ARIA Valuspa

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

(D3.1). Implementation of multi-lingual task-based dialogue system for the chosen scenario

Due date of deliverable: Month 11 Actual submission date: 1/12/2015

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

(November, 2015)

1. Purpose of Document

This document describes the current version of the Dialogue Management System that is used for experimentation with the ARIA-VALUSPA Agents. Besides some documentation of the software that is being delivered at this stage (Section 3), it also provides some background on dialogue management systems, the options that we are considering, and the steps we are making to establish the requirements that enable us to move forward (Section 2). Finally, we make a number of suggestions for next steps in the next period of the project.

2. Dialogue management

In the following paragraphs we present a short introduction to Dialogue Management systems in general and then present some of the options for dialogue management that we are exploring for the ARIA agent in various test cases.

The basic architecture of a spoken dialogue system is as follows. 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.

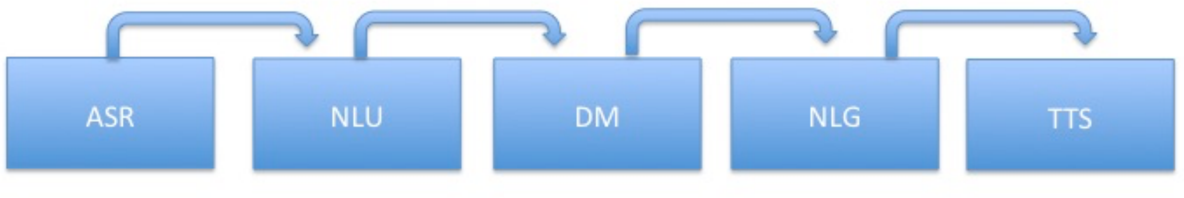

Figure 1: Steps involved in generating dialogue

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. Furthermore some components are spelled out further.

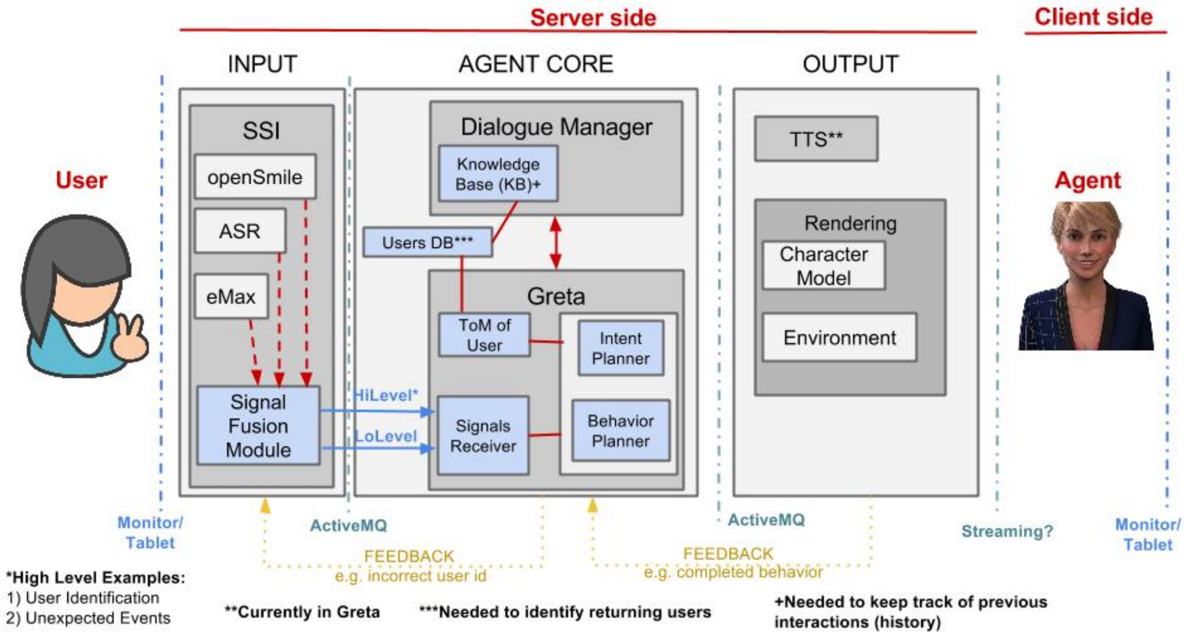

Figure 2: The ARIA-VALUSPA architecture

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[1], 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.

2.1 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 conversation. This is they 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[2], which uses the ANSI/CEA-2018 task model specification[3].

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

---

[1] Hemphill, C.T., Godfrey, J.J. and Doddington, G.R. (1990). The ATIS spoken language systems pilot corpus. In *Proceedings of the workshop on Speech and Natural Language*. Association for Computational Linguistics, p. 96-101

[2] Rich, C. (2009). Building task-based user interfaces with ansi/cea-2018. Computer 42(8), p. 20-27.

[3] See also: https://trac.telecom-paristech.fr/trac/project/greta/wiki/Disco

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.

2.2 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[4] [12], 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. This approach is applied in TrindiKIT.[5]

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[6].

Another approach that we have tried to some extent is the information Retrieval approach. This approach has been successfully implemented in NPCEditor[7], 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

---

[4] Larsson, S., Traum, D.R. (200). Information state and dialogue management in the trindi dialogue move engine toolkit. *Natural Language Eng*ineering 6(3-4), p. 323-340

[5] www.ling.gu.se/projekt/trindi/trindikit/

[6] 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*, Lecture Notes in Computer Science 6895, p. 470-472

[7] Leuski, A., Traum, D.R. (2010). Npceditor: A tool for building question-answering characters. In: *LREC*

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, we also implemented a probabilistic version of the information retrieval model and are testing that for its suitability to use for the encoding of dialogues. This model and its implementation constitute Deliverable 3.1.

3. System Implementation

We created a scenario to demonstrate the dialogue manager in the context of the Book-ARIA. The code is publicly available from https://github.com/ARIA-VALUSPA/ARIA-System. The particular demonstrator for D3.1 is run on Windows machines by double-clicking the file "RUN-Dialogue-Manager.bat". This starts a chat with Alice about the book 'Alice in Wonderland' by Lewis Carol. The chat is text-only. For a full system demonstration, including emotion recognition and behaviour generation, you can run "RUN-All.bat", but please note that this is essentially deliverable D1.1, which is not finalised until end of December 2015.

The current dialogue management system implements probabilistic graphical models (Dynamic Bayesian Networks) to determine the next states for the dialogue. Figure 3 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[8], 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 4 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[9].

[8] 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

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

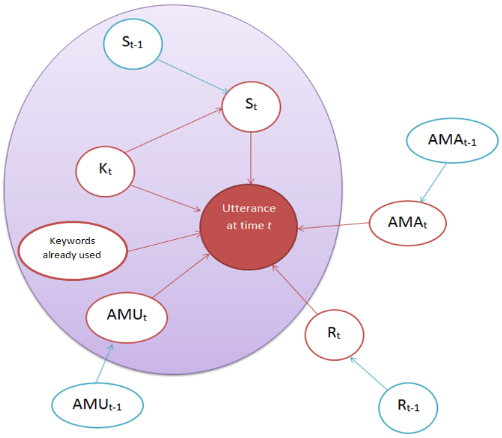

Figure 3: Dialogue Models

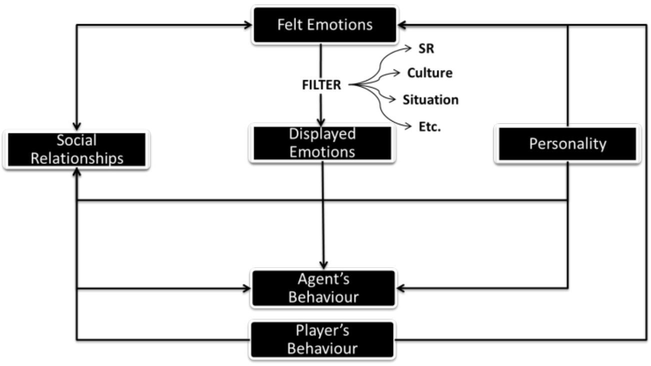

Figure 4: Models of Personality, Emotions, and SR

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.
```

4. Plans for Next Period

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 Jurafsky-Martin[10]).

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 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.

[10] Jurafsky, D. and Martin, J.H. (2009). Speech and Language Processing (2nd Edition). Prentice-Hall, Inc., Upper Saddle River, NJ, USA

(November, 2015)

In the next phase we will test the various toolkits further based on the types of dialogue that we have collected in the initial data collection phase and will then make a choice of toolkit and how to develop it further for the project.

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.

Intelligent Virtual Agents

Chowanda, A., Blanchfield, P., Flintham, M., Valstar, M.F. (2014). ERiSA: Building Emotionally Realistic Social Game-Agents Companions, *Proc. Int'l Intelligent Virtual Agents (IVA)*, 2014.