in HCI (Blog).
URL <http://comm6480rpi.blogspot.com/2009/09/clippy-misunderstood-animated.html> (Cited on page 33.)
- [97] Silberschatz, A.; Gagne, G. and Galvin, P. B.: *Operating System Concepts*. John Wiley & Sons, 7th edition, 2005. ISBN 0-471-69466-5. (Cited on page 49.)

- [98] Vallance, C.: AI Aims to Solve In-game Chatter. Online, 2009. Accessed on 4 January 2010.
URL <http://news.bbc.co.uk/2/hi/8426523.stm> (Cited on page 1.)
- [99] Van Zijl, L. and Chamberlain, M.: A Generic Development Platform for ASD Therapy Tools. In Cordeiro, J. A. M.; Shishkov, B.; Verbraeck, A. and Helfert, M., editors, *Proceedings of the Second International Conference on Computer Supported Education, Valencia, Spain, 7–10 April 2010*, vol. 1, pp. 82–89. INSTICC Press, 2010. ISBN 978-989-674-023-8. (Cited on page 26.)
- [100] Van Zijl, L. and Venter, W. J.: An Embodied Conversational Agent with Asperger Syndrome. In Verbraeck, A.; Helfert, M.; Cordeiro, J. and Shishkov, B., editors, *Proceedings of the Third International Conference on Computer Supported Education, Noordwijkerhout, The Netherlands, 6–8 May 2011*, vol. 1, pp. 153–158. Science and Technology Publications, 2011. ISBN 978-989-8425-49-2. (Cited on page 3.)
- [101] Venter, W. J.: 'n Gestaltegespreksagent met Outistiese Gedrag. *Die Suid-Afrikaanse Tydskrif vir Natuurwetenskap en Tegnologie*, vol. 29, no. 3, pp. 170–171, 2010. (Cited on page 3.)
- [102] Vermeulen, P.: *Autistic Thinking—This is the Title*. Jessica Kingsley Publishers, London, 2001. (Cited on pages 3 and 86.)
- [103] Vosinakis, S. and Panayiotopoulos, T.: SimHuman: A Platform for Real-Time Virtual Agents with Planning Capabilities. In De Antonio, A.; Aylett, R. and Ballin, D., editors, *Intelligent Virtual Agents*, vol. 2190 of *Lecture Notes in Computer Science*, pp. 210–223. Springer, Berlin/Heidelberg, 2001. ISBN 978-3-540-42570-0. (Cited on pages 25 and 26.)
- [104] Wakabayashi, A.; Baron-Cohen, S. and Wheelwright, S.: Are Autistic Traits an Independent Personality Dimension? A Study of the Autism-Spectrum Quotient (AQ) and the NEO-PI-R. *Personality and Individual Differences*, vol. 41, no. 5, pp. 873–883, 2006. ISSN 0191-8869. doi:DOI:10.1016/j.paid.2006.04.003.
URL <http://www.sciencedirect.com/science/article/B6V9F-4K5SSV8-1/2/756955f376a64f41639dd379ee8591a5> (Cited on page 89.)
- [105] Weizenbaum, J.: ELIZA—A Computer Program for the Study of Natural Language Communication between Man and Machine. *Communications of the ACM*, vol. 9, no. 1, pp. 36–45, 1966. ISSN 0001-0782. doi:<http://doi.acm.org/10.1145/365153.365168>. (Cited on page 37.)
- [106] Wilcox, B.: Beyond Façade: Pattern Matching for Natural Language Applications. Online, 2011. Accessed on 11 December 2011.
URL <http://gamasutra.com/view/feature/6305/> (Cited on page 53.)

- [107] Wobcke, W. R.: Reasoning about BDI Agents from a Programming Languages Perspective. In *Proceedings of the AAAI 2007 Spring Symposium on Intentions in Intelligent Systems*. 2007. (Cited on page 30.)
- [108] Wooldridge, S.: Bayesian Belief Networks. *Technical report*, CSIRO Centre for Complex System Science, Canberra, 2003. Australian Institute of Marine Science. (Cited on page 12.)