



Universität Karlsruhe (TH)

- Institut für Theoretische Informatik (IThI) -

Interactive Systems Lab

Conceptualization and Development of a  
Conversational, Mediating and Animated Receptionist:

COSIMAR

Diploma Thesis  
of  
Kerstin Dittmar

Supervised by  
Prof. Dr. Alexander Waibel  
Dr. Matthias Kaiser (SAP)  
Dipl.-Inform. Hartwig Holzapfel

THE BEST-RUN BUSINESSES RUN SAP





## DECLARATION

Herewith I confirm that I independently prepared this thesis. No further or auxiliary means as the ones indicated in this work were used for the preparation.

June 25, 2005

A handwritten signature in blue ink that reads "Kerstin Dittmar". The signature is written in a cursive style with a long horizontal stroke at the end.

Kerstin Dittmar





## **ABSTRACT**

A wide variety of complex and powerful software applications surrounds us in our day-to-day business. More and more people use software applications, and everyone has different skills, needs and interests in using specific software, but not everyone wants to get expertise on how to use those applications.

While looking for a short scenario realizable in a prototype which covers the purpose of making user interfaces more user friendly and additionally support SAPs or customers business processes, we came up with the idea of the Conversational, Mediating and Animated Receptionist (COSIMAR), which combines different modalities to enable people carrying out dialogs to benefit from the service. It concentrates on the most common way people communicate: natural language in combination with the elemental communication modality speech. The major characteristics of this virtual receptionist are multimodal capabilities, including virtual face presentation with facial expressions, sensing potential communication partners, the ability to conduct a dialog with the visitor using speech input and output, and moreover to embed another dialog with a third party. Accordingly, a non-expert user interface is built, which can understand the user's intention and proactively recommend, guide and explain the next steps in a natural way. This work will present the development, realization and evaluation of such a system.

## **ACKNOWLEDGEMENT**

This work was developed in collaboration with the HCI Research group at SAP in Palo Alto and the Interactive Systems Laboratories (ISL) at the University of Karlsruhe (TH).

I want to thank both the HCI group in Palo Alto and Prof. Alex Waibel for making the combination of internship and research work possible.

Special thank goes to my supervisors Matthias Kaiser at SAP and Hartwig Holzapfel from the University of Karlsruhe, who supported me in all respects, kept an eye on the progress of my work and always have been available when I needed their advise.

I also want to thank my colleagues from the Research Department for all their help, interest and valuable hints. Especially I want to thank Frankie James, Townsend Duong, Samir Raiyani, Rama Gurram and Sally Lawler for sharing their knowledge in their research areas with me. Thank also goes to my team colleagues Christine Müller and Andriy Sobol for the fruitful discussions, and to Lars Dittmann and Kai Sachs for helping me in tricky questions. Furthermore, I want to thank the receptionists who allowed me taking a look over their shoulders, and all who took the time for helping evaluating the system.

I owe special thank to my friend Alexander Sofalvi and to my family, who encouraged and motivated me continuing my work in hard times.

# CONTENTS

<b>1</b>	<b>Introduction.....</b>	<b>1</b>
1.1	Motivation.....	1
1.2	Scope of COSIMAR .....	3
1.2.1	Benefits of COSIMAR .....	3
1.2.2	Field of Application.....	3
1.2.3	Realization .....	5
1.3	Introduction to the Chapters .....	6
<b>2</b>	<b>Requirement Analysis .....</b>	<b>7</b>
2.1	Business Process.....	7
2.2	Business Area .....	9
2.3	System Use Cases.....	12
2.3.1	Overview .....	12
2.3.2	Mediate Customer.....	12
2.3.3	Call & Talk to Person .....	17
2.3.4	Greet .....	20
2.4	Non – Functional Requirements .....	21
<b>3</b>	<b>Related Work.....</b>	<b>23</b>
3.1	University of Karlsruhe – Architecture for an Emotion-Sensitive Dialog System .....	23
3.2	Projects at MIT .....	25
3.2.1	Gandalf .....	25
3.2.2	REA.....	26
3.2.3	MACK.....	27
3.3	EU project MagiCster .....	29
3.4	University of Rochester – An Architecture for More Realistic Conversational Systems (TRIPS-System) .....	31
3.5	University of Reading – Queue-Based Agent Architecture for Multimodal Interfaces .....	33
3.6	Related Work in Comparison to the Requirements .....	34



<b>4</b>	<b>Technical Bases .....</b>	<b>37</b>
4.1	Speech Recognition and Text-to-Speech .....	37
4.2	Call System .....	39
4.3	Natural Language Understanding and Dialog Management .....	40
4.3.1	Domain dependent knowledge .....	40
4.3.2	Dialog State .....	43
4.4	Social Agents FATA .....	44
<b>5</b>	<b>Architecture .....</b>	<b>47</b>
5.1	Components of the Virtual Receptionist.....	47
5.1.1	Virtual Receptionist Control .....	48
5.1.2	Customer-Receptionist Component.....	49
5.1.3	Employee-Receptionist Component .....	50
5.2	Application Layering .....	51
5.3	Interaction and Process Flow .....	52
<b>6</b>	<b>Design and Implementation.....</b>	<b>55</b>
6.1	Virtual Receptionist Control .....	55
6.2	Basic Services.....	56
6.2.1	Social Agent.....	58
6.2.2	Speech Recognition Application .....	61
6.2.3	Dialog Management for the Customer and Employee Interaction.....	62
6.2.4	Telephony Application .....	65
<b>7</b>	<b>User Tests .....</b>	<b>67</b>
7.1	Differences between prestudy and main study .....	67
7.2	Evaluation of the User Studies .....	68
7.2.1	Evaluation of the Kiosk Interaction .....	68
7.2.2	Evaluation for Telephone Interaction .....	78
7.3	Conclusions drawn from the User Test Evaluation .....	82
<b>8</b>	<b>Conclusion and Outlook .....</b>	<b>83</b>
8.1	Conclusion .....	83
8.2	Results .....	84

8.3	Future Work .....	85
8.3.1	Scenarios .....	85
8.3.2	Research .....	85
<b>9</b>	<b>Literature and Abbreviation .....</b>	<b>87</b>
9.1	Literature .....	87
9.2	Abbreviations und Definitions .....	92
<b>10</b>	<b>Appendix .....</b>	<b>95</b>
10.1	Questionnaire User Test I .....	95
10.2	Questionnaire User Test II .....	103

## TABLE OF FIGURES

FIGURE 1: Business Process: Reception.....	7
FIGURE 2: Business Area: Reception reduced with regard to the requirements of the Virtual Receptionist.....	9
FIGURE 3: System use cases of the business use case: Mediate Customer .....	12
FIGURE 4: Mediation Object .....	15
FIGURE 5: System Activity Diagram - Mediates Customer.....	16
FIGURE 6: MediationResult Object .....	18
FIGURE 7: System Activity Diagram: Call & Talk to Person .....	19
FIGURE 8: System Activity Diagram: Greet.....	20
FIGURE 9: Data Flow of an Emotion-Sensitive Dialog System [Ho03] .....	23
FIGURE 10: Ymir Architecture [Th99 S.12] .....	25
FIGURE 11: Architecture of REA [CC02].....	27
FIGURE 12: Architecture of MACK [CS02] .....	28
FIGURE 13: Architecture of MagiCster [CC02].....	29
FIGURE 14: Architecture of TRIPS-System [AG01] .....	31
FIGURE 15: Queue-Based Multi-Agent Architecture [KW01].....	33
FIGURE 16: SpeechKit Architecture for Speech Recognition and Speech Synthesis [Ch96].....	38
FIGURE 17: SpeechStudio Architecture for Speech Recognition, Speech Synthesis and Telephony .....	39
FIGURE 18: Knowledge Sources of the Dialog Manager TAPAS .....	40
FIGURE 19: Typed Feature Structure .....	42
FIGURE 20: Application Tier Architecture [DY03].....	45
FIGURE 21: UML-Component-Diagram of COSIMAR .....	48
FIGURE 22: Application Layering .....	51
FIGURE 23: Sequence Diagram of the Components.....	52
FIGURE 24: UML Class Diagram of the Virtual Receptionist Control.....	55
FIGURE 25: UML Class Diagram of Basic Services .....	57
FIGURE 26: COSIMARs Graphical User Interface .....	58
FIGURE 27: UML Class Diagram Speech Recognition.....	61
FIGURE 28: UML Class Diagram Telephony.....	65
FIGURE 29: Establishment of the Mediation of the different scenarios .....	68
FIGURE 30: Amount of time required to complete the task (distribution and average).....	70
FIGURE 31: Look and Feel (Test 1) of COSIMAR.....	71
FIGURE 32: Application in General (Test 2).....	71
FIGURE 33: Speech Recognition of COSIMAR.....	71
FIGURE 34: COSIMARs Amount of Understanding .....	72
FIGURE 35: COSIMARs response time to your utterance.....	74
FIGURE 36: COSIMARs voice .....	74
FIGURE 37: COSIMARs Facial Expressions .....	75

FIGURE 38: Using the solution was a satisfying experience.....	75
FIGURE 39: Interaction with the system was natural .....	76
FIGURE 40: Ease to learn the interaction.....	76
FIGURE 41: Usage of the Virtual Receptionist .....	77
FIGURE 42: I think COSIMAR supports the mediation .....	77
FIGURE 43: Establishment of the Mediation .....	78
FIGURE 44: Amount of time required to complete the task .....	79
FIGURE 45: Speech Recognition .....	79
FIGURE 46: COSIMARs Response Time to Your Utterance .....	80
FIGURE 47: COSIMARs Voice.....	80
FIGURE 48: Interaction with the system.....	80
FIGURE 49: I think the telephone interaction supports the mediation of visitor and employee .....	81



## LIST OF TABLES

TABLE 1: Overview about the System Use Cases of the Virtual Receptionist ...	12
TABLE 2: Comparison Overview .....	34
TABLE 3: Summary of the Components .....	51
TABLE 4: Snippet of VR_veeper.jsp.....	59
TABLE 5: Snippet of VR_speechInput.jsp .....	59
TABLE 6: Dialog Goals of Mediation .....	63
TABLE 7: Dialog Moves of Mediation .....	64
TABLE 8: Example Dialog 1 .....	73
TABLE 9: Example Dialog 2 .....	73



# 1 Introduction

## 1.1 Motivation

A wide variety of complex and powerful software applications surrounds us in our day-to-day business. More and more people use software applications, and everyone has different skills, needs and interests in using specific software, but not everyone wants to get expertise on how to use those applications. So the big challenge in the field of human-computer interaction consists in satisfying all those individual user requirements. In other words, we need to make user interfaces more user friendly and thus scale up accessibility and usability.

The motto of usability experts is, that "software must be accessible to be usable, and it must be usable to achieve user goals successfully" [KD05]. A software product is then accessible, if the user is aware of his actual situation and of the options which are available to change this situation. Additionally, he needs to have the executive capabilities to perform those actions. Usability means that the user can reach his context specific goals in an effective, efficient, and satisfying way.

For compensating the user's disfavor reading a comprehensive handbook before they can use an application, we need to reduce the technical training. In addition, users must be supported according to their needs and expertise. A novice user for example needs more guidance through an application workflow than an expert. Furthermore, interfaces should be easily adaptable to optimize accessibility and usability. To make an interface more user friendly, also the graphical user interface needs to be transformed to a social and enjoyable experience.

SAP AG, as the provider of innovative and collaborative business solutions for all types of industries, goes towards the future by performing research in this field. In SAP Labs Palo Alto, the department of Human Computer Interaction (HCI) is engaged in research of new interface technologies and in optimizing customer experience through research in human computer interaction.

While looking for a short scenario realizable in a prototype which covers those purposes and additionally support SAPs or customers business processes, we came up with the idea of the Conversational, Mediating and Animated Receptionist (COSIMAR), which combines different modalities to enable people carrying out dialogs to benefit from the service. It concentrates on the most common way people communicate: natural language in combination with the elemental communication modality speech. The major characteristics of this virtual

receptionist are several hypermedia and multimodal capabilities, including virtual face presentation with facial expressions, sensing potential communication partners, providing several references, the ability to conduct a dialog with the visitor using speech input and output and moreover to embed another dialog with a third party. Accordingly, a non-expert user interface is built, which can understand the user's intention and proactively recommend, guide and explain the next steps in a natural way.

This diploma thesis will focus on the architecture of such a system, which covers the coordination of the involved components and the information transfer between them. The design of the architecture includes a generic design, so that the visual representation, the external systems, the dialog strategy, the topic of the dialog and the relating components can be chosen depending on the thematic specification. Just as much attention will be given to the user studies, which play an important role in evaluating the system regarding usability and accessibility.

On the one hand, the diploma thesis will be part of the research project of SAP and on the other hand it will be written in association with the current research at the department Interactive Systems Laboratories (ISL) at the University of Karlsruhe (TH).

ISL focuses on the main research areas of speech and multimodality that helps to improve user interfaces with regard to human-computer and human-human communication. In the area of speech, ISL explores speech recognition, machine translation, speech understanding, dialog management, discourse analysis and multilingual speech processing. Within multimodality, ISL concentrates on person tracking, person recognition, gaze tracking, gesture, pointing, sign translation and lip-reading.

Both SAP and ISL will benefit from the combination of the two different expertises – economic and educational knowledge – and furthermore, in the near future, a usable and accessible software product which provides a natural language user interface could be offered to the user.



## **1.2 Scope of COSIMAR**

COSIMAR combines different modalities to enable people using an application with natural language. The major characteristics of the virtual receptionist are several capabilities which include virtual face presentation with facial expressions, sensing potential communication partners, the ability to carry out a dialog with the visitor using speech input and output and moreover, embedding another dialog with a third party. Thus, COSIMAR not only acts as a communication partner to retrieve and provide information, but also as a communication mediator.

### **1.2.1 Benefits of COSIMAR**

The motivation in developing a virtual receptionist at SAP is founded on economic, usability and "state-of-the-art" purposes.

Typically, information regarding a meeting or orientation in a building or information in general about the company, sought by customers, is retrieved from human receptionists. But up to now, a twenty-four-seven availability was compensated in favor of the high costs of ownership. Thus, the virtual receptionist is planned to support the conventional reception for providing a round-the-clock availability with regard to a high economy.

Another benefit using a virtual receptionist, which provides information in a natural manner, is the enhanced usability and accessibility. People who are interested in company information don't want to study a comprehensive handbook to get to know how to use a virtual receptionist. They just want to use the system to achieve their goals effectively, efficiently and in a satisfying way, but without any training and waste of time.

More advantages for SAP using a virtual receptionist are justified in customers' acquisition. On the one hand, COSIMAR can attract customers and satisfy their needs by providing information, and on the other hand SAP presents itself as an innovative and future-oriented software company, which is the best business partner to collaborate with.

### **1.2.2 Field of Application**

At SAP it is planned to use the virtual receptionist in a calm lounge. When a customer approaches the virtual receptionist, it is able to track him via infrared sensor, and thereupon it welcomes the visitor at a reception site. Given that the receptionist is the first point of contact to the company, it must be primarily designed to appear friendly and pleasingly, and particularly it must give helpful information. COSIMAR is able to engage the user proactively in a dialog and to

respond reactively if the user speaks to it. With respect to the customer needs, there are several tasks that the virtual receptionist has to fulfill.

The main task of the virtual receptionist comprises locating specified persons or meetings a visitor seeks at SAP. To establish the contact between the third party and the visitor, the virtual receptionist needs to call the required person, so that there is a dialog between COSIMAR and the employee. The following example should give you an impression of this scenario:

COSIMAR: Welcome to the SAP Research Center. I am the virtual receptionist COSIMAR. If you want to talk to me, please use the headphone in front of you.

Visitor: Hello.

COSIMAR: Hello. Who are you here to see today?

Visitor: I have a meeting with Charlie Brown.

COSIMAR: What is your name?

Visitor: Mr. Green.

COSIMAR: Okay, I will call your contact person, Charlie Brown. Please be patient.

The virtual receptionist starts an application that calls Mr. Brown, so that there is a dialog between Mr. Brown and the COSIMAR:

Mr. Brown: Hello.

COSIMAR: Hello, this is the virtual receptionist COSIMAR. Mr. Green is waiting for you in the lounge. When do you come to pick him up?

Mr. Brown: In a few minutes.

COSIMAR: Okay, I will tell your meeting party, that you will come in a few minutes. Good bye.

COSIMAR: Mr. Green, I have finished the phone call. Your meeting party will come in a few minutes. He will meet you at the reception. It was a pleasure to help you.

Visitor: Thank you.

COSIMAR: You are welcome.

Beyond the scope of diploma thesis, future enhancements of the virtual receptionist could be the ability to answer questions regarding the orientation in the building or helping to locate the facilities. In addition, the virtual receptionist could know about transportations and directions in the surrounding area. This includes giving the fastest way to the airport, explaining important routes and providing telephone numbers of public transportation, taxi and car services. But if the conversation becomes too complex, the virtual receptionist will be able to contact a person who will take over.



Deriving from the field of application, there are several different tasks the virtual receptionist has assigned. Hence, to allow changing or extending the application, also depending on new thematic specifications, the architectural design of COSIMAR needs to be open and flexible.

### 1.2.3 Realization

To realize the virtual receptionist, different components from several research areas have to cooperate with each other. The main components are speech recognition, dialog management – involving language understanding, response generation and communication with knowledge sources and other components –, speech output and visual representation with an appropriate animation.

The speech recognition component converts the speech input to a string of words. There are several requirements relating to this component. Since there are several customers, it must be speaker independent. Supporting continuous speech and spontaneous conversational speech is also an important fact, because a familiar communication is aspired.

The dialog management component controls the interaction between the system and the user, including the coordination of the other components of the system. To understand the user's utterance it is not only necessary to get the string of words from the speech recognizer or from the keyboard, but rather a meaningful knowledge representation which is generated by the language understanding component. Through the usage of knowledge sources and an external service, the dialog manager generates a response output string passed to the speech output component. The Virtual Receptionist requires the following knowledge sources:

- a database: storing information about employees
- a task record model: representing the information which has been disclosed by the user

Moreover it is desired that the dialog manager compensates ill-formed or incomplete input to support the speech recognition engine.

As external service an application that realizes the phone call between the Agent and the third party is needed.

There are two different ways, how the speech output can be realized, either with prerecorded speech or with text-to-speech synthesis. Because the answers will be composed of different phrases a text-to-speech synthesis will be applied. The main attention of this component rests on a preferably natural speech output with a consistent voice.

In combination with the speech output, the visual representation and animation will be realized by the SAP Social Agents. Here it is important that the visual

representation looks human-like which has adapted facial expressions depending on the response.

### **1.3 Introduction to the Chapters**

First we concentrate on the user requirements regarding the Conversational, Mediating and Animated Receptionist by performing a comprehensive requirement analysis in chapter 2. Then, chapter 3 gives an overview about the related work and compares it with the requirements of COSIMAR. Thereby, different architectures for conversational systems are taken into account.

In chapter 4, an overview about the software products used for the virtual receptionist is given. This includes the Chant SpeechKit realizing the speech recognition component, TAPAS responsible for the dialog management, SpeechStudio implementing the call functionality and realizing the speech recognition on the phone, and the Social Agent FATA system, which is concerned with the display of the virtual character and the generation of the speech output. Afterwards, in chapter 5 the architectural design of COSIMAR is described. Chapter 6 explains the design and implementation of the system.

In chapter 7, the user studies regarding the use of the virtual receptionist are evaluated, analyzed and summarized.

Conclusions and outlooks drawn from the insights of this research project are presented in chapter 8.



## 2 Requirement Analysis

### 2.1 Business Process

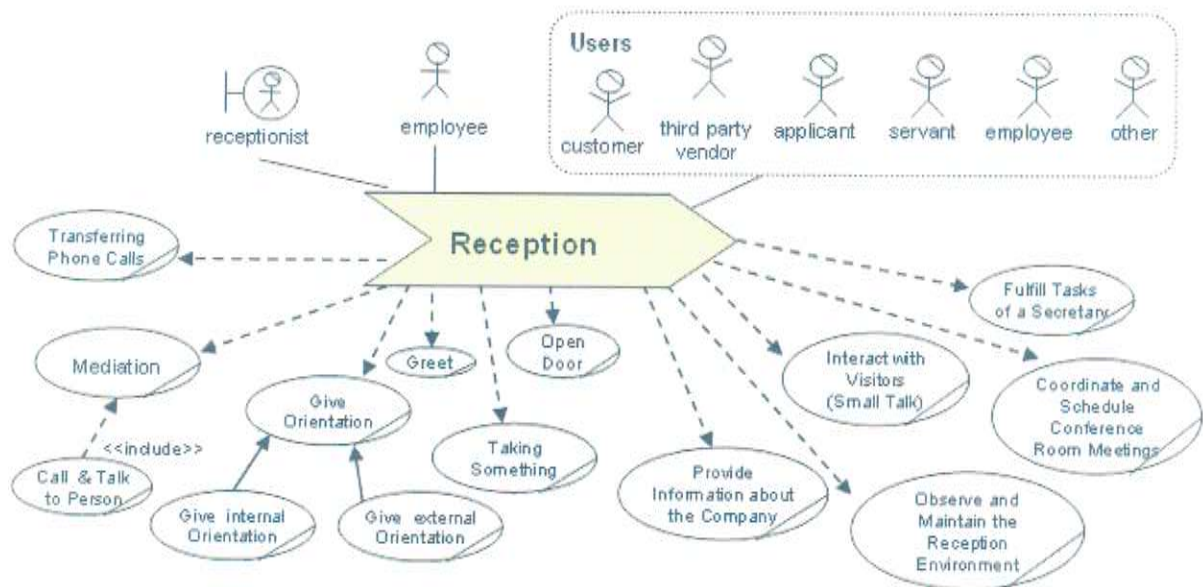


FIGURE 1: Business Process: Reception

FIGURE 1 gives an overview about all business use cases realizing the business process *Reception*. Because receptionists are the first representatives of an organization a visitor encounters, they need to be polite, helpful, and professional.

Name	Reception
Short description	<p>The business process <i>Reception</i> comprises several tasks. It includes establishing the contact between the customer and the employee, insofar that it calls the employee or a representative person of the department to come over.</p> <p>In the following chapters, both business use cases <i>Mediate Customer</i> and <i>Call &amp; Talk to Person</i> are considered separately, because <i>Mediate Customer</i> includes <i>Call &amp; Talk to Person</i> and couldn't be realized without this. <i>Call &amp; Talk to Person</i> is no independent business use case; it's always embedded in another one.</p> <p>Furthermore orientations regarding both the company area (e.g. restroom, meeting room) and the surrounding external area (e.g. restaurant, shopping center) are given.</p> <p>The business process also includes opening the door, greeting visitors, transferring calls and providing information about the company (e.g. HR,</p>

	<p>printer, lunch). Since many people visit the reception, observation and maintaining of the reception environment (e.g. slippery outside), as well as interacting with visitors like talking about the weather, sight seeing and gambling must be realized.</p> <p>Beyond, many "secretary" tasks like taking phone calls, answering mails, looking up phone numbers have to be fulfilled.</p> <p>Since often more than one people visit the reception, the receptionist must be able to handle multiple tasks at the same time.</p> <p>Many of the business use cases are dependent from the user roles. Thus, from the perspective of security, not all actions are allowed, that can be performed.</p>
<b>Contained business use cases</b>	<ul style="list-style-type: none"> <li>• Mediation (<i>includes</i> Call &amp; Talk to Person)</li> <li>• Give Orientations</li> <li>• Give internal Orientations</li> <li>• Give external Orientations</li> <li>• Transferring Phone Calls</li> <li>• Greet</li> <li>• Taking something</li> <li>• Open door</li> <li>• Provide Information about the company</li> <li>• Interact with visitors (Small Talk)</li> <li>• Observe and maintain the reception environment</li> <li>• Coordinate and Schedule Conference Room Meetings</li> <li>• Fulfill tasks of a Secretary</li> </ul>
<i>Officer</i>	Director of Facilities
<i>Actors</i>	<p>user:</p> <ul style="list-style-type: none"> <li>• SAP employee, management, senior management</li> <li>• customer; potential customer</li> <li>• applicant</li> <li>• third party vendor</li> <li>• servant (taxi driver, parcel or flower deliverer)</li> </ul> <p>business employees:</p> <ul style="list-style-type: none"> <li>• receptionist</li> <li>• employee</li> </ul>



## 2.2 Business Area

In FIGURE 2 the business area of the *Reception* is displayed, which is reduced with regard to likely functionality of the virtual receptionist. The reception is a business area, which mediates the customers, employees, potential customer or objects by a receptionist to an employee. For realization it uses the business use case *Call & Talk to Person* to tell the employee that a user is waiting for him in the foyer. Due to security aspects, third party vendors or applicants are mainly not mediated. Furthermore in the business area all users are given orientations either internal orientation (regarding the SAP/ company area) or external orientations (e.g. the environment). Further important business use cases are *Greet*, *Open door* and *Taking Something*.

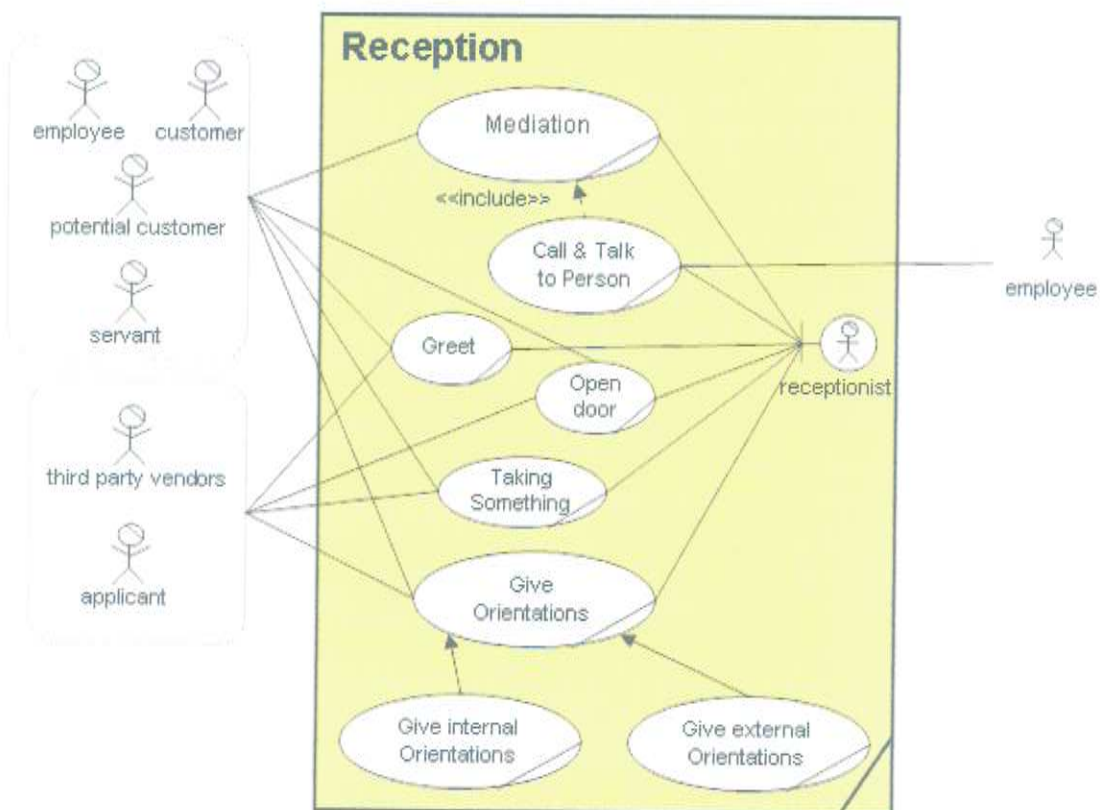


FIGURE 2: Business Area: Reception reduced with regard to the requirements of the Virtual Receptionist

The next tables summarize the business uses cases which should be realized by the virtual receptionist. Whereas {excluded} means, that these cases are not realized in the first prototype version.

<i>Name</i>	<b>Mediation</b>
<i>Type</i>	business use case
<i>Description</i>	The actor or object of an actor is mediated to a specific employee or a responsible person of a department by the receptionist. To do so it's important that the visitor register himself (name, company and his contact person).
<i>Motivation</i>	<ul style="list-style-type: none"> <li>a) Actor has a meeting, or wants to talk to a person.</li> <li>b) Actor has an object (e.g. flowers, packet) for an employee</li> </ul>
<i>Result</i>	The actor or object of an actor was mediated.
<i>Actors</i>	users: customer, employee, potential customer, servant receptionist employee

<i>Name</i>	<b>Call &amp; Talk to Person</b>
<i>Type</i>	business use case
<i>Description</i>	This business use case comprises the call to an employee and the talk between the receptionist and the employee. The receptionist tells the employee who is waiting and receives the answer for the visitor.
<i>Motivation</i>	To establish the contact between the user and the employee, the receptionist has to make a phone call and tell the employee.
<i>Result</i>	<ul style="list-style-type: none"> <li>a) result of the conversation with the employee</li> <li>b) couldn't reach the employee</li> </ul>
<i>Actors</i>	receptionist employee

<i>Name</i>	<b>Greet</b>
<i>Type</i>	business use case
<i>Description</i>	This business use case comprises greeting and offering help to visitors.
<i>Motivation</i>	<ul style="list-style-type: none"> <li>a) the visitor greets</li> <li>b) the visitor is near the reception</li> <li>c) the visitor says what he wants the receptionist to do</li> </ul>
<i>Result</i>	greeting or starting to help the visitor
<i>Actors</i>	users: customer, employee, potential customer, third party vendor, servant, applicant receptionist

<i>Name</i>	<b>Open the door {excluded}</b>
<i>Type</i>	business use case
<i>Description</i>	Within this business use case the front door will be opened.
<i>Motivation</i>	Person wants to get into the building.
<i>Result</i>	Person is in the building.
<i>Actors</i>	users: customer, employee, potential customer, third party vendor, servant, applicant receptionist (or someone else opens the door)



<i>Name</i>	<b>Taking something {excluded}</b>
<i>Type</i>	business use case
<i>Description</i>	In this business use case the receptionist receives an object (like the badge) from a visitor.
<i>Motivation</i>	Visitor wants to give something to the receptionist.
<i>Result</i>	Receptionist got the object.
<i>Actors</i>	users: customer, employee, potential customer, third party vendor, servant, applicant receptionist

<i>Name</i>	<b>Give Orientations {excluded}</b>
<i>Type</i>	business use case
<i>Description</i>	Gives orientations about an area in general.
<i>Motivation</i>	Visitor wants to know where to find some place.
<i>Result</i>	Visitor received the orientation information.
<i>Actors</i>	users: customer, employee, potential customer, third party vendor, servant, applicant receptionist

<i>Name</i>	<b>Give internal Orientations {excluded}</b>
<i>Type</i>	business use case
<i>Description</i>	Orientations regarding the company area are given. For example where the restroom or the meeting room are located.
<i>Motivation</i>	Visitor wants to know where a specific room is.
<i>Result</i>	The visitor found the room.
<i>Actors</i>	users: customer, employee, potential customer, third party vendor, servant, applicant receptionist

<i>Name</i>	<b>Give external Orientations {excluded}</b>
<i>Type</i>	business use case
<i>Description</i>	Orientations regarding the surrounding external area (e.g. restaurant, shopping center) are given.
<i>Motivation</i>	Visitor wants to know something about the external area.
<i>Result</i>	The visitor received the external orientation information
<i>Actors</i>	users: customer, employee, potential customer, third party vendor, servant, applicant receptionist

## 2.3 System Use Cases

### 2.3.1 Overview

TABLE 1 gives an overview about the system use cases, which will be realized within the first version of COSIMAR. To simplify matters, the system use cases UC-02 and UC-03 of the virtual receptionist are given the same denotation as the business use cases.

S. No.	Use-Case ID:	Use-Case Name:	Priority <sup>1</sup>
1.	UC-01	Mediate Customer	High
2.	UC-02	Call & Talk to Person	High
3.	UC-03	Greet	High

TABLE 1: Overview about the System Use Cases of the Virtual Receptionist

### 2.3.2 Mediate Customer

FIGURE 3 shows the system use cases that realize the business use case of *Mediate Customer*. On the one hand there is the common system use case, up to now, realized with a human receptionist: *Human Receptionist mediates Customer*. On the other hand, the business use case can be realized within the system COSIMAR as the system use case: *(Virtual Receptionist) - Mediate Customer*.

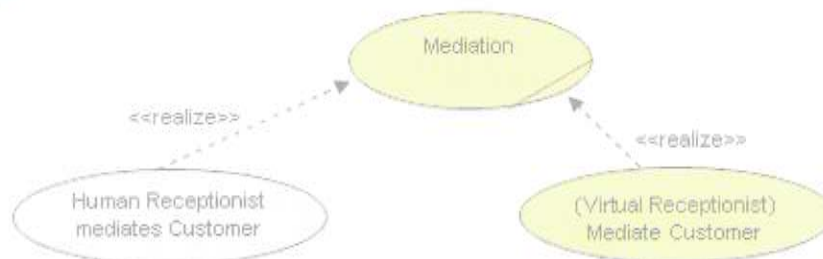


FIGURE 3: System use cases of the business use case: Mediate Customer

The proceeding identifying the system use cases is the same for the other (UC-02 and UC-03). Therefore in the next chapters only the system use cases of COSIMAR are considered.

In the first version of COSIMAR not all the functionalities of the business use case *Mediate Customer* will be realized, therefore UC-01 differs from its business use case.

---

<sup>1</sup> Priority: High, Medium or Low



Name	<b>Mediate Customer</b>	UC-01
Type	system use case	
Description	The Virtual Receptionist mediates a customer to an employee or a responsible person of a department. To establish the contact, it's important that the visitor registers himself by saying his name and contact person. Then the Virtual Receptionist calls the contact person. It is realized by using the system use case <i>Call &amp; Talk to Person</i> .	
Actors	customer employee	
Motivation	The customer has a meeting, or wants to talk to a person.	
Preconditions	User is one of the persons: customer Selection of this system use case.	
Postconditions	Establishment of the mediation: COSIMAR tells customer the answer of the employee.	
Result	Mediation of the customer.	
Frequency of use	High	
Activities	A1 Register the customer A1.1 Tell customer about security A2 Identify the task A2.1 Identify the meeting A2.2 Identify a representative employee A2.3 Identify the employee A3 Get the telephone numbers of the employee A4.1 Entertain customer with a video A4.2 Call & Talk to employee (includes the system use case UC-02) A5 Try another number A6 Tell customer the answer	

### Detailed description of the system activities

The system activities A1, A2, A2.1, A2.3, also visualized in the system activity diagram in FIGURE 5, are considered separately from each other to emphasize the important activities of the interaction between the customer and the system. But because we use a natural language user interface, there is no given linear way, how COSIMAR has to proceed. So we separated the activities just for visualizing.

#### A1 Register the customer

The registration of the customer occurs not in a given linear way, because we use a natural language user interface which allows "free" user input. But it's important that the system detects the user name. If the system doesn't understand the name, in future versions there will be the possibility for the customer to enter his name via keyboard or to spell it.

#### **A1.1 Tell customer about security**

If the customer doesn't want to disclose his name, COSIMAR is not allowed to mediate the customer due to security aspects. Therefore the system informs the user about security and the customer can't be mediated.

#### **A2 Identify the task**

To mediate the customer it's also important to get to know the name of the meeting party. If the customer doesn't know the name, he can say the meeting name or meeting time. Then the proceeding continues with A2.1.

##### **A2.1 Identify the meeting**

To identify the meeting, the virtual receptionist looks up in the internal database for a meeting that takes place at this time. If no meeting can be found, the customer can't be mediated.

##### **A2.2 Identify a representative employee**

If a meeting was located in the database, the responsible person for this meeting will be identified. If there is no responsible person, the customer can't be mediated.

##### **A2.3 Identify the employee**

If the customer says the name of the contact person, the system has to identify the employee by searching in the internal database. If it locates him, the mediation process can be continued. Otherwise the customer can't be mediated, what about the customer will be informed.

#### **A3 Get the telephone numbers of the employee**

If the system identified the employee, it can search for his telephone numbers in the internal database. If it finds no number, the customer can't be mediated.

##### **A4.1 Entertain customer with a video**

Because COSIMAR calls the employee, the customer should be synchronistically entertained with a video about the actual projects of his meeting party's department.

#### **A5 Try another number**

If the employee isn't reachable, then the system looks up, if there is another number of the employee stored in the mediation object. If so, it goes back to A4.2, otherwise the contact can't be established and the system use case ends with A6.

In future enhancements COSIMAR could try to find another person or send an email.

### A6 Tell customer the answer

If COSIMAR finished the phone call, it tells the customer the answer. That could be for example:

- Mr. Green, I have finished the phone call. Your meeting party will come in a few minutes. He will meet you at the reception. It was a pleasure to help you.
- I'm sorry to inform you, that your meeting party has no time to meet you today. Please contact him per email to arrange a new meeting.
- I'm sorry to inform you, that I couldn't contact your meeting party.

### System Activity Diagram

As mentioned in 3.2.1, the System Activity Diagram in FIGURE 5 just visualizes the actions taken by the system, but it's not the main approach, since there is the input modality natural language.

The business object *Mediation*, visualized in FIGURE 4, encapsulates all information required for the realization of the system use case *Mediate Customer*. These are the name, department and telephone number of the employee, the name of the customer, the meeting time and name, and the *MediationResult* (q.v. FIGURE 6).



FIGURE 4: Mediation Object

The activities A1, A1.1, A2, A2.1, A4.1 and A6 will have as graphical user interface the social agent with appropriate facial expressions and a display for the video.



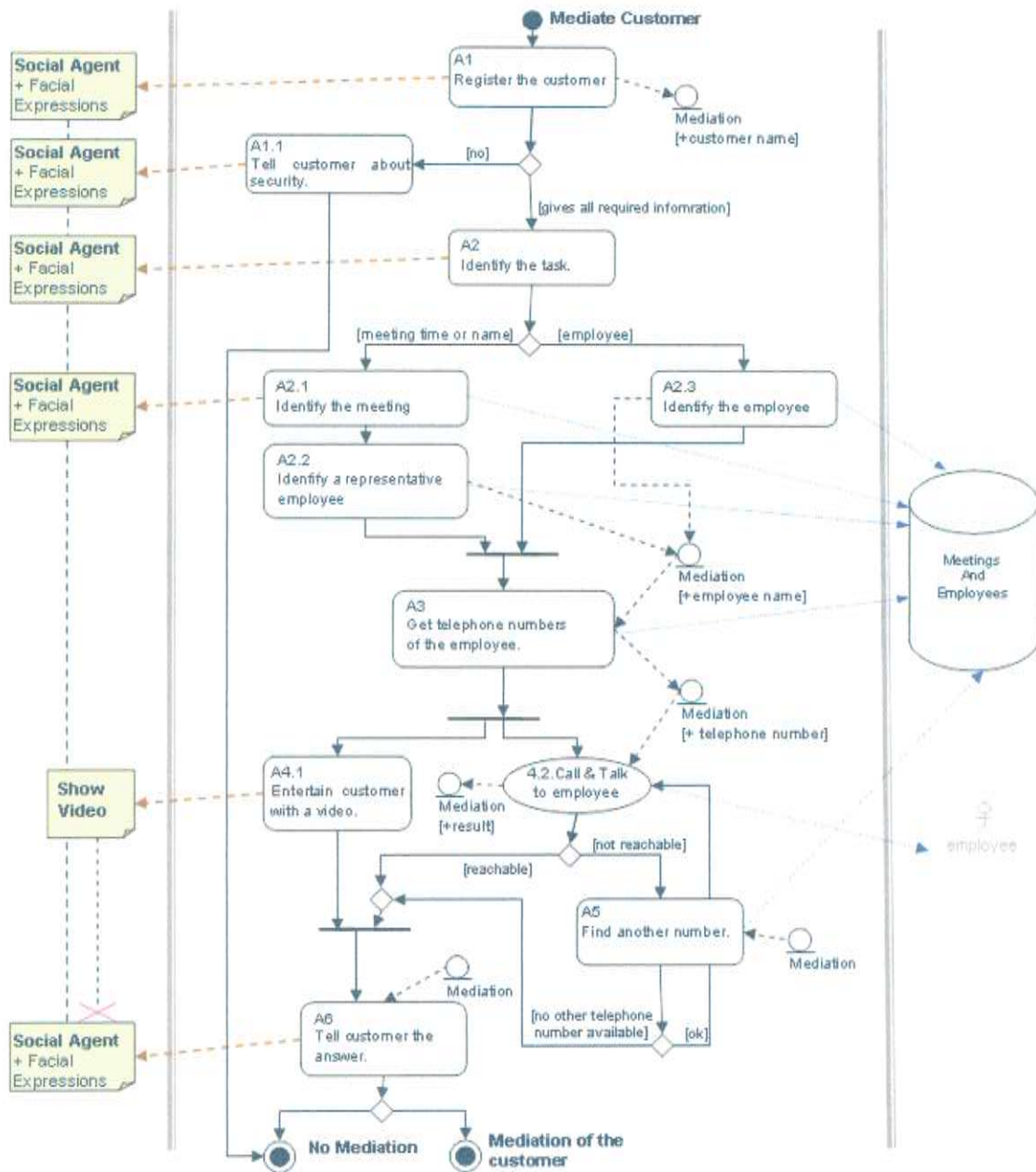


FIGURE 5: System Activity Diagram - Mediates Customer



### 2.3.3 Call & Talk to Person

Name	Call & Talk to Person	UC-02
Type	system use case	
Description	This system use case comprises the call to an employee and the talk between COSIMAR and the employee. The virtual receptionist tells the employee who is waiting and receives the answer from the employee.	
Actors	Employee	
Motivation	To establish the contact between the user and the employee, the receptionist has to make a phone call and tell the employee.	
Preconditions	<ul style="list-style-type: none"> <li>- Customer name is identified</li> <li>- Employee number is identified</li> </ul>	
Postconditions	<ul style="list-style-type: none"> <li>a) Receptionist reached the employee and talked to him</li> <li>b) Couldn't reach the employee</li> </ul>	
Result	<ul style="list-style-type: none"> <li>a) result of the talk</li> <li>b) couldn't reach the employee</li> </ul>	
Frequency of use	high	
Activities	<ul style="list-style-type: none"> <li>A1 Dial Number</li> <li>A2 Wait for answer</li> <li>A3 Listen</li> <li>A4 Talk to Person</li> <li>A5 Wait for beep</li> <li>A6 Leave a message</li> <li>A7 Save Mediation Result</li> </ul>	

### System Activity Diagram

The business object *MediationResult*, visualized in FIGURE 6, encapsulates all information required for the realization of the system use case *Call & Talk to Person*. This information is:

- `meeting place`: in the prototype version this could either be the reception lounge or customer area in the second floor
- `meeting time`: when the employee will be at the meeting place
- `establishment`: if the COSIMAR talked to the employee
- `result type`: information whether this was the wrong number, it was busy, the employee wasn't on his place, it left a message on the mailbox or if it talked to the employee
- `try other number`: will be true, if the COSIMAR didn't got the employee



FIGURE 6: MediationResult Object

FIGURE 7 shows the system activity diagram of the use case *Call & Talk to employee*. First the number of the employee is dialed [A1], which is encapsulated in the *Mediation* object. Then, there are three possibilities: the number doesn't exist, it's busy or there is a free tone. If there is no free tone the mediation result should be saved [A7]. Otherwise it rings and COSIMAR has to wait [A2] until it rings a specific time (which means that nobody is in the office – continue with A7) or somebody answers. The voice COSIMAR is listening to [A3], could either be the mailbox or a human. If it's the mailbox (recognizable by the announcement beginning which will be the same), it will wait for the beep [A5], leave a message [A6] and save the mediation result [A7]. If it reached the employee, it conducts a dialog [A4], including the information of the *Mediation* object (e.g. customer name). The information figured out in the dialog is saved in the *MediationResult* object.

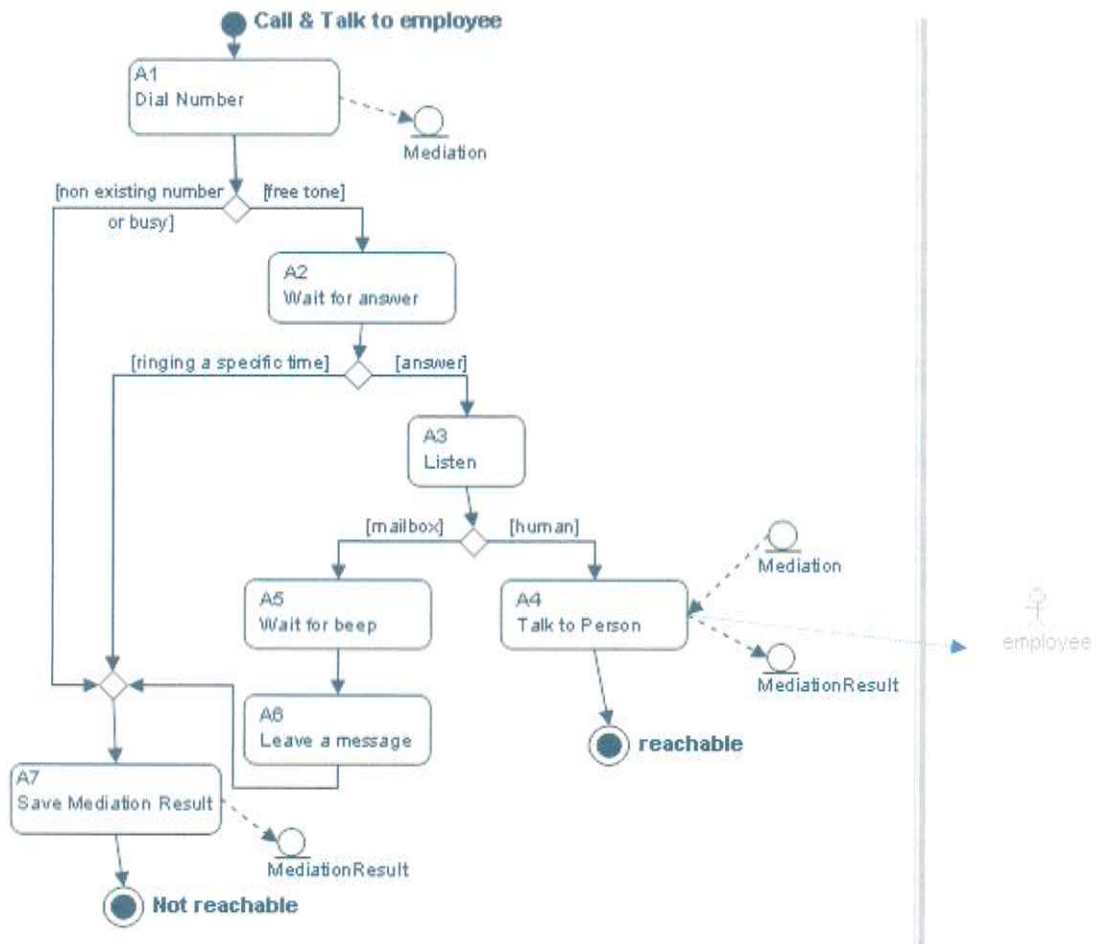


FIGURE 7: System Activity Diagram: Call & Talk to Person

### 2.3.4 Greet

Name	<b>Greet</b>	UC-03
Type	system use case	
Description	This system use case comprises the greeting and offering help to customers.	
Actors	customer	
Motivation	A customer enters the reception environment or the customer greets the receptionist.	
Preconditions	a) COSIMAR has sensed a customer b) Customer greets COSIMAR	
Postconditions	The customer was greeted and offered help.	
Result	Greeting of the customer or selecting of the next system use case.	
Frequency of use	high	
Activities	A1 Say hallo to the customer A2 Offer help A3 Say good bye A4 Select a use case	

FIGURE 8 presents a system activity diagram of the system use case *Greet*. If COSIMAR senses the customer, it will introduce itself and conduct a welcome dialog [A1]. Within this dialog the customer can say what he wants the system to do. In this case the virtual receptionist selects accordingly the use case, or apologizes for not offering the task [A4]. If the customer doesn't select a task himself, the system offers proactively help [A2]. Now the user can decide weather he wants to use the virtual receptionist [A4] or not. In this case it says good bye to the visitor [A3].

The actions A1, A2 and A3 will show the social agent with appropriate facial expressions.

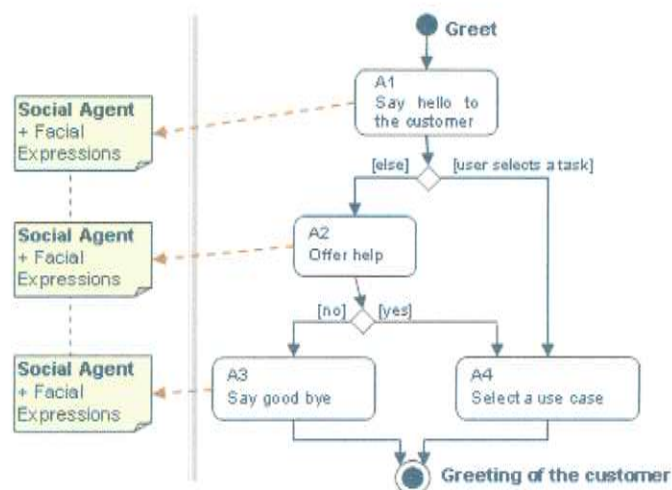


FIGURE 8: System Activity Diagram: Greet



## 2.4 Non – Functional Requirements

As already mentioned, the main aim of the virtual receptionist project is offering customers a usable software product which provides a natural language user interface for receiving information. Therefore, the focus lies on those non-functional requirements optimizing accessibility and usability, and transforming the graphical user interface to a social and enjoyable experience. But before defining the non-functional requirements of COSIMAR, first let's have a look at the definition of usability and accessibility: [KD05]

Usability comprises the following characteristics:

1. Effectiveness: the accuracy and completeness with which a user achieves a goal
2. Efficiency: the qualitative and quantitative amount of resources that a user needs to achieve goals
3. Satisfaction: the emotional effect a user experiences when interacting with the software, including the user's impression that the goal was properly completed

Accessibility is defined by the following aspects:

1. Situational awareness: user knows in what situation he currently is
2. Awareness of options: user knows which actions can be taken to change the situation
3. Executive capabilities: user can execute these actions

From these characteristics the non-functional requirements performance, availability, and interfaces can be derived. To support efficiency, the performance of the system must be high enough to enable real-time understanding and response generation. A high availability in cooperation with the functional requirements guarantees effectiveness and accessibility. Moreover, high availability, high performance, a graphical user interface which has a pleasant, human-like appearance with appropriate facial expression as to the speech output, assure a high user satisfaction and a social and enjoyable experience.

Beyond the requirements concerning usability and accessibility, security is another important non-functional requirement. Since the virtual receptionist is the first point of contact, it must be ensured that non-authorized users are not mediated to a person or allowed to enter the internal campus. In addition, internal information, like SAP employee telephone numbers, must not be revealed.

The non-functional requirements not discernable at runtime play an important role with respect to the future enhancements of COSIMAR. Most essential for the architectural design is modifiability, for the reason that the virtual receptionist should be installed in different reception environments. Regarding the embedding of the virtual receptionist in SAP applications, reusability as well as portability of the components should be concerned.



### 3 Related Work

There are many challenges in embodied conversational agent projects. In this section a short review is given about related work in the following three areas: multimodal interfaces, models of conversation and conversational agent interfaces.

#### 3.1 University of Karlsruhe – Architecture for an Emotion-Sensitive Dialog System

The work of [Ho03] focuses on both user emotions in addition with its interpolation over the history (emotion tendency), and system emotions. The emotion-sensitive architecture is flexible and can realize various scenarios. These scenarios using emotions are varying from Emotion Mirror to Humanoid Robot [GF03]. The data flow of this architecture is shown in FIGURE 9.

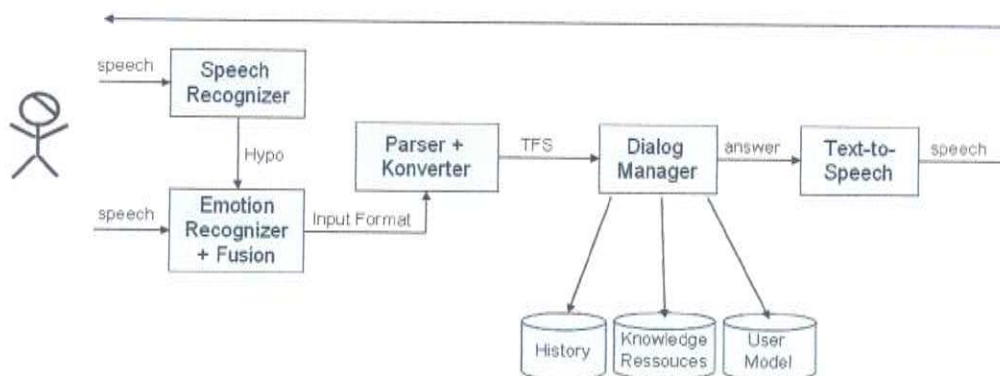


FIGURE 9: Data Flow of an Emotion-Sensitive Dialog System [Ho03]

For the speech recognition the Janus Recognition Toolkit (JRtk) is used, which runs in about 0.5 x real-time [FH04]. JRtk can easily switch between different domain grammars. This helps to deal with spontaneous speech, dialects, bad acoustic conditions, and unknown words. Furthermore, emotions from the speech input are recognized. These two detected input hypothesis, emotion and speech utterance, are thereupon fused, parsed, and converted into a semantic representation – a typed feature structure (TFS) – which is send to the dialog manager. By using history, knowledge resources and a user model, an answer is generated, which is thereupon sent to the speech synthesizer.



The different scenarios are realized with the dialog system TAPAS. TAPAS follows an information-based approach, instead of a model-based approach to dialog processing. Information-based is understood as the determination of actions undertaken by the dialog system based on the information available at any given point in the dialog [DW97]. The partial information is processed by representing them in generalized descriptions, which allow representing more than one dialog goal. The missing information, to reach exactly one communicative goal, is gathered depending on the dialog strategy defined as a sequence of actions. Furthermore, TAPAS has language understanding algorithms allowing incorrect speech recognition and incorrect grammatical utterances.

Regarding multimediality not only the fusion of emotions and speech is realized. As well, pointing gestures both 2D and 3D can be fused with the speech input [HN04].

TAPAS is a multilingual spoken dialog system [Ho05]. The multilingualism is realized by separating the language-specific grammar parts – used for parsing the user input – and the spoken output from the semantic representations, which are equal for different languages. Thus the development and maintenance of such an internationalized application is supported.

TAPAS does not only cover the areas of speech processing, dialog management and language generation, but also offer a tighter integration with the speech recognizer [FH04].



## 3.2 Projects at MIT

At Massachusetts Institute of Technology (MIT) there are different generations of conversational agents. First, research in this area at MIT started with *Animated Conversation*. In this system two animated characters conducted a conversation with each other. The next generations, considered in this chapter, are *Gandalf* followed by *REA* and *MACK*, the information kiosk, which derived from *REA*.

### 3.2.1 Gandalf

*Gandalf* is an embodied multimodal conversational agent capable to conduct a conversation with users in real-time. The scenario of *Gandalf* deals with the solar system thematic. The user can ask him questions about the solar system and travel to planets using natural speech and gesture, whereas the user has to wear a body suit. *Gandalf*'s visual representation consists of a hand and a face, which appear on a small monitor in front of the user. To display the solar system, there is a large-screen on the left.

*Gandalf* is implemented in an architecture called *Ymir* which supports multimodal input and output. As described in [Th99 S.19], *Ymir* has a layered architecture which provides a hierarchy for the complexity and incremental nature of perception and interpretation, for decision making prioritization and for bottom-up analysis and top-down control.

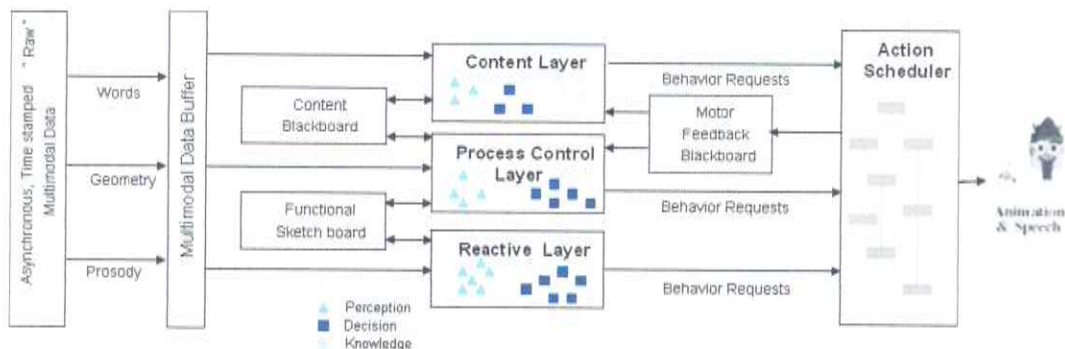


FIGURE 10: Ymir Architecture [Th99 S.12]

The *Ymir* architecture, as displayed in FIGURE 10, consists of three different layers. The lowest layer – *Reactive Layer* – makes decisions about behaviors that require recognizing actions shorter than one second.

The *Process Control Layer* concerns with behaviors that usually takes more time. It has a sophisticated analysis of the input and initiates task-level actions, that take more time to execute. Accordingly, it is concerned with recognizing the global context of dialogue and managing the communicative behavior of the

agent. The *Content Layer* contains knowledge bases which produce plans, multimodal actions and content-related responses in the dialogue.

Each of the layers includes the modules *Perception* and *Decision*. There are different categories of unimodal *Perceptions* which perceive for example speech, prosodies and deictic gestures. Then with the help of multimodal integrators it is possible to recognize backchannel feedback and turn-interruption. The *Decision* modules [Th98] have to decide how to process the information from the *Perceptions* and how to communicate its status to other modules as an action request. Furthermore *Deciders* in each layer are allowed to turn on and off modules in other layers. Whereas the behavior requests from the Reactive Layer takes the highest priority followed by Process Control Layer and Content Layer.

To separate the decisions from the execution of the motor actions, all layers send its behavior requests to the *Action Scheduler* which arranges the action requests by priority and expected lifetime. Then the *Behavior* is chosen from a behavior lexicon or behavior tree.

To realize the communication between the different layers, the Ymir architecture includes several blackboards. The *Functional Sketchboard* blackboard realizes the communication between Reactive Layer and Process Control Layer and it stores intermediate and final results of low-level high-speed perceptual processes. Moreover it keeps a record of all decisions that have been initiated from the Reactive Layer. The *Content Blackboard* realizes the communication between Process Control Layer and Content Layer. The *Motor Feedback Blackboard* keeps track of which part, in a stream of actions, is currently being planned and/or executed by the Action Scheduler. Both Process Control Layer and Content Layer can read from this blackboard the status of initiated behaviors.

### 3.2.2 REA

In comparison to Gandalf, REA (Real Estate Agent) can track the user passively through cameras and its embodiment comprises speech and a fully animated body including hand gesture, body posture, facial expressions and eye gaze. Another enhancement is the synthesizing of her responses including speech with appropriate gestures using propositional information corresponding to the content of the conversation, and interactional information consisting of cues that regulate the conversational process and includes non-verbal behaviors.

As described in [Ca99] and [Ca00], the general approach of the system includes integrating input from several modalities into a single semantic representation that is passed from module to module. To maintain the regulatory and content-oriented contribution of every conversational act throughout the system, the semantic representation has slots for interactional and propositional information. The overall communication of the components is realized by using KQML (*Knowledge Query and Manipulation Language*) that is a speech-act based inter-



agent communication protocol which serves to make the system modular and extensible.

The main components of the architecture of REA, displayed in FIGURE 11 are *Input Manager*, *Hardwired Reactions*, *Deliberative Module* and *Action Scheduler*.

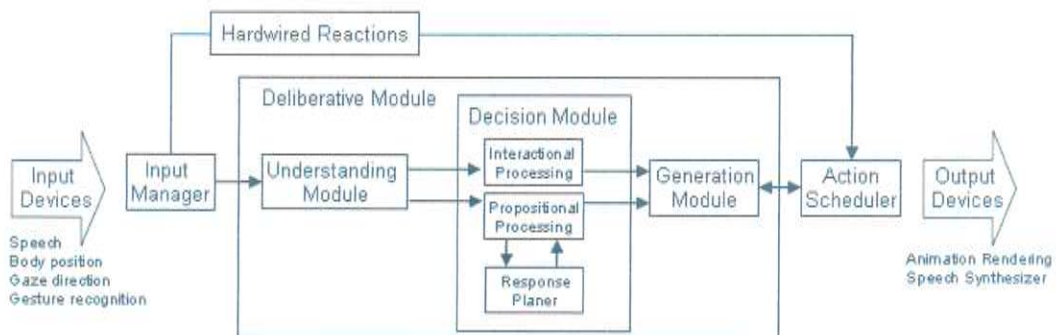


FIGURE 11: Architecture of REA [CC02]

The *Input Manager* collects input from all modalities and decides whether the data requires instant spontaneous reaction, which is realized in the *Hardwired Reaction* module, or deliberate discourse processing. The *Deliberative Module* handles all input that requires a discourse model for proper interpretation to produce interactional behaviors (e.g. “hmm”) as well as propositional behaviors. In detail, the information from the *Input Manager* is passed to an *Understanding Module*, followed by the *Decision Module* containing the modules *Response Planner*, *Interactional* and *Propositional Processing* which pass their information to a *Generation Module*. Both the *Hardwired Reaction* module and *Deliberative Module* send their action requests to the *Action Scheduler* which schedules the motor events, as known from Gandalf.

### 3.2.3 MACK

MACK (Media Lab Autonomous Conversational Kiosk) is a virtual reception kiosk, who can answer questions regarding research groups at MIT, and who can give campus directions by the use of a projector that highlights areas on a map.

For the purpose of MACK the planning and complex generation module of REA's architecture was replaced with a simple template-based sentence generator [CS02].

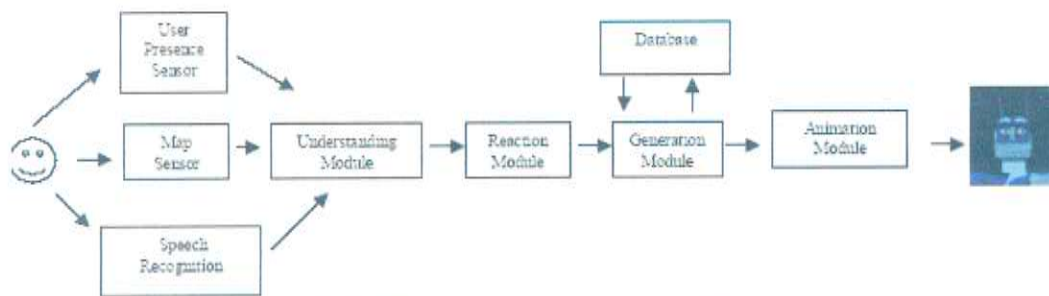


FIGURE 12: Architecture of MACK [CS02]

As FIGURE 12 shows, all information from the input devices: *User Presence Sensor*, *Map Sensor* and *Speech Recognition* are passed to the *Understanding Module*. The speech input is recognized by grammar-based speech recognizer (MIT LCS' *SpeechBuilder*). To recognize the presence of the user, a pressure-sensing chair mat is used. Moreover users' pen gestures on a paper map can be interpreted – for example the user action: "Tell me about this" while pointing to a specific research group on the map. After understanding the user input a process line starts, which pass through the components *Reaction Module*, *Generation Module* using a *Database* to generate an answer, and *Animation Module*. The *Animation Module* BEAT (Behavior Expression Animation Toolkit) can automatically annotate text with hand gestures, eye gaze, eyebrow movement and intonation. Furthermore, the multimodal output comprises the speech synthesis using Microsoft Whistler TTS and a projector which highlights areas and draws paths.

### 3.3 EU project MagiCster

[CC02] and [CR01] describe an architecture of a multimodal agent MagiCster which takes the role of a doctor and makes use of gaze, facial expression, gesture, body posture and speech in a synchronized way.

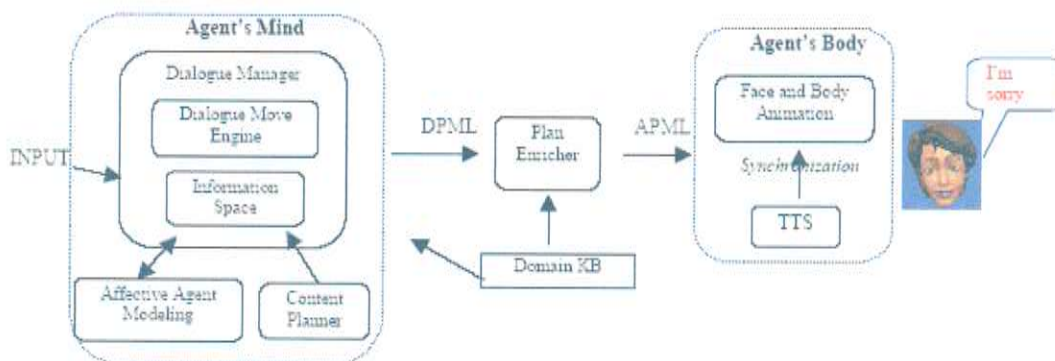


FIGURE 13: Architecture of MagiCster [CC02]

The architecture of MagiCster, as shown in FIGURE 13, includes the main components *Agent's Mind* and *Agent's Body* which are interfaced by a *Plan Enricher*.

The *Agent's Mind* decides which dialog move should be performed. Its result is a discourse plan formalized according to the markup language DPML, which enables tagging the discourse plans. The *Agent's Mind* includes the modules *Affective Agent Modeling*, *Content Planner* and *Dialogue Manager*.

The *Affective Agent Modeling* decides which emotional state should be activated and in which intensity. Accordingly, it is responsible for updating the agent's mental state, which is represented as a Dynamic Belief Network. It is build automatically at every dialog turn from both a network that represents the agent's state at previous turn and a network that represent the events occurring between two turns. These networks are stored in the module named *Information Space*.

The module *Content Planner* generates the discourse structure that is appropriate in a given situation by the use of DPML. On an abstract high level, a plan is generated representing the steps needed to achieve the goals. From these plan steps less complex dialog moves are expressed on a lower level.

The *Dialogue Manager*, built on top of the TRINDI (Task Oriented Instructional Dialogue) architecture, uses the *Information Space* for storing the symbolic representation of the user's move. Furthermore, the *Information Space* is used by the *Dialogue Manager* to decide the next dialogue action by choosing a sub plan.



The *Plan Enricher*, as an interface between *Agent's Mind* and *Agent's Body*, translates the symbolic representation of a dialog move into an agent's behavior specification. So, it translates DPML into APML, a markup language for believable behavior generation.

The *Agent's Body* comprises a speech synthesizer and a *Body Generator* which displays a 3D face. The *Body Generator* interprets the APML-tagged dialog move, sent by the *Plan Enricher*, and decides how to convey the meanings regarding facial expressions, gaze and head movement.

### 3.4 University of Rochester – An Architecture for More Realistic Conversational Systems (TRIPS-System)

The architecture, presented in [AG01], is intended for conversational systems, that enable human-like performance along the three dimensions: interpretation, generation and behavior, but yet without including visual representation.

The three main components of the TRIPS-System architecture, shown in FIGURE 14, are *Interpretation Manager*, *Behavioural Agent* and *Generation Manager*. The task of the *Interpretation Manager* comprises interpreting arising user input and updating the discourse context. The *Behavioural Agent* plans the behaviour of the agent based on its goals and obligations. Accordingly, it is called the autonomous “heart” of the agent. The *Generation Manager* receives actions that involve communication and collaboration with the user. Furthermore, it plans the specific content of utterances and displays updates.

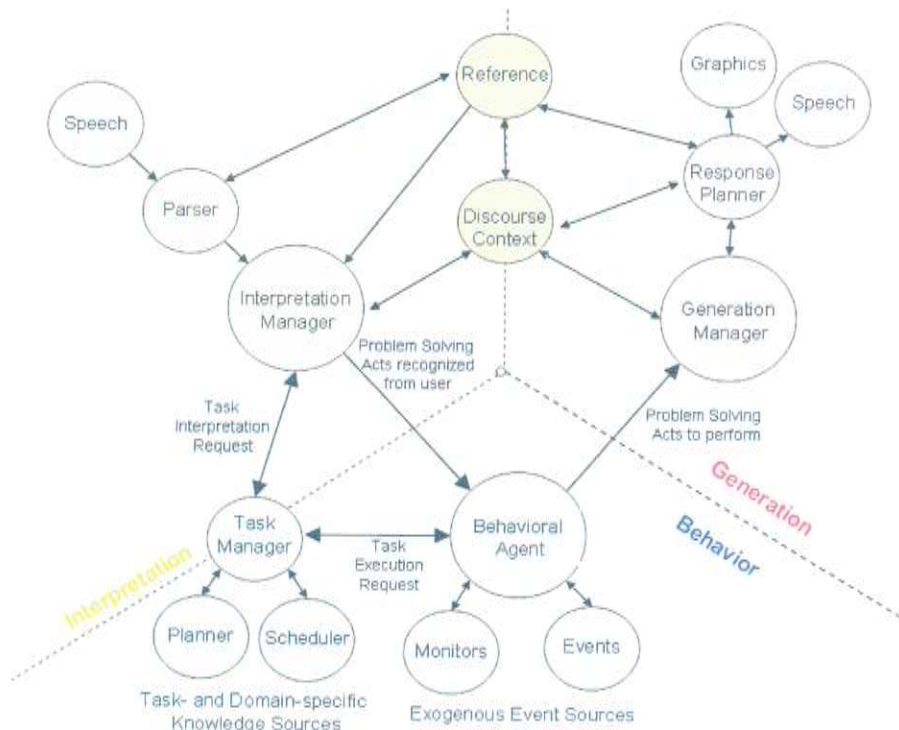


FIGURE 14: Architecture of TRIPS-System [AG01]

To support both *Interpretation* and *Generation* of expressions, there is a *Discourse Context* which contains a model of the current salient entities in the discourse. Furthermore, it includes the current status of the turn, the discourse history and current discourse obligations. To support ellipsis resolution and clarification questions, the *Dialogue Context* also contains the structure and

interpretation of the preceding utterance. Hence, the *Discourse Context* provides information to coordinate the system's conversational behavior.

To assist both the *Interpretation* and the *Behaviour*, there is the *Task Manager* that has the ability to answer queries about objects and their role in the domain, provide intention recognition services as well as an interface between the generic problem solving acts and actual task-specific agents.

The *Abstract Problem Solving Model*, remaining at an abstract level and used by *Interpretation Manager*, *Behaviour Agent* and *Generation Model*, is formalized as a set of actions that can be performed on problem solving objects. These objects include objectives (goals), solutions (proposed courses of action), resources (objects used in solutions) and situations.

### 3.5 University of Reading – Queue-Based Agent Architecture for Multimodal Interfaces

To solve the problems of fixed speech input grammar with associated performance drawbacks and expansion limitations, fixed system interaction time scales, and non-continuous architecture without provision for overlapping and concurrent input interpretation initiatives, [KW01] suggested a Queue-Based Multi-Agent Architecture.

Data received by the input sensors are small communication messages so called *MessagePackets*, which can only provide a complete picture, if a series of messages was sent. These *MessagePackets* are transported in the *MessagePacket Queue* that operates like a sophisticated blackboard. In this queue *MessagePackets* have a chronological order and contain their expiry time. Furthermore, agents can post their own messages on the queue, so that these can be read by all other agents.

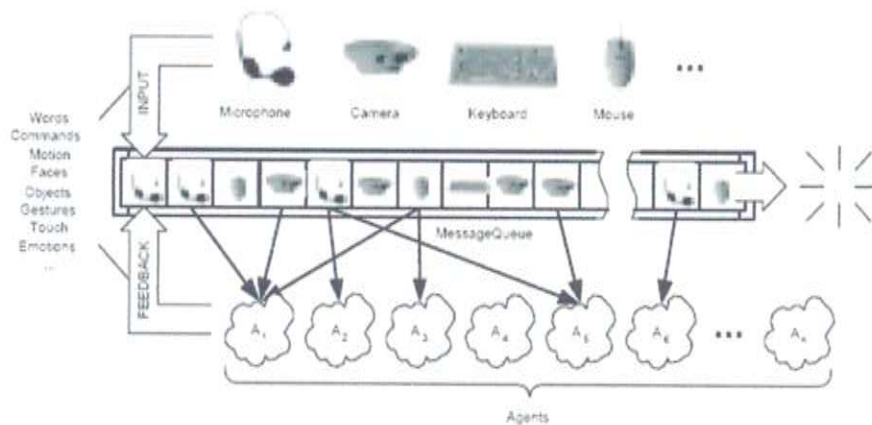


FIGURE 15: Queue-Based Multi-Agent Architecture [KW01]

Regarding the access, every agent may access from any part of the queue by recognizing certain types of *MessagePackets*. But concerning the timing and ordering of when and where agents access the *MessagePacket* queue, there is the need of an *Agent Scheduler*. This approach taken by this system is to only allow certain agents to examine certain parts of the queue and to dynamically vary the volume of agents according to the overall performance.

The arrangement is critical for the successful operation of the system, both for performing its normal functions and maintaining user interaction fluency.



### 3.6 Related Work in Comparison to the Requirements

TABLE 2 gives an overview about the functionality of the introduced systems and the requirements of the Virtual Receptionist in comparison.

		UKA	MIT			Magi-Cster	TRIPS	Queue
			Gan.	REA	MACK			
Input	Multimodal integration	x	x	x	x	–	–	x
	Speech input	x	x	x	x	o	x	x
	Context free grammar	x	x	x	x	o	x	_1
<b>Dialog management</b>		x	o <sup>2</sup>	x	x	x	x	o <sup>3</sup>
Output	Speech synthesis	x	–	x	x	x	o	o
	Human-like face	_ <sup>5</sup>	–	x	–	x	–	o <sup>4</sup>
	Facial expression	_5	x	x	x	x	–	o
<b>Modifiability</b>		x	–	x	o	x	–	x
<b>Performance</b>		x	x	o	x	o	o	x
Mediation	Integrated Call-System	–	–	–	–	–	–	–
	Conduct 2 dialogs within one task	–	–	–	–	–	–	–

- x                    yes
- no
- o                    not specified or focused on in the architecture description

TABLE 2: Comparison Overview

On the one hand, the Conversational, Mediating and Animated Receptionist COSIMAR needs multimodal integration of user sensing and speech input, but on the other hand the system architectures, providing this functionality, are mainly focused on multimodal integration in a more enlarged domain. They support for example user gestures via body suit or camera, pen gestures, mouse, speech, keyboard and emotional input. For the virtual receptionist, these too complex input modalities are not required and would raise the complexity of the system, and thus influence the system performance.

<sup>1</sup> Uses speech recognition in dictation mode

<sup>2</sup> "An (admittedly skimpy) dialogue knowledge base and a topic knowledge base [...]." [TH90]

<sup>3</sup> Evaluation system used continuous user dialog, but in the presented architecture it wasn't considered

<sup>4</sup> Evaluation system used a human-like face with facial expressions, but in the presented architecture it wasn't considered in detail.

<sup>5</sup> But they have an interface to the MS Agent and an integration of system emotions, which can be enhanced to facial expressions.



Speech recognition is primarily supported via context free grammars, except for the Queue-Based Agent Architecture which uses speech recognition in dictation mode. The system of University of Karlsruhe has a speech recognizer which allows decoding along context free grammars in addition to statistical n-gram language models. Moreover, the dialog manager offers a tighter integration with the speech recognizer. The MagiCster architecture just assumes user input, it's not specified whether speech or text.

All the systems have a kind of dialog management. Gandalf's dialog management is only partially developed [Th99]. REA and MACK use a Collagen™ dialogue manager [CN01]. The dialog system used by the University of Karlsruhe is TAPAS. The dialog management of MagiCster is based on TRINDI [CC02]. The TRIPS system itself focuses on the sophisticated dialog management. In the Queue-Based Agent Architecture the dialog management is not closer considered.

With regard to the output, Gandalf uses pre-recorded speech which doesn't fulfill the requirement of the virtual receptionist using speech synthesis. For TRIPS and the Queue-Based Architecture it isn't stated explicitly whether speech synthesis is used. However, REA, MACK and MagiCster present architectures that include speech synthesis with appropriate facial expression generation. Moreover, a human-like appearance is only found in REA and MagiCster, whereas REA has a full embodiment and MagiCster just a 3D head. The systems from Karlsruhe have speech synthesis and offer an interface to the MS Agent; furthermore, there is the openness of the architecture, which allows integrating system emotions and thus appropriate facial expressions.

As the completely new architecture of REA within the MIT projects shows: Gandalf seems not to be easily modifiable, whereas MACK gives an example of REA's modifiable architecture. MagiCster decouples *Agent's Mind* and *Agent's Body* by using a *Plan Enricher* as interface; therefore it's more modifiable than a solely sequential architecture, but within these two components, modifications can be expensive. TRIPS offers no modifiable architecture, because there is a lot of traffic between the components, whereas the Queue-Based Agent Architecture was designed with regard to support modifiability on the components communication level. The different systems of Karlsruhe state an example, that their architecture is easily modifiable.

The MIT projects, the projects of Karlsruhe as well as the Queue-Based Agent Architecture focused in their approach on another important requirement of the virtual receptionist: the real-time performance.

Unfortunately, all these presented systems neither integrate a calling system to hold conversations via phone, nor conduct a dialog with two parties within the same task. But just, these utterly necessary requirements of COSIMAR constitute the fundament for a serviceable receptionist system-architecture.



## 4 Technical Bases

COSIMAR is based on different third party vendor products, which are described more detailed in the following chapters. The decision for the use of these components is founded, besides its enclosing functionality, on economic and efficiency development purposes.

### 4.1 Speech Recognition and Text-to-Speech

The Microsoft Speech SDK contains both a Speech Recognition engine and a Text-to-Speech engine. The Speech Recognition engine WHISPER (Windows Highly Intelligent Speech Recognizer) was extended from the CMU Sphinx II recognizer [HA95] and covers the outlined requirements of the virtual receptionist such as, continuous speech recognition, speaker-independence, noise robustness, and context free grammars.

Furthermore, the Microsoft Speech SDK includes a speech application programming interface (SAPI), which is a software layer sitting between applications and speech engines to allow them communicating in a standardized way. SAPI covers among other things the following aspects [MS05]:

- Controlling audio input from a microphone, files, or a custom audio source and converting this input to a valid engine format.
- Loading grammar files and resolving grammar imports and grammar editing.
- Sharing of recognition across multiple applications using the shared engine, as well as all marshaling between engine and applications.
- Returning results and other information back to the application.
- Ensuring that applications do not cause errors – preventing applications from calling the engine with invalid parameters, and dealing with applications hanging or crashing

To handle the complexities of speech recognition and speech synthesis, and referring to minimize the programming, the Chant SpeechKit is used. The Chant SpeechKit consists of a component library, which supports different speech recognition and speech synthesis engines including Microsoft SAPI 5 and which is available for different programming languages, such as Java, C#, C++.



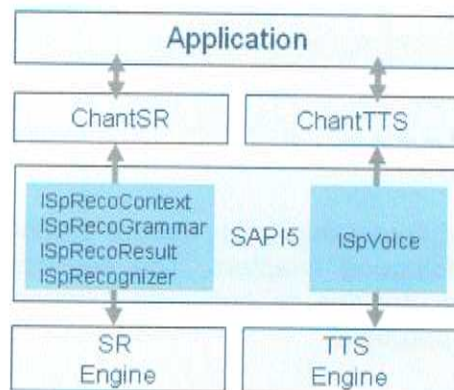


FIGURE 16: SpeechKit Architecture for Speech Recognition and Speech Synthesis [Ch96]

FIGURE 16 displays the Chant SpeechKit architecture. The session object *ChantSR* encapsulates all necessary speech recognition objects and handles low-level activities directly with a recognizer, so that the development of speech recognition applications is simplified. The application itself just receives the recognized speech as text and the notification of other processing states through event callbacks.

To define the context from which to recognize speech there are different possibilities Chant SpeechKit supports [Ch96]:

- Command* Command vocabulary consists of words and phrases that are spoken as commands
- Grammar* A grammar vocabulary consists of words and phrases and combinations of words and phrases.
- Dictation* Dictation vocabulary represents a dictionary of all possible words from which speech is recognized.
- Dictation topic* A dictation topic is a dictionary of words for a specific subject area. Topics improve recognition accuracy by increasing the probability the spoken words are in the domain of the topic dictionary.

Just like the *ChantSR*, the text-to-speech session object *ChantTTS* handles the low-level functions with text-to-speech engines and encapsulates the objects necessary for the synthesizing process.

## 4.2 Call System

One of the required system use cases of the virtual receptionist is the ability to conduct a dialog via phone. To realize the phone call we decided to use a third party vendor product.

All Microsoft Windows operating systems include a telephony application programming interface (TAPI) which "abstracts the call-control functionality and exposes a common interface to applications" [MS99]. TAPI constitutes an entry point, which represents all telephony resources of the local computer, so that services provided by telephony vendors can be used. With the aid of TAPI, a telephony application can be written, which dials telephone numbers and handles media tasks such as speech recognition, text-to-speech and tone-detection.

To realize the programming of the telephony application, on top of TAPI, the SpeechStudio TAPI control [SS03] is used, which implements the TAPI control interfaces. It supports TAPI devices, which dial regular phone numbers and maintain audio streaming. This control also includes a telephone event mechanism which notifies the application about ringing, caller ID, connect status, hanging-up, and touch-tone (DTMF) input and output. Furthermore, the integrated telephony connects the SAPI 5 speech recognition and text-to-speech to the telephony devices.

Whenever a SpeechStudio telephony application is started, the SpeechPlayer component of the SpeechStudio Suite is launched. Among other functionalities, it provides the voice interface and checks the system for a TAPI capable device and thus enables its support for telephony. FIGURE 17 shows the SpeechStudio architecture for speech recognition, text-to-speech and telephony.

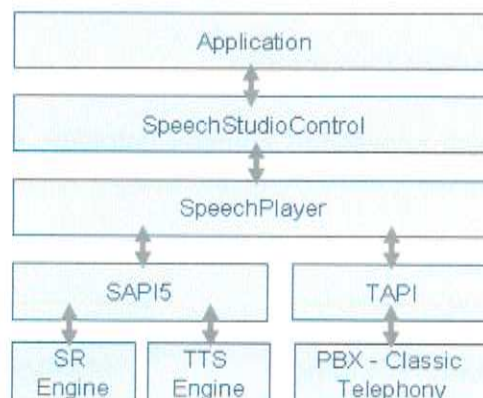


FIGURE 17: SpeechStudio Architecture for Speech Recognition, Speech Synthesis and Telephony



## 4.3 Natural Language Understanding and Dialog Management

There are several requirements to the dialog system for the COSIMAR. Primarily, it should support a natural dialog, so that the user not only achieves its goals through telling phrases or yes/no-statements, but rather through complete sentences. In order to that, it's important to have language understanding algorithms allowing incorrect speech recognition and incorrect grammatical utterances. Since a user not always discloses all necessary information within one utterance, it's important to have the ability to process partial information and to provide a dialog management gathering the missing information in sub dialogs. Furthermore, the dialog management needs to communicate with external applications, such a database storing the information about the employees and customers, the Social Agents FATA system generating the audio answer and the animation of the virtual character, and the telephony application realizing the connection to the employee. Comprising, there is the need of expressive knowledge representations and a dialog management that interacts with external applications and the user through the information content disclosed by the user.

The dialog system TAPAS of the University Karlsruhe meets these requirements. Moreover, it is language and domain independent, and it realizes a separation of domain dependent knowledge and interactive specific knowledge – the dialog state. TAPAS works task orientated, goal based and turn based. As semantic knowledge representation multidimensional typed feature structures are used. Thus, the overall dialog system comprises natural language understanding, dialog management, and answer generation. The next section gives an introduction to the dialog system. More detailed information can be found in the dissertation of M. Dennecke who invented the dialog system ARIADNE [De02], TAPAS derived from.

### 4.3.1 Domain dependent knowledge

TAPAS uses the different knowledge sources ontology, dialog goals, semantic context free grammars, data bases and generation templates, as displayed in FIGURE 18.

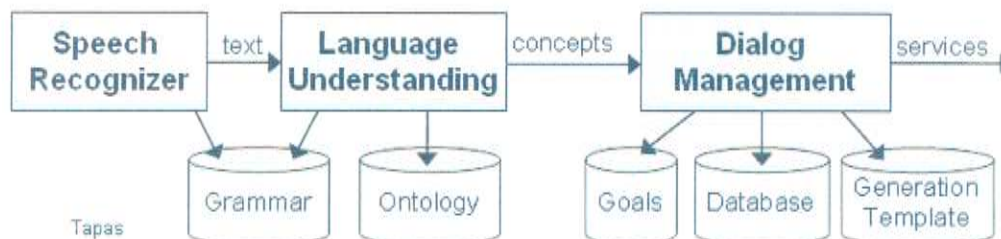


FIGURE 18: Knowledge Sources of the Dialog Manager TAPAS



## Ontology

The ontology introduces the concepts which the system needs to know in order to understand the user's utterance. There are the concepts: *object*, *action*, and *property* which can inherit of each other and thus built up concept hierarchies. The following example illustrates a simplified ontology:

```
class act_greet inherits generic:action;
class act_howAreYou inherits act_greet;

class obj_customer inherits generic:object{
    base:string : NAME;
};

class obj_employee inherits generic:object{
    base:string : NAME;
    base:string : TELEPHONE;
};

class act_mediation inherits generic:action{
    obj_customer : CUSTOMER;
    obj_employee : EMPLOYEE;
};
```

## Grammar

The user's utterance received from the speech recognizer is analyzed according to the rules of a semantic context free grammar. This has the advantage that besides syntactic information also semantic information is extracted. But to allow the reuse of domain independent information specified in syntactic grammars, TAPAS uses *vectorized context free grammars*. These grammars can also be used by the speech recognizer, and thus restricting the search space of a grammar based speech recognizer through information given by the dialog manager [FH04]. The following snippet shows a vectorized context free grammar specification:

```
<act_customerIdentification,VP,_>
    = <identification,V,_>*
      <obj_customer,N,_>{CUSTOMER obj_customer}

<identification,V,_> = 'I am' 'called'*
                      : 'my name' 'is'*;

<obj_customer,N,_> = 'Brown';
```

Through the grammar specification, the user's utterance can be mapped onto a typed feature structure, which represents the semantic input. Furthermore, typed feature structures are used to model the discourse state, referenced object representations and dialog goal descriptions. Thus they represent the semantic knowledge of the dialog. FIGURE 19 shows an example of the typed feature structure:

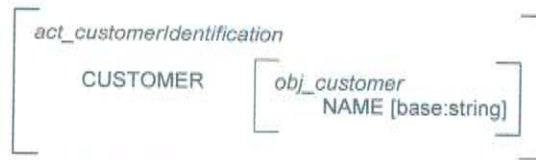


FIGURE 19: Typed Feature Structure

## Database

TAPAS allows retrieving information from a database [HG04]. Thus, by using databases it is possible to separate grammar development from database information and to resolve references of nominal phrases in grammars. The following snippet visualizes the integration of an object database, which stores the names of the customers, and which was compiled into java classes by the dialog manager:

```

database Customer obj_customer jpkg://localhost:5454/Customer?jpkg
{
  dbtable Customer obj_customer {
    dbfield name = [NAME];
  };
};

<obj_customer,N,_> = import jpkg://localhost:5454/Customer?jpkg
  Customer name {NAME import};
  
```

## Dialog Goals

A dialog goal is the link between the proceeded dialog and the services, which should be invoked to realize the user's goal. The dialog goal specification can be considered as a description of a filled out form.

The following example describes the goal *mediation*, which is reached when the customer disclosed his name and the name of the employee he wants to talk to. To execute the goal, the dialog manager communicates with the required applications:

```

goal mediation{
  precondition:[ act_mediation
    CUSTOMER [ obj_customer
      NAME[ base:string ]
    ]
    EMPLOYEE [ obj_employee
      NAME[ base:string ]
    ]
  ]
  -> bindings:
    jpkg://localhost:5454/answerSocialAgent "I will call your
      meeting party. Please be patient.";
    jpkg://localhost:5454/callEmployee
      $objs.[EMPLOYEE|TELEPHONE];
};
  
```

### Generation Templates

Generation templates are used to form the spoken output. They define which concepts of the typed feature structures should be generated into natural language. Different components can be used to generate the output:

```
Strings "I will call your meeting party."  
Path references $objs.[EMPLOYEE|NAME]  
Labels #Welcome  
Python scripts °{result = DlmFacade.formatTTSOutput(objs,  
"NAME")}°
```

### 4.3.2 Dialog State

To realize language and domain independent dialog strategies TAPAS provides an abstraction of the actual dialog state. The description of the dialog state consists of different status variables [HG04, Ho03] :

<i>Intention</i>	estimated level of the user's intention
<i>TurnQuality</i>	estimated quality of the semantic representation of the actual input
<i>OverallQuality</i>	estimated whole quality of the proceeded dialog
<i>SpeechAct</i>	speech act of the actual utterance
<i>Reference</i>	state of the process for solving references
<i>ReferringExpressions</i>	describes the relations of the referred discourse expression to the objects
<i>HoldState</i>	a single wrongly recognized utterance does not lead to aborting the dialog goal

To gather information necessary to reach a dialog goal, a dialog strategy is used. The dialog strategy defines the actions to be executed depending of the dialog state, such as database queries, adaptation of the discourse regarding new information and clarification questions. This strategy constitutes an association between a state of the dialog and – in our case – the virtual character to get more information for reaching the goal. To adapt the discourse there are different interaction pattern, which define the information exchange. There are four base types:

<i>Questions</i>	information getting
<i>Corrections</i>	deletes information
<i>Substitution</i>	replaces discourse information
<i>State</i>	internal state transition



## 4.4 Social Agents FATA

As already pointed out in the previous chapters, a user interface with a pleasant, human-like appearance and appropriate facial expressions is an essential requirement for COSIMAR. SAP's Feed Automatically Talking and Animate (FATA) architecture [DY03] offers a solution regarding these issues. With FATA it is possible to create web applications that need to display a speaking and appropriate animated human-like talking head.

The FATA system comprises four different tiers: Client Tier, Application Tier, Pulse Web Services Tier and Text-to-Speech Tier.

The task of the TTS Tier is generating sound files from a given text. The Pulse Web Services Tier runs on a proprietary server that must reside on a configured Linux 8.0 server. It communicates with the TTS Tier and creates a pulse audio file (\*.pwc), which is available for a SOAP-compliant client via an URL. The FATA web application runs in a Tomcat server and communicates via SOAP with the Pulse Web Service. Concerning the Client Tier, a web browser is required, which supports the JavaScript Pulse Veeper plug-in to play and animate the talking head.

The animated talking head is created from any image. To do so, you have to use the Pulse Veeper Tool, where you can define areas of the face, like mouth, eyes and eye brows. Then the software calculates the user input and creates a character, which is able to do facial expression and communicate over the internet [PW05].

FIGURE 20 gives a detailed overview about the FATA Web Application Tier Architecture, which is the main entry point for web clients.

The *Web Content Subsystem* of the FATA system includes static and dynamic HTML content. The dynamic contents such as JSP pages can access the services of *FataDataFactory*, which provide access to the Web container, *PulseBehaviorHandlerClient*, *FataURLCache* and *FataData* container.

The *FataData* comprises for example information about the emotions and expressions of the talking head, and the text to say. It is stored in an XML file, which is dynamically loaded at runtime. The following snippet shows an example of the XML-description:

```

< talking-head-data name="Introduction" emotion="neutral"
    voice="Mary">
    < speech-files type="set">
        < speech-data type="static" value="Hello." />
        < speech-data type="static" expression="Smile"
            look="downright"
            value="I am Cosimar the Virtual
                Receptionist." />
        < speech-data type="static" expression="confused"
            emotion="happy"
            value="Who are you?" />
    < /speech-files>
< /talking-head-data>

```

The *FataURLCache* is used to speed up identical requests through pointing to previous used audio files on the shared-storage of the Pulse Web Server.

The task of the *PulseBehaviorHandlerClient* is to marshal all client requests to the Pulse Web Service Server and turning them into SOAP payloads.

Responsible for initializing, destroying or tearing down all these FATA subsystems is the *FataDataServlet*.

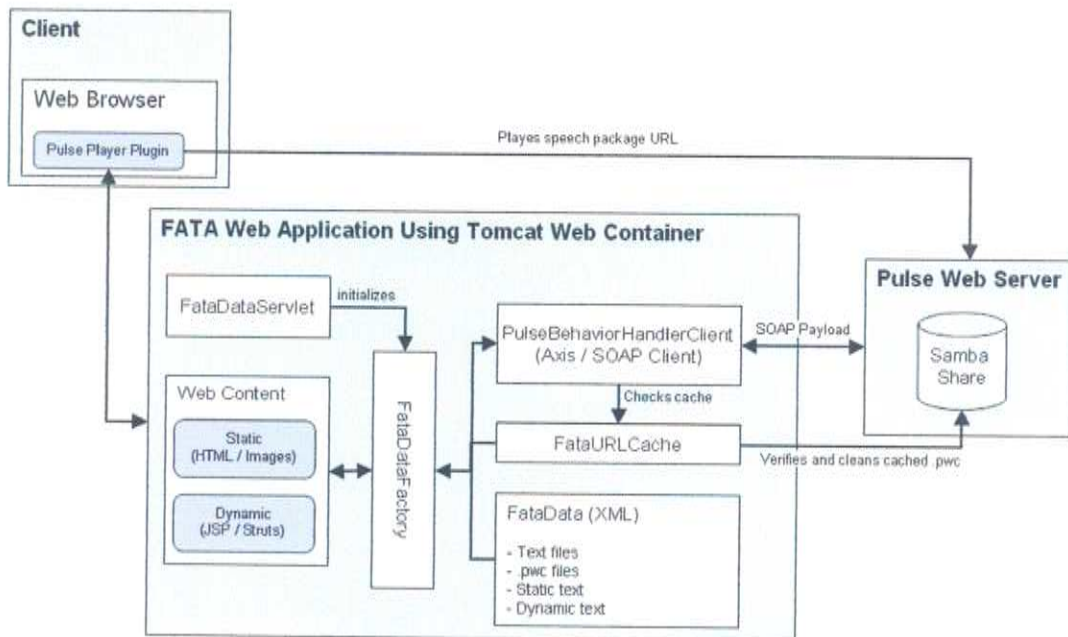


FIGURE 20: Application Tier Architecture [DY03]





## 5 Architecture

This chapter describes the software architecture of COSIMAR. The main focus lies on the interaction of the different software components.

### 5.1 Components of the Virtual Receptionist

The architecture of COSIMAR comprises different components, which are realized with different systems characterized in the previous chapter.

On the one hand, different components have to work together to realize the interaction with the customer. These components are *Sensing Component* (Sensing-Comp), *Speech Recognition Application* (SR-App), *Dialog-Customer Component* (D-C-Comp) and *Social Agent Component* (SA-Comp) encapsulated in *Customer-Receptionist Component* (C-R Comp). The global knowledge sources required for these components are:

- grammar: defines what can be said within the reception dialog
- database: stores the information about the meetings, employees and customers

On the other hand, there is the need of a component realizing the interaction with the employee: *Employee-Receptionist Component* (E-R Comp), which contains a *Dialog-Employee Component* (D-E-Comp) and a *Telephony Application* (T-App). The required knowledge sources are a grammar, which identifies what can be said in the telephony dialog with the employee, and a data storage keeping the mediation information.

The coordination of both main components is handled by the *Virtual Receptionist Control* (VR-Control). It is informed by the Sensing-Comp when a customer reaches the kiosk. Thereupon, it starts, resumes, pauses or stops the Customer-Receptionist Component, the Employee-Receptionist Component and its sub components respectively.

In FIGURE 21 an overview about the interaction and interfaces of the components is given.

### 5.1.1 Virtual Receptionist Control

The VR-Control realizes the interaction of both main components by using its interfaces and these of the subcomponents to start, stop, pause and resume them.

On its part, it provides three interfaces. The first is an event handler, which receives the Event1, when the Sensing-Comp senses the user. If the D-C-Comp wants the VR-Control to call the employee, it sends the mediation request to the port 2001. The mediation result, which should be told to the customer, is send respectively by the D-E-Comp to the port 2002.

For the D-E-Comp, the VR-Control generates the grammar depending on the mediation request of the C-R-Comp. In this way there is a more limited grammar supporting the speech recognition on the phone.

The advantage of having a VR-Control on top of the architecture is founded on the requirement supporting modifiability. This control allows easily adding other components which could for example print the visitor badge or provide company information by playing a video.

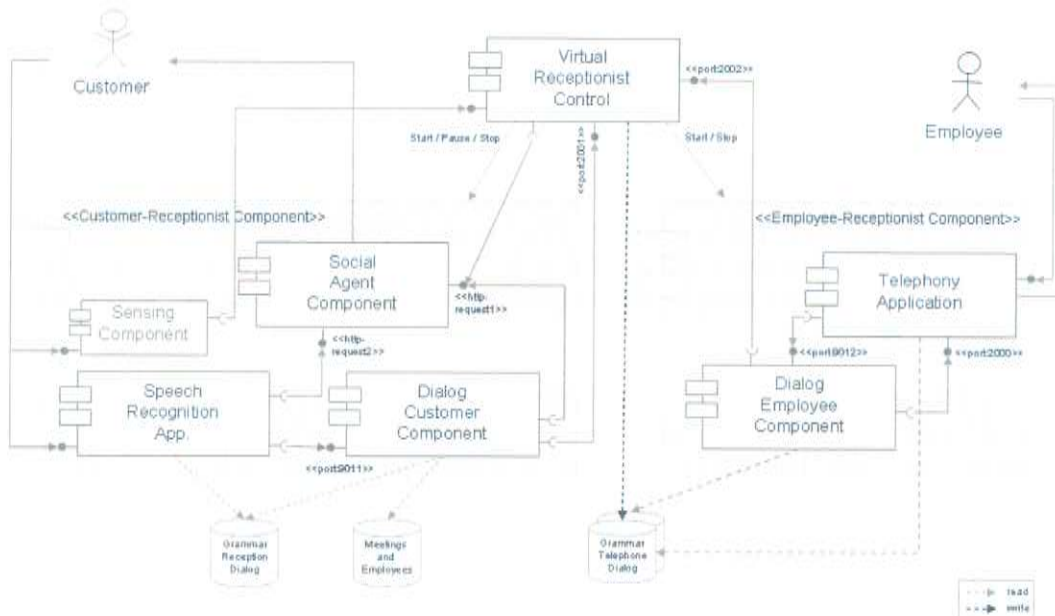


FIGURE 21: UML-Component-Diagram of COSIMAR



## 5.1.2 Customer-Receptionist Component

This component ties all required components for the interaction of the virtual receptionist with the user. It is responsible for starting, stopping, pausing and resuming of its embedded components. In this way both the modifiability of its subcomponents and the reusability of its entirety functionality are lightened.

To not affect the performance of the system negatively, some subcomponents communicate directly with the VR-Control but not with the C-R-Comp. In this way parts of modifiability will be lost in favor of performance.

### Sensing Component

The sensing component senses the approaching customer by an infrared sensor. If it senses that a person approaches the kiosk, it will send an event to the VR-Control. While the customer stands at the kiosk, it is in listening mode, until the person leaves the kiosk. Thereupon it sends again an event to the VR-Control.

### Speech Recognition Application

The speech recognition application realizes the recognition of the user's utterance. It is realized with the Chant SpeechKit. To support a satisfactory recognition and thus a good availability and usability, a grammar is used, which only defines those words and utterances, the DC-Comp understands.

On the one hand, the SR-App requests the SA-Comp, to display the recognition result to the user. In this way, the user is aware of what the system has recognized and may understand when misrecognition leads to a wrong behavior of the virtual receptionist. On the other hand, the SR-App sends its result to the port 9011, which is provided by the DC-Comp, so that an answer for the user can be generated.

### Dialog Customer Component

This component realizes the natural language understanding, dialog management and natural language generation. Generally, it realizes the dialog between the customer and COSIMAR. As interface, it provides a port, where the user input can be sent in form of a string. This input is processed, and thereupon the DC-Comp requests the SA-Comp to generate the audio file and the animation by sending a textual answer.

The D-C-Comp is implemented with the TAPAS system and uses a grammar which is shared with the SR-App. Furthermore there is the need of data storage where the information about the customer, employee and meetings are kept.

To invoke the telephone call and accordingly to receive the mediation result, the D-C Comp sends an request over an provided socket (port: 2001) to the VR-Control, after all necessary information for mediation has been gathered from the customer.

### Social Agent Component

The speech output, the visual representation with appropriate animation and the display of the speech input will be realized by the SA-Comp. It provides



interfaces for both the recognition result and the answer that the virtual receptionist wants to communicate to the customer. These interfaces are accomplished by a JSP site.

By this means we realize a user interface with a pleasing 3D humanlike face which adapts facial expressions depending on its audio response, and a display of the speech input giving the user feedback about the recognition of his utterance.

### **5.1.3 Employee-Receptionist Component**

In this component Telephony Application and Dialog Employee Component are encapsulated for the interaction of COSIMAR with the employee. Just like the Customer-Receptionist Component, it is responsible for starting and stopping, of its subcomponents, and so modifiability and reusability are supported.

#### **Dialog Employee Component**

Compared to the D-C Comp, the D-E Comp has the same task, namely realizing a dialog though between the employee and the virtual receptionist.

Not only because this dialog pursues other goals within a different situation and with other kind of users, but also because this dialog should be conducted while another dialog takes place, we decided to decouple both dialogs and to create two dialog components. In this way modifiability and reusability of the two dialog components, of the Employee-Receptionist Component and Customer-Receptionist Component respectively can be supported.

Another advantage in decoupling both dialogs consists in security. Since both dialogs are divided, the user has no chance to outwit the system by saying something the employee could say or to found out, what the employee said to the receptionist.

The D-E Comp also provides a port to where the T-App can send its recognition result. The port number differs from the one of D-E Comp, because both components run on the same machine. D-E Comp sends its answer back to the T-App, which offers also a socket as interface.

The used data storage of the D-E Component are a grammar generated for the telephone dialog and the mediation object storing the required mediation information such as the telephone number, customer and employee name.

After the required mediation result was gathered, the D-E Comp sends the mediation result object to the VR-Control over the socket port 2002.

#### **Telephony Application**

The T-App realizes the telephone call to the employee and provides a speech user interface. It uses the Speech Studio which realizes the call functionality, speech recognition and text-to-speech respectively.

On the one hand, it provides an interface, port 2000, to where an answer from the D-E Comp can be send. On the other hand, it uses the interface of the D-E Comp to send the recognition result.

Since the speech recognition is poorer at the phone than per microphone, we need to use a quite limited grammar which was generated by the VR-Control.

## 5.2 Application Layering

The architecture of COSIMAR consists of three layers presented in FIGURE 22. Both the Java Virtual Machine and .NET runtime provide a basis for the system. Above these is the Basic Services layer, which includes Customer-Receptionist Component and Employee-Receptionist Component with their subcomponents. The SA-Comp, included in the Customer-Receptionist Component, is running in a Tomcat Server. On top is the Application Control layer, which includes the Virtual Receptionist Control. The advantage of application layering lies in modifiability and maintainability of the system.

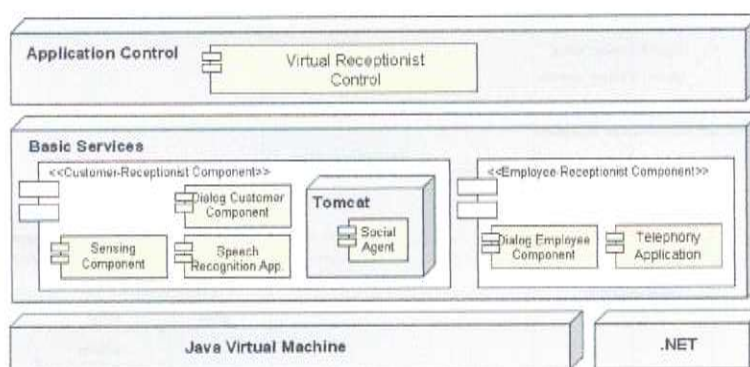


FIGURE 22: Application Layering

TABLE 3 gives a summary of the components, their addressed quality attributes, implementation language and assigned layers.

Component	Quality Attributes Addressed	Implementation	Layer
Virtual Receptionist Control	Modifiability	Java	Application Control
Customer-Receptionist Component	Modifiability, Reusability	Java	Basic Services
Sensing Component	User Interface		Basic Services
Speech Recognition Application	User Interface	Java Chant	Basic Services
Dialog Customer Component	Reusability	TAPAS, Java	Basic Services
Social Agent Component	User Interface	JSP, JavaScript, Java	Basic Services
Employee-Receptionist Component	Modifiability, Reusability	Java	Basic Services
Dialog Employee Component	Reusability	TAPAS, Java	Basic Services
Telephony Application	User Interface	C# SpeechStudio	Basic Services

TABLE 3: Summary of the Components



### 5.3 Interaction and Process Flow

Now, let's gain insight the interaction of the components and the process flow. For this purpose take a look at the sequence diagram in FIGURE 23.

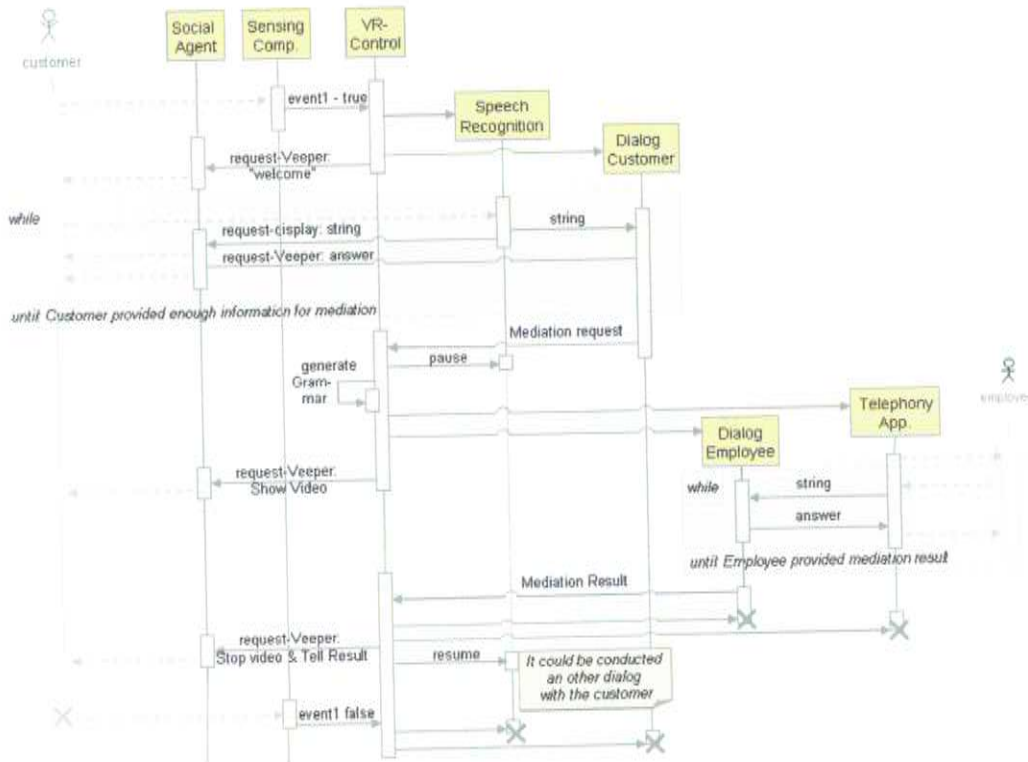


FIGURE 23: Sequence Diagram of the Components

The three components *SA-Comp*, *Sensing-Comp* and *VR-Control* are alive after COSIMAR was started. Since the *VR-Control* is the controller of the whole application, it goes without saying, that it is present all the time. The *Sensing-Comp* is also required to be alive to observe the receptionist environment: if there is a customer or not. The decision, having the *SA-Comp* present all the time, follows from the requirement to attract the customer.

If a customer approaches the kiosk, an event with value *true* is sent, to inform the *VR-Control*. Thereupon it starts the *SR-App* and *D-C-Comp*. Furthermore it sends an http-request to the *SA-Comp* to welcome the customer and ask for his intentions. In this way a dialog between COSIMAR and the customer begins. The customer says something and the *SR-App* processes it. Then the recognition result is send to both the *SA-Comp* via http-request and the *D-C-Comp* via a socket-connection on port 9011. The appearance of the recognized speech input on the screen in a textual style, realized by the *SA-Comp*, is important for



usability reasons, since nowadays speech recognition does not yet meet the desired functionality, and thus the customer will need a response if his utterance has been misrecognized or not. The D-C-Comp is responsible for processing the recognition result and sending an answer to the SA-Comp, so that the virtual character will answer and be animated. The loop of recognizing speech input and generating an answer will proceed until the customer provides all necessary information for mediation.

If all mediation information has been received from the customer, the D-C-Comp sends a request to the VR-Control with the mediation object. The prototype version COSIMAR does not plan to conduct another dialog with the customer while the virtual receptionist has a phone call with the employee. Even though no speech input from the customer is expected, the SR-App needs to be paused to avoid misrecognitions. However, the architecture would also allow realizing two dialogs simultaneous.

Instead of conducting two dialogs, the customer will be entertained with a video about the actual projects of his meeting party's department. Thus the VR-Control will request the SA-Comp to display the video.

Before the Virtual Receptionist can call the employee, a limited grammar for both the *T-App* and *D-E-Comp* needs to be generated by the VR-Control. This step is required to support the recognition proceeding at the telephone. Now, the prerequisite is fulfilled and both components can be started.

The T-App calls the employee and if he picks up, COSIMAR greets, tells that someone is waiting for him in the lounge and ask when he has time to pick him up. If the employee answers, the recognition result will be sent via a socket-connection on port 9012 to the D-E-Comp. This component processes the input and sends an answer back to the T-App via a socket-connection on port 2000. This procedure takes until the employee tells the mediation result. Thereupon, this mediation result is processed to the VR-Control via a socket-connection on port 2002. This result will also be send, if for example the employee did not pick up the phone. The content of the mediation result just differs.

After that, both components, T-App and D-E-Comp, are stopped by the VR-Control. Furthermore VR-Control requests the SA-Comp to tell the mediation result and resumes the SR-App, so that another dialog with the customer can be carried out again. If the customer leaves the kiosk, the Sensing-Comp will send an event with value *false* to the VR-Control, so that it stops both D-C-Comp and SR-App. This "no-customer-at-the-kiosk" event could also be send during the whole procedure, but then the VR-Control has to react in another way depending of its actual status.



## 6 Design and Implementation

### 6.1 Virtual Receptionist Control

The *VR-Control* is designed with the mediation pattern [DO05a]. In this way, all objects required to fulfill the mediation task can interact with each other and be varied in their interaction independently. This allows modifiability for enhancing the functionality of the Virtual Receptionist. FIGURE 24 gives an overview about the class diagram of the VR-Control.

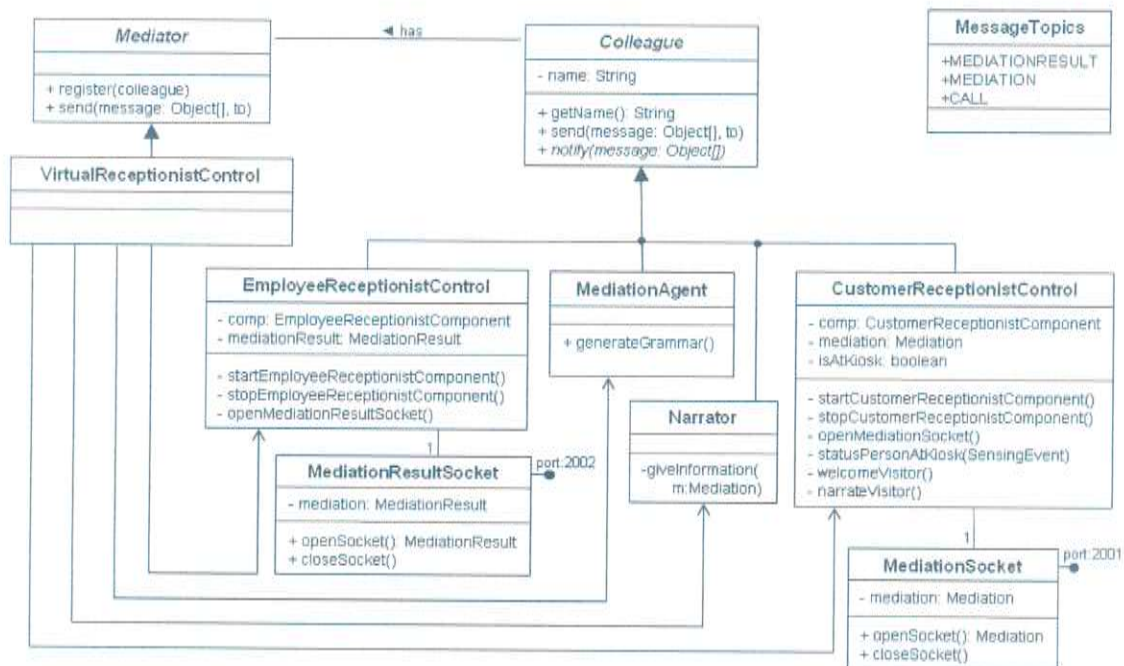


FIGURE 24: UML Class Diagram of the Virtual Receptionist Control

The mediation pattern suggests having several *Colleagues* and one *Mediator*, which registers the colleagues and sends the messages between them. In this case, the *VirtualReceptionistControl* is the mediator and encapsulates all registered colleagues.

The *Colleagues* send on the one hand messages to the *Mediator* and define to whom they should be sent. And on the other hand they notify the messages from the *Mediator* and react accordingly. The main communication of the Virtual Receptionist occurs between the *EmployeeReceptionist-Control* and *CustomerReceptionist-Control*.



As described in the architecture chapter, the *VirtualReceptionistControl* provides interfaces to notify when mediation is requested and the mediation result received respectively. These interfaces are realized by both a *MediationSocket* which is used by the *Customer-ReceptionistControl* and a *MediationResultSocket* which is used by the *EmployeeReceptionist-Control*.

Furthermore there are the colleagues *MediationAgent*, which generates the grammar for the E-R-Comp, and the *Narrator* which entertains the customer, while he is waiting, with a video about the actual projects of the meeting party's department.

To know which message can be understood and received there is the object *MessageTopics* that encapsulates all message titles.

As outlined in the requirement analysis and architecture chapter, there is a need of a *Sensing Component*. Unfortunately the technical resources for this prototype have been limited, so that this component couldn't be implemented. But it could be easily added by creating another *Colleague*, which sends a message to the *CustomerReceptionist-Control* if a visitor is sensed or not.

## 6.2 Basic Services

As outlined in the architecture chapter, the *Basic Services Layer* contains several components. All these components have in common that they need to be started and stopped, and they are executed by a single thread. Therefore we introduce the interface *RunComponent* which extends *Runnable* [Ja03], by adding the method *stopComp()*, to stop the corresponding component. This interface is implemented by the components *CustomerReceptionistComponent*, *EmployeeReceptionistComponent*, and *RunExe*, as you can see in FIGURE 25.

*RunExe* realizes the start and stop of the components which are executable programs started by a single exe-file. These components are for example the telephone application and the dialog. To realize this task there is a need of two *StreamGobbler* [Da05], which reads either the *InputStream* or the *ErrorStream* of the process started.

*RunExe* is used by both *CustomerReceptionistComponent* and *Employee-ReceptionistComponent*, because they are responsible for managing the lifetime of its included components.

Since there is only the need of a single *C-R-Comp* and a single *E-R-Comp* on the desktop, both are realized with the *Singleton Pattern* [DO05b].

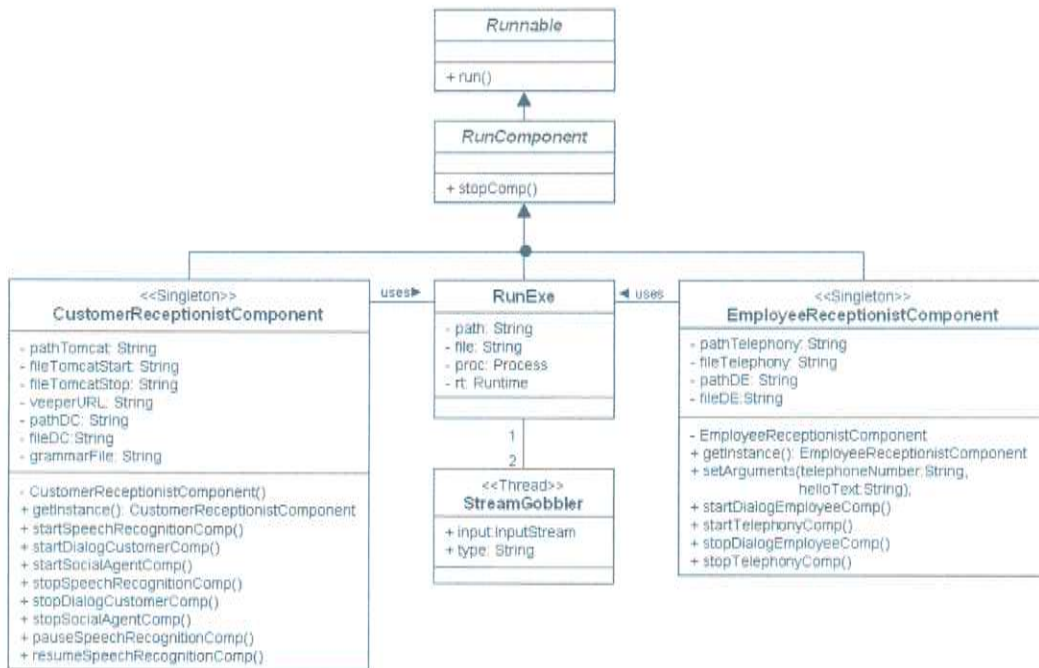


FIGURE 25: UML Class Diagram of Basic Services

### 6.2.1 Social Agent

For the customer, we need an appealing graphical user interface representing the receptionist and its environment. Mainly we need to display a human like character having facial expressions while speaking. This is realized with the Social Agents FATA. The image we used for the Social Agent was edited from some pictures found at the web pages DerivantART [Am05a] and [Am05b]. Since we want to entertain the customer with a video about the actual SAP projects while he is waiting, we will display a TV on the screen. To give the customer feedback of what the system recognized, we display the speech input as text beneath the Social Agent. The user also needs to have feedback, if the utterance couldn't be recognized with a high confidence (the speech recognizer's hypothesis). In this case the virtual receptionist won't answer and the speech input is displayed as text in the following form: "Did you mean: <text>? Please repeat." to the customer. The GUI we designed for COSIMAR is displayed in FIGURE 26.



FIGURE 26: COSIMARs Graphical User Interface

Because we decided to use the SAP Social Agents, we need to run the application in an internet browser. The challenge here is to update the screen without having user events directly on the screen like mouse clicks or keyboard input, but through the speech input.

For updating the internet sites we use the pushlet mechanism described in [Br02]. The idea of pushlets is sending continuously JavaScript content back to the client in a timer loop from a JSP (or Servlet). "In other words, we just stream



in lines of JavaScript from a JSP or Servlet. These lines get interpreted by the browser who may do something interesting.”

Applying pushlets to our needs, we implement a parent JSP-site (VR\_veeper.jsp) displaying the Social Agent and the TV, and embedding another JSP-site (VR\_speechInput.jsp) that displays the utterance of the user. To update the user speech input, VR\_veeper.jsp has the JavaScript method pushSpeechInputDisplay(time) that updates VR\_speechInput.jsp in regular time loops. Whereas the method pushSpeechInputDisplay(time) is called, when VR\_speechInput.jsp is executed. You can find the shortened snippets of these JSP-sites in TABLE 4 and TABLE 5, which illustrate this mechanism.

```

VR_veeper.jsp

<html>
...
<script LANGUAGE="JavaScript">
    function pushSpeechInputDisplay(time) {
        setTimeout("document.getElementById('speechInput').src=
            'VR_speechInput.jsp'", time);
    }
</script>

<body bgcolor="ffffff" border="0" leftmargin="0" rightmargin="0">
...
<iframe src="VR_speechInput.jsp" name="speechInput" frameborder="0"
    width="587" height="100" scrolling="no"></iframe>
...
</body>
</html>

```

TABLE 4: Snippet of VR\_veeper.jsp

```

VR_speechInput.jsp

<%@ page language="java" %>
<%@page import="VirtualReceptionist.BasicServices.SocialAgent.*" %>
<HTML>
<%
    int time = 300;
    String input = VRSocialAgentData.getSpeechInput();
    if(input == null){
        input="";
    }
%>
<BODY>
YOU: <i><%=input%></i>
<script language=JavaScript>
    parent.pushSpeechInputDisplay(<%=time%>)</script>
</BODY>
</HTML>

```

TABLE 5: Snippet of VR\_speechInput.jsp

The open question remaining is how the `VR_speechInput.jsp` knows about the updated speech input. As mentioned in architecture chapter [Components of the Virtual Receptionist], the Social Agent component provides an interface for both the speech recognition component and the dialog management component through a JSP-site which can be requested. Within this site the answer is saved in a JavaBean (`VRSocialAgentData`). Through the pushlet mechanism, the content of JavaBean can be read in regular time intervals and be displayed on the screen. The same technique is used for playing COSIMAR answers and displaying the video.

With the help of the Pulse Veepers Tool you can create your own Social Agent which is saved as a proprietary `.pwr` data. The FATA system offers an easy integration of this data through its FATA tag library. You just add the following tag:

```
<fata:renderVeeperJS path="veepers/newLady/veeper.pwr"/>
```

To let the Social Agent speak you can add one of the following tags:

```
<fata:renderFataJS name="Introduction"/>
<fata:renderRawJS voice="Mary" text="Hello Mr. Green."/>
```

The difference between these tags is, that the value of the attribute `name` of the tag: `<fata:renderFataJS name="Introduction"/>` points to the identification of the talking-head-data stored in an XML-file explained in chapter Social Agents FATA. Whereas the value of the attribute `text` of the tag: `<fata:renderRawJS voice="Mary" text="Hello Mr. Green."/>` is the text, which should be said directly by the Social Agent.

The advantage in using an XML is that the static text can be stored in one repository. But to allow the Virtual Receptionist saying something dynamically, like varying names and locations, we use the tag `<fata:renderRawJS />`.

## 6.2.2 Speech Recognition Application

The speech recognition enables recognizing the customer's utterances. The application uses the Chant SpeechKit, which provides an interface to the SAPI by the session object *ChantSR*. The *ChantSR* encapsulates all necessary speech recognition objects and handles low-level activities directly with a recognizer [Ch96].

When starting the application, a grammar file is assigned. This grammar file provides the base for the recognition. Thereupon the speech engine is initialized and the recognition can be started (*startRecognitionGrammar*).

To notify the application of a speech recognition event, *SpeechRecognition* implements *JChantSREvents* and its method *hasEvent*, as you can see in FIGURE 27.

If a grammar is used, the events the application can notify are *CCSRHasPhrase* and *CCSRHasPhraseHypothesis*. *CCSRHasPhrase* is the speech recognition engine's highest confidence about what was said in the last utterance from a grammar vocabulary, whereas *CCSRHasPhraseHypothesis* is just the speech recognition engine's hypothesis.

Since the *CCSRHasPhraseHypothesis* has not such a high confidence, this phrase is send to the *Social-Agent-Component* (*postSRResultToDialogManager*), to display the user his potential phrase which he should repeat.

If the user's utterance could be recognized with the highest confidence, then the result is send to the dialog manager to process it.

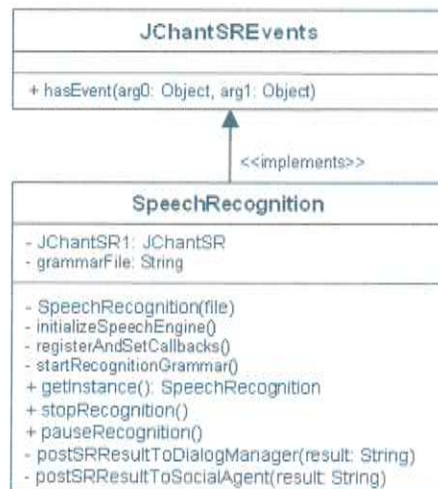


FIGURE 27: UML Class Diagram Speech Recognition

The *SpeechRecognition* is started by the *Customer-Receptionist-Component*. To allow it stopping and pausing, there are the public methods *stopRecognition* and *pauseRecognition*.



### 6.2.3 Dialog Management for the Customer and Employee Interaction

As outlined in chapter *Natural Language Understanding and Dialog Management* we need to have expressive knowledge representations and a dialog management interacting with external applications and the user through the information content disclosed by the user. These requirements are met by the dialog system TAPAS. Thus in the next steps the ontology, grammar, database, dialog goals, dialog moves and the interfaces to the other applications are defined.

In the previous chapter 4.3.1 examples regarding the ontology, grammar and database were given; therefore we pass on this and continue describing the dialog goals, dialog moves and interfaces for the receptionist application in more detail.

For establishing the mediation, the system needs to know the name of the customer and either the name of the employee or the name (or time) of the meeting. Therefore we identify two dialog goals of the mediation task displayed in TABLE 6.

If the necessary information is provided for one dialog goal, it is executed. Thereby it invokes the next actions to be taken by calling methods of a java application assigned to the dialog manager. Within both mediation dialog goals the method `mediationRequest` is called. The assigned parameters are the values of the mediation object specified in FIGURE 4. Thus within the method `mediationRequest` the mediation object is created and the next actions for calling the employee are taken. That means the mediation object is send to the Virtual Receptionist Control over the provided socket connection, and thereupon the next actions are executed by the VR-Control.

#### Dialog goals of Mediation

```
goal mediation{
  precondition:[ act_mediation
                 CUSTOMER [obj_customer
                           C_LASTNAME[ base:string ]
                           ]
                 EMPLOYEE [obj_employee
                           E_LASTNAME[ base:string ]
                           ]
                 ]
  -> bindings:
    jpkg://localhost:5454/mediationRequest
    $sem.[CUSTOMER|C_FIRSTNAME], $sem.[CUSTOMER|C_LASTNAME],
    $sem.[EMPLOYEE|E_FIRSTNAME], $sem.[EMPLOYEE|E_LASTNAME],
    $sem.[EMPLOYEE|TELEFON], $sem.[EMPLOYEE|DEPARTMENT];
};
```

```

goal mediationMeeting{
  precondition:[ act_mediation
                 CUSTOMER [obj_customer
                           C_LASTNAME[ base:string ]
                           ]
                 MEETING [obj_meeting
                           NAME[ base:string ]
                           ]
                 ]
  -> bindings:
    jpkg://localhost:5454/mediationRequest
    $sem.[CUSTOMER|C_FIRSTNAME], $sem.[CUSTOMER|C_LASTNAME],
    "null", "null",
    $objs.[MEETING|TELEPHONE], $objs.[MEETING|DEPARTMENT];
};

```

TABLE 6: Dialog Goals of Mediation

Since not always, all information necessary for a dialog goal is provided by the user within one utterance, it is necessary, to ask for the missing information. This is realized by the dialog moves. Some of the dialog moves necessary for requesting the missing information are displayed in TABLE 7.

To speak to the customer, the Dialog Component needs to request the Social Agent Component. This is realized by the methods `requestSocialAgent` and `requestSocialAgentRawJS` of the java application. Thereby the prefix `FataJS` and `RawJS` respectively specify whether the predefined text stored in an XML file should be said, or if the audio file should be generated directly from the text [vs. 58].

#### Dialog moves of Mediation

```

move Mediation_AskForIdentification on variable Intention changed to
  selected {
  goal:(mediation = selected),
  path:($sem.[CUSTOMER|C_LASTNAME]is undefined),
  path:($sem.[EMPLOYEE|E_LASTNAME]is defined),
  -> bindings:
    jpkg://localhost:5454/requestSocialAgentRawJS "I will inform ",
    $sem.[EMPLOYEE|E_LASTNAME], " that you are waiting for him.
    Would you please tell me therefore your name?";
};

move Mediation_AskForContactPerson on variable Intention changed to
  selected{
  goal:(mediation = selected),
  path:($sem.[CUSTOMER|C_LASTNAME]is defined),
  path:($sem.[EMPLOYEE|E_LASTNAME]is undefined),
  path:($sem.[EMPLOYEE|E_KNOWN]is undefined),
  -> bindings:
    jpkg://localhost:5454/requestSocialAgent FataJS:askContact;
};

```

```

move Mediation on variable Intention changed to finalized (
  goal:(mediation = finalized),
  path:($sem.[CUSTOMER|C_LASTNAME]is defined),
  path:($sem.[EMPLOYEE|E_LASTNAME]is defined),
  -> bindings:
    jpkg://localhost:5454/requestSocialAgentRawJS "Okay, I will
      call your contact person, ", $sem.[EMPLOYEE|E_FIRSTNAME],
      " ", $sem.[EMPLOYEE|E_LASTNAME], ". Please be patient.";
};

move Mediation_DontKnowContactPerson on variable Intention changed to
  selected(
  goal:(mediation = selected),
  path:($sem.[CUSTOMER|C_LASTNAME]is defined),
  path:($sem.[EMPLOYEE|E_LASTNAME]is undefined),
  path:($sem.[EMPLOYEE|E_KNOWN]is false),
  path:($sem.[MEETING|M_KNOWN]is undefined),
  -> bindings:
    jpkg://localhost:5454/requestSocialAgent
      FataJS:dontKnowContact;
};

move MediationMeeting on variable Intention changed to finalized (
  goal:(mediationMeeting = finalized),
  path:($sem.[CUSTOMER|C_LASTNAME]is defined),
  path:($sem.[MEETING|TIME]is defined),
  -> bindings:
    jpkg://localhost:5454/requestSocialAgent
      FataJS:mediationMeeting;
};

```

TABLE 7: Dialog Moves of Mediation

The definitions of the ontology, grammar, database, dialog goals, dialog moves and the interfaces to the Dialog-Employee-Component are realized in a similar way.



## 6.2.4 Telephony Application

The telephony application realizes the call to the employee and the speech recognition on the phone. It is written in C# and uses the SpeechStudio API.

If the application is started, the arguments telephone number and *welcome*-text need to be assigned. In the *Main*-method [FIGURE 28] these arguments are read. If the arguments are complete, then the grammar is assigned and the request getting a line is started.

The SpeechStudio Control provides a *TelephoneLine* event, which will arrive when we get or fail to get a line. If we got a line, the call to the employee is initiated on the telephony device. The *ConnectTimeout* timer is necessary to detect when the call never connects. The *TelephoneConnect* event will follow if someone answers. But if it gets disconnected the *TelephoneDisconnected* event will occur. These events are handled by the method *SpeechCtrl\_Telephone* [SS00].



FIGURE 28: UML Class Diagram Telephony

If the connection to the employee was established, we need to have an event handler, which receives the speech input. This is the *RecognizedHandler*. Thereupon the recognized phrases are forwarded to the dialog manager in the method *SendPhraseToDM*. The answer from the dialog manager is received in the method *OnClientConnect*. Here the result is spoken through the *SpeechPlayer* to the employee.

If the connection to employee couldn't be established or if the phone call was finished, the dialog manager is informed and the application shuts down with *Dispose*.



## 7 User Tests

For measuring the success of the Conversational, Mediating and Animated Receptionist, we take a look at three issues [SS00]: correct task completion, cost of completing the task and quality of the interaction between the system and the end user. We evaluated these aspects by giving the users different tasks with an additional questionnaire (q.v. appendix). Furthermore the questionnaire collects the ratings of ease of use, efficiency, naturalness of the interaction and the user's opinion about the functionality of the single components.

To gather first information about how the user interact and experience the system we made a prestudy. Furthermore this study should help to identify the weak points of COSIMAR. After we have made some modifications we performed the main user study.

In both tests, the tasks of the user have been divided into two parts. In the first part the users were asked to play three times the role of the visitor approaching the kiosk. The first of these three scenarios was a simple one, where the user just provided his name and the name of the meeting party. The second scenario was more complex, because the last name of the meeting party wasn't known and so they were asked to say an alternative like the meeting time or meeting name. Within the third scenario the user were given all information and he could choose how to perform the task.

The second user task was to take two times the role of the employee on the phone.

### 7.1 Differences between prestudy and main study

The main difference of the prestudy and the main study was, that within the prestudy the users have been told what they can expect from the system. Furthermore, they have been guided through the interaction by providing the user's answers and sentences.

For the prestudy we wanted the user to get used to the system, to learn the interaction and to get to know the functional range of COSIMAR, but also to keep him focusing not only to the dialog but also to the whole interaction. Therefore we predefined in the first two scenario steps *detailed scenario* and *more complex scenario* some sentences and phrases the user could say. In the last step: *personal interaction* the user only got the information about his name, the name of his meeting party and the meeting time and meeting name.

The main study was performed as real customer-receptionist-interaction, whereas the users haven't been introduced to and guided through the system. It



was just how we would expect the visitors to interact with COSIMAR at the reception.

## 7.2 Evaluation of the User Studies

The prestudy took place with 11 participants (2 female and 9 male). The age-group was 20-30 years and all of them had advanced computer knowledge, although mostly no experience with speech user interfaces or virtual characters.

The main study proceeded with 11 participants (3 female and 8 male). The age-group was wider: from 11-50. The computer knowledge of the participants was also varying, some users are very experienced whereas some are working at the computer just a few hours per week. The majority of the participants had no experience with speech user interfaces or virtual characters, but still some experienced the speech interface of the SAP telephone support and the virtual character of games and yellowstrom.

### 7.2.1 Evaluation of the Kiosk Interaction

#### Establishment of the Mediation

As you can see in FIGURE 29 the mediation could not always be established. In the prestudy in 9-27% of the cases the mediation failed. The failure was caused by the telephony component, which couldn't call out or just hang. However, when the phone call was established, also the mediation was established.

During the main study the same problems with the telephone component appeared. Additionally in the more complex scenario some users couldn't establish the contact, because they didn't know how to proceed when they don't know the last name of their contact person.

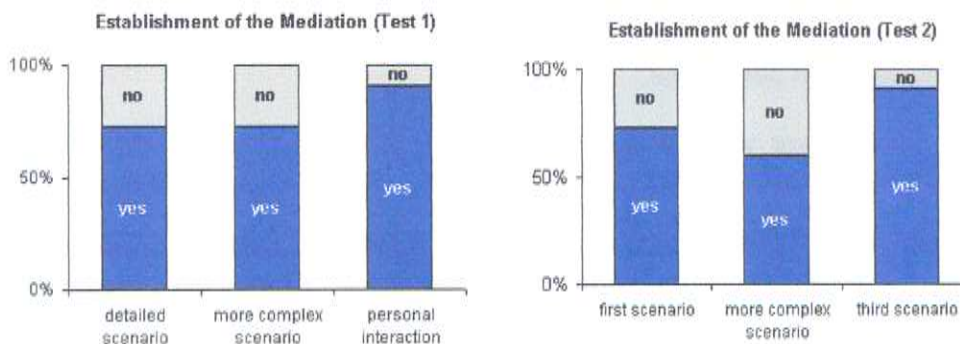


FIGURE 29: Establishment of the Mediation of the different scenarios

## Amount of time required to complete the tasks

To evaluate how satisfied the users are with the amount of time required to complete the tasks, we ask after every task completion for their opinion. FIGURE 30 depicts the user satisfaction about the task completion time for the different scenarios and the overall time for both tests.

In the prestudy, the users impressions while interacting with the system are, that *“after getting used to the system the amount of time required to complete a task speeded up”*, but they also felt that *“too many repetitions have been necessary”* until their utterance was recognized.

For the first scenario we noticed for both user tests (prestudy and main study) that users, in average, have almost the same evaluation for the time required to complete the task. Notable is also the fact, that the users of the main study, which didn't have the predefined sentences, even gave the system a better rating. This supports the preassigned requirement providing an intuitive user interface which can be used without technical training.

As mentioned before, in the more complex scenario of the main study, some users couldn't establish the contact to their meeting party, because COSIMAR didn't understand the users' intention. This fact reflects in the evaluation of the time required to complete the task. So the task was really complicated for some users, so that they rated with *“1 – more than expected”*. But on the other hand some users have been very satisfied and rated with *“4”* or *“5 – less than expected”*.

In the prestudy the users were asked to fulfill the third scenario in their own words. This was completely new for the users and they tried to discover the functional range of COSIMAR. This results in the less satisfied ratings compared to the previous ratings.

Whereas the users of the main study became experts in using the system and so the time required to use the system decreased. This results in the average value *“3.8”*.

We also asked the users to tell us their overall perspective of the time required to complete the three tasks. Here the results of both user studies are quite similar. In average, the users are satisfied with the time required to complete the task, but still some improvements are necessary.

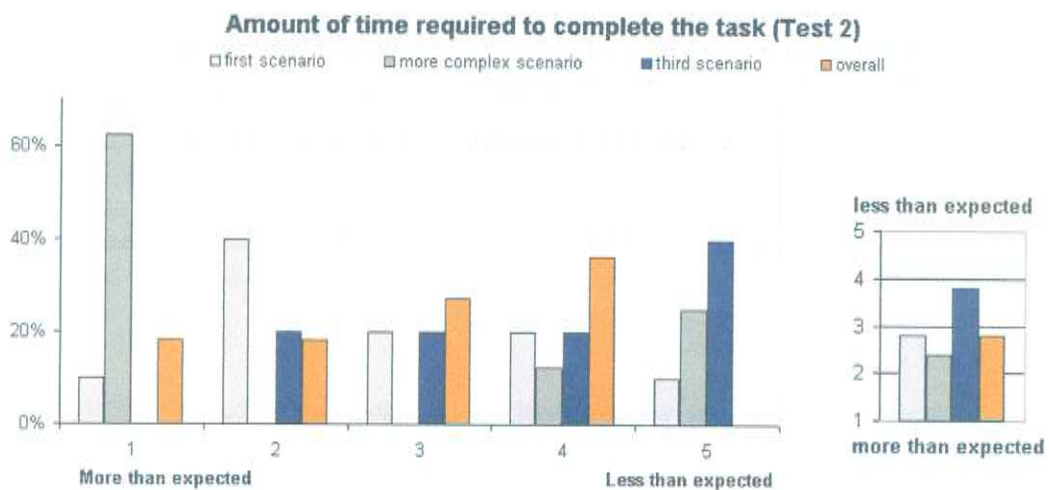
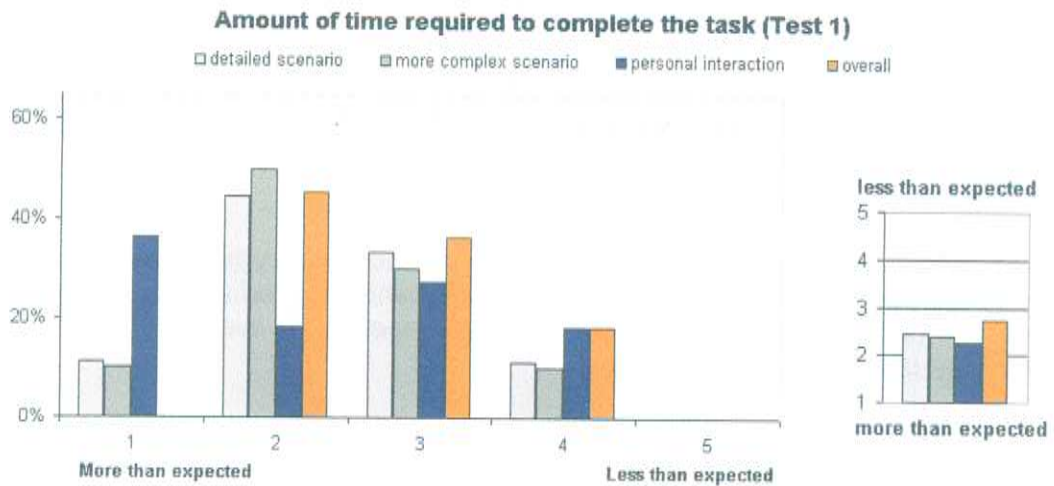


FIGURE 30: Amount of time required to complete the task (distribution and average)

Furthermore we asked the users questions about the whole application. We noticed if they haven't been that satisfied with the speech recognition, they generally answered all questions more negative.

### Look and Feel of COSIMAR

In the prestudy, overall the users liked the look and feel of COSIMAR as shown in FIGURE 31. But some users mentioned that the *“feedback for non-understood utterances”* should rather be audio feedback of the virtual character, like *“I’m sorry, I didn’t understand.”* Good was the natural kind of the interaction and the *“nice women”*.



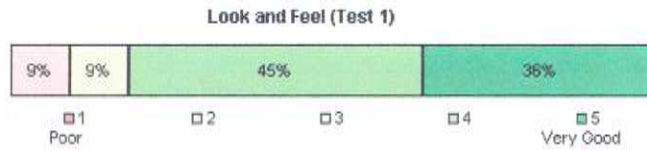


FIGURE 31: Look and Feel (Test 1) of COSIMAR

To gain more detailed results of the application in general, in the main study, we substituted the question to the look and feel for the two questions to the application in general [FIGURE 32]:

- a) very bad – very good
- b) hard to figure out – very intuitive.

27% of the users think that it's a bad application, whereas 45% think that it's at least a good application. 27% think that *"this is a very neat application, but may need a more fine tuning"*. 50% think that the application is intuitive. In contrast 20% find it somehow hard to figure out. One user thought: *"it was hard first; I didn't know I could speak naturally"*.

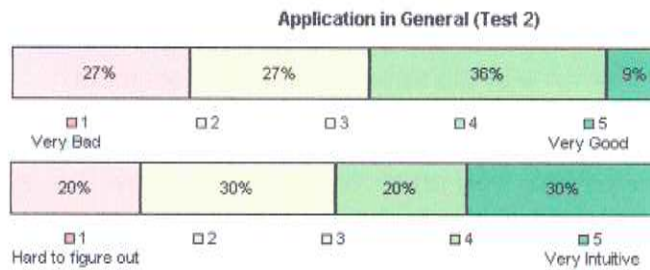


FIGURE 32: Application in General (Test 2)

## Speech Recognition

For both user test, the speech recognition was in average regarded as okay (average: 3). Even 9% in the first test thought, it's *"pretty good."* But, at least 36% thought that the speech recognition should be improved [FIGURE 33].

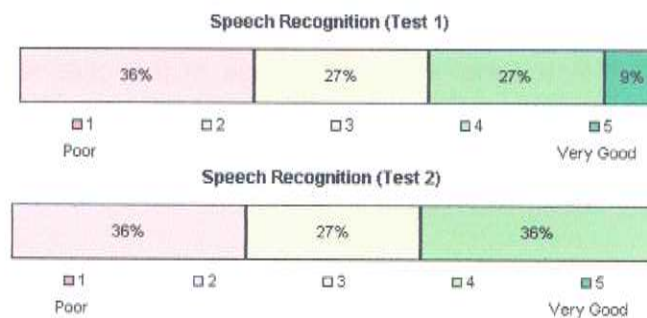


FIGURE 33: Speech Recognition of COSIMAR

## COSIMARs Amount of Understanding

In average for both user tests, COSIMARs amount of understanding was mainly considered as satisfying (average: 3). In 18% of the cases the users didn't felt understood. This is mainly either due to the insufficient speech recognition or the more complex scenario users had problems with.

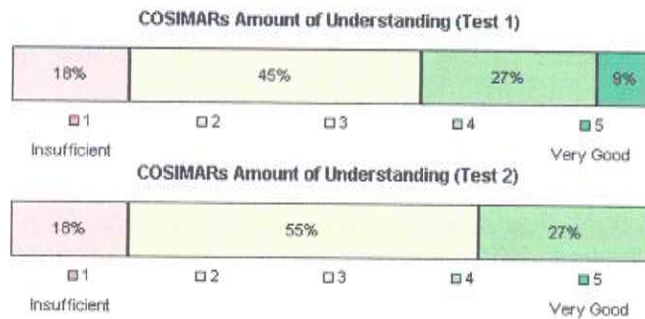


FIGURE 34: COSIMARs Amount of Understanding

## Objective Dialog Component Evaluation

To evaluate the dialog component decoupled from the speech recognition, we fed the dialog application with the transcripts of the user input.

The result of the prestudy was, that 295 of 341 utterances could be parsed (and 264 of them completely) and thus be understood by the system. The 46 utterances, which couldn't be parsed and understood, comprise sentences like: *"are you still with me?"*, *"oh I see"* or aborted sentences. We could also find statements like: *"I don't know"* and *"yes"*, which should be included in the future dialog.

In the main study 247 of 289 utterances could be parsed (and 183 of them completely). To the 42 phrases which couldn't be parsed count for examples: *"can I talk to you?"*, *"yes"*, *"no"* and names not stored in the data base.

These results show that the dialog component can handle spontaneous speech including sentence aborts and incorrect syntax not covered by the grammar. Furthermore it can also handle speech recognition errors, as proved in the user tests. For example was recognized: *"My name Miss Green."* instead of *"My name is Green."*

TABLE 8 and TABLE 9 show example dialogs of the user study. In the first example you can see, that on the one hand two phrases couldn't be recognized, but on the other hand, the spontaneous speech with its wrong syntax like *"ehm, I am like to see I'd like to see Charlie Brown."* could be understood although with an incorrect speech recognition result.

COSIMAR: Welcome to the SAP Research Center. I am the Virtual Receptionist Cosimar. If you want to talk to me, please use the headphone in front of you.

User: Hello Cosimar.

COSIMAR: Hello. Who are you here to see today?

User: I am Mary White <pause> and I would like to see Charlie Brown.

COSIMAR: Who are you here to see today?

User: I am here to see Charlie Brown.

COSIMAR:

User: ehm, I am like to see I'd like to see Charlie Brown.

COSIMAR: Okay, I will call your contact person, Charlie Brown. Please be patient.

To pleasing your waiting time, I will give you some information about the actual projects of Charlie Brown department: SAP Research. Please have a look at the SAP News Screen.

I am sorry to interrupt, but meanwhile I finished the phone call. Your meeting party will come in a few minutes He will meet you at the reception. It was a pleasure to help you.

User: Thanks. Bye bye.

COSIMAR: You are welcome.

TABLE 8: Example Dialog 1

The second example shows the big advantage of TAPAS. The user was able to establish the contact to its meeting party by just saying one utterance.

User: Hi Cosimar, my name is Ryan Arthur and I would like to speak to Grace Rainbow.

COSIMAR: Okay, I will call your contact person, Grace Rainbow. Please be patient.

To pleasing your waiting time, I will give you some information about the actual projects of Grace Rainbow department: SAP Research. Please have a look at the S A P News Screen.

I am sorry to interrupt. But meanwhile I finished the phone call. Your meeting party will come in soon. He will meet you at the reception. It was a pleasure to help you.

TABLE 9: Example Dialog 2



## COSIMARs Response Time

As you can see in FIGURE 35, COSIMARs response time to the user's utterance was considered as slow or okay (average: 2.45) in the prestudy. COSIMARs response time in the main user study was in average better (average: 3). But still, some more improvements should be done at this issue.

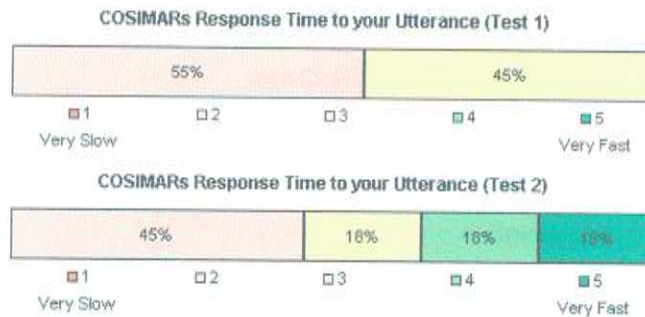


FIGURE 35: COSIMARs response time to your utterance

## COSIMARs Voice

In the prestudy COSIMARs voice was mainly considered as synthetic (average: 2.1), so it is not that pleasing to the user [FIGURE 36]. Just a few users have been okay with it, or found it pleasing.

In the main study, some users also thought, that it is an *“unnatural voice, which seems unfriendly”*. They *“didn't like the robotic voice”*. But some users found the voice pleasing or satisfying, so that the average here is better: 2.7.

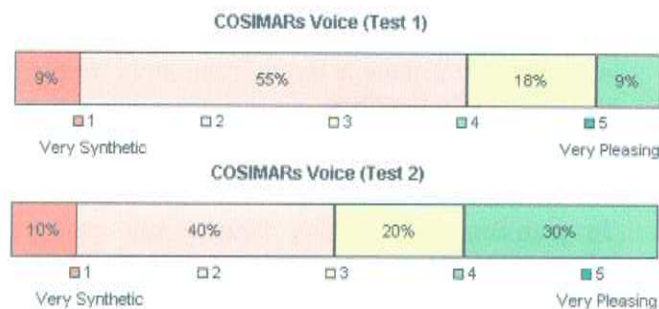


FIGURE 36: COSIMARs voice

## COSIMARs Facial Expressions

COSIMARs facial expressions [FIGURE 37] have been considered as normal to very pleasant. This was the feature the users have been liked a lot: *“the social agent seemed very professional”*.

To evaluate in the main study how the users found the facial expression, we divided the question into two parts:

- a) very unrealistic – very realistic
- b) very unpleasant – very pleasant

The users in the main study thought, that the facials expressions are not that realistic nevertheless they found them pleasing and liked the “facial expressions”. But also there are some users how found them unpleasant and too “comic-like”.

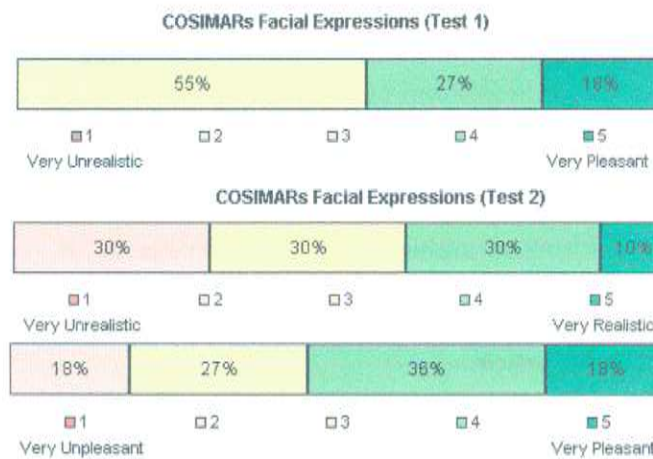


FIGURE 37: COSIMARs Facial Expressions

### Using the solution was a satisfying experience

Regarding the question, if using the solution was a satisfying experience [FIGURE 38], the user of the prestudy mainly have been satisfied or agreed (average: 3.1): “To interact with a virtual person is a satisfying experience” and “it was a novel experience. I liked the option for learning more about the division”. On the other hand some thought it is “too slow and too awkward to find out about the computer’s actual capabilities”.

In the main study the users haven’t been that satisfied with the usage experience (average: 2.9). They thought, that the “voice recognition does not work as expected”.



FIGURE 38: Using the solution was a satisfying experience

## Interaction with the system was natural

The users' opinions to the naturalness of the interaction [FIGURE 39] was in average 2.9 (prestudy) and 2.8 (main study). But some thought "as long as utterances have to be repeated several times it's not natural" and that "you will always realize the difference while talking with a virtual person".

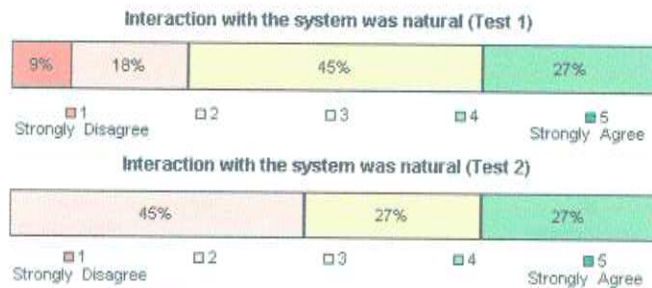


FIGURE 39: Interaction with the system was natural

## Ease to learn the interaction

The ease to learn the interaction with the system [FIGURE 40] was in the prestudy mainly considered as easy (3.7). In the main study the users thought it is mainly okay or easy (3.5), even if they haven't been guided through the interaction as the users from the prestudy.

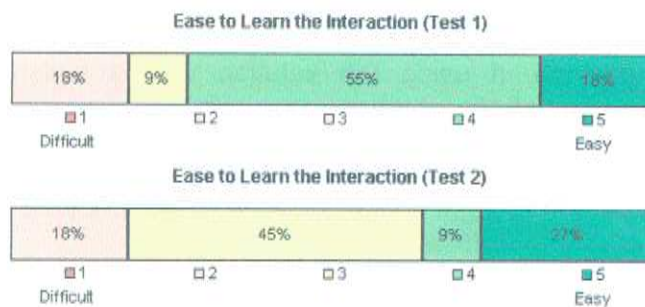


FIGURE 40: Ease to learn the interaction

## Usage of the Virtual Receptionist

The usage of COSIMAR was considered by the prestudy users as intuitive (average: 4). Even the users which didn't know how to use it initially found it intuitive (average:3.7) [FIGURE 41].



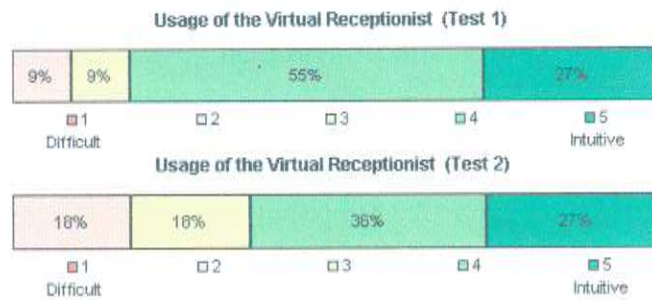


FIGURE 41: Usage of the Virtual Receptionist

### I think COSIMAR supports the mediation

Finally we asked, if the users think, that COSIMAR supports the mediation. There are two different positions. One user group thinks, that *“little improvements will make it very helpful”* and guarantee the *“availability”* at the reception. The other group is the opinion that visitors *“prefer a real person”* and *“don’t want to talk to a computer”*.

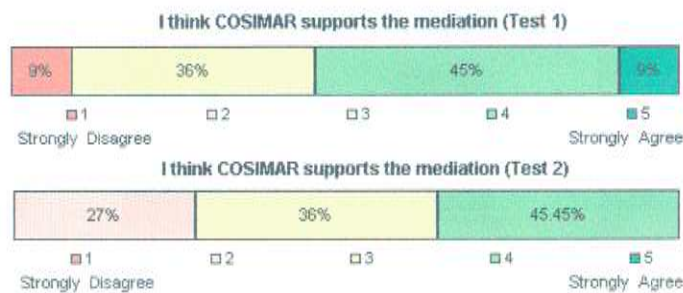


FIGURE 42: I think COSIMAR supports the mediation

## 7.2.2 Evaluation for Telephone Interaction

For the interaction on the phone the users in the prestudy were asked to take two times the role of the employee. In the first scenario they were given the phrases they should say:

- |                 |                  |                        |
|-----------------|------------------|------------------------|
| 1. Greet:       | Hello.           | → Cosimar's response.  |
| 2. Answer with: | In a few minutes | → Cosimar's response.  |
| 3. Answer with: | At the reception | → Cosimar says goodbye |

Within the second task they could decide how they would like to answer. Since the used SpeechStudio application only supports simple grammars, the user's utterances have been limited to words or simple phrases.

But even these restrictions haven't led to the desired results. As you can see in FIGURE 43 only in about 50% of the cases the mediation could be established. The problem was either caused by the speech recognition or the response time, so that the users after a while just hang up.

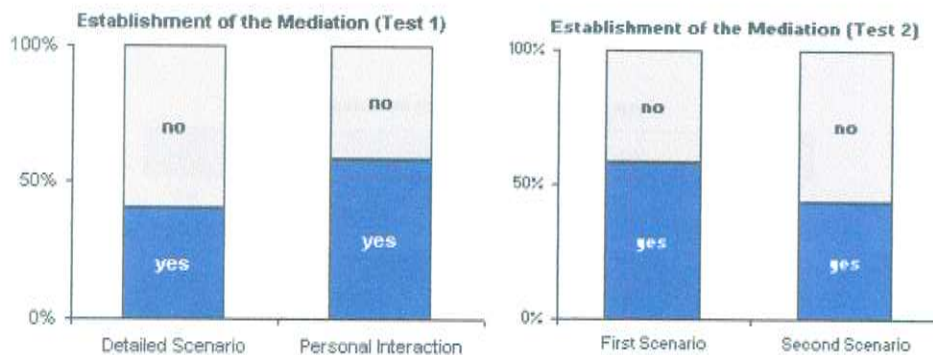


FIGURE 43: Establishment of the Mediation

Due to the unsatisfactory results of the prestudy, some changes in the telephony application for the main study have been made.

If the greeting of the user couldn't be recognized, COSIMAR started speaking after a certain time slice. For example:

```
This is Cosimar, the Virtual Receptionist. James Green is waiting for you in the lounge. When do you come to pick your meeting party up. You can say: immediately, in a few minutes, half an hour, at the negotiated time or I can not make it.
```

Because in the main study, the phrases, the employee could say on the phone, weren't disclosed to the users in the questionnaire, it was necessary to include the expected answers, like *immediately*, *in a few minutes*, in this utterance.

COSIMARs question regarding the meeting point was excluded in the main study, because this was the main obstacle to establish the contact in the prestudy. So the meeting point in this scenario has always been the reception.

Unfortunately, these changes didn't improve the user test results regarding the establishment, as you can see in FIGURE 43. The main study users have been more impatient and hang up, when there was a long pause – which could either be caused through the response time or the unrecognized phrase. If the user's utterance wasn't recognized he had to repeat it. But the SpeechStudio offered no possibility to give any unrecognized-response to the user; so the user had to repeat intuitively, what some didn't do and so they hang up.

If we take a look at the amount of time required to complete the task [FIGURE 44], we can see that this could indeed be improved, but it is still not desirable.

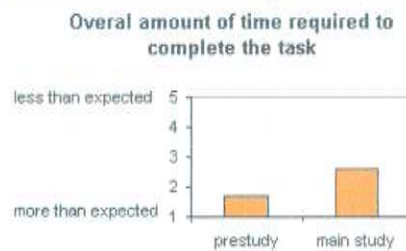


FIGURE 44: Amount of time required to complete the task

Also the user's impressions regarding the speech recognition could be improved. As outlined before, in the prestudy the speech recognition was really dissatisfying [FIGURE 45].

In the main study, only the phrases regarding the meeting time needed to be recognized. If they have been recognized, the users were ok with it.

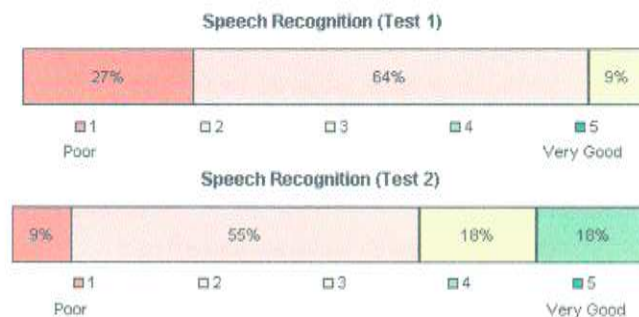


FIGURE 45: Speech Recognition

The response time of COSIMAR to the user's utterance is no factor which can be improved without changing the SpeechStudio API, which we haven't done. Therefore the results are equal [FIGURE 46].



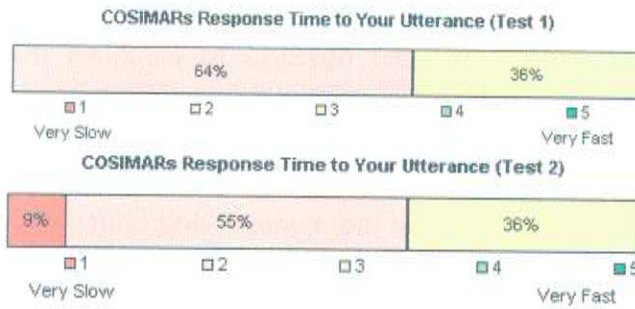


FIGURE 46: COSIMARs Response Time to Your Utterance

Surprisingly the users of the main study liked COSIMARs voice more than the users of the prestudy [FIGURE 47], whereas no changes has been done.

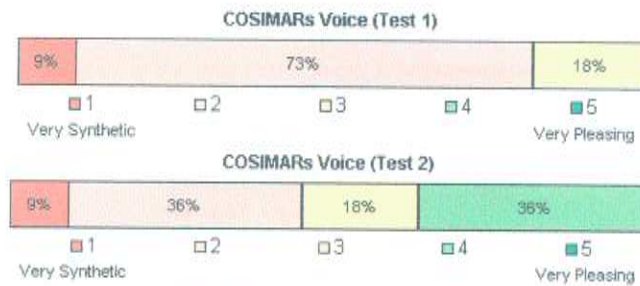


FIGURE 47: COSIMARs Voice

First we tried to keep the interaction as natural as possible, but the results of the prestudy prompted changes, which restricted the naturalness of the interaction [FIGURE 48]. Furthermore the users commented that *"COSIMARs voice is sometimes hard to understand over the phone"* and that *"the long pause before COSIMAR started talking made the experience more confusing."*

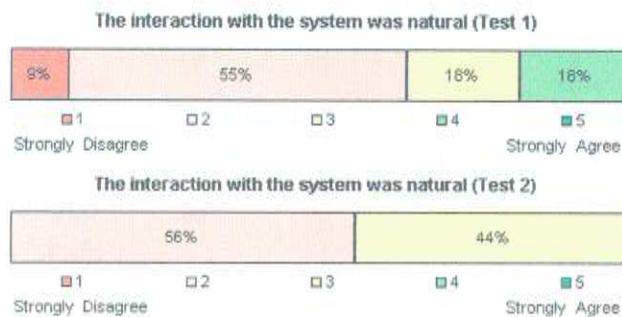


FIGURE 48: Interaction with the system

FIGURE 49 shows the user evaluation regarding mediation functionality. The users are the opinion, that – in this state – the system isn't applicable for the mediation task. But improvements will make it ready for use. Some even *"like the idea of a computer calling me"*.

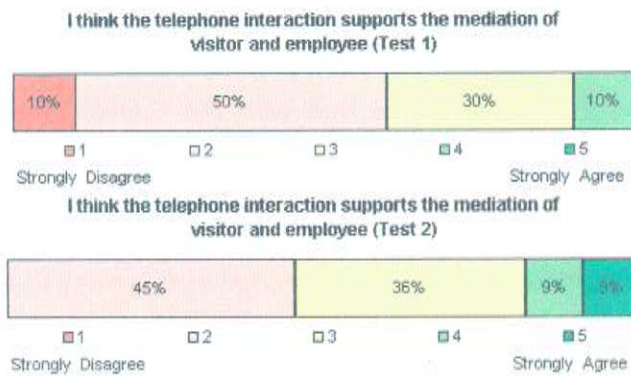


FIGURE 49: I think the telephone interaction supports the mediation of visitor and employee

### 7.3 Conclusions drawn from the User Test Evaluation

On the one hand the results show that with the Conversational, Mediating and Animated Receptionist, we succeeded to provide a user friendly interface. On the other hand, the user tests emphasize that some more improvements need to be done until the system is serviceable for the reception use case.

Both user groups agreed that the Virtual Receptionist is easy to use, although the second group has had no instructions how to proceed. Thus, with this system we meet the requirement to reduce technical training and to provide an interface which is easy to use.

Regarding the design of the GUI, we can draw the conclusion from the user tests, that the GUI was transformed to a social and enjoyable experience. The main feedback thereto was namely, that the users liked the social agent with its facial expressions and the video component with its entertaining functionality. Improvements necessary for the social agent component are the performance and the voice.

Although the speech recognition results have been quite unsatisfying, we can say that the speech recognition component proved to be suitable for this scenario, because the users gave the whole system a quite good note and emphasized its intuitiveness. But in any case the application needs a better speech recognizer, and needs to give better feedback if something couldn't be recognized.

Through allowing a natural interaction we supported users according to their needs and expertise. The more expertise the users had using the application, the less guidance and thus less time did they need to reach their goals. Therefore, the dialog manager TAPAS turned out to be very suitable for the virtual receptionist scenario. Especially should be emphasized that TAPAS compensated the incorrect speech recognition by allowing incorrect input and thus increased the usability of the whole system.

Summarizing the interaction on the telephone, we can say, that major improvements needs to be done regarding the naturalness of the interaction and the speech recognition. So we can say, that the SpeechStudio is not suitable at all and that in future versions the use of another system is advisable.



## 8 Conclusion and Outlook

### 8.1 Conclusion

The result of this work is a prototype version of a Conversational, Mediating and Animated Receptionist, which combines different modalities to enable people using an application with natural language. The major characteristics of the virtual receptionist are several capabilities which include the ability to carry out a dialog with the visitor using speech input and output, virtual face presentation with facial expressions, and moreover to embed another dialog with a third party via telephone to establish a contact between both parties.

This work presented the development and realization of such a system. As a first step, we collected the requirements for such a system by observing the reception environment and conducting interviews with receptionists and their clients. The results became system use cases for optimizing the development by taking actual needs and requirements into account, and thus supporting the reception business use case.

After that, we took a look at the actual research projects in the field of conversational and embodied systems. In this area a lot of important and fundamental work is going on, but none of this research projects considered the integration of two different dialogs within one task and the mediation of visitors by using a call system.

Motivated by the interesting ongoing research work, we performed the next step: identifying the third party vendor products relevant for the virtual receptionist. Restricted through economic purposes, we decided us for the speech recognition engine of Microsoft accessible through the Chant SpeechKit; the SpeechStudio realizing the call functionality and speech recognition on the phone; the natural language understanding and dialog management system TAPAS; and the FATA system of SAP displaying the virtual character and generating the speech output.

After defining the software systems, which should be used, we concerned with the question how to realize the interaction of these systems to meet the system's requirements. The result was the architecture of COSIMAR, which has been thereupon designed and implemented.

The last action, which was taken, was a comprehensive user study to evaluate COSIMARs functionality and the user needs, interests and opinions regarding the whole system.

## 8.2 Results

The results drawn from the user evaluation are, that we succeeded to provide a user friendly interface, which offers mediation functionality and can be easily used without giving the visitor any instructions, because it is equal to the way people communicate with each other. These advantages were mainly achieved by the strength of the whole architecture and the third party systems FATA and TAPAS, which turned out to be very suitable for this system. Nevertheless some more improvements are necessary until COSIMAR is serviceable to support the reception business use case. Especially the speech recognition and the call system need major improvements.

Furthermore, we noticed that the users personalized the virtual character and perceived COSIMAR as social actor. This conclusion has also been drawn in [NI00]. In addition, they discovered, that the appearance, social rules and expectations to embodied conversational agents are a very important of how people access these characters. If the virtual character shares the ethnicity of the user, they have been perceived as more competent. And if they have a consistent personality, they have been perceived as more useful. Thus we can reason, that a beauty appearance, a nice voice and a consistent personality are the key for designing a successful conversational embodied character, which plays an important role supporting the communication between user and system.



## **8.3 Future Work**

COSIMAR lays the foundation for a conversational, embodied and mediating agent, which is suitable for different scenarios. Nevertheless there is a lot of space open for further research work.

### **8.3.1 Scenarios**

COSIMAR is applicable in areas, where people wants get information and orientations, or wants to be mediated to a specific person. Below some example of such business cases are given.

#### **Shopping mall**

Everybody knows the feeling of being lost in a shopping mall, or the long time it takes searching for an expert to get advice on a question. On the one hand COSIMAR could support these customers by giving orientations and providing information; and on the other hand COSIMAR could call the appropriate expert.

#### **City Guide**

COSIMAR could also be used as a tourist information system that provides information and orientations about a city. LingWear [FW01], a project of ISL, already offers such a mobile information system, but without integrating a calling system and the display of a virtual character. To offer the user a social and enjoyable experience, COSIMAR could be integrated into LingWear. In this way, also the possibilities to call a taxi, reserve a table in a restaurant and book a hotel could be provided.

#### **Hotel**

In a hotel, we also have receptionists, which cannot guarantee a 24-7-availability. To provide the visitors with information about the hotel and its rates, COSIMAR could be used, and if the visitor decides to stay, COSIMAR could call a person to come over.

#### **Expert Finder**

Within a huge company, employees often have difficulties to find experts in specific areas. To keep the privacy of the experts, COSIMAR could be utilized as a mediator.

### **8.3.2 Research**

Developing and designing conversational, mediating and animated agents is a wide open and multidisciplinary research topic. The next paragraphs list some research topics which would benefit, improve and enhance COSIMAR.



### **Sensing Communication Partner**

In the current version of COSIMAR, the sensing of communication partners isn't included and needs to be completed. To not only distinguish, if a communication partner is at the kiosk or not, research in identifying interested users could be a continuative topic, as currently done in [Sch05]. Furthermore face recognition, on the one hand, would allow conducting a dialog specific to the users needs, and on the other hand, it would support security.

### **Speech Recognition**

Besides improving speech recognition regarding context free grammars, for COSIMAR it would also be beneficial, if names not stored in the database (that means out of vocabulary words) could be recognized as well. Other aspects performing research in the area of speech recognition to improve COSIMARs capabilities are spontaneous speech and dialects.

### **Dialog Management**

"Bodily behaviors are the most important means through which human participants in a face-to-face conversation accomplish turn-taking behaviors and assess each other's continued attention and understanding" [MC00]. To enhance the dialog functionality and the social component it would be desirable to integrate gesture and emotion recognition, and correspondingly the response with gestures and emotions. Furthermore, research in the area of conversational functions, like the behavior for initiation and termination, turn taking and backchannel feedback on both sides: user and system, could also benefit the interaction between user and COSIMAR. To support the user according to the user needs, a user model and a dialog history could also be integrated.

### **Social Agent**

For a more natural interaction between user and system, more research in the area of text-to-speech and the generation of facial expressions is needed. Also the performance of the Social Agents needs to be improved.

### **Call System**

With respect to the call system, research in supporting speech recognition on the phone is needed.

### **Entire interaction**

Consideration also is needed of the entire interaction between the user and system, to evaluate how the user is influenced by the animated character and if it differs from human-human interaction. This research would benefit the entire system's design.

## 9 Literature and Abbreviation

### 9.1 Literature

- [AG01] Allen, J., F. George, A. Stent: *An architecture for more realistic conversational systems*. In: *International Conference on Intelligent User Interfaces. Proceedings of the 6th international conference on Intelligent user interfaces table of contents*, Santa Fe, New Mexico, pp. 1-8, 2001.
- [Am05a] Amazinggrace: *Reyna's Smile*, at DevaintArt. 2005, January 17 [online] – URL: <http://www.deviantart.com/deviation/14286786/>.
- [Am05b] Amazinggrace: *Amused*, at DevaintART. 2005, January 17 [online] – URL: <http://www.deviantart.com/deviation/13763053/>.
- [Br02] Just van den Broecke: *Pushlets – Whitepaper*, 2002. URL: <http://www.pushlets.com/doc/whitepaper.html> (May 20, 2005)
- [Ca00] Justine Cassell: *More than Just Another Pretty Face: Embodied Conversational Interface Agents*. In: *Communications of the ACM* 43(4), pp. 70-78, 2000.
- [Ca99] Justine Cassell: *Embodied Conversation: Integrating Face and Gesture into Automatic Spoken Dialogue Systems*. In: Luperfoy (ed.), *Spoken Dialogue Systems*. Cambridge, MA: MIT Press, 1999.
- [CB01] Cassell, Bickmore, Vilhjalmsson, Yan: *More Than Just a Pretty Face: Conversational Protocols and the Affordances of Embodiment*. In: *Knowledge-Based Systems* 14, pp. 55-64, 2001.
- [CC02] De Carolis, B., V. Carofiglio, F. de Rosis, C. Pelachaud, I. Poggi: *Multimodal Dialogs With a Conversational Agent*. In: *AI\*IA, 8th Conference of the Italian Association for Artificial Intelligence. Atti del Workshop su "NLP e Web: la sfida della multimodalità tra approcci simbolici e approcci statistici"*, 10-13 September, 2002.
- [Ch96] Chant SpeechKit Developer Guide. Copyright: 1996-2004.
- [CN01] Cassell, J., Nakano, Y., Bickmore, T., Sidner, C., Rich, C.: *Non-Verbal Cues for Discourse Structure*. In: *Proceedings of the 41st Annual Meeting of the Association of Computational Linguistics*, pp. 106-115, July 17-19, Toulouse, France, 2001.



- [CR01] De Carolis, B., F. de Rosis, V. Carofiglio and C. Pelachaud: *Interactive Information Presentation by an Embodied Animated Agent*. In: *International Workshop on Information Presentation and Natural Multimodal Dialogue*, Verona, Italy, December 14-15, 2001.
- [CS02] Cassell, Stocky, Bickmore, Gao, Nakano, Ryokai, Tversky, Vaucelle, Vilhjálmsón: *MACK: Media lab Autonomous Conversational Kiosk*. In: *Proceedings of Imagina02*, February pp. 12-15, Monte Carlo, 2002.
- [CV99] Cassell, Vilhjálmsón, Chang, Bickmore, Campbell, Yan: *Requirements for an Architecture for Embodied Conversational Characters*. In: *Computer Animation and Simulation '99 (Eurographics Series)*, pp. 109-120, 1999.
- [Da05] Michael C. Daconta: *When Runtime.exec() won't. Navigate yourself around pitfalls related to the Runtime.exec() method*, 2005, March 10 [online] – URL: <http://www.javaworld.com/javaworld/jw-12-2000/jw-1229-traps.html>
- [De02] Matthias Denecke: *Generische Interaktionsmuster für aufgabenorientierte Dialogsysteme*. Doktorarbeit, Universität Karlsruhe, 2002.
- [DO05a] Data & Object Factory: *Mediator*. 2005, March 10 [online] – URL: <http://www.dofactory.com/Patterns/PatternMediator.aspx>
- [DO05b] Data & Object Factory: *Singleton*. 2005, March 10 [online] – URL: <http://www.dofactory.com/Patterns/PatternSingleton.aspx>
- [DW97] Matthias Denecke and Alex Waibel: *Dialogue Strategies Guiding Users To Their Communicative Goals*. In: *Eurospeech 1997*, Rhodes, Greece, 1997.
- [DY03] Townsend Duong and Andrew Yinger: *FATA Engine Architectural Description*, Revision 2003 (internal).
- [Ja03] Java™ 2 Platform, Standard Edition, v 1.4.2 API Specification. URL: <http://java.sun.com/j2se/1.4.2/docs/api/index.html>
- [FG02] Fügen, C., P. Gieselmann, H. Holzapfel, T. Schaaf, A. Waibel: *Multimodaler Mensch-Maschine-Dialog*. In: *Human Centered Robotic Systems (HCRS)*, Karlsruhe, Germany, December 2002.
- [FH04] Fügen, C., H. Holzapfel, A. Waibel: *Tight coupling of speech recognition and dialog management - dialog-context dependent grammar weighting for speech recognition*. In: *INTERSPEECH-2004*, 169-172, 2004.



- [FW01] Christian Fügen, Martin Westphal, Mike Schneider, Tanja Schultz, and Alex Waibel: *LingWear: A Mobil Tourist Information System*, HLT 2001, San Diego.
- [Ga04] Gandalf: The Communicative Humanoid. 2004, December 22 [online] – URL: <http://xenia.media.mit.edu/%7Ekris/gandalf.html>
- [GF03] Gieselmann, P., C. Fügen, H. Holzapfel, T. Schaaf, and A. Waibel: *Towards Multimodal Communication with a Household Robot*. In: *Proceedings of the Humanoids*, Karlsruhe/Munich, 2003.
- [HA95] Huang, X., A. Acero, F. Alleva, D. Beeferman, M. Hwang, and M. Mahajan. *From CMU Sphinx-II to Microsoft Whisper: Making Speech Recognition Usable*. In: *Advanced Topics in Speech Recognition*, Kluwer Academic Publishers, 1995.
- [HD04] Hartwig Holzapfel and Matthias Denecke: *Tapas Tutorial Step 1 – Step 3*, Version 0.8.3, August 2004.
- [HG04] Hartwig Holzapfel and Petra Gieselmann: *A Way Out of Dead End Situations in Dialogue Systems for Human-Robot Interaction*. In: *Humanoids 2004*, Los Angeles, 2004.
- [HG04b] Hartwig Holzapfel and Petra Gieselmann: *Dynamic Database Information for Grammar Generation and Environment Modeling in Dialogue Management*. 2004 (internal).
- [HN04] Hartwig Holzapfel, Kai Nickel and Rainer Stiefelhagen: *Implementation and Evaluation of a Constraint-Based Multimodal Fusion System for Speech and 3D Pointing Gestures*. In: *Proceedings of the International Conference on Multimodal Interfaces, (ICMI)*, State College, 2004.
- [Ho03] Hartwig Holzapfel: *Emotionen als Parameter der Dialogverarbeitung*. Diploma Thesis, University of Karlsruhe, 2003.
- [Ho05] Hartwig Holzapfel: *Towards Development of Multilingual Spoken Dialogue Systems*. In: *Proceedings of the 2nd Language and Technology Conference (L&T'05)*, Poznan, 2005.
- [KD05] Matthias Kaiser and Townsend Duong: *Improving Accessibility of Complex Applications with Knowledge-Based Proactive User Interfaces*. (internal)
- [KW01] Kille, M. and K. Warwick: *Queue-Based Agent Architecture for Multimodal Interfaces*. In: *Proceedings of IEEE International Conference on Multimedia & Expo 2001*, August 22-25, Tokyo, Japan, 2001.

- [MC00] Dominic W. Massaro, Michael M. Cohen, Jonas Beskow, and Ronald A. Cole: *Developing and Evaluating Conversational Agents*. In: *Embodied Conversational Agents*, MIT Press, 2000.
- [Mc02] Michael F. McTear: *Spoken Dialogue Technology: Enabling the Conversational Interface*. In: *ACM Computing Surveys*, 34(1):90 -- 169, Mar 2002.
- [MS05] Microsoft Speech SDK Documentation. 2005, January 31 [online] – URL: <http://download.microsoft.com/download/speechSDK/SDK/5.1/WXP/EN-US/sapi.chm>
- [MS99] Microsoft Windows 2000 Server: *IP Telephony with TAPI3.0. White Paper*. Copyright 1999.
- [NI00] Clifford Nass, Katherine Isbister and Eun-Ju Lee: *Truth is Beauty: Researching Embodied Conversational Agents*. In: *Embodied Conversational Agents*, MIT Press, 2000.
- [NK00] Patrick Niemeyer and Jonathan Knudsen: *Learning Java*, O'Reilly, 2000.
- [Th95] Kristinn Thórisson: *Computational Characteristics of Multimodal Dialogue*. In: *AAAI Fall Symposium on Embodied Language and Action*, Massachusetts Institute of Technology, Cambridge, Massachusetts, November 10-12, pp. 102-108, 1995.
- [Th97] Kristinn Thórisson: *Layered, Modular Action Control for Communicative Humanoids*. In: *Computer Animation*, pp.134-143, 1997.
- [Th98] Kristinn Thórisson: *Real-Time Decision Making in Multimodal Face to Face Communication*. In: *Second ACM International Conference on Autonomous Agents*, Minneapolis, Minnesota, May 11-13, pp. 16-23, 1998.
- [Th99] Kristinn Thórisson: *A Mind Model for Multimodal Communicative Creatures & Humanoids*. In: *International Journal of Applied Artificial Intelligence*, pp. 449-486, 1999.
- [TP04] Thórisson, Pennock, List, DiPirro: *Artificial Intelligence in Computer Graphics: A Constructionist Approach*. In: *Computer Graphics Quarterly*, February, pp. 26-30, 2004.
- [Oe01] Bernd Oestereich: *Objektorientierte Softwareentwicklung: Analyse und Design mit der UML 2.0*, Oldenbourg Wissenschaftsverlag, 2001.

- [Oe03] Bernd Oestereich et al.: *Objektorientierte Geschäftsprozessmodellierung mit der UML*, dpunkt.verlag, 2003.
- [PW05] Pulse Website. 2005, May 22 [online] – URL: <http://www.pulse3d.com>.
- [Sch05] Christoph Schaa: *Automatisches Erkennen und Interessenklasifikation von Benutzern für humanoide Roboter*. [not published yet]
- [SD05] Sun Developer Network. 2005, May 22 [online] – URL: <http://developers.sun.com/>.
- [SS00] Gregory A. Sanders and Jean Scholtz: *Measurement and Evaluation of Embodied Conversational Agents*. In: *Embodied Conversational Agents*, MIT Press, 2000.
- [SS03] SpeechStudio Suite VB.NET and C# Users. Documentation and Tutorials. Copyright: 2000-2003.
- [UI03] Christian Ullenboom: *Java ist auch eine Insel. Programmieren für die Java 2-Plattform in der Version 1.4*, Galileo Computing, 2003. 2005 May 30 – URL: <http://www.galileocomputing.de/openbook/javainsel3/>



## 9.2 Abbreviations und Definitions

ActiveX	A set of technologies that enables software components to interact with one another in a networked environment, regardless of the language in which the components were created.
APML	Affective Presentation Markup Language: a Markup Language for Believable Behavior Generation.
BEAT	Behavior Expression Animation Toolkit
Collagen	Collaborative interface Agents
COSIMAR	<b>C</b> onversational, <b>M</b> ediating and <b>A</b> nimated <b>R</b> eceptionist
COTS	Commercial-Off-the-Shelf
DPML	Markup language that enables tagging the discourse plans so that the main items are identified (goals, rhetorical relations, focus).
DTMF	Dual Tone Multi-Frequency
FATA	Feed Automatically Talking and Animate
HTML	Hyper Text Markup Language: a coding language used to create Hypertext documents for use on the World Wide Web.
InputStream	JAVA: This abstract class is the superclass of all classes representing an input stream of bytes.
JavaBean	A reusable component that can be used in any Java application development environment.
JSP	Java Server Pages: a scripting language based on Java for developing dynamic Web pages and sites. [Ja03]
KQML	Knowledge Query and Manipulation Language that provides a syntax and high-level semantics for messages exchanged between agents.
OutputStream	JAVA: This abstract class is the superclass of all classes representing an output stream of bytes. [Ja03]

Payload	In a set of data, such as a data field, block, or stream, being processed or transported, the part that represents user information and user overhead information, and may include user-requested additional information
Pulse	Software company, which provides technology and consulting services that enable its customers to create and deploy virtual characters for Web and wireless applications. [PW05]
Samba	SAMBA is a collection of free software developed to provide Microsoft file system services from UNIX file servers.
SOAP	Simple Object Access Protocol XML-based protocol for exchange of information in a decentralized, distributed environment.
Speech Recognition	Speech recognition is the process of converting an acoustic signal (i.e. audio data), captured by a microphone or a telephone, to a set of words.
Speech Synthesis	Speech synthesis is the process of converting words to phonetic and prosodic symbols and generating synthetic speech audio data.
tag	A piece of text that describes the semantics or structure of a unit of data in a markup language. Tags are surrounded by angle brackets (< and >) to distinguish them from text.
tag library	In JSP technology tag libraries define declarative, modular functionality that can be reused by any JSP page. Tag libraries reduce the necessity to embed large amounts of Java code in JSP pages by moving the functionality of the tags into tag implementation classes.
TAPI	Telephony Application Programming Interface
Thread	A <i>thread</i> is a thread of execution in a program. [Ja03]
Tomcat	Java based Web Application container that was created to run servlets and JSPs.
TRINIDI	Task Oriented Instructional Dialogue
Web Service	A Web service is an application that makes itself available over the Internet and uses a standardized XML messaging system.
WHISPER	Windows Highly Intelligent Speech Recognizer





# 10 Appendix

## 10.1 Questionnaire User Test I

### A) Background Questionnaire

Please answer the following questions below in order to help us understand your background and experience.

Gender:  Male  Female

Age: \_\_\_\_

#### RECEPTION EXPERIENCE

1. How often do you ask the reception of a company to help or mediate you?
    - never
    - once a year
    - once a month
    - once a week
    - every day
  2. What do you generally ask at the reception?
  3. What are you usually asked by a receptionist?
  4. From your experience: How are you helped by them?
  5. What do you do while waiting?
- 

#### COMPUTER EXPERIENCE

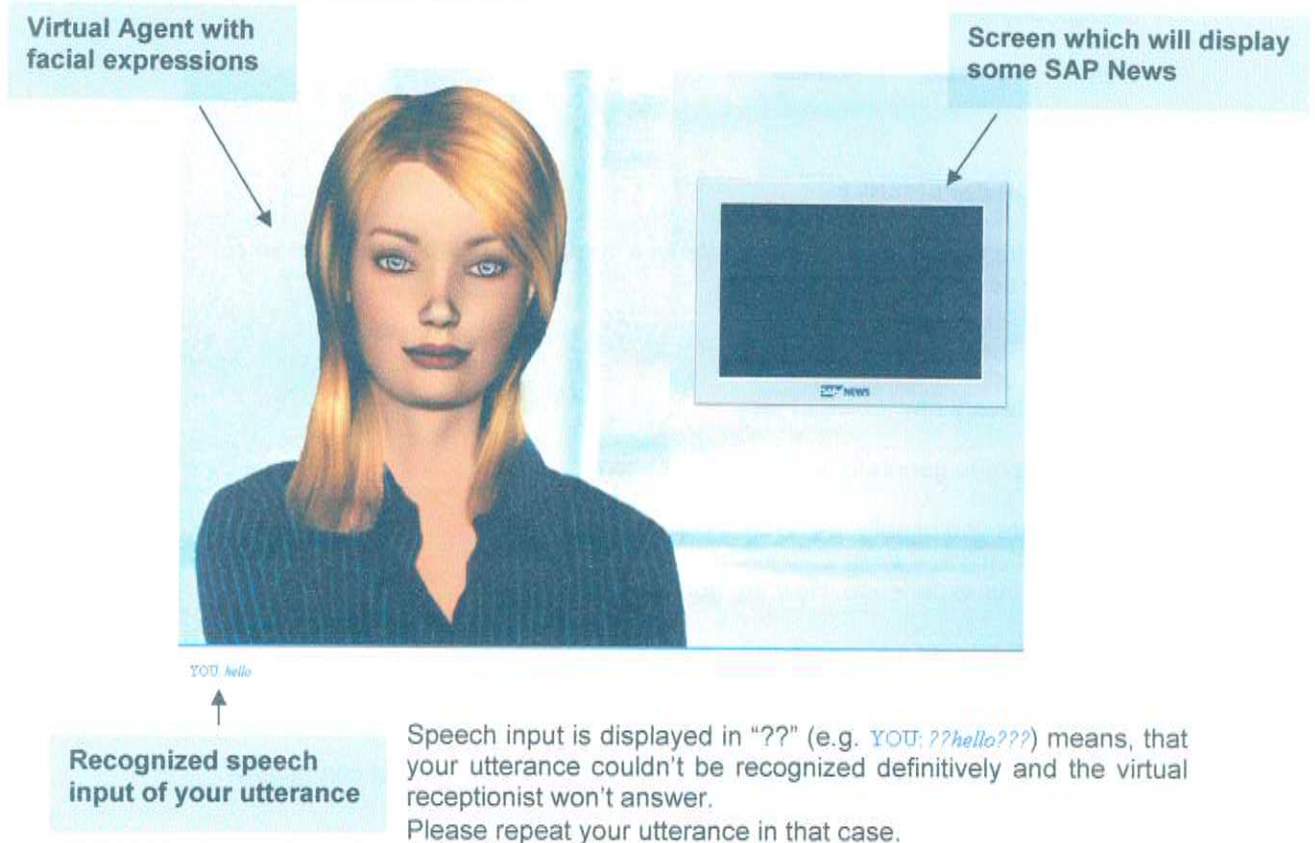
1. What is the total length of time you have been using personal computers? \_\_\_\_
  2. How much hours do you work at the computer per week? \_\_\_\_
  3. How much experience do you have with speech user interfaces?
    - none
    - some
    - plentyIf you have at least some experience, where do you use a speech user interfaces?
  4. How much experience do you have with virtual characters?
    - none
    - some
    - plentyIf you have at least some experience, where have you used virtual characters?
-

## B) Scenario Interaction with the Virtual Receptionist at the Kiosk

This prototype version of the Virtual Receptionist is designed to establish the contact between a visitor, approaching the reception of a company, and his meeting party.

The *Virtual Receptionist* combines different modalities to enable people using an application with natural language. The major characteristics of the *Virtual Receptionist* are several capabilities which include the ability to carry out a dialog with the visitor using speech input and output, virtual face presentation with facial expressions, and moreover to embed another dialog with a third party via telephone.

### What you will see on the screen:



### Some hints

- If the speech recognition recognizes your utterance, it will take some seconds until the virtual agent responds. Please be patient.
- If she recognizes something wrong, please repeat your utterance clearly.
- COSIMAR only knows the names and meetings stored in the database. So, please only use the suggested names.
- The functionality of COSIMAR is restricted to the mediation functionality. She neither does small talk nor gives general information.
- The scenario might not take place step by step as described below scenarios, in that case just answer COSIMAR's questions.

### Detailed scenario steps

Imagine you are Mary White and you approach at the kiosk of the virtual receptionist. You are here to see Charlie Brown and you ask the virtual receptionist, to establish the contact between you and Charlie Brown. To do so, please fulfill the following steps.

1. Greet the Virtual Receptionist: `Good morning.`  
And listen to COSIMAR's question.
2. Answer her question: `I want to talk to Charlie Brown.`  
And listen to Cosmiar's response.
3. Answer her question: `My name is Mary White.`
4. While you are waiting she will show you a video on the SAP News screen.
5. Wait until she finishes the phone call and listen to her result.
6. Was the contact established?  yes  no
8. Amount of time required to complete all tasks was:  

<i>More than expected</i>					<i>Less than expected</i>
1	2	3	4	5	

### More complex scenario steps

Now you take the role of James Green who wants to join the Research Forum at 10 am. He was invited by the employee Karin. Unfortunately Mr. Green only knows the first name of his contact. But to allow Mr. Green to attend the forum, the virtual receptionist needs to know at least the meeting information. To successfully attend the Research Forum, please fulfill the following steps:

1. Say that you are James Green. Choose an expression which you are comfortable with. For example:  
`Hi, I am James Green`  
`Mr. Green`  
`My name is James Green`  
And listen to Cosmiar's question.
2. Answer her question that you want to talk to Karin. For example:  
`I am here to see Karin.`  
`I would like to meet Karin.`  
`I want to talk to Karin.`  
And wait for COSIMAR's response.
3. Say that you don't know the name of your contact. For example:  
`I forgot the name`  
`I don't know the name`  
`I am sorry; I don't know the name of my contact person.`  
And wait for COSIMAR's response.



4. Answer that you know the meeting time or meeting name. For example:  
 Yes, I want to join the Research Forum  
 I am invited to the Research Forum  
 Yes, 10 am.
5. While you are waiting she will show you a video on the SAP News screen.
6. Wait until she finishes the phone call and listen to her result.
7. Was the contact to the responsible person established?  yes  no
9. Amount of time required to complete all tasks was:  

<i>More than expected</i>					<i>Less than expected</i>
1	2	3	4	5	

### Personal Interaction

Now you can decide on your own, how to proceed. Here is the following information you might need:

Your Name	Ryan Arthur
Your Meeting Party	Grace Rainbow
Meeting Time	2 pm
Meeting Name	Status Meeting

1. Feel free to complete the task as you would do with a real receptionist.
2. Was the contact to the responsible person established?  yes  no
3. Amount of time required to complete all tasks was:  

<i>More than expected</i>					<i>Less than expected</i>
1	2	3	4	5	

### C) Evaluation Questionnaire for Kiosk Interaction

To help evaluating the Virtual Receptionist, please answer the following questions:

1. Overall, I am confident COSIMAR supports the mediation of visitor and third party (Cycle one)

*Strongly Disagree* 1 2 3 4 *Strongly Agree* 5

Comments:

2. Amount of time required to complete all tasks was:

*Too slow* 1 2 3 4 *really fast* 5

Comments:

3. Using the solution was a satisfying experience.

*Strongly Disagree* 1 2 3 4 *Strongly Agree* 5

Comments:

4. The interaction with the system was natural:

*Strongly Disagree* 1 2 3 4 *Strongly Agree* 5

Comments:

5. General look and feel is:

*Poor* 1 2 3 4 *Very Good* 5

Comments:

6. COSIMARs Facial Expressions are:

*Very Unrealistic* 1 2 3 4 *Very Pleasant* 5

7. COSIMARs amount of understanding was:

*Insufficient* 1 2 3 4 *Very Good* 5

8. Speech Recognition is:

*Poor* 1 2 3 4 *Very Good* 5

9. Response Time of COSIMAR to your utterance is:

*Very Slow* 1 2 3 4 *Very Fast* 5

10. COSIMARs Voice is:  
*Very Synthetic* 1 2 3 4 *Very Pleasing* 5

11. Ease to learn the interaction:  
*Difficult* 1 2 3 4 *Easy* 5

12. Usage of the Virtual Receptionist was:  
*Difficult* 1 2 3 4 *Intuitive* 5

Comments:

13. I found the following aspects of the Virtual Receptionist very good:

- a. \_\_\_\_\_
- b. \_\_\_\_\_
- c. \_\_\_\_\_

14. I disliked the following aspects of the Virtual Receptionist:

- a. \_\_\_\_\_
- b. \_\_\_\_\_
- c. \_\_\_\_\_

15. What topics would you never talk about with the Virtual Receptionist?

- a. \_\_\_\_\_
- b. \_\_\_\_\_
- c. \_\_\_\_\_

16. What topics would you like the Virtual Receptionist to talk about during the interaction?

- a. \_\_\_\_\_
- b. \_\_\_\_\_
- c. \_\_\_\_\_

16. Further Comments

---

---

---



## D) Scenario Interaction with the Virtual Receptionist on the Phone

When the customer encloses all necessary information for mediation, the Virtual Receptionist calls the employee, tells him that a visitor is waiting for him, and asks when and where he will pick him up.

Now you are asked to take the role of the employee on the telephone.



### Some hints

- COSIMAR needs some time to response, so wait a few seconds until you repeat your utterance.
- If there is a really long pause, please repeat your utterance or try another one.
- COSIMAR has a small vocabulary, please use phrases instead of sentences.
- The functionality of COSIMAR is restricted to the telephone dialog. She neither does small talk nor gives general information.

### Detailed Scenario Steps

Imagine you are Charlie Brown and you have a meeting with Mary White at 2pm. It's 1:55pm.

1. If the phone rings – pick it up
2. Greet: Hello.  
And wait for answer and listen.
3. Answer: In a few minutes  
And wait for response.
4. Answer: At the reception  
And wait for response.
5. Did the Virtual Receptionist mediate the correct answer?  yes  no
6. Amount of time required to complete all tasks was:  

<i>More than expected</i>						<i>Less than expected</i>
1	2	3	4	5		

### Personal Interaction

Repeat previous scenario and use your own words, how you would react with a real receptionist.

Did the Virtual Receptionist mediate the correct answer?  yes  no

Amount of time required to complete all tasks was:  

<i>More than expected</i>						<i>Less than expected</i>
1	2	3	4	5		

## E) Evaluation Questionnaire for Phone Interaction

To help evaluating the Virtual Receptionist, please answer the following questions:

1. Overall, I am confident the telephone interaction supports the mediation of visitor and employee (Cycle one)

*Strongly Disagree* 1 2 3 4 *Strongly Agree* 5

Comments:

2. Amount of time required to complete the tasks was:

*More than expected* 1 2 3 4 *Less than expected* 5

Comments:

3. Using the solution was a satisfying experience.

*Strongly Disagree* 1 2 3 4 *Strongly Agree* 5

Comments:

4. The interaction with the system was natural:

*Strongly Disagree* 1 2 3 4 *Strongly Agree* 5

Comments:

5. Speech Recognition is:

*Poor* 1 2 3 4 *Very Good* 5

6. Response Time is of COSIMAR to your utterance:

*Very Slow* 1 2 3 4 *Very Fast* 5

7. COSIMARs Voice is:

*Very Synthetic* 1 2 3 4 *Very Pleasing* 5

8. Further Comments

---



---



---

## 10.2 Questionnaire User Test II

### A) Background Questionnaire

Please answer the following questions below in order to help us understand your background and experience.

Gender:  Male  Female

Age: \_\_\_\_

#### RECEPTION EXPERIENCE

1. How often do you ask the reception of a company to help or mediate you?
    - never
    - once a year
    - once a month
    - once a week
    - every day
  2. What do you generally ask at the reception?
  3. What are you usually asked by a receptionist?
  4. From your experience: How are you helped by them?
  5. What do you do while waiting?
- 

#### COMPUTER EXPERIENCE

1. What is the total length of time you have been using personal computers? \_\_\_\_
2. How much hours do you work at the computer per week? \_\_\_\_\_
3. How much experience do you have with speech user interfaces?
  - none
  - some
  - plenty

If you have at least some experience, where do you use a speech user interfaces?

---

4. How much experience do you have with virtual characters?
  - none
  - some
  - plenty

If you have at least some experience, where have you used virtual characters?

---



## B) Scenario Interaction with the Virtual Receptionist at the Kiosk

This prototype version of the Virtual Receptionist is designed to establish the contact between a visitor, approaching the reception of a company, and his meeting party.

The *Virtual Receptionist* combines different modalities to enable people using an application with natural language. The major characteristics of the *Virtual Receptionist* are several capabilities which include the ability to carry out a dialog with the visitor using speech input and output, virtual face presentation with facial expressions, and moreover to embed another dialog with a third party via telephone.

Now you are asked to play the role of the customer approaching the kiosk. Please fulfill the steps as described below.

### Some hints

- COSIMAR only knows the names and meetings stored in the database. So, please only use the suggested names.
- If the speech recognition recognizes your utterance, it will take some seconds until the virtual agent responds. Please be patient.
- If she recognizes something wrong, please repeat your utterance clearly.
- The functionality of COSIMAR is restricted to the mediation functionality. She neither does small talk nor gives general information.

### First Scenario

Imagine you are Mary White and you approach at the kiosk of the virtual receptionist. You are here to see Charlie Brown and you ask the virtual receptionist, to establish the contact between you and Charlie Brown. Please go ahead, take the headphone and talk in your own words to COSIMAR, so that she calls Charlie Brown and wait until she finishes.

Your Name                  Mary White  
Your Meeting Party      Charlie Brown

---

Was the contact established?    yes    no

Amount of time required to complete the task was:

*More than expected*                  *Less than expected*  
1                  2                  3                  4                  5

### Second Scenario

Now you take the role of James Green who wants to join the Research Forum at 10 am. He was invited by the employee Karin. Unfortunately Mr. Green only knows the first name of his contact. But to allow Mr. Green to attend the forum, the virtual receptionist needs to know at least the meeting information.

Your Name                    James Green  
Your Meeting Party        Karin  
Meeting Time                10 am  
Meeting Name                Research Forum

---

Was the contact to the responsible person established?         yes     no

Amount of time required to complete the task was:

*More than expected*

1

2

3

4

*Less than expected*

5

### Third Scenario

Now, you take the role of Ryan Arthur. He is here to see Grace Rainbow.

Your Name                    Ryan Arthur  
Your Meeting Party        Grace Rainbow  
Meeting Time                2 pm  
Meeting Name                Status Meeting

---

Was the contact to the responsible person established?         yes     no

Amount of time required to complete the task was:

*More than expected*

1

2

3

4

*Less than expected*

5

### C) Evaluation Questionnaire for Kiosk Interaction

To help evaluating the Virtual Receptionist, please answer the following questions:

1. Overall, I am confident COSIMAR supports the mediation of visitor and third party  
(Circle one)

*Strongly Disagree* 1                      2                      3                      4                      *Strongly Agree* 5

Comments:

2. Amount of time required to complete all tasks was:

*More than expected* 1                      2                      3                      4                      *Less than expected* 5

Comments:

3. Using the solution was a satisfying experience.

*Strongly Disagree* 1                      2                      3                      4                      *Strongly Agree* 5

Comments:

4. The interaction with the system was natural:

*Strongly Disagree* 1                      2                      3                      4                      *Strongly Agree* 5

Comments:

5. To me, the application in general is:

*Very bad* 1                      2                      3                      4                      *Very Good* 5

*Hard to figure out* 1                      2                      3                      4                      *Very Intuitive* 5

Comments:

6. COSIMARs Facial Expressions are:

*Very Unrealistic* 1                      2                      3                      4                      *Very Realistic* 5

*Very Unpleasant* 1                      2                      3                      4                      *Very Pleasant* 5

7. Speech Recognition is:

*Very Poor* 1                      2                      3                      4                      *Very Good* 5

8. COSIMAR understood me: (if your utterance was correctly was recognized)

*Not at all* 1                      2                      3                      4                      *All the time* 5



9. Response Time of COSIMAR to your utterance is:

<i>Very Slow</i>					<i>Very Fast</i>
1	2	3	4	5	

10. COSIMARs Voice is:

<i>Very Synthetic</i>				<i>Very Natural</i>
1	2	3	4	5

11. Learning the interaction was:

<i>Very Difficult</i>				<i>Very Easy</i>
1	2	3	4	5

12. Using the Virtual Receptionist was:

<i>Very Difficult</i>				<i>Very Easy</i>
1	2	3	4	5

Comments:

---



---

13. I liked the following aspects of the Virtual Receptionist:

a. 

---

b. 

---

c. 

---

14. I disliked the following aspects of the Virtual Receptionist:

a. 

---

b. 

---

c. 

---

16. What topics would you like the Virtual Receptionist to talk about during the interaction?

a. 

---

b. 

---

c. 

---

15. What topics would you never talk about with the Virtual Receptionist?

a. 

---

b. 

---

c. 

---

16. Further Comments

---



---



---

## D) Scenario Interaction with the Virtual Receptionist on the Phone

When the customer encloses all necessary information for mediation, the Virtual Receptionist calls the employee, tells him that a visitor is waiting for him, and asks when and where he will pick him up.

Now you are asked to take the role of the employee on the telephone.

### Some hints

- COSIMAR needs some time to response, so wait a few seconds until you repeat your utterance.
- If there is a really long pause, please repeat your utterance or try another one.
- COSIMAR has a small vocabulary, please use the suggested phrases



### Detailed Scenario Steps

Imagine you are Charlie Brown and you have a meeting with Mary White at 2pm. It's 1:55pm.

1. If the phone rings – pick it up.
2. Proceed as you would do it if the phone rings.

---

Did the Virtual Receptionist mediate the correct answer?  yes  no

Amount of time required to complete all tasks was:

*More than expected*

1

2

3

*Less than expected*

4

5

### Personal Interaction

Repeat the *Detailed Scenario Steps*

Imagine you are Charlie Brown and you have a meeting with Mary White at 2pm. It's 1:55pm.

1. If the phone rings – pick it up.
2. Proceed as you would do it if the phone rings.

---

Did the Virtual Receptionist mediate the correct answer?  yes  no

Amount of time required to complete all tasks was:

*More than expected*

1

2

3

*Less than expected*

4

5

## E) Evaluation Questionnaire for Phone Interaction

To help evaluating the Virtual Receptionist, please answer the following questions:

1. Overall, I am confident the telephone interaction supports the mediation of visitor and employee (Circle one)

*Strongly Disagree* 1      2      3      4      *Strongly Agree* 5

Comments:

2. Amount of time required to complete the tasks was:

*More than expected* 1      2      3      4      *Less than expected* 5

Comments:

3. Using the solution was a satisfying experience.

*Strongly Disagree* 1      2      3      4      *Strongly Agree* 5

Comments:

4. The interaction with the system was natural:

*Strongly Disagree* 1      2      3      4      *Strongly Agree* 5

Comments:

5. Speech Recognition is:

*Very Bad* 1      2      3      4      *Very Good* 5

6. Response Time is of COSIMAR to your utterance:

*Very Slow* 1      2      3      4      *Very Fast* 5

7. COSIMARs Voice is:

*Very Synthetic* 1      2      3      4      *Very Natural* 5

8. Further Comments

---

---

---



