European Union's Horizon 2020 research and innovation programme 645378, ARIA-VALUSPA
(November, 2015)

Artificial Retrieval of Information Assistants – Virtual Agents with Linguistic Understanding, Social skills, and Personalised Aspects

Collaborative Project

Start date of project: 01/01/2015 Duration: 36 months

(D4.1). Implementation of overall dynamic audio-visual communicative behaviour generation

Due date of deliverable: Month 11 Actual submission date: 01/12/2015

ARIA Valuspa

European Union's Horizon 2020 research and innovation programme 645378, ARIA-VALUSPA
(November, 2015)

Project co-funded by the European Commission		
Dissemination Level		
PU	Public	X
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

STATUS: [DRAFT]

Deliverable Nature		
R	Report	
P	Prototype	
D	Demonstrator	X
O	Other	

Participant Number	Participant organization name	Participant org. short name	Country
Coordinator			
1	University of Nottingham, Mixed Reality/Computer Vision Lab, School of Computer Science	UN	U.K.
Other Beneficiaries			
2	Imperial College of Science, Technology and Medicine	IC	U.K.
3	Centre National de la Recherche Scientifique, Télécom ParisTech	CNRS-PT	France
4	Universitat Augsburg	UA	Germany
5	Universiteit Twente	UT	The Netherlands
6	Cereproc LTD	CEREPROC	U.K.
7	La Cantoche Production SA	CANTOCHE	France

Table of Contents

1. Purpose of Document

This document provides a description of the design, architecture and implementation of components in the ARIA system. This system allows us to generate the dynamic audio-visual communicative behaviour displayed by the ARIA agent in the scenarios described in D5.1.

This deliverable is organized as follows. In Section 2 we describe the ARIA system design and architecture. In Section 3 we provide a brief overview of existing components developed by each partner and we describe new components that have been implemented for the ARIA system building on existing ones. Section 4 describes where to download the demonstrator and how to run it. Finally, we present other deliverables that we obtained in this first period of the project and a plan for the next period.

2. System Design and Architecture

2.1 Overview

The ARIA system architecture, shown in Figure 1, is divided in two: a Server and a Client part. The split is designed to offload as much computational effort as possible to the server, whereas the client side deals with the audio-visual input and output. In addition the Client side performs such computations as to reduce the amount of data that needs to be sent to the server for analysis. It is to be understood that there is always a trade-off between computational offload and data transmission load.

Three modules operate on the server side: Input, Agent Core and Output. In the Input module there are components dealing with user's audio-visual multimodal input detection. These components detect in real-time user's speech and multimodal behaviour (e.g. facial expressions) and provide low level signals (i.e. raw detected signals such as a user's head nod) and high level one (i.e. interpreted information, such as a user's emotional state) to the Agent Core module. The Agent Core's components keep track of the user socio-emotional and mental states, and plan the dialogue content and multimodal behaviour of the Agent. Finally, the produced speech and multimodal behaviour is handled by the Output module where a text to speech (TTS) component and rendering one finalize the output that need to be sent to the Client side where it is displayed to the user on a chosen platform and device (e.g. mobile).

European Union's Horizon 2020 research and innovation programme 645378, ARIA-VALUSPA
(November, 2015)

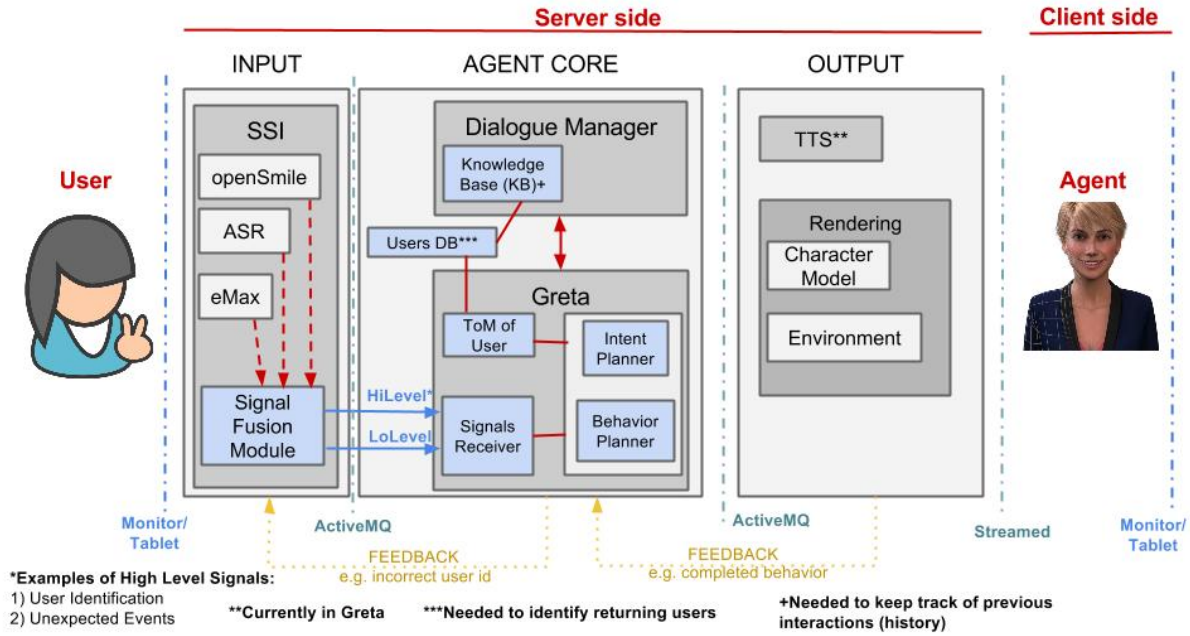

Figure 1: The ARIA system architecture.

2.2 COMPONENTS

2.2.1 Audio-visual multimodal input detection (UA, UN and IC)

The audio-visual multimodal input detection is implemented using the Social Signal Interpretation (SSI) framework[1] used in the whole ARIA-VALUSPA project, and adding it new components for advanced automatic speech and facial expression detection (ASR and eMax). SSI (described later) supports a large range of sensor devices, filters and feature algorithms, as well as machine learning and pattern recognition plugins (cf. D5.1, section 2.2.4 for more details on SSI).

Automatic Verbal and Non-Verbal Speech Analysis (IC)

The automatic speech analysis component has two main tasks: detection of verbal elements of speech (i.e. Automatic Speech Recognition) and the recognition of non-verbal aspects of speech, including voice activity detection, turn detection, and recognition of emotion and other paralinguistic elements such as gender or age. In addition it will facilitate person-specific adaptation of the ARIA agents by providing speaker recognition.

Automatic Face Analysis (UN)

The automatic face analysis component has a number of tasks. Its primary goal is the recognition of non-verbal aspects of language such as facial expressions of emotion and social signals, including gaze. But in addition it will be used to determine if someone is facing the agent by determining if a face is present in front of the camera, empower the

[1] Wagner, J., Lingenfelser, F., Baur, T., Damian, I., Kistler, F., André, E. (2013). The social signal interpretation (SSI)

European Union's Horizon 2020 research and innovation programme 645378, ARIA-VALUSPA
(November, 2015)

ARIA agents with person-specific adaptation by providing face recognition, age estimation, and gender estimation.

2.2.2 Multimodal behaviour planning and realization (CNRS-PT and UT)

Greta (CNRS-PT)

The Greta component uses the detected user's audio-visual multimodal features together with an internal component indicated as ToM User (i.e. Theory of Mind User) for generating communicative intents and, therefore, behaviour plans accomplishing those intents via multimodal behaviour. The ToM user allows this component to build a theory of mind of the user reflecting, for example, the user's emotional state, attitude and beliefs. A shared Users DB component allows the Greta and Dialogue Manager to store previous interactions information about the users as well as previous user's socio-emotional states.

Dialogue Manager (UT)

The basic architecture of a spoken dialogue system is as follows (see Figure 2). Speech recognizers (ASR) process the input from the user and output the result to the Natural Language Understanding (NLU) component, which in turn provides input to the Dialogue Manager (DM). The DM decides what utterance needs to be produced and lets the Natural Language Generator (NLG) produce this utterance, which is then turned into speech by the Text-To-Speech (TTS) module.

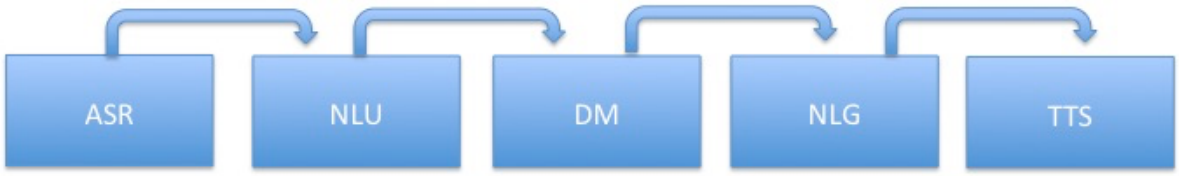

Figure 2: A basic Dialogue Manager architecture.

The components of this basic architecture can also be found in the architecture of the ARIA agent. The most important difference is that the ARIA agent is multimodal in its input and output.

2.2.3 Audio-visual Multimodal output (CANTOCHE and CEREPROC)

Animation (CANTOCHE)

The animation and rendering of our virtual assistant is done via Cantoche's Living Actor technology. This is a rendering engine that allows the ARIA system to display an animated virtual character in a 3D environment. This component receives synchronization information from the Agent Core module in order to display the character's multimodal behaviour in synchrony with synthesised speech.

Speech Synthesis (CEREPROC)

The speech synthesis is obtained via the CereVoice TTS engine developed by CEREPROC and shown in the OUTPUT part of Figure 1. The TTS engine upon receiving the dialogue content (e.g. sentences that the agent needs to utter) it produces the corresponding synthetized speech in synchrony with the multimodal behaviour plan built through the Greta system (e.g. gestures and facial expressions accompanying speech).

3. System Implementation

3.1 Existing components

3.1.1 SSI: The Social Signal Interpretation Framework (AU)

The Social Signal Interpretation (SSI) framework offers tools to record, analyse and recognize human behaviour in real-time, such as gestures, mimics, head nods, and emotional speech. Following a patch-based design pipelines are set up from autonomic components and allow the parallel and synchronized processing of sensor data from multiple input devices. In particularly SSI supports the machine learning pipeline in its full length and offers a graphical interface that assists a user to collect own training corpora and obtain personalized models. In addition to a large set of built-in components SSI also encourages developers to extend available tools with new functions. For inexperienced users an easy-to-use XML editor is available to draft and run pipelines without special programming skills.

- Synchronized reading from multiple sensor devices, e.g. microphone, asio audio interface, web-cam, dv-cam, wiimote, kinect and physiological sensors.
- General filter and feature algorithms, such as image processing, signal filtering, frequency analysis and statistical measurements in real-time.
- Event-based signal processing to combine and interpret high level information, such as gestures, keywords, or emotional user states.
- Pattern recognition and machine learning tools for on-line and off-line processing, including various algorithms for feature selection, clustering and classification.
- Patch-based pipeline design (C++-API or easy-to-use XML editor) and a plug-in system to integrate new components.

European Union's Horizon 2020 research and innovation programme 645378, ARIA-VALUSPA
(November, 2015)

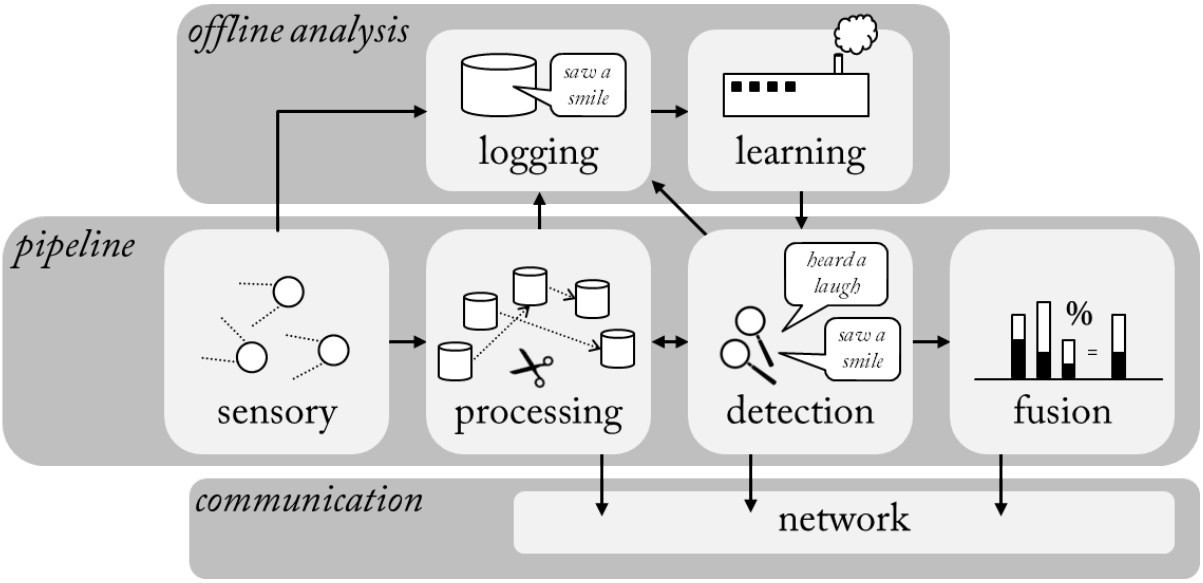

Figure 3: The SSI framework architecture.

SSI is open-source and provides possibilities to add further sensors and algorithms. In the ARIA-VALUSPA project, besides other improvements, University of Nottingham's e-max and automated speech recognition from Imperial College will be added. Those will be described in the next sections.

3.1.2 E-Max: Facial Expression Detection (UN)

The eMax visual Facial Expression Recognition (FER) module has been trained to recognise the six prototypical emotions (anger, disgust, fear, happiness, sadness, and surprise). The output of this module is a per-frame and per-label score indicating the confidence of each of the possible labels. This includes the absence of any emotion, i.e., the presence of a neutral face, leading to 7 real-valued scores. A base system existed for ARIA, but this has been improved in a number of ways. See Deliverable D2.1 for details.a

3.1.3 Automatic Speech Recognition (IC)

We created an SSI pipeline for online speech analysis and recognition of various user's states and traits (cf. D2.1, section 2.1). This pipeline captures the user's speech through a standard computer microphone, extracts relevant acoustic features, feeds the adequate features to the respective recognition modules (pre-trained offline), and outputs the predicted class or value that later can be used by the Signal Fusion Module.

The general architecture of this component is described in more detail in D2.1 (Section 3.2). The component is implemented using state-of-the-art approaches. Feature extraction is based on Mel-frequency Cepstral Coefficients (MFCCs) with additional delta and acceleration coefficients. The language model is a modified Kneser-Ney smoothed backoff 4-gram language model. Acoustic models are context-dependent triphone models trained using a hybrid Deep Neural Network – Hidden Markov Models (DNN-HMM) setup. The DNN part is based on deep maxout neural networks with p-norm pooling strategy. These neural networks are trained to predict the posterior

probabilities of each context-dependent state, which are then divided by the corresponding state prior probability to provide a "pseudo-likelihood" that is used in place of the state emission probabilities in the triphone HMMs. The neural network training is performed on top of feature space maximum a posteriori linear regression (fMLLR) speaker adapted features. The decoder is based on Weighted Finite State Transducers (WFSTs).

3.1.4 The GRETA/VIB Platform

The Greta/VIB system allows a virtual or physical (e.g. robotic) embodied conversational agent to communicate with a human user[2][3]. The global architecture of the system is depicted in Figure 4. It is a SAIBA compliant architecture (SAIBA is a common framework for the autonomous generation of multimodal communicative behaviour in Embodied conversational agents[4] [3]). The main three components are: (1) an Intent Planner that produces the communicative intentions and handles the emotional state of the agent; (2) a Behaviour Planner that transforms the communicative intentions received in input into multimodal signals and (3) a Behaviour Realizer that produces the movements and rotations for the joints of the ECA. A Behaviour Lexicon (i.e. Multimodal Behaviour Lexicon in Figure 4) contains pairs of mappings from communicative intentions to multimodal signals. The Behaviour Realizer instantiates the multimodal behaviours, it handles the synchronization with speech and generates the animations for the ECA.

VIB (Virtual Interactive Behaviour) is an enhancement of Greta with additional components that allow the ECA to detect its environment (i.e. Perceptive Space), and to interact with the user while constantly updating the agent's mental and emotional states. Thus, an ECA's mental state includes information such as beliefs, goals, emotions and social attitudes. Different external tools plugged-in the VIB platform (i.e. SHORE facial expressions, SEMAINE facial action units and acoustics, and speech recognition as shown on the right side of Figure 4) allow an agent to detect and interpret user's audio-visual input cues captured with devices such as cameras, Microsoft's Kinect and microphones. This information is provided to the agent via the Perceptive Space module. A direct link between this module and the Behaviour Realizer allows the agent to exhibit reactive behaviours by quickly producing the behaviour to exhibit in response to user's behaviour, as for back-channels for example. Finally, the Motor resonance manages the direct influence of the socio-emotional behaviours of the user (agent perceptive space) to the ones of the agent (agent production space) without cognitive reasoning. In particular, it allows the ECA to dynamically mimic the behaviour of the user.

[2] Ochs M, Prepin K, Pelachaud C (2013). From emotions to interpersonal stances: Multi-level analysis of smiling virtual characters. In: Affective Computing and Intelligent Interaction (ACII), 2013 Humaine Association Conference on, IEEE, pp 258-263.

[3] Niewiadomski R, Obaid M, Bevacqua E, Looser J, Anh LQ, Pelachaud C (2011). Cross-media agent platform. In: Proceedings of the 16th International Conference on 3D Web Technology, ACM, New York, NY, USA, Web3D, pp 11-19.

[4] Kopp S, Krenn B, Marsella S, Marshall AN, Pelachaud C, Pirker H, Thorisson KR, Vilhjalmsson HH (2006) Towards a common framework for multimodal generation: the behavior markup language. In: Proceedings of the 6th international conference on Intelligent Virtual Agents, Springer-Verlag, Berlin, Heidelberg, IVA, pp 205-217.

European Union's Horizon 2020 research and innovation programme 645378, ARIA-VALUSPA
(November, 2015)

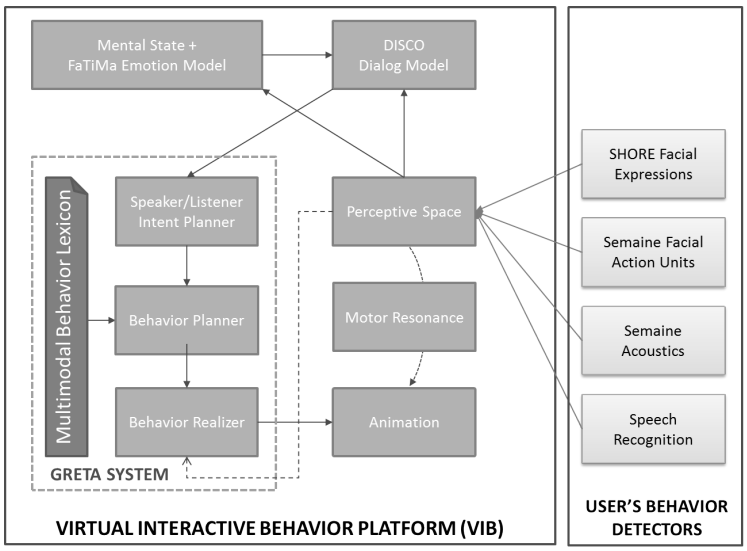

Figure 4: The Greta/VIB System architecture

3.1.5 Dialogue management (UT)

The Dialogue Management approach that was used in SEMAINE[5] and that is reflected in the current system architecture resembles an information state architecture. According to Larsson and Traum[6] an information-state dialogue manager is composed of five main components: a description of the informational components of the theory of dialogue modelling (beliefs, intentions, conversational structure, etc.), their formal representation, a set of update rules for the data in the information state, dialogue moves that trigger these update rules, and an update strategy that determines when to apply which rule.

The responsibility of the DM in the classical sense is to control the structure of the dialogue. Given the input it needs to decide what output should be produced. In the earliest systems such as ATIS[7], the conversation consisted of a sequence of questions by the system and answers by the users. The ordering of the question was more or less fixed and implemented as a finite state machine.

Finite state machines

The simplest way to model a conversation is by generating a graph of all its possible states, thus allowing a limited set of responses to each of the agent's utterances. This approach is perfect for relatively simple task-oriented dialogues, but can be very cumbersome for the designer to create, as they need to plan every single state of the

[5] ter Maat, M. and Heylen, D.K.J. (2011) Flipper: An Information State Component for Spoken Dialogue Systems. In: Proceedings of the 10th international conference on Intelligent Virtual Agents (IVA 2011), 15-17 Sep 2011, Reykjavik, Iceland. pp. 470-472. Lecture Notes in Computer Science 6895.

[6] Larsson, S., Traum, D.R.: Information state and dialogue management in the Trindi dialogue move engine toolkit. Nat. Lang. Eng. 6(3-4), 323-340, (Sep 2000).

[7] Seneff, S., Hirschman, L., & Zue, V. W. (1991). *Interactive problem solving and dialogue in the ATIS domain*. MASSACHUSETTS INST OF TECH CAMBRIDGE LAB FOR COMPUTER SCIENCE.

conversation. This is the type of dialogue management used by Role Playing Games or automated train reservation terminals. One system for generating this kind of tree with an integrated NLG unit is the DISCO system by Rich, which uses the ANSI/CEA-2018 task model specification[8].

These architectures work somewhat for specific kinds of conversations, which concern very structured tasks such as asking for information regarding flights as in the ATIS system. However, other types of tasks will require a more complicated dialogue. In the case of the ARIA Agent, the user is also allowed to ask questions for instance. Furthermore, the system needs to deal with errors, unexpected situations, and with the mental - including emotional - state of the user. It should also be able to engage in chat, question-answering, and meta-dialogue amongst others. This requires a more complex system in which more knowledge needs to be stored and updated on the one hand and some structure of the dialogue needs to be accounted for.

Information state approach

This approach contains several components, and is the one that tries to bring the most complete semantic representation of information and language. According to Larsson and Traum[9], an information-state dialogue manager is composed of the main components: a description of the informational components of the theory of dialogue modelling (beliefs, intentions, conversational structure, etc.), their formal representation, a set of update rules for the data in the information state, dialogue moves that trigger these update rules, and an update strategy that determines when to apply which rule. This approach is applied in TrindiKIT[10].

The Update Rules typically have a Condition-Action format. If the Condition holds (that is if a subset of the variables in the Information State has particular values) than either the Information State is updated or some communicative action is selected for execution. In the SEMAINE project we developed the FLIPPER toolkit for this[11].

Deciding on Dialogue Management Architecture and Toolkit / Current Experimentation

The more-or-less standard procedure to design dialogue systems is by following a number of steps (see[12]).

1. Study the user and task
2. Build simulations and prototypes - using a Wizard of Oz set-up possibly
3. Iteratively test the design on users

The initial studies of the users and the building of prototypes through WOz experiments is on-going. This serves the purpose of (1) establishing the requirements for the Dialogue

[8] See also: https://trac.telecom-paristech.fr/trac/project/greta/wiki/Disco
[9] Larsson, S., Traum, D.R.: Information state and dialogue management in the Trindi dialogue move engine toolkit. Nat. Lang. Eng. 6(3-4), 323-340, (Sep 2000).
[10] www.ling.gu.se/projekt/trindi/trindikit/
[11] Ter Maat, M. and Heylen, D.K.J. (2011) Flipper: An Information State Component for Spoken Dialogue Systems. In: Proceedings of the 10th international conference on Intelligent Virtual Agents (IVA 2011), 15-17 Sep 2011, Reykjavik, Iceland. pp. 470-472. Lecture Notes in Computer Science 6895.
[12] Martin, J. H., & Jurafsky, D. (2000). Speech and language processing.*International Edition*.

Management Architecture and system (or toolkit) that is needed and (2) providing input for the development of actual dialogues and the information that needs to be stored in the Information State (such as the Theory of Mind module, the Knowledge Base etc.)

We have collected an initial corpus of Wizard-of-Oz type interactions that we are currently analysing. Furthermore we have tested several possible toolkits for use in the development of the ARIA Agent. This includes DISCO, mentioned above. The current feeling is that the task-based format makes it less suitable for ARIA-style dialogues.

Another approach that we have tried to some extent is the information Retrieval approach. This approach has been successfully implemented in NPCEditor[13], the DM toolkit shipped with the VHToolkit. It fit the main purpose of the current ARIA agents that have a major information retrieval task as this approach sees the response selection as an information retrieval problem, in which the user's utterance is a query, and an appropriate answer is the most fitting result. This approach is a good fit for question-answering systems (for user guidance, for example). All possible questions and answers are stored in a database and linked with a many-to-many relationship. When the DM receives a user utterance, it looks for the stored question that is the most similar to it, gets the list of possible answers to this question, and picks one depending on some annotations on the answers and the dialogue state. If the creation of the complete database of questions-answers can be cumbersome for the IVA designer, the information retrieval approach makes DM very robust to unexpected questions: the agent will always find an answer to give, even wrong, bringing the conversation back to where it is knowledgeable. It might be used for testing purposes and some prototypes but may need to be extended or replaced by more involved systems.

Finally, in the current phase of the project we are testing and evaluating some other approaches as well, such as task-based system that is integrated with Greta, called DISCO[14] and a custom-built system described in Deliverable 3.1 and Section 3.2 of this document.

3.1.6 Living Actor: Character Animation (CANTOCHE)

The Living Actor™ Technology is a software suite that offers the best-of-breed in next generation human-computer interaction. Based on high quality 3D interactive Avatars, artificial intelligence algorithms, and open web-services, Living Actor™ offers standardized SaaS solutions for clients with their own SDK or API, or for partners and third parties who want to include Living Actor™ in their applications or R&D projects.

3.1.7 CereEngine: Speech Synthesis Engine (CEREPROC)

The CereEngine can perform characterful synthetic speech in real-time and on a wide variety of platforms. It can be deployed on any platform from smartphones to high-end servers.

The expressivity of the TTS output can be controlled through the use of specific XML tags inserted in the text to be pronounced. The current system support several natural

[13] Leuski, A., Traum, D.R.: Npceditor: A tool for building question-answering characters. In: LREC (2010)
[14] https://trac.telecom-paristech.fr/trac/project/greta/wiki/Disco

emotional modalities in almost all voices: happy, sad, tense, calm. Through the use of standard SSML tags, the prosody can be further altered to more finely control emphasis and other high-level prosodic features. Some natural sounding non-verbal conversation fillers such as hesitations or encouragements can also be inserted in the conversation flow.

3.2 Implemented components

3.2.1 Dialogue Management

The current dialogue management system implements probabilistic graphical models (Dynamic Bayesian Networks) to determine the next states for the dialogue. Figure 5 illustrates the current model of dialogue transition. Only the parts within the purple area are currently implemented in the system. The remaining parts will be extended using the ERISA Framework[15], which will introduce emotion, personality, and social relationships models as variables in the Agent Model of the User and Agent Model of the Agent to determine the next agent's utterance. Figure 6 shows how emotion, personality and social relationships interact with each other on influencing the agent's behaviour in the ERISA Framework. We distinguish internal emotions, which the agents actually "felt", from those displayed by it. This mimics the fact that a number of aspects, such as their social relationship with the user, cultural display rules and the current situation can influence people to mask their emotions[16].

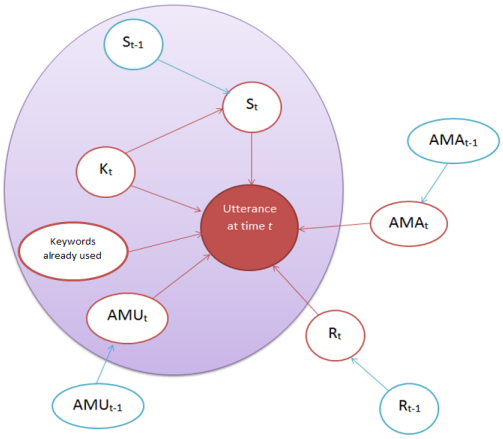

Figure 5: Dialogue Models

[15] Chowanda, A., Blanchfield, P., Flintham, M. and Valstar, M.F. (2014). Erisa: Building emotionally realistic social game-agents companions. In Proc. *Intelligent Virtual Agents*, 8637 p. 134-143

[16] Ekman, P. (2007). Emotions revealed: Recognizing faces and feelings to improve communication and emotional life. Macmillan.

European Union's Horizon 2020 research and innovation programme 645378, ARIA-VALUSPA
(November, 2015)

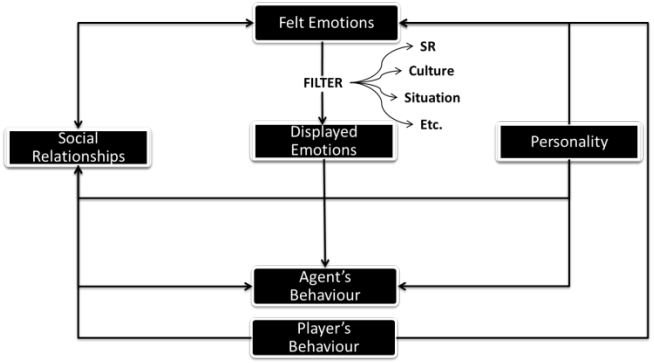

Figure 6: Models of Personality, Emotions and SR

The current utterance is determined by:

- Current dialog state (S_t) and the previous one (S_{t-1}).
- Keyword input by the user at time t, K_{t-1}.
- The list of keywords already used during the conversation to handle the repetitions.
- Agent Model of the User (AMU_t, AMU_{t-1}), where :

$$AMU = \{ID, gender, name, surname, age, Social\ Relationships, Emotions, Engagement, Personality\}$$

- Social relationships at time t, R_t and $t-1$, R_{t-1}.
- Agent Model of the Agent (AMA_t, AMA_{t-1}), where :

$$AMA = \{ID, gender, name, surname, age, Social\ Relationships, Emotions, Engagement, Personality\}$$

To determine the current utterance, the input from either the Automatic Speech Recognition module or text input is parsed to keywords. The system then matches the recognised keywords with those from the keyword database, which will return the probability values of each state given that particular keyword. The Levenshtein Distance is applied to determine the confidence value for the spotted keywords. The agent will ask a confirmation of the user or ask the user to rephrase their sentence if the confidence value is relatively low.
Simultaneously, the system also calculates the probability values of the next possible states, updates the social relationships with the user, and the user emotions perceived by video analysis components. The system then selects a file that has the highest probability value. Afterwards, the selected sentences and emotions are sent to GRETA system using the FML mark-up language. Finally, the system updates all the utterance data states (e.g. last user keyword spotted, the list of keywords already used to prepare the next utterance etc.).

All the agent's utterances are stored in a file format, where each file contains the following information:

- User's emotion as perceived by the system
- A two-dimensions vector of Social Relationship (Affinity and Familiarity)
- Keywords
- List of possible next states
- The utterance

Example of an utterance file:

```
0.5
0.5 1
wonderland information opinion ?
wonderland
You look happy today, let's me tell you about the Wonderland where Alice goes after falling into the Rabbit Hole.
It seems to look like a big garden where animals can talk just like humans.
The Cheshire Cat says to Alice that she would not have come there if she was not mad.
```

3.2.2 Speech Synthesis

In the current ARIA implementation, the TTS engine is integrated within the Greta component, through the use of a Java class wrapping the CereEngine low-latency API. Through that API, the speech output can be generated and played back to the user in real-time, allowing a human-like reactivity.

4. Case Scenario and Real-Time Demonstration

We created a scenario to demonstrate the behaviour generation in the context of the Book-ARIA The code is publicly available from https://github.com/ARIA-VALUSPA/ARIA-System. The particular demonstrator for D4.1 can be run on Windows machines by double-clicking the file "RUN-All.bat". This is essentially a full system demonstration, including emotion recognition, dialogue management and behaviour generation, as we felt that this was the best way to demonstrate the functioning of the behaviour generation.

5. Outputs

In what follows, we indicate the outputs with pertinence to this deliverable (categorised by topics) that have been published (or are *in press*) in the first 11 months of the project.

Virtual Human Evaluation

Baur, T., Mehlmann, G. Damian, I. Lingenfelser, F., Wagner, J., Lugrin, B., André, E. and Patrick Gebhard (2015). Context-Aware Automated Analysis and Annotation of Social Human-Agent Interactions.

6. Plans for Next Period

For the next period (short-term) we plan to start integrating Cantoche's Living Actor technology in the ARIA system integration. Moreover, by the end of month 13, we plan to have a first integrated prototype of the system that can be used to screen a system demo for a German TV documentary.

In the mid-term, we want to enhance the Dialogue Manager (DM) component by improving the current implementation and ultimately have it capable of planning the agent's utterances needed in our use case scenarios (see deliverable XX). We also plan to implement interruptible agent multimodal behaviour and TTS calls, i.e. if the user starts

speaking; the system needs to gracefully offer the floor. For example, current TTS systems just halt in mid-sentence or wait for a sentence break to stop speaking. In order to achieve human-like behaviour, the ARIA-VALUSPA Agent Core will need to be able to offer alternative completions and/or alter the speech synthesis and multimodal behaviour as the agent is speaking.

Furthermore, in year 3, we plan to work on gradual, higher quality emotional speech. Currently the quality degrades when emotional voice qualities (such as tense voice) are chosen. We will investigate the use of closed phase LPC analysis to carry out source filter modelling in order to modify neutral speech corpora into tense or lax, allowing us to improve the coverage of emotional data. That DSP approach would also allow us to perform gradual changes in the emotional modality.