KRISTINA

A Knowledge-Based Information Agent with Social Competence and Human Interaction Capabilities

H2020-645012

D2.2

Integration of the OwlSpeak Platform into the KRISTINA Architecture

Dissemination level:	Restricted
Contractual date of delivery:	Month 12, 29.2.2016
Actual date of delivery:	Month 12, 25.2.2016
Workpackage:	WP2 Adaptive Dialogue Management
Task:	T2.2 Design and implementation of the basic dialogue manager
Type:	Report
Approval Status:	Draft
Version:	1.0
Number of pages:	23
Filename:	D2.2-Integration of the OwlSpeak Platform into the KRISTINA Architecture_2016-01-27_v1.0.docx

Abstract

This document describes the original architecture and functionality of OwlSpeak, discusses them in the light of the requirements of KRISTINA and specifies the extensions of OwlSpeak and the interfaces via which OwlSpeak is integrated into KRISTINA.

at its sole risk and liability.

co-funded by the European Union

History

Version	Date	Reason	Revised by
0.1	27.01.2016	Initial draft	Louisa Pragst
0.2	08.02.2016	Integrated comments by Stefan Ultes, Wolfgang Minker, Stamatia Dasiopoulou	Louisa Pragst
0.3	15.02.2016	Integrated comments by Wolfgang Minker, Leo Wanner, Ludo Stellingwerff	Louisa Pragst
1.0	23.02.2016	Integrated comments by Eleni Kamateri, Georgios Meditskos and internal reviewer Gregor Mehlmann	Louisa Pragst

Author list

Organization	Name	Contact Information
UULM	Louisa Pragst	louisa.pragst@uni-ulm.de
UULM	Stefan Ultes	stefan.ultes@uni-ulm.de
UULM	Wolfgang Minker	wolfgang.minker@uni-ulm.de
UPF	Leo Wanner	leo.wanner@upf.edu
UPF	Stamatia Dasiopoulou	stamatia.dasiopoulou@upf.edu
AL	Ludo Stellingwerff	ludo@almende.org
CERTH	Eleni Kamateri	ekamater@iti.gr
CERTH	Georgios Meditskos	gmeditsk@iti.gr
UAU	Gregor Mehlmann	gregor.mehlmann@informatik.uni-augsburg.de

Executive Summary

The dialogue manager is a core component of the KRISTINA system: It is responsible for managing the interaction with the user by keeping a dialogue history and selecting the system's dialogue contribution in response to the contribution of the user. Furthermore, the dialogue manager is responsible for generating a system emotion and considering it when selecting the system's contribution, in order to make the behaviour of the system more human-like. The OwlSpeak dialogue manager (Heinroth, et al., 2010) (Ultes & Minker, 2014) has been chosen as basis for the dialogue management of the KRISTINA system and therefore needs to be integrated into the overall KRISTINA architecture. This document describes the adaptations and extensions of OwlSpeak that are necessary to complete this task.

We describe the original architecture of OwlSpeak and illustrate its functionality with the help of a use case example. Furthermore, a short overview of the KRISTINA architecture is given. The integration requirements are identified using those descriptions. On the basis of the determined requirements, we introduce the new architecture of OwlSpeak and specify the interactions of OwlSpeak with other KRISTINA components. The new dialogue management process is illustrated using the same use case example as before. Concluding, we provide a summary of the findings of this document.

Abbreviations and Acronyms

DM	Dialogue Management
EA	Emotion Analysis
KI	Knowledge Integration
LA	Language Analysis
SDO	Spoken Dialogue Ontology
SLG	Spoken Language Generation
STT	Speech-To-Text
VA	Valence/Arousal

Table of Contents

1 INTRODUCTION

Dialogue Management is a key component of every dialogue system. It shapes the system behaviour, e.g. by selecting the semantic content of the system's contribution to the dialogue, keeping a dialogue history in order to reference previous dialogue contributions, and recovering from misunderstandings. Thereby, it strongly influences the course of the system-user-interaction.

The OwlSpeak dialogue manager has been chosen as a starting point to handle the interaction with the user in the KRISTINA system. It has many advantages that can be useful for the KRISTINA system, in particular the ability to adapt the selection of the system's contribution, e.g. to the current situation or the user profile. Further advancements of the functionality are planned in the course of the project. However, OwlSpeak has been developed for the use with VoiceXML and is therefore restricted to spoken language. The KRISTINA system, on the other hand, includes an animated avatar as well as emotion and gesture recognition as additional modalities with the goal of providing a more empathic and relatable communication partner. Changes are required in order to adapt OwlSpeak to the multimodal setup of the KRISTINA system.

This document aims at identifying the requirements that come with the integration of OwlSpeak into the KRISTINA system, and defining the adaptations necessary to meet those requirements. In Section 2, we present the original architecture of OwlSpeak and illustrate its functionality with the help of a use case example. Section 3 describes the characteristics of the KRISTINA architecture, and thereby provides the basis for Section 4, in which the required adaptations of OwlSpeak are identified and described in detail. Additionally, we specify the interaction of OwlSpeak with other KRISTINA components and illustrate the functionality of the new architecture using the same use case example as in Section 2. Finally, the findings of this document are summarised in Section 5.

2 FUNDAMENTALS OF OWLSPEAK

OwlSpeak is an ontology-based dialogue manager initially developed by Heinroth et al (Heinroth, et al., 2010). Since then, it has been constantly improved and extended, e.g. by Ultes and Minker (Ultes & Minker, 2014). This section describes the architecture as well as the functionality of the current OwlSpeak implementation.

2.1 Architecture of OwlSpeak

OwlSpeak follows the model-view-presenter design pattern, which supports a strict separation of data management, dialogue logic and dialogue interface. It is implemented as a Java Servlet, so that easy interaction with speech devices is possible. The general architecture is shown in Figure 1.

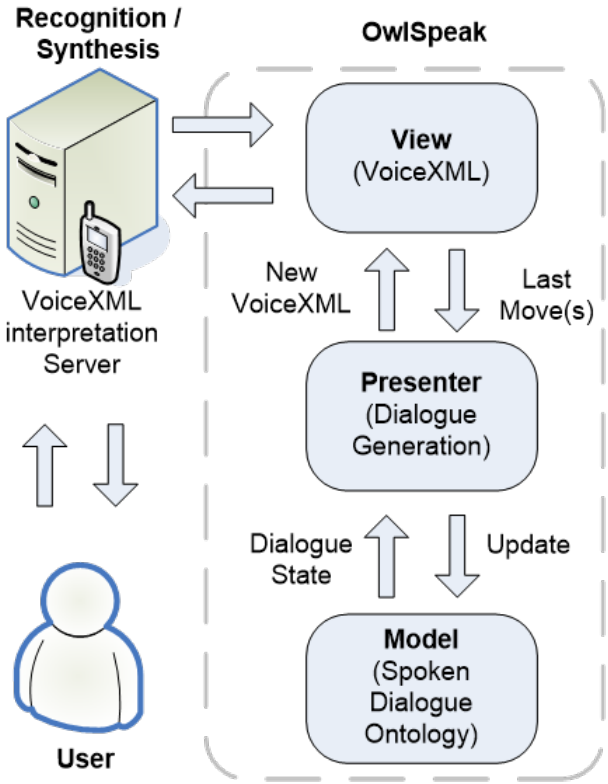

Figure 1: The architecture of OwlSpeak. (Heinroth, et al., 2010)

In the following, a short overview of the three components, model, view, and presenter, is given.

Spoken Dialogue Ontology

The model of OwlSpeak is based on the Web Ontology Language (OWL) (Antoniou & Van Harmelen, 2004) and referred to as Spoken Dialogue Ontology (SDO). A schematic representation of an SDO is shown in Figure 2. SDOs consist of two parts: the static description of the dialogue domain, called speech part, and the current dialogue state, called state part, which is updated after each dialogue turn.

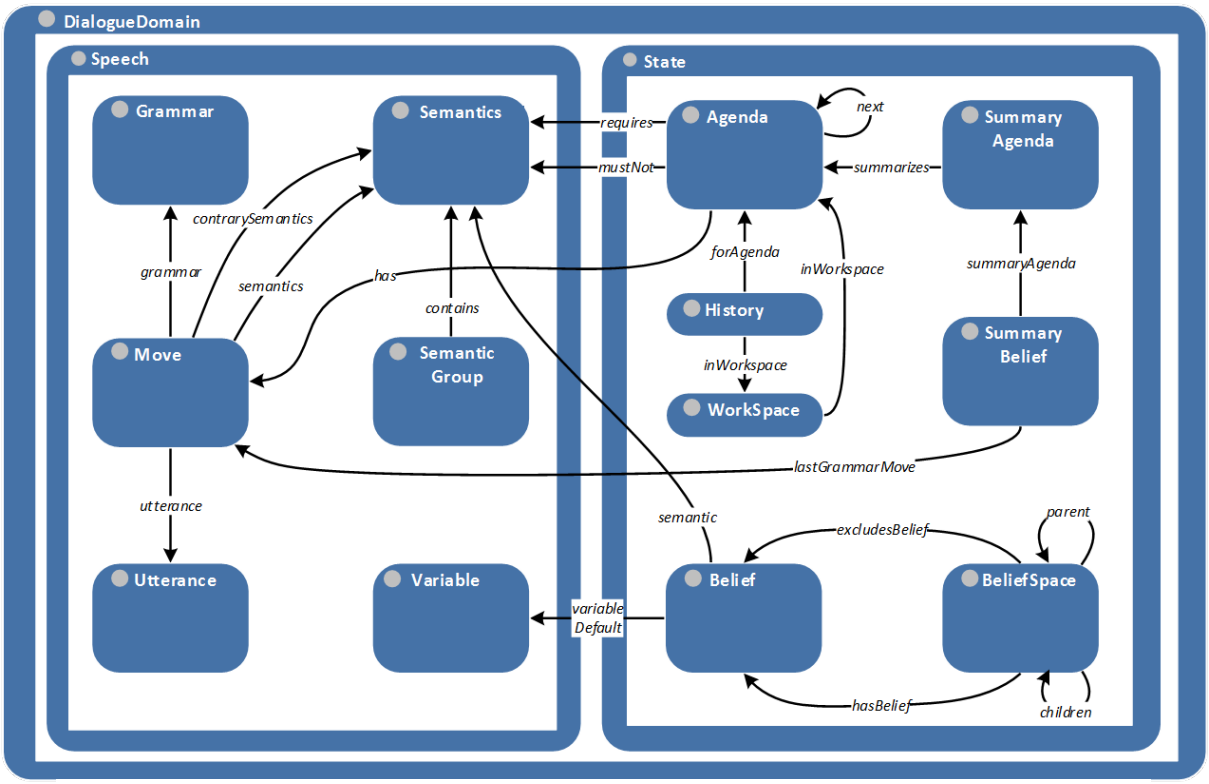

Figure 2: Schematic representation of the Spoken Dialogue Ontology.

The most important concepts of an SDO are described in the following:

- *Utterance:* The utterance concept represents, what the system may say in one turn. This can consist of one or more sentences. An utterance could for example be 'Hello. What can I do for you?'
- *Grammar:* The grammar concept determines what the user can say or rather the system can understand. A grammar could look like this: 'What's the (optimal|ideal|best) water temperature to bathe a baby?' In this case, the user could say either of the words optimal, ideal and best, but has to phrase the rest of the sentence as specified in order for the system to recognise the sentence.
- *Semantics:* Semantics represents information that is important from a dialogue perspective. This can be the meaning of what was said or how often question has already been asked, for example. A semantics *Semantics_Temperature* could be used to indicate that the user has requested the best temperature for a bath.
- *Move:* The move concept represents one atomic step of the dialogue. All moves are associated with either a grammar or an utterance by the respective relations *grammar* and *utterance*. A grammar move acts as user action while an utterance move stands for a system action. The relations *semantics* and *contrarySemantics* define semantics which is set, or unset respectively, when the move is performed.
  An SDO could incorporate the moves *SystemMove_RequestTask*, which is marked as system move by its relation to the utterance 'Hello. What can I do for you?', and *UserMove_RequestTemperature*, which has a relation to the grammar 'What's the (optimal|ideal|best) water temperature to bathe a baby?' and the semantics *Semantics_Temperature*.
- *Belief:* Beliefs are closely related to semantics. However, while semantics is static, a belief is generated during the course of the dialogue when a move with semantics is

enacted. It represents the semantics which is valid in the current dialogue state. If the user move *UserMove_RequestTemperature* is enacted the Belief *Belief_Temperature* that is related to the semantics *Semantics_Temperature* is generated.

- *Beliefspace:* The beliefspace contains all beliefs which are held valid in the current dialogue state. If the user move *UserMove_RequestTemperature* is enacted the Belief *Belief_Temperature* is put in the beliefspace.
- *Agenda:* An agenda represents a system move as well as the user moves that are expected in response to that system move, and possible next agendas.
  An agenda can contain zero or one system moves as well as several user moves. For example, the agenda *Agenda_RequestTask* could be related with *SystemMove_RequestTask* and *UserMove_RequestTemperature*.
  The agendas that can be enacted following the current agenda are defined by the next relation. The agenda *Agenda_InformTemperature* could be related to *Agenda_RequestTask* with the relation *next*.
  In each turn, an agenda is selected and executed. Preconditions which have to be true in order for the agenda to be executed are defined by the relations *requires* and *mustNot*. The semantics associated with an agenda by these relations must (or must not) exist as a belief in the beliefspace in order for the precondition to be considered true. *Agenda_InformTemperature* could be related with *Semantics_Temperature* by *requires*.
- *Workspace:* The workspace contains all agendas that are scheduled for execution. It is not necessary that the preconditions of an agenda are true in order for the agenda to be scheduled. At the beginning of a dialogue, the Agenda *Agenda_RequestTask* could be in the workspace. This would indicate, that the agenda can be executed.

Voice Document

OwlSpeak's View is realized as VoiceXML (Oshry, et al., 2007). This results in OwlSpeak providing VXML-documents as output. Those are dynamically generated from the selected agenda, using the system move's utterance as system prompt and the combined grammars of all user moves as grammar for speech recognition. The names of the user moves are utilised as grammar tag: if the user utterance fits the grammar of a user move, the name of this user move is send as input to OwlSpeak.

VXML-documents can be interpreted by a VoiceXML Interpretation Server, which handles the generation of audio output and the analysis of the user input.

Dialogue Control

The dialogue control logic of OwlSpeak is located in the presenter. This section describes the process of selecting a system move. A concrete example can be found in Section 2.2.

The first agenda to be executed is marked by the flag *masteragenda*. This flag may only be used once per SDO. The *next* relation of this agenda determines the possible first agendas that can be executed by the system. They are the first agendas to be written in the workspace.

The first step of the presenter is to check the preconditions of all agendas in the workspace. An agenda may only be executed if its precondition is true. If more than one agenda is available for execution the decision is based on a priority score. This priority can either be predefined and/or generated dynamically, depending e.g. on the time the agenda has already been in the workspace.

The view is generated based on the selected agenda. The user utterance is mapped to the corresponding user move and this move is the new input for the presenter. The beliefspace is updated in compliance with the *semantics* and *contrarySemantics* of the user move: if the move contains a *semantics*, a corresponding belief is put in the beliefspace, while *contrarySemantics* indicate beliefs that should be removed from the beliefspace. Additionally, the workspace is updated by removing the selected agenda and adding the agendas that are marked as *next*.

2.2 A Use Case Example

The functionality of OwlSpeak is demonstrated using the following example dialogue:

System: Hello. What can I do for you?

User: What's the best water temperature to bathe a baby?

System: The optimal water temperature is 38 degrees.

A SDO has to be defined before any dialogue can be started. This short example dialogue could be generated using the SDO depicted in Figure 3.

The workspace of OwlSpeak initially contains only the agenda *Agenda_RequestTask* as it is marked as starting point. It is the only choice and has no prerequisites, therefore OwlSpeak selects it as next system action. The agendas that are marked as next to *Agenda_RequestTask, Agenda_InformTemperature* and *Agenda_InformHoney,* are put into the workspace.

The system move *SystemMove_RequestTask* is enacted and the corresponding utterance 'Hello. What can I do for you?' is generated as speech by the VoiceXML interpretation server. The grammars of the user moves are used by the speech recognition to identify the enacted user move.

The user enacts the user move *UserMove_RequestTemperature* by saying 'What's the best water temperature to bathe a baby?' The corresponding semantics *Semantics_Temperature* is put into the beliefspace as a belief.

The prerequisites of the agenda *Agenda_InformTemperature* are fulfilled, while *Agenda_InformHoney* cannot be enacted. Therefore, OwlSpeak chooses *Agenda_InformTemperature* as next agenda.

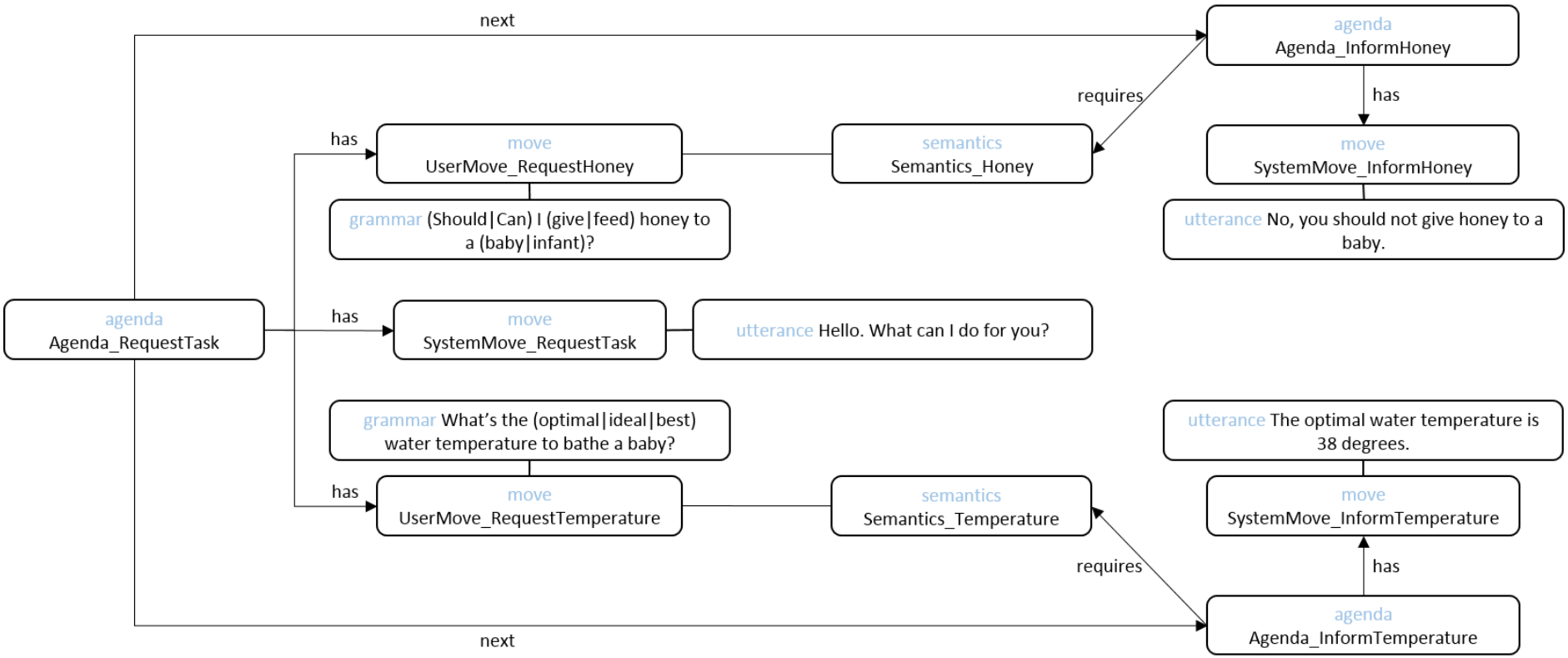

Figure 3: SDO that enables the processing of the example dialogue.

3 ARCHITECTURE OF KRISTINA

While OwlSpeak in its current version is able to manage spoken dialogues by utilising VoiceXML and a VoiceXML interpretation server, changes are necessary to adapt it to the multimodal dialogues required by the KRISTINA system. New components replace the VoiceXML interpretation server and need to be considered for the functionality of OwlSpeak.

In this section, the overall architecture of the KRISTINA system is taken into consideration from a Dialogue Management (DM) point of view. The KRISTINA architecture is described in detail in D7.1 (Stam, et al., 2015). The functional layout depicted in Figure 4 can be found there. A more general overview is given in this section, which serves as the foundation for the considerations regarding the integration of OwlSpeak into the KRISTINA architecture in Section 4.

As can be seen in Figure 4, a Speech-to-Text (STT) component and a Language Analysis (LA) module are utilised on the input side to analyse the user's utterance. These components make use of statistical models to cover a wide range of possible user utterances and thereby enable the user to interact with the system by way of natural speech. In addition, video data is interpreted by face and gesture analysis components and used in combination with an analysis of the audio data to estimate the current emotion of the user.

The turn control component determines when a user move is completed and proceeds to forward the semantic information of the user utterance as well as the detected user emotion to the DM. A sophisticated Knowledge Integration (KI) component provides the DM with knowledge about the domain and the current context. All this information can be utilised by the DM to select the next system move, as will be explained in further detail in Section 4.

On the output side, the Spoken Language Generation (SLG) component is responsible for phrasing the selected system move into an utterance and providing prosody for that. It is not necessary to prepare all possible system utterances in advance, instead new phrasings can be generated dynamically and provided with natural prosody that accounts for the current situation. Therefore, this approach supports a more flexible and situation aware output than prescripted system utterances can provide. Furthermore, an avatar is rendered to present visual feedback. An idle behaviour component provides continuous instructions to the rendering component in addition to the turn based DM. This way, the avatar shows natural behaviour at all times, not only when actively participating in the dialogue by enacting a system move.

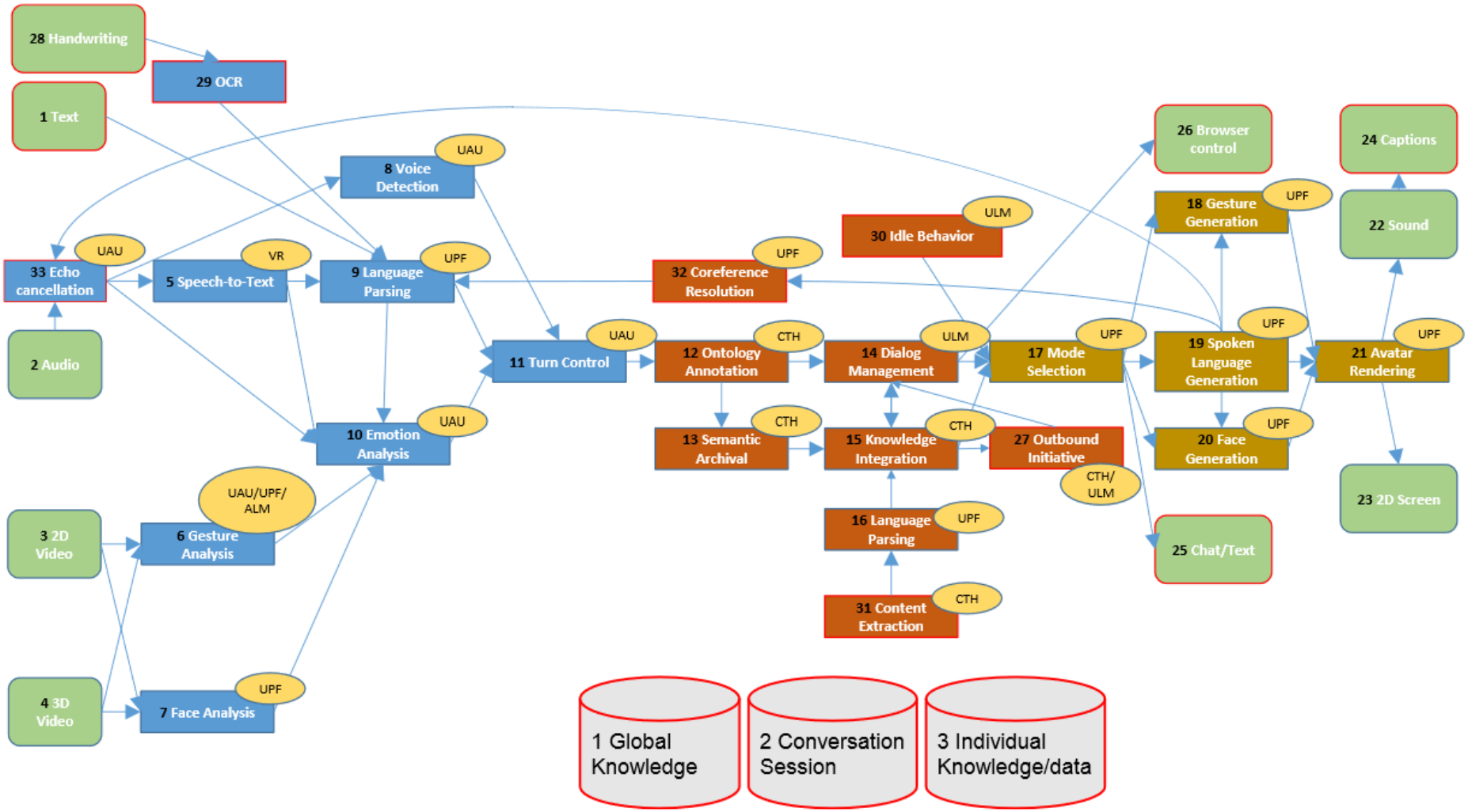

Figure 4: KRISTINA Functional Architecture. (Stam, et al., 2015)

4 INTEGRATION OF OWLSPEAK INTO KRISTINA

With the introduction to OwlSpeak and the KRISTINA architecture in Sections 2 and 3 as foundation, we continue by describing the integration of OwlSpeak into KRISTINA. This section starts by identifying the requirements for integration, and then proceeds to describe the adaptation that have to be implemented in order to fulfil those requirements. Afterwards, the interaction of OwlSpeak with the other KRISTINA components is specified. Finally, the entire process of dialogue management is illustrated using the example dialogue from Section 2.3.

4.1 Integration Requirements

After describing the KRISTINA architecture in Section 3, this section aims at defining the requirements OwlSpeak needs to fulfil in order to be integrated into KRISTINA. Considering the components that are available in the KRISTINA system, six major requirements for the integration of OwlSpeak can be identified:

- OwlSpeak in its original architecture interacted with a VoiceXML interpretation server using utterances and grammars, while the interactions in KRISTINA are based on semantics. Therefore, the interface of OwlSpeak needs to be adjusted to support *semantic interactions with other components*.
- Neither the STT nor the LA components are restricted regarding the possible user input. It is therefore unnecessary to provide prescripted grammars that are associated with user moves to define what the user can say at a given time. It would even constrict the potential flexibility of the system to do so. This entails that the association of a grammar with a specific user move can no longer be utilised to identify the enacted user move. A new mechanism that enables the *classification of a user move* without relying on grammars needs to be established.
- Having components that are able to determine the user emotion provides the system with useful additional information. It can help to detect misunderstandings early and thereby making the system more robust: If the user says angrily ‘I am mad.’ but the system understands ‘I am glad.’ the detected emotion stands in contrast to the sematic content of the utterance and indicates a misunderstanding. Moreover, the system can react to the user more empathically: For example, it might refrain from correcting minor mistakes of the user while he is angry or show compassion if he is sad. Therefore, the *user emotion should be incorporated in the decision making process* of OwlSpeak in order to use this information to its full capacity.
- The KRISTINA system incorporates a sophisticated KI. *Some tasks of the SDO should be assumed by the KI ontology* in order to separate the representation of the domain state from the representation of the dialogue state. This allows both components to focus on their respective area of expertise. Furthermore, a redundant representation of data can be avoided by this separation.
- Predefined system moves restrict the possibility of integrating content of the ontology dynamically. Hence, *more general features of system moves* (e.g. *inform* move with topic *medicine* instead of the concrete *SystemMove_InformTakeAspirin*) should be considered in the decision process in order to utilise the KI component to its full extent.

- As output the DM should *provide a system emotion* in addition to the selected system move. This will enable the SLG to add appropriate prosody to the system utterance and the Face and Gesture Generation components to animate the avatar accordingly.

While the focus of this document is the integration of OwlSpeak into the KRISTINA architecture, we also briefly consider the requirements to OwlSpeak derived from the use cases in D2.1 (Ultes & Pragst, 2015):

- *Adaptivity to both emotion and culture* is desirable to provide a trustworthy communication partner to the user of the KRISTINA system. This emphasises the need to consider emotion in the decision making process and adds the cultural aspect as additional influence to this process.
- *System emotions* have to be generated in order to make the avatar more relatable and human-like.
- Moreover, the large scope of the covered dialogue domain further supports the utilisation of the *KI ontology as external knowledge base* instead of modelling the domain directly in the DM.

Overall, the requirements derived from the use cases are closely related to the requirements of the integration. They further emphasise the importance of the intended changes to OwlSpeak.

4.2 New Architecture of OwlSpeak

The components of the KRISTINA architecture yield the potential for a higher degree of flexibility in regards to possible user utterances and system responses than the VoiceXML interpretation server currently employed by OwlSpeak. Furthermore, the interaction can be enriched by the utilisation of emotions of both user and system. Changes to OwlSpeak are clearly necessary in order to preserve this flexibility and make best use of all the provided modalities.

A conception of the extended functionality of OwlSpeak has been presented by Pragst et al. (Pragst, et al., 2015). Here, we concretise this conception. The revised architecture continues to follow the model-view-presenter pattern. However, the modules themselves have to be modified.

Spoken Dialogue Ontology

The SDO is modified as some components are no longer needed in the new architecture and others are relocated to the KI ontology to enable a more specialised task handling and prevent redundant data storage. The new SDO is depicted in Figure 5. A list of all previous components of the SDO and their role in the new architecture follows:

- *Grammar:* The grammar concept is unnecessary in the new architecture as the speech to text component does not rely on grammars. The new architecture does therefore not incorporate grammars.
- *Utterance:* Prescripted utterances are no longer necessary as the SLG is responsible for phrasing the information of the system move into coherent sentences. Hence, the utterance concept is not part of any ontology.

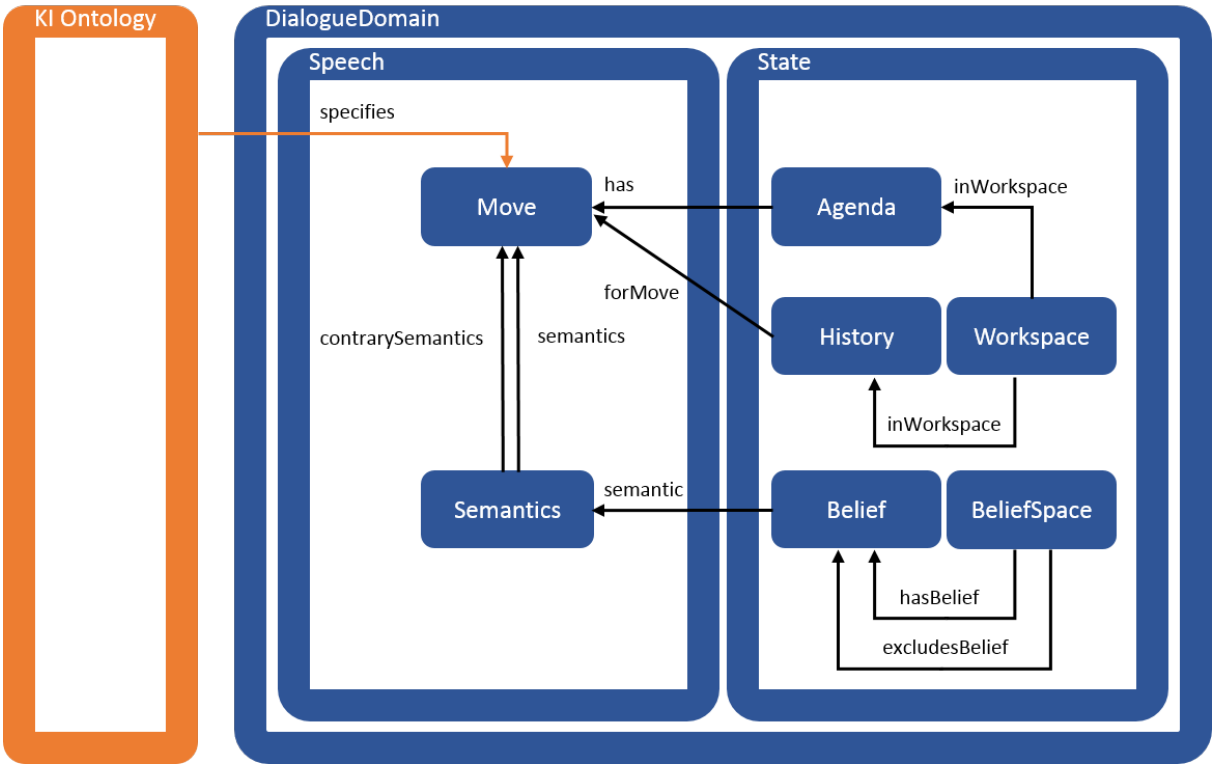

Figure 5: Schematic representation of the new Spoken Dialogue Ontology.

- *Semantics:* The meaning of a user move is no longer prescripted. Instead it is derived directly from the language analysis of the user utterance and the extracted information is stored in the ontology of the KI. However, semantics can not only be used to describe the meaning of a user move, but also for dialogue related tasks, e.g. how often a specific request has already been made. Therefore, semantics are still part of the SDO, but no longer used to define the semantic meaning of a move.
- *Move:* Modelling each singular user and system move (e.g. *UserMove_RequestTemperature*) is no longer feasible considering the potential flexibility of the interpretation of user input as well as the rich information of the KI ontology. Instead, moves contained in the SDO will be more general. We consider existing classification schemes for dialogue acts, such as DAMSL (Core & Allen, 1997), DiAML (Bunt, et al., 2012), or DIT++ (Bunt, 2009), to generate a list of general dialogue acts suitable for the use in KRISTINA. Such general dialogue acts might include:
  - Request
  - Inform
  - Confirm
  - Affirm
  - Deny
  - Acknowledge
  - Comment

  Additionally, purely emotional dialogue acts can be defined, e.g.:
  - Calm down
  - Cheer up
  - Console

In the course of the dialogue, these moves are instantiated with semantic content: User utterances are analysed by the LA and labeled as one of the basic dialogue acts *request* and *inform*. The DM specifies this label using the dialogue history. The semantic content of a user move is provided by the analysis of the KI.
Furthermore, at each turn the KI component indicates information that is missing in the ontology and should be requested from the user, and/or information that should be provided to the user. Missing information is used to create a *request* system move with the respective semantic content, while provided information is used as semantic content of an *inform* system move.

- *Belief/Beliefspace:* The current world state is no longer represented in the SDO but in the KI ontology instead. This means that the KI ontology replaces the beliefspace where general knowledge is concerned. Facts in the KI ontology resemble the beliefs of the original architecture of OwlSpeak. The original beliefspace and the corresponding beliefs are still part of the SDO, but are used exclusively for dialogue related information, e.g. how often a specific request has already been made.
- *Agenda:* Agendas remain a part of the SDO. User moves that are associated with a specific agenda might be considered 'expected' answers and be used to detect misunderstandings. They are however not the only viable user move anymore. Whenever the DM creates new system moves based on the feedback of the KI, it also creates corresponding agendas and places them in the workspace.
- *Workspace:* The workspace is still part of the SDO. The DM is responsible for adding new agendas and selecting the next agenda from the workspace.
- *History:* A History of the enacted user and system moves as well as the user and system emotion is maintained by OwlSpeak. This is realised as a temporal sequence of the respective moves and Valenca-Arousal (VA) values.

Semantic Content/Valence-Arousal

OwlSpeak utilised VoiceXML as view in its original architecture. A VoiceXML document defines a sequence of system utterances and corresponding user responds in form of grammars. It is not well suited for the tasks presented in KRISTINA, e.g. representing the semantic content of an arbitrary user utterance on the input side. A view based on semantics rather than speech is required. A possible solution for representing semantic information is RDF. It is commonly used in Semantic Web and knowledge management applications, and well suited for the tasks of the KRISTINA system. Therefore, the partners agreed on RDF/XML (Gandon & Schreiber, 2014) as exchange format between the LA, DM, KI, and SLG. The new architecture of OwlSpeak utilises RDF/XML to represent the semantic content of the user as well as the system move. Additionally, the user and system emotion is given as Valence and Arousal (Russell, 1980).

Dialogue Control

The dialogue logic controlled by the presenter has to be adapted to the new information available. This section describes the new decision making process of OwlSpeak. This process is further illustrated using an example dialogue in Section 4.3.

Two modes of functionality can be differentiated: a system and a user initiated dialogue. In the user initiated dialogue, the interaction is started by a user utterance. The user utterance

is labelled by the LA with one of the dialogue acts *request* or *inform*. These dialogue acts can be refined by the DM by considering the dialogue history. The results of the LA are forwarded to the KI component in order to perform the domain state update. At the same time, the DM updates the dynamic part of its SDO. The KI component then provides a list containing either information missing from the ontology or information that can be provided to the user. The DM creates agendas for these suggestions by using request moves for missing information and inform moves for provided information, and puts the generated agendas in the workspace.

The DM selects an agenda from the workspace based on a priority score. This score can either be predefined and/or dynamically generated based on the dialogue state, the user emotion, the user culture or other relevant factors. The assignment of priority scores to dynamically generated system moves is accomplished by taking into account the object hierarchy of the ontology. In order to find a good dialogue strategy, we will test rule based approaches based on findings in communication sciences (e.g. (Feghali, 1997) for adaptation to culture) as well as machine learning approaches for the dynamic generation of priority scores.

In addition, a system emotion is generated considering the enacted user move, the user emotion and the system culture. The appraisal theory (Arnold, 1960) is utilised to determine the impact of the user move and emotion on the system emotion. For example, the system's valence might decrease if the user informs the system that he is ill and a low valence is detected. The system's emotional reaction to such events is influenced by the simulated culture of the system.

The system emotion and the selected system move might influence each other. Both are provided as output and the DM waits for the next user move.

If the interaction is initiated by the system, the DM starts by requesting the required data for one specific aspect, e.g. biographical data, from the KI. The KI then returns a list of biographical properties that are still missing in the ontology and can be requested from the user. The DM chooses one of the properties based on the culture and emotion of the user, and schedules a request for this property. The user's response is then forwarded to the KI, which updates its ontology accordingly and sends a list of further properties that can be requested in case there are new properties available by adding the new information.

4.3 Interaction of OwlSpeak with other Components

This section focuses on the interaction between OwlSpeak and other components of the KRISTINA system. Four components have been identified to have a direct interaction with the DM as can be seen in Figure 6.

The semantic input is provided by the LA, while the user emotion is produced by the Emotion Analysis (EA), which uses audio features as well as facial expression and a gesture analysis to obtain a robust estimation of VA. Both are fed to Visual SceneMaker, a shell for OwlSpeak that enhances dialogue management by handling idle behaviour and turn taking. The decision process of OwlSpeak is supported by its interaction with the KI module. The resulting semantic system move and the system emotion are distributed through Visual SceneMaker to the components responsible for producing the video and audio output of the KRISTINA system.

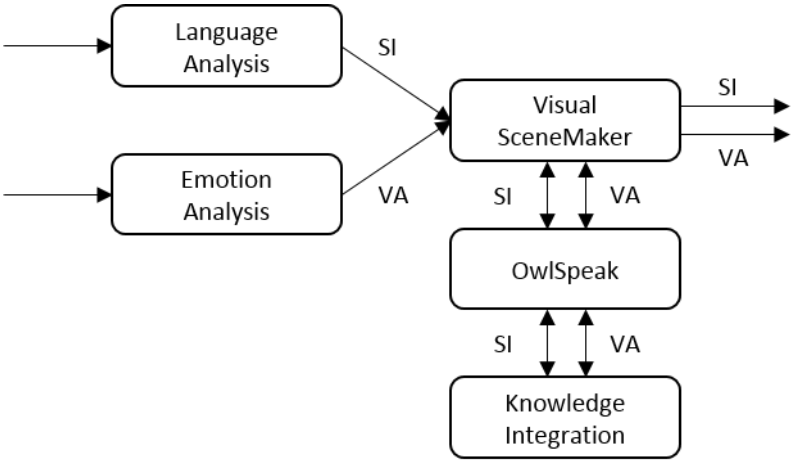

Figure 6: The interaction of OwlSpeak with other KRISTINA components. Semantic information (SI) as well as valence and arousal (VA) are transmitted between the components.

The interaction between OwlSpeak and the connected KRISTINA components is described in the following.

Language Analysis

While the LA is not directly interacting with OwlSpeak, it provides important information for the DM: it is responsible for labelling the user move with one of the basic dialogue acts *request* and *inform* and thereby provides crucial information for the decision making process. It supports classifying the user utterance in terms of the on-going dialogue.

Visual SceneMaker

Visual SceneMaker is a DM-shell into which OwlSpeak is embedded. It is the main interface between OwlSpeak and the remaining KRISTINA components. It is responsible for both turn control and idle behaviour, and serves as OwlSpeak's connection to the input as well as the output side.

This interaction between OwlSpeak and Visual SceneMaker is realised as a REST service offered to Visual SceneMaker by OwlSpeak, and is therefore unidirectional. Visual SceneMaker forwards the semantic user move as determined by the LA and the VA by the EA to OwlSpeak via POST request. As answer it receives an RDF representation of the semantic system move and the system emotion as VA. Visual SceneMaker forwards this information to the Mode Selection component.

While OwlSpeak is responsible for turn based reactions to the user input, Visual SceneMaker provides continuous idle behaviour as output for the avatar animation during the time in which no system move is enacted. Furthermore, it monitors the turns of system and user and triggers OwlSpeak when a system move is needed.

Knowledge Integration

The interaction between OwlSpeak and the KI component is very close. Apart from Visual SceneMaker, KI is the only component with direct interaction to OwlSpeak. Moreover, it partly replaces the SDO, which is an integral part of the original architecture of OwlSpeak. The KI provides OwlSpeak with all needed information regarding the dialogue domain.

The interaction between OwlSpeak and the KI is realised as a REST service offered to OwlSpeak by the KI, and is therefore unidirectional. OwlSpeak can issue either a GET or a POST request, depending on whether the user or the system initiates the dialogue.

Using a GET request, OwlSpeak can query information – such as missing user information regarding biographical data – from the ontology. This information is represented in RDF format. OwlSpeak can utilise the received information to generate, e.g., *request* moves.

With a POST request, OwlSpeak forwards the results of the LA and the EA to the KI component for storage in the ontology. The KI component further analyses the received data and returns a list of information missing in the ontology that should be requested from the user, and/or information from the ontology that can be given to the user. This is represented in RDF format. OwlSpeak can create *request* or *inform* moves from the provided information and provide these moves as output.

4.4 A Use Case Example

In this section, the example dialogue from Section 2.3 is discussed again considering the proposed new functionality of OwlSpeak. The example dialogue has been described as the following:

System: Hello. What can I do for you?

User: What's the best water temperature to bathe a baby?

System: The optimal water temperature is 38 degrees.

As before, there needs to be a foundation before this dialogue can be processed. The KI ontology needs to incorporate all the necessary concepts, such as there exists something called water, it can have a temperature, persons can bathe in it etc. Furthermore, the SDO needs to include the basic moves *inform* and *request*, as well as an agenda *Agenda_RequestTask* that is marked as starting point.

The DM would begin by selecting *Agenda_RequestTask* from the workspace and forwarding it to the Mode Selection component. In addition, a starting point emotion, such as valence 0.25, arousal 0 to indicate a balanced and friendly behaviour, would be send. After the system has enacted the system move, the user can answer with 'What's the best water temperature to bathe a baby?' The LA determines that this is a *request* move and encodes the contained information into an RDF structure. The EA detects a valence of -0.25 and an arousal of 0.5, which might indicate impatience. Both results are sent to the DM. The DM decides to keep the *request* label as the dialogue history does not contain any moves yet, and then stores the user request and the emotion in the history. As next step, the information is forwarded to the KI. The KI component determines that the exact age of the baby might be useful, but finds an answer for the general question nonetheless. It sends a list to the DM containing the information 'missing information: age' and 'result: 38 degrees'. The DM creates corresponding agendas, puts them into the workspace and selects the *inform* move as users with high arousal often prefer fast answers over lengthy conversations. Furthermore, it adjusts the system emotion to be more serious with a valence of 0 and an arousal of 0, and sends emotion and system move to the Mode Selection component.

5 CONCLUSIONS

This document described the changes in OwlSpeak that are necessary to integrate it into the KRISTINA system. It was established that the original OwlSpeak did not fit the KRISTINA requirements in mainly two regards: multimodality and flexibility.

Multimodality requires that OwlSpeak can utilise user emotions in its decision making process and provide a system emotion as output. Furthermore, it is necessary to provide continuous feedback for the animation of the avatar in addition to turn based feedback. OwlSpeak is used in combination with Visual SceneMaker to enable this continuous role-based behaviour.

The STT and LA components provide a lot of flexibility regarding the possible user moves as they do not rely on prescripted grammars. Prescripted user moves would limit this potential. Similarly, the sophisticated KI component would not be able to be employed to its full potential with predefined system moves. Hence, OwlSpeak utilises more general user and system moves and integrates the KI component closely in the dialogue management process to benefit from its reasoning capabilities.

6 REFERENCES

Antoniou, G. & Van Harmelen, F., 2004. Web ontology language: Owl. In: *Handbook on ontologies.* Berlin Heidelberg: Springer, pp. 67-92.

Arnold, M. B., 1960. *Emotion and Personality. Vol. 1, Psychological Aspects.* New York: Columbia University Press.

Bunt, H., 2009. The DIT++ taxonomy for functional dialogue markup. *AAMAS 2009 Workshop, Towards a Standard Markup Language for Embodied Dialogue Acts,* pp. 13--24.

Bunt, H. et al., 2012. ISO 24617-2: A semantically-based standard for dialogue annotation. *LREC,* pp. 430-437.

Core, M. G. & Allen, J., 1997. Coding dialogs with the DAMSL annotation scheme. *AAAI fall symposium on communicative action in humans and machines,* pp. 28-35.

Feghali, E., 1997. Arab cultural communication patterns. *International Journal of Intercultural Relations,* 21(3), pp. 345-378.

Gandon, F. & Schreiber, G., 2014. *RDF 1.1 XML Syntax: W3C Recommendation, 25 February 2014.* [Online] Available at: https://www.w3.org/TR/rdf-syntax-grammar/ [Accessed 24 02 2016].

Heinroth, T., Denich, D. & Schmitt, A., 2010. Owlspeak - adaptive spoken dialogue within intelligent environments. *8th IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops),* pp. 666-671.

Oshry, M. et al., 2007. Voice extensible markup language (voicexml) 2.1. *W3C Recommendation.*

Pragst, L., Ultes, S., Kraus, M. & Minker, W., 2015. Adaptive Dialogue Management in the KRISTINA Project for Multicultural Health Care Applications. *SEMDIAL 2015 goDIAL,* p. 202.

Russell, J. A., 1980. A circumplex model of affect. *Journal of Personality and Social Psychology,* Volume 39, pp. 1161-1178.

Stam, A. et al., 2015. *D7.1 Roadmap for the development of the KRISTINA agend,* EU Horizon 2020: KRISTINA.

Ultes, S. & Minker, W., 2014. Managing adaptive spoken dialogue for intelligent environments. *Journal of Ambient Intelligence and Smart Environments,* 6(5), pp. 523-539.

Ultes, S. & Pragst, L., 2015. *D2.1 Requirements for Dialogue Management in adaptive Human-Machine-Communication,* EU Horizon 2020: KRISTINA.