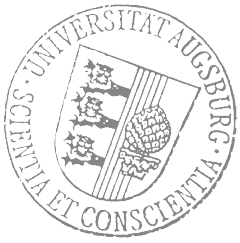
This is actually the first page of the thesis and will be discarded after the print out. This is done because the title page has to be an even page. The memoir style package used by this template makes different indentations for odd and even pages which is usally done for better readability.

University of Augsburg
Faculty of Applied Computer Science
Department of Computer Science
Bachelor's Program in Computer Science



Bachelor's Thesis

Engagement Detection

Inferring conversational engagement from verbal and nonverbal behaviour

submitted by
Amr Abdelraouf
on 31.7.2014

Supervisor:
Prof. Dr. Elisabeth André aus Augsburg

Adviser:
MSc. Tobias Baur

Reviewers:
Prof. Dr. Elisabeth André

Abstract

Interview skills are of utmost important for a person's career and personal image. Furthermore it is an essential matter to exhude conversational engagement in an interview to give the impression of confidence and attentiveness. This thesis aims to track the engagment level of an interviewee in a mock interview situation. It tracks the verbal and nonverbal behaviour of the interviewee with respect to the ongoing context of the interview. The gathered engagement data can be further used to assess the interviewee's performance.

Statement and Declaration of Consent

Statement

Hereby I confirm that this thesis is my own work and that I have documented all sources used.

Amr Abdelraouf

Augsburg, 3.7.2014

Declaration of Consent

Herewith I agree that my thesis will be made available through the library of the Computer Science Department.

Amr Abdelraouf

Augsburg, 3.7.2014

Contents

Chapter 1

Introduction

1.1 Motivation

This thesis was proposed to help measure the engagement of an interviewee in a job interview situation. Through a simple mock interview the performance of the interviewee will be assessed. One of the most important attributes of that performance is whether or not the interviwee is engaged with and attentive to the interviewer. A simple playback of the interview coupled with the measurment of the engagement level will easily highlight the ups and downs of his/her demonstration in the mock interview.

1.2 Objectives

This thesis aims to measure the engagement levels of an interviewee through verbal and non verbal behaviour of said interview. It studies the conversational interaction with the interviewer, the responses to certain commands and behaviour during certain segments of the interview.

1.3 Outline

This thesis will first discuss the theoretical background and the information gathered on this subject. It will go into the details of the work previously done. Next it will describe the setup of it's mock interview from both the interviewer and the intervieweee's perspectives.

In the following section details of constitution and software workings of this thesis will be covered. First it will describe the general structure of the pipeline. Then it will describe how the inputs are processed from a bottom up approach; starting with raw sensor data and working up level by level to demonstrate how the engagement is calculated.

Chapter 2

Theoretical Background

2.1 Previous Work

2.1.1 Recognizing Engagement in Human Robot Interaction

In [1] Rich et al. studied the engagement behaviour in a conversation between a human and a humanoid robot. Their target was to build a robot architecture that would allow the robot to mimic the conversational behaviour of a human. The attributes of the measured engagement were divided into 4 main parts:

Mutual Facial Gaze Ability of both parties to maintain eye contact.

Directed Gaze Ability of one party to follow the gaze of the other when the other party points or gazes at a certain object in the environment.

Adjacency Pair The ability of one party to reply sensibly to the other party's speech.

Backchanneling During the speech of one party, the other communicates a small gesture to indicate attentiveness.

You will find that this thesis is heavily influenced by Rich et al.'s work, with a slight modification to the definition of the four main modules.

2.1.2 Engagement Rules for Human-Robot Collaborative Interactions

Like the previously mentioned paper, Sidner et al. [2] worked on a robot that is to mimic human social behaviour.

The paper builds the human-robot interaction on three main pillars. Namely initiating, maintaining and terminating engagement behaviour during a conversation, understanding the speech utterence and replying accordingly, and finally the ability to take a decision and point the conversation into a certain direction.

The paper uses a set of rules to allow the robot to understand both how to engage and how to recognize engagement. The robot uses facial gaze and greeting utterences to indicate the start of a conversation. It maintains its gaze with the human conversation participant as long as the participant is speaking. To detect that the human participant wishes to disengage from the conversation the robot monitors the interlocuter's gaze; if it is set away from the robot then the desire to disengage is detected. And lastly when the robot runs out of things to say and wishes to disengage itself, it closes the conversation using a set of rules for conversation closing.

Chapter 3

Setup

3.1 Subject

The subject of our experiment is the interviewee in our mock interview. As shown in figure 3.1 the subject is seated aproximately 70 cm from a screen. A number of sensors are then set up to capture the needed inputs. Namely a Microsoft Kinect, an SMI Eyetracker and a microphone.

3.2 Agent

There are two main softwares used to simulate the virtual interview environment. First there is Charamel. Charamel is responsible for creating the interviewers (or agents) and their surrounding environment. The scene used for this thesis consists of two virtual characters, namely Curtis and Gloria. They stand behind a desk to mimic an office interview. On the left lies a white board that is used as an object in our environment. The setting is demonstrated in figure 3.2.

The second software used is Scenemaker. Scenemaker is responsible for sending the agents actions to perform. The program consists of a state machine, each state containing a command to be executed by the agents accordingly. These commands include ordering the agents to utter a certain sentence, stop and wait for the subject to reply, perform a certain hand gesture, and so on and so forth. Scenmaker also contains a script that defines dialogues to be acted out by the agents. The agents follow the dialogue when it is referenced in one of the state machines.

Figure 3.1: Interview setup

Figure 3.2: Virtual environment

Chapter 4

Events

Events are the backbone of the software workings of this thesis. Raw sensor data are converted to events that can be further processed in the software's pipeline. Furthermore external software send events to our own software over a network. These events can be displayed by themselves as output or can be used as inputs to trigger other events.

4.1 Event Structure

Events are constructs of several attributes:

Time The clock signature of when the event was triggered.

Duration The time duration of the event.

Ptr (Pointer) Meta data about the event.

Type Indicates the nature of the meta data wrapped by the event.

State A boolean flag to indicate whether the event is starting or ending.

In the software's pipeline events are measuerd every time cycle. A cycle of 500 ms is used.

4.2 Sensors

4.2.1 Microsoft Kinect

Kinect is a device made by Microsoft to sense 3D movements. It uses a number of cameras to detect a 3D environment; A normal VGA camera and an infrared camera that is used to measure and create a depth map.

Kinect can also track the skeletal movements of a person standing in its range. It measures the joint orientation and sends the data to the api. Kinect detects the skeletal movements of a user in two main stages. First it uses the infrared camera to create a depth map. Secondly the machine learning is used to decide whether or not the image in the depth map represents a skeleton.

In our software we are mostly interested in the movement of the subject's head. Kinect is used to detect the prepetual displacement of the subject's head which indicates that s/he is nodding. This triggers an event called *HeadNod*. HeadNod is an event measured and outputed every 500 ms and its pointer contains a value from 0 to 1 which represents the probability that the subject is nodding.

4.2.2 SMI Eyetracker

The SMI Eyetracker is a plug-and-play hardware that is used to pinpoint where the user is looking. It is placed below th screen approximately 65-70 cm away from the user. After calibrating the user position with an eye tracking test a stream of x and y coordinates is sent via the SMI API to our software's pipeline. Since the machine used in this thesis runs an 64 bit operating system (which is incompatible with the SMI software), we connect the eye tracker to a separate machine. We then receive the stream via the LAN network.

Since we are dealing with a virtual agent on a screen we consider the top left corner of the screen as the (0,0) coordinate. Displacement of the subject's gaze point to the right alters the x coordinate and to the bottom affects the y coordinate.

The software defines two main rectangular areas on the screen. First is the area of the Agent's face. Second is the area of the board that is present in the environment.

When the subject's gaze point falls on the area defined for the agent's face it triggers an event called *SubjectFacialGaze*. SubjectFacialGaze's pointer contains a value of either 0 or 1 indicating whether or not the subject is looking at the agent's face. When the gaze enters the facial

area SubjectFacialGaze is triggered with the value 1 indicating that it has started and when the gaze leaves the facial area it is triggered with the value 0 indicating that it is complete.

If the subject's gaze falls in the area of the board the event *SubjectObjectGaze* is triggered. Similar to SubjectFacialGaze, the event carries a value of either 0 or 1 indicating whether or not the subject is looking at the board. The event is prompted with pointer value 1 when the subject starts looking at the object, and triggered again with 0 when the subject directs his/her gaze away.

4.2.3 Microphone

A microphone is used to record the verbal utterances produced by the subject. When the microphone detects a voice the event *vad* (which is short for Voice Activity Duration) is fired. When the voice is first detected the event's pointer carries a value of 1. When the voice activity ends the same event is triggered but with value 0 to indicate that the event is complete.

In our expariment we used a microphone mounted on a head gear to eliminate noise produced by the agents' speech utterences.

4.3 Scenemaker

As mentioned before in subsection 3.2 Scenemaker is the software used to send commands to the virtual agents. Scenemaker is also responsible for sending the events that are triggered to represent the agents' behaviour to our software's pipeline. The events can be subcategorized into two main parts: Gaze and speech.

4.3.1 Gaze

Firstly we are concerned with where the agent is looking. When the script commands the agent to looks at the subject in front of the screen the event *AgentFacialGaze* is triggered. Similar to the subject's gaze events the event pointer holds the value 1 when the agent starts looking at the subject and holds the value 0 when the agent looks away from the subject's face.

Furthermore when the agent is commanded to look at the board, the event *AgentObjectGaze* is triggered with a pointer value 1 or 0 indicating that the agent has started or stopped looking at the board.

4.3.2 Speech

As mentioned in 3.2 it was mentioned that Scenemaker contains a script. A script contains a number of dialogues and each dialogue is split into a number of sentences. When the agent starts reading a sentence the event *AgentSpeech* is triggered with a pointer value of 1. When the agent finishes reading that sentence AgentSpeech is triggered with a pointer value of 0.

Chapter 5

Main Modules

So to review our events, we have:

- HeadNod
- SubjectFacialGaze
- SubjectObjectGaze
- vad
- AgentFacialGaze
- AgentObjectGaze
- AgentSpeech

Those events will be used as inputs for our four main modules.

5.1 Mutual Facial Gaze

Mutual Facial Gaze is defined as the eye contact between the subject and the agent. It is necessary for the subject to direct his/her gaze at the agent's face when being addressed. The event *MutualFacialGaze* is triggered with a pointer value 1 (indicating that it started) when both SubjectFacialGaze and AgentFacialGaze are ongoing. When either of the two input events are triggered with the value 0 (event ends) the event MutualFacialGaze is also ends and therefore is triggered with pointer value 0.

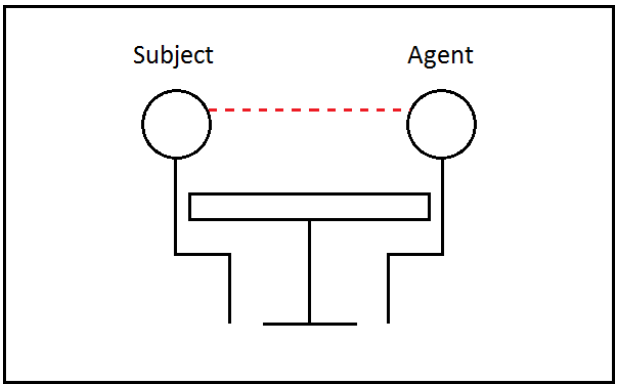


Figure 5.1: Mutual Facial Gaze

5.2 Directed Gaze

Directed Gaze occurs when the agent points or looks at a certain object and then the subject follows as shown in figure 5.2. In our environment the white board acts as the object. The event *DirectedGaze* is triggered with pointer value 1 when both SubjectObjectGaze and AgentObjectGaze are ongoing. And triggered again with pointer value 0 when one of the two input events ends.

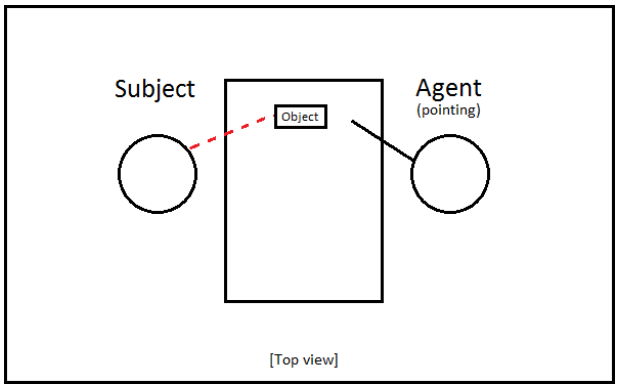


Figure 5.2: Agent points to object while subject is looking at it

5.3 Backchanneling

Backchanneling is the small responses given by the subject during the time where the agent is speaking. These responses indicate that the subject is

following what the agent is saying.

Here we introduce the concept of a *Backchanneling Pulse*. Since backchanneling has a very short duration that usually lasts only one event cycle which is not enough time to influence the bayesian network. So instead when a backchanneling event is provoked it is outputed on 5 consecutive event cycles with pointer value 1, 0.75, 0.5, 0.25 and 0 respectively, as shown in figure 5.3.

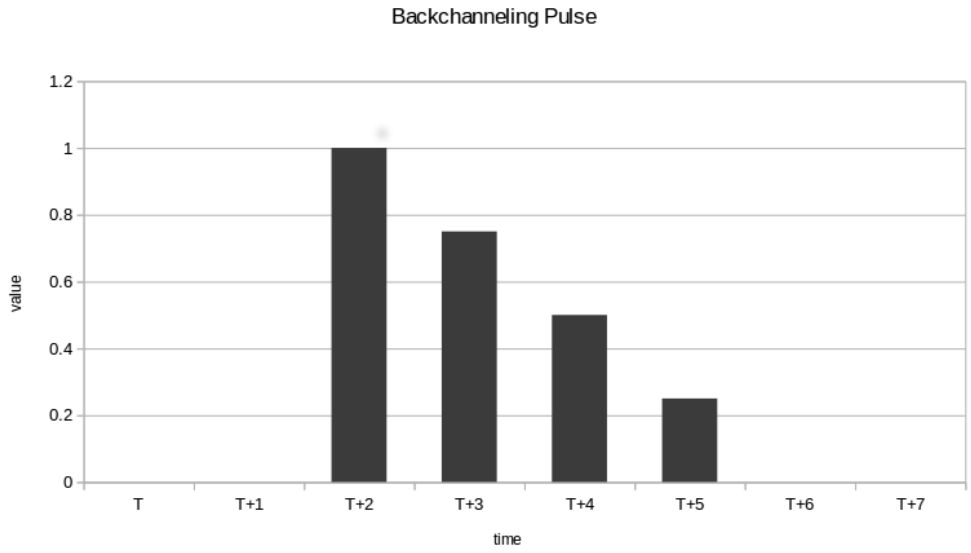


Figure 5.3: Backchanneling event triggered at T+2

A BCPulse can be triggered by two different ways. It can be prompted when a HeadNod is detected during AgentSpeech. Or it can be set off when a vad event with a duration less than 2 seconds is detected during an AgentSpeech.

5.4 Adjacency Pair

Adjacency Pairs are defined as a speech utterance which is provoked by a previous speech utterance. For instance the answer to a question is an adjacency pair. This thesis did not go into the the semantics of natural language processing. The inputs where when the agent started and stopped speaking, and when the subject started and stopped speaking. Naturally we redefined the meaning of adjacency pairs to match our inputs.

This thesis defines adjacency pairs as one of two conversational situations:

- If the subject starts speaking within a two second time window after the agent has finished its sentence it is considered an adjacency pair and a *AdjacencyPair* start event is initiated. This is demonstrated in the second line in figure 5.4.
- An AdjacencyPair start event is triggered the subject starts speaking right before the agent finished speaking as dipicted in the second line of figure 5.4. This is to mostly account for the delay in the pipeline between the triggering of the AgentSpeech end event and the vad start event.

AdjacencyPair end events are triggered when the vad event associated with the AdjacencyPair event ends.

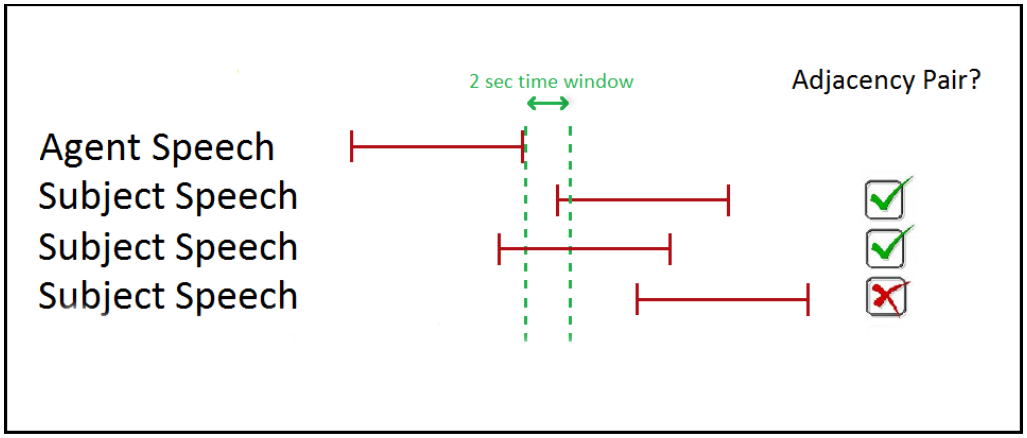


Figure 5.4: Adjacency Pair

Chapter 6

Bayesian Network

The final stage in our pipeline is to input our four main modules as factors to calculate the engagement. This is done using a bayesian network. The bayesian network was out of this thesis work's scope. Yet it is essential for this thesis to mention and explain how the bayesian network operates as it is the bridge between the work done in this thesis and the calculation of the final engagement value.

A Bayesian network is a statistical model that is represented using a directed graph. Each node in the graph represents a variable. Each directed edge from node A to node B represents a relation between the two variables. It denotes that the value of A directly affects the value of B. Moreover when an a change occures to node B (a change independent of node A) the bayesian networks adjusts the value of node A according to the relationship defined between A and B. So a directed edge indicates both a direct and indirect relation between two nodes.

Each node has an associated probability table. Each entry in the table contains the probability of the value of its associated node given the values of its input nodes. If node B has no other directed edges pointing to it then it would have 2 entries in its probability table indicating the value of B when A is present and when A is unpresent.

Figure 6.1 shows the start state of the bayesian network used in our software. The starting values of the probability are set intuitively but non of them are used since all of the leaves in this bayesian network's tree are connected to external softwares which provide the real time values. This thesis is mainly concerned with the relationship between the nodes of the four main states (nodes MutualFacialGaze, DirectedGaze, Backchanneling,

and AdjacencyPair) and the Enagegement node.

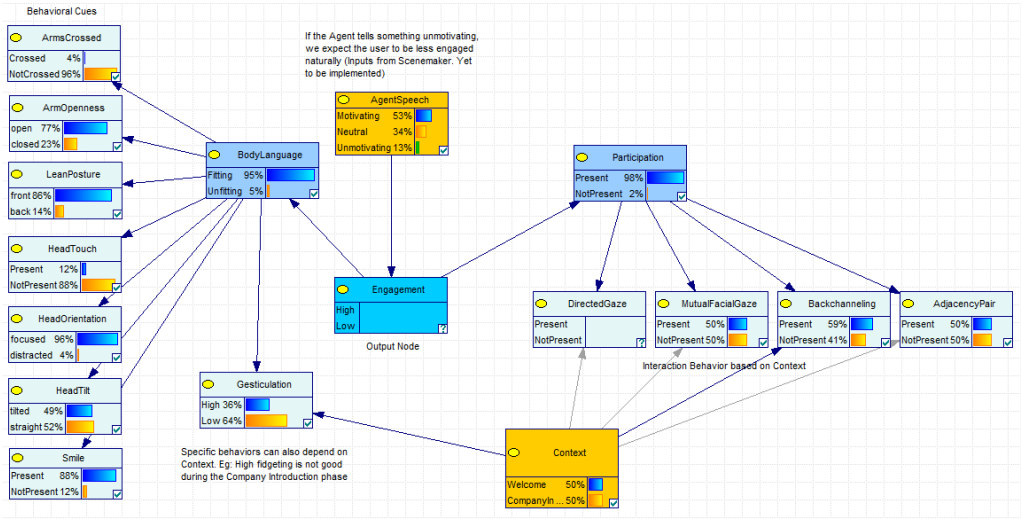


Figure 6.1: Bayesian network start state

As mentioned before each node has an associated probability table. Figure 6.2 shows the probability table of node Backchanneling. The node has two input directed edges coming from the nodes Participation and Context. The set probability of Backchanneling when participation is present and the context is welcome is 0.51. The probability of Backchanneling when participation is not present and context is companyintro (company introduction phase) is 0.01, and so on and so forth.

To set the values for the nodes our software communicates the output events to the bayesian network. As mentioned before each event carries an associated value from 0 to 1. These values are used to set evidance for the values in the nodes. For example if a mutual facial gaze occurs a MutualFacialGaze event is prompted with a pointer value of 1. The bayesian network then sets the evidance for the node MutualFacialGaze as Present = 100%, notPresent = 0%.

Figure 6.3 dipicts an example of the state of the bayesian network at a given intance. It mimics a hypothetical situation during the interview from the point of view of the nodes of the four main modules only. All the other nodes are left untouched and contain their default values. In this case the MutualFacialGaze and AdjacencyPair is present (and therefore set at 100%) while the DirectedGaze and Backchanneling (both set at 0%) are not. This indicates a situation where the subject is facing the Agent and for example answering its question. By only setting these two variables the

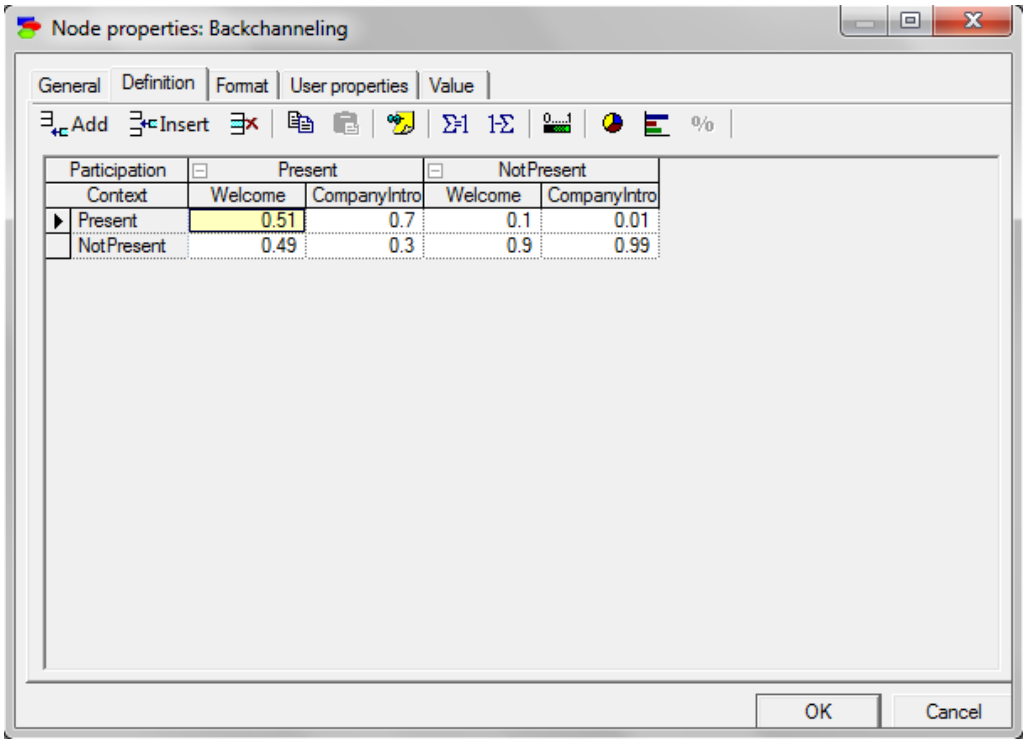


Figure 6.2: Probability table of Backchanneling node

bayesian network back propagated throught the Participation node to the Engagement node, calculating an engagement probability of 78%. This is of course very intuitive since a subject facing and replying to an agent is clearly engaged in conversation.

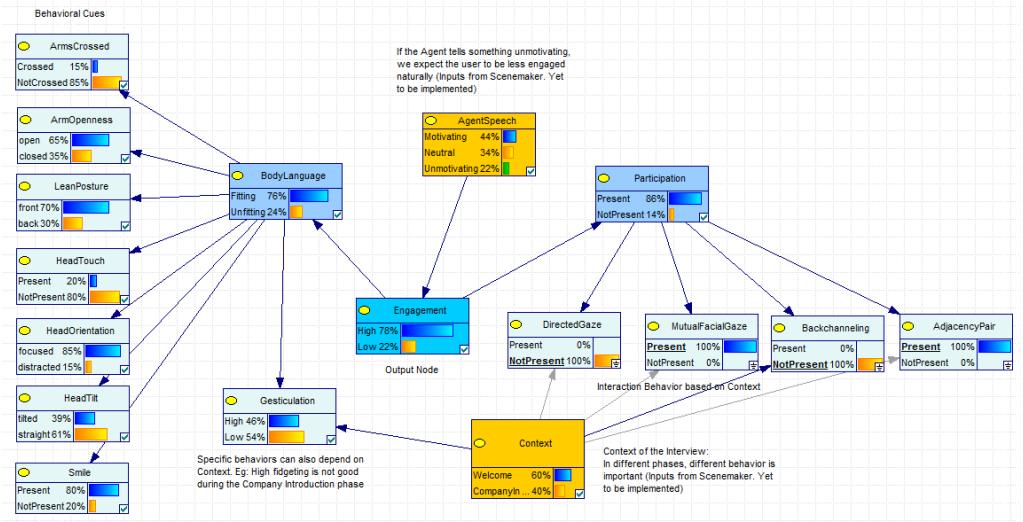


Figure 6.3: Instance of bayesian network

Chapter 7

Summary

To reach the thesis's aim of detecting the engagement of an interviewee in an interview situation we have devised a setup which consists of a hardware and software component.

The hardware component refers to the setup of the expariment. The interviewee is seated across two virtual agents present on a screen. Attached to the top of the screen a Microsoft Kinekt which mainly aims to track the movement of the subject's head. On the bottom of the screen lies an SMI Eyetracker used to track the subject's gaze and feed gazepoint coordinates to our software. And finally a microphone that is used to record the subject's speech utterences.

The pipeline of our software is topological. As shown in figure 7.1 the software's pipeline starts with the raw input data given by both the subject and the agent. Namely the subject's head nod, gaze, and speech which are captured by the Microsoft Kinekt, the SMI Eyetracker and the microphone respectively, and the agent's gaze and speech, which are both data that is generated by the external software Scenemaker.

The next level of the pipeline includes the basic events. *HeadNod* which is controlled by the subject's nodding, *SubjectFacialGaze* and *SubjectObjectGaze*, triggered by the location of the subject's gaze. *vad* (Voice Activity Duration) that is prompted when the subject speaks. As for the agent there are *AgentFacialGaze*, *AgentObjectGaze*, and *AgentSpeech* which are activated by where the agent's looking and whether or not it is speaking.

Next comes the four main modules in this thesis. Those are the four main phenomena that the software tries to capture. First there is *MutualFacialGaze*, which can be thought of as the eye contact between tha

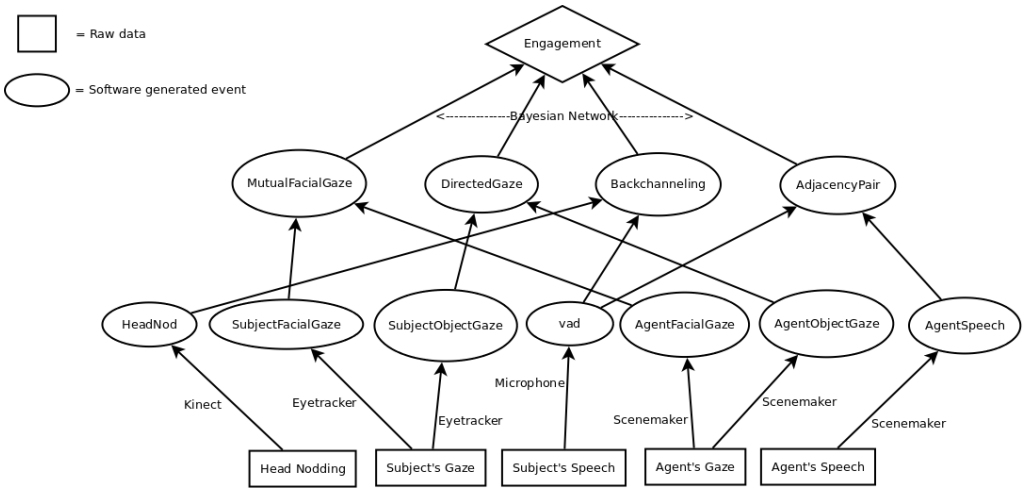


Figure 7.1: Software pipeline

subject and the agent. This is controlled by the basic events SubjectFacialGaze and AgentFacialGaze. Next there is *DirectedGaze*. It is triggered when the subject follows the gaze of the agent when it directs him to a certain object. This event is controlled by SubjectObjectGaze and AgentObjectGaze. Then comes *Backchanneling*. Backchanneling is an event in which the subject directs a brief verbal event or gestural communication back to the agent during the agent's speech. This event indicates that the subject is following what the agent is saying. Backchanneling is controlled by HeadNod, vad, and AgentSpeech. Finaly comes *AdjacencyPair* which is a speech communicated by the subject following a speech uttered by the agent. AdjacencyPair is controlled by vad and AgentSpeech.

Last level in our topology is the Enagagement level. Contrary to figure 7.1 the four main modules are not the only contributors to the calculation of the engagement level. They are merely so from the point of view of this thesis. The main modules are inputed into the bayesian network and together with other factors calculate the level of engagement.

Bibliography

[1] Charles Rich, Brett Ponsler, Aaron Holroyd, and Candace L Sidner. Recognizing engagement in human-robot interaction. In *Human-Robot Interaction (HRI), 2010 5th ACM/IEEE International Conference on*, pages 375–382. IEEE, 2010. [cited at p. 3]

[2] Candace L Sidner, Christopher Lee, and Neal Lesh. Engagement rules for human-robot collaborative interactions. In *IEEE INTERNATIONAL CONFERENCE ON SYSTEMS MAN AND CYBERNETICS*, volume 4, pages 3957–3962, 2003. [cited at p. 4]

List of Figures