

Towards a Persuasive Dialog System Supporting Personal Health Management

Bachelor Thesis
of

Vera Götzmann

at the
Institute for Anthropomatics and Robotics,
Interactive Systems Lab

KIT

Advisor: M.A. Maria Schmidt
Reviewer: Prof. Alexander Waibel
Second reviewer: Dr. Sebastian Stüker

NAIST

Advisor: Prof. Graham Neubig
Second Advisor: M.E. Takuya Hiraoka

Duration: 13. February 2015 – 12. June 2015

I declare that I have developed and written the enclosed thesis completely by myself, and have not used sources or means without declaration in the text.

PLACE, DATE

.....
(VERA GÖTZMANN)

Abstract

This work is aimed toward a system offering assistance regarding personal health management through persuasive spoken dialog.

As health awareness is a topic receiving much attention, the interest in self management of the personal health has risen in the age of the Internet. For this, people require a large amount of motivation, which is a promising area of application for a persuasive agent. With spoken dialog technologies having spread to global markets, it is an ideal choice to combine a spoken agent with the ability to be perceived more personally with the need for individual encouragement and support.

The objective of this work is to explore the potential for persuasion on a daily basis through spoken dialog for the common issue of a healthy lifestyle. A prototype system was created and evaluated. It implements scheduler features to support the regularity required by many health measures. Coupled with this is a persuasive routine, aiming to keep the user from veering off the path towards the goals he set for himself. Through encouraging feedback and suggestions toward making up for missed plans, the system attempts to steer the user back on track.

For this purpose the psychology behind motivation is explored, evaluating what kind of utterances are perceived as encouraging. With the results of these efforts implemented, the prototype was evaluated in a final user study measuring user interest and acceptance of the system.

The study found that overall user satisfaction correlated with the satisfaction concerning the persuasive part, suggesting it to have been more important to the user than the scheduler. Even more encouraging, scores concerning a general interest in this field were high, meriting deeper investigations. There was consensus that the persuasion was not annoying and the agreement of motivational effectiveness was high. Many participants commented on drawbacks concerning the scheduler or the used voice. As many of the encountered problems were linked to the framework used, an adaption to another environment seems a suitable approach for improvement. Future work should include employing more interactive persuasive suggestions and a broader sentence base.

Acknowledgments

This work was created partly at *Nara Institute of Science and Technology*, Japan as part of the InterACT exchange program.

Therefore I would not only like to thank my supervisor at the KIT, Maria Schmidt, who always remained patient and flexible, but also Prof. Satoshi Nakamura and the staff from the *Augmented Human Communications Laboratory* for welcoming me kindly. An additional thanks is due to Prof. Graham Neubig for his motivated and encouraging supervision at NAIST. I also want to give a special thanks to my second supervisor at NAIST for his continuous assistance and kindness. Takuya Hiraoka was usually the first person I turned to with questions or problems. Another person who deserves my thanks in more ways than one is my father Prof. Walter Götzmann for providing me with insights of academic approaches in scientific work, as well as a secluded working place when needed. The same goes for my mother Andrea Götzmann, introducing me to valuable sources on motivational psychology and insights on real life experience working with it.

Not to be forgotten are the *InterACT* program and the *Baden-Württemberg Stiftung*, without both of which the student exchange to Japan would not have been possible.



Contents

1. Introduction	1
1.1. Motivation	1
1.2. Introduction to Spoken Dialog Systems	1
2. Background	7
2.1. Application Fields of Spoken Dialog Systems	7
2.2. Persuasive Dialog Systems	8
2.3. Personal Health Care Systems	9
3. Introduction to the Health Assistant System	11
3.1. Application Scenario	11
3.2. Applied Dialog Management Principles	12
3.3. Usage Environment	12
4. Features	13
4.1. Start Up Routine and General System Behavior	13
4.2. Scheduler Interface	14
4.3. Persuasive Advising on Physical Exercising	15
4.4. Dialog Framework	15
5. Implementation	17
5.1. Working with the Jindigo Framework	17
5.2. Natural Language Understanding and Generation	18
5.3. Dialog Management	19
5.3.1. State Handling	19
5.4. Scheduler Functionality	20
5.4.1. Grammar Design	21
5.4.2. Preliminary User Evaluation of the Scheduler	21
5.4.2.1. Scenario and Realization	21
5.4.2.2. Results	22
5.5. Persuasion for Exercising	23
5.5.1. Motivational Sentence Study	24
5.5.2. Results and Impact on the Implementation	25
5.6. Encountered Framework Limitations	26
6. User Evaluation	29
6.1. Evaluation Targets	29
6.2. Questionnaire	30
6.3. Scenario	31
6.4. Results and Discussion	32
7. Future Works	35

Bibliography	37
Appendix	43
A. Code Examples	43
A.1. Grammar and Configuration Example XML Files	43
A.1.1. XML File with Program Configurations	43
A.1.2. Input Grammar for General Program Functions	44
A.1.3. Output Grammar for Persuasion Specific Responses	47
B. Persuasive Phrases	51
B.0.4. Motivational Phrases	51
B.0.5. Appraisal Phrases	51
C. Evaluation Files	51
C.0.6. Final Evaluation Questionnaire	51
C.0.7. Final Evaluation Dialog Example	56
C.0.8. Final Evaluation Statistical Results	57

1. Introduction

1.1. Motivation

Since early works of the Science Fiction genre, humans always imagined talking to machines¹. Interaction through spoken conversation is the most natural sort of communication for us humans. Children are able to talk years prior to learning to read and write. Yet interaction with computers is still mostly limited to text and graphic manipulation---though the recent years have seen some advances in this sector. Thus, as a continuing development, *Spoken Dialog Systems (SDS)* offer many intriguing future uses, and research in this direction can be expected to stay interesting in years to come.

One prevalent vision are virtual personal assistants with a natural language interface, devices to assist with mundane everyday tasks (as in picture 1.1 from 1968). Ideally they should act as attentive secretaries, taking spoken instructions while the user can be focused on something else. It is easy to access that a personal human secretary with around-the-clock attendance for everyone would be unfeasible. Yet with computing devices starting to accompany us every step of our lives, virtual assistants offer great possibilities. Apple's Siri program is a prominent example of efforts towards this goal. The purpose of this thesis lies in a similar direction.

So in what areas of life does the user require assistance? Possible usage is not limited to secretary tasks only. Reflecting on domains of human assistants, another example comes to mind: Personal fitness trainers and health advisers. With the rise of public awareness towards the importance of a healthy lifestyle, a general willingness to integrate healthy practices into their lives is demonstrated by many people. Yet putting those actions into place often falls victim to the hassle of everyday life. Whether a person feels too stressed, simply forgets about health examination appointments, can not conjure up enough motivation---there are abundant reasons. A supportive and helpful sentence from time to time, a friendly reminder to avoid procrastination, these are tasks an assisting spoken dialog agent could attend to.

This work aims at evaluating user interest in and acceptance of a spoken agent designed for motivating healthy behavior through the development and testing of a prototype of such a program.

1.2. Introduction to Spoken Dialog Systems

While everyone has an intuitive understanding of what a *dialog* is, computer science calls for a more specific definition. A dialog is defined as a communicational exchange between two

¹For one among an innumerable quantity of examples see "I, Robot" from 1939 [Bin39], a short story influential to famous SciFi authors like Isaac Asimov, featuring the dilemma of a robot capable of speech---yet out of fear no one will listen.



Figure 1.1.: A vision from Licklider and Taylor [LT68, p.39] about computers as communicative Devices.

participants² taking turns speaking. What is spoken during a turn can be separated into one or more utterances,³ which can then be classified as different dialog acts. Such a classification “is a means of capturing and encoding different levels of discourse structure, and identifying how they relate to one another at the pragmatic level”([GMM00, p.57]) and thus helps to identify purpose and effect of what is said.

Being able to participate in a dialog, SDS are generally virtual agents that can not only output spoken language (an easy task with prerecorded voices), but can accept, interpret and react to spoken input by the user, too. Thus services can be provided to the user in a natural and intuitive way, leaving eyes and hands free for other occupations.

The functional components of these systems are customarily classified ([McT04]) into automatic speech recognition, natural language understanding, dialog management, natural language generation and text-to-speech synthesis (the relations of which are depicted in 1.2). The purpose of the *automatic speech recognition (ASR)* is to convert an audio signal into to virtual words and phrases. This is done by it matching the incoming audio signal with stored acoustic patterns of phonemes, searching for the best match. In order to do so, the most relevant features have to be extracted from speech audio signals. Resultant features are classified into phonemes, and through the combination of those phonemes words can be expressed in an *acoustic model*. The acoustic model assigns probabilities to the different phonemes making up the pronunciation of single words. These probabilities are then supplemented by a *language model* holding information on possible word sequences through grammar rules or assigned probabilities in N-grams⁴.

The data containing probable word sequences is passed along to the *natural language understanding (NLU)* unit to interpret and assign meaning to what was spoken. Allowed sentences can be described by rules of allowed grammatical compositions including given words in a language. Or

²In this context: a human user and a computer.

³“An uninterrupted chain of spoken or written language.” [Dic]

⁴An N-gram is a sequence of n words in texts. Combining a large amount of N-grams, assumptions about the probability of a word in a given context can be made.

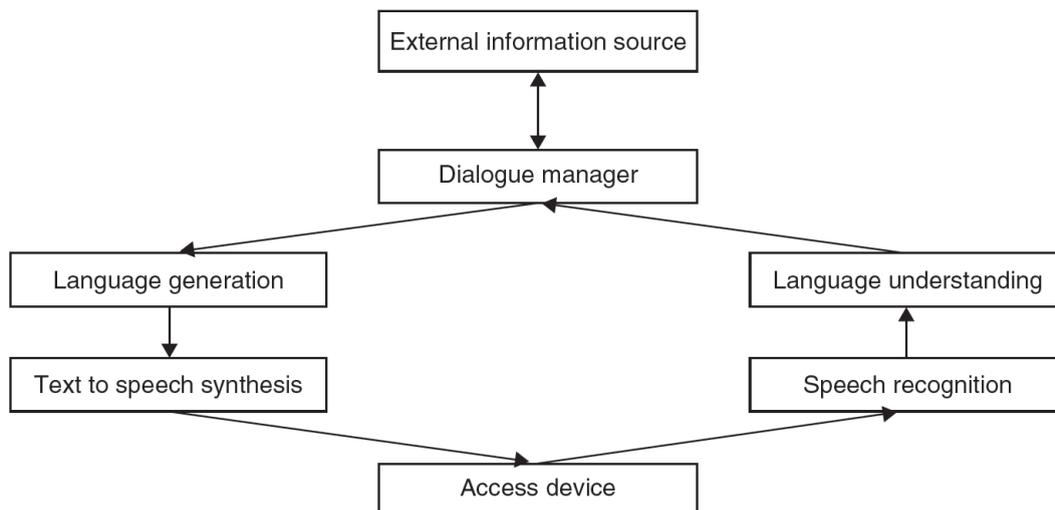


Figure 1.2.: Typical component layout of an SDS. [McT04, p.80]

for another approach, by rules describing how the meanings behind words can interact (disregarding the question of grammatical correctness). Through *parsing*, the various ways to combine the defined rules are explored and stored in a parse tree.

Through parsing it with these rules meaning is assigned to input, passing the resulting concepts of speech along to the *dialog management (DM)*. Using *rule-based* or *statistical* approaches, concepts are processed. During processing, they are frequently matched with a previous dialog history or enhanced by utilizing external knowledge sources. Based on policies specific to each application, it is ruled what course of action would be an adequate response to the input.

After the action to take has been decided on, an utterance is created accordingly by the *natural language generation (NLG)* component. In many cases, *Canned text*, a predetermined response structure where specific data can be added as needed, is used for this. The phrases created through merging the canned text and the data are then handed over to the *text-to-speech (TTS) synthesizer* in order to generate the output audio.

For strongly limited outputs, prerecordings of the canned text might simply be combined together. More flexible output needs speech synthesizing however. In a first *text analysis* step, phonemes for output are produced. This involves several steps. Depending on the structure of retrieved text, *text segmentation* separates it into smaller units (e.g. sentences). Any abbreviations or standard forms like measurement units are examined by *normalization*.

To deal with the problem of storing large amounts of word pronunciation information, *morphological analysis* is used to create different related forms dynamically from the stored root form of a word. Next *syntactic tagging* determines which part of speech a word belongs to, as this can have an effect on the words pronunciation and the emphasis on its syllables. Having prepared little speech segments like this, *continuous speech effects* need to be modeled to achieve natural sounding speech and reduce chipping sounds between the segments. At this point, we arrive at an abstract linguistic representation of the text. A *speech generation* process will then compute a parametric continuous representation out of the phonemes given by either modeling characteristics of the vocal tract or characteristics of acoustic signals. The continuous representation's waveform is then used as output.

Theoretically a text-only dialog agent can be converted to a spoken one with a manageable amount of adjustments through inserting a *ASR* and *TTS* unit as an additional process layer between user and system. The main differences are other occurring error types and a change in how utterances are phrased to which the *NLU* should be adapted accordingly.

To limit the scope of this work, the focus is on the decisive routines of the dialog system, namely *DM*, *NLU* and *NLG*.

Technical incarnations of SDS are numerous. Existing services deploy a large variety of different platforms; Ranging from the usage of computers (both locally and over the Internet via VoiceXML) over phone services, to smartphones and other small gadgets connected to a cloud offering continuously updated service with presentable quality⁵. With this, the location of usage also varies from locally tied (e.g. on the PC) to available everywhere, anytime. This also poses interesting challenges concerning different sound recording techniques, background noises and multi-user-conversations.

An important aspect of dialog modeling is who is taking the leading role, as explained by Jokinen and McTear [JM09]. Programs where initiating topics can only be done by the system tend to be robust, but also feel unnatural and limiting to the user, requiring him to adapt his speech to the system's needs. With the varying degree of system initiative, mixed initiative and user initiative, the user's role in the conversation changes from information providing over issuing commands and question queries to taking part in a mutual exchange.

Internally, this effects the principles of how dialog control is implemented. For programs led by the system, a basic rule based approach are *graphs*. All dialog acts are described in a state graph, where user responses traverse pre-designed states of the program (e.g. Levin and Levin [LL05]). The approach limits what the user can say, offering robust recognition however.

Yet besides the limitation of the user's speech, a main drawback is the amount of definitions and states needed for more complex dialogs. An example by Jokinen and McTear [JM09, p.25] states: "given five items of information to be elicited, each with three possible values (unknown, known, verified), 3^5 nodes would be required, giving rise to an even larger number of transitions."

Toward avoiding this, system designers can use so-called *frames* to illicit required information from the user. Frames define what information the system requires from the user, but no fixed states in which way to achieve them. Instead of designing which answer will lead to which question directly, the system will check and ask for missing information in its frame, but is still able to process and store further information the user might offer on his own accord. This greatly reduces the complexity of transitions, as only one graph state per frame is necessary. Moreover, saying a whole sentence including all information is a more natural way to communicate than just offering information snippets on their own⁶. Frames still predefine the topics and general flow of the conversation and so complexity remains a major factor as well. If the handled dialog topics increase, the amount of frames needs to be increased accordingly.

To alleviate this, other approaches discard of the notion of manually designed rules and deploy machine learning instead. Using spotting keywords, the system tries to arrive at its own strategy on how to respond and what utterances seem the most appropriate in the given context.

No matter how extensive graphs and frames implement their rules, they always remain limited to a special domain consisting of previously known topics and concept, only spotting keywords or recognizing rule based utterances.

Statistical Models on the other hand are mostly limited on the database used to train them. Still profiting from limited domain in terms of robustness (and training time), given enough text corpora as learning examples, nothing keeps statistical models to be trained for all kinds of unexpected user inputs in an open domain. To further this, the system can be continually improved during usage, using statistical online learning on the dialogs it has already encountered.

Through constant improvements speech enabled systems have become wide spread, nonetheless there are still many challenges. There are general factors limiting the use of SDS under

⁵Most of the speech featuring products prominently in the news lately deploy cloud servers to increase database and processing power quality, thus reducing hardware limitations. (Cp. the examples mentioned in 2.1.)

⁶For an example please refer do the *Philips Automatic Train Timetable Information System* [Aus+95].

certain conditions, unrelated to the specific implementation, stemming from spoken dialog as a whole [McT04, p.40-p.41].

One problem is of a social nature: privacy and consideration towards other people. Being what it is, the usage of spoken systems is not silent. A user could find himself in a situation where silence is required or expected, rendering a spoken service inaccessible. Or there might be people to whom talking with a system about private topics is embarrassing when being surrounded by other people. Furthermore, looking at the coin from the other side, not everyone is happy to listen to nearby passengers chatting vividly with their agent while commuting home after a long day of work.

Also, dialog systems are not suitable for all service types. Imagine a system reading out a list of a few dozen information messages matching the users query. In some circumstances, the system can try to guess the information the user is most interested in and to output only that one, but this is not always possible. Humans invented figures, graphs and diagrams for a purpose. Some information can be delivered within a few moments when represented in a graphical fashion, while a detailed text description of the same can be lengthy, tedious and difficult to understand.

Leaving those general limits aside, technical challenges are abundant as well. Massive research efforts are unceasingly targeted at improving the speech recognition in a variety of adverse conditions, such as dialects⁷, background noises, echoes and multiple people talking [SA12]. Letting dialogs sound more natural and adapting to the conversational partner in a conversation are challenges for the dialog management keeping researchers busy for years to come.

Above mentioned limitations have sparked much research toward improvement of these conditions. Yet meeting the vivid imagination of humans concerning what programs should be able to do is hard to beat. For decades the science fiction genre has put out new ideas regarding technologies of the future, showing robots and other speech enabled systems capable of complex dialogs and astonishing problem solving abilities.

Anthropocentric research and the drive to understand the inner workings of humans is as old as mankind, yet the how and why of human behavior are still not understood in a satisfactory manner. So it constitutes a mayor challenge for a field as (comparatively) young as Computer Science to produce systems mimicking this complexity.

For a more complete introduction to the world of SDS, extensive literature is available. Please refer to Schmitt and Minker [SM13], Jokinen and McTear [JM09] or similar works.

⁷According to Kudo et al. [Kud+96] on Japanese dialects, the speech recognition accuracy is (only) influenced by 2-4%. However for many languages spoken around the world a greater variety of dialect exists, not to mention speech input from non-native speakers or children for example.

2. Background

2.1. Application Fields of Spoken Dialog Systems

The beginning of the 1990s saw the increased use of telephone based service systems as the first representatives of large scale commercial dialog systems. To this day many other representatives of such systems have been made available. Information retrieval services for weather inquiries or public transportation schedules, [Rau+03] customer support systems handling the most common problems automatically, etc.---the uses are versatile.

These applications are available from everywhere just by dialing a phone number. Through the usually very limited functionality, which in turn allows for a limited recognition grammar, high accuracy of the understanding can be achieved. Examples for other suggested or already employed fields for the use of spoken dialog agents are in-car systems like hands-free navigation control, [Sch96] educational programs acting as private tutors [LF05] and providing services to illiterate people. [Raz+12]

Classical tasks carried out by SDS found in literature during or as a result of a dialog are providing information, manipulation of data or the environment and mere chatting. These jobs aim at fulfilling the user goal, thus becoming the system goal as well. If the application *provides information*, the user files a request and retrieves either simply data from a database or, in slightly more complex scenarios, is given an instruction of actions to take. In case of *environment manipulation*, the program accepts user input and revises a database or carries out actions in the physical world as a result.¹ In this case the user goal is the machine's completion of actions in the user's stead.

In these scenarios the distinguishing feature from dictation or spoken command applications is derived from the system's ability for dialog, enabling it to pro-actively request required information and check back with the user in case of problems or uncertainties.

Another much researched domain are *chatbots*. Not task-oriented like the previous examples but systems of entertainment value mostly, trying to fascinate users with their human-like behavior. They are often put to the *Turing Test*, first introduced by Alan Turing in 1950. [Tur50]

He suggested playing an 'imitation game' with two human participants and a computer. The first human acts as the judge. He will interact with both the human and the machine in a way that permits him from directly knowing which is which². His task is to decide after five minutes of interaction which conversational partner is human and which machine. The computer system

¹As is often done in the field of robotics.

²Turing suggested separate rooms and an intermediate, today usually digital text is used.

and the second human should each try to convince the judge that they are the human. In the end, the judge announces his deduction who might be the human or the machine.

In his article, Turing stated his believe that in the year 2000, systems would be advanced enough for it to be common that judges can only decide correctly in 70% of the time. For today's Turing Tests, this figure is used as the passing threshold, requiring a program to fool people at least 30% of the time. A popular example of a text only conversational agent available on the Internet is the website *cleverbot*. [Car] According to some sources, it was the first program to ever pass the Turing Test in 2011. [Aro11b] Other sources disagree and name the program *Eugene Goostman* to be the first to succeed in 2014. [Rea14]

The aforementioned purposes for SDS can be combined in one application to enable a broader spectrum of uses. One of the most basic combinations is a system for database handling, both providing information and manipulating data on user demand. When thinking of how a chatbot might produce interesting conversations, the idea to use external datasources to retrieve up to date information about the world lies at hand³. Researchers have also developed systems with their own agenda. For example conversational agents who shall gather information about their environment through the conversations they hold. [Hol+06]

Yet human imagination has long foreseen more advanced uses for SDS---agents that act as assistants and companions. In 2004, McTear depicted a fantasy scenario in his book, describing well what many people have expected talking agents to be able to accomplish for a long time. "In the fantasy scenario Rachel was able to call on a virtual assistant to check and rearrange her appointments, place calls and check her email." [McT04, p.13]

Undoubtedly Apple's *Siri*, introduced to iPhones in 2011, fulfills many expectations of this kind⁴ and brought public awareness to possibilities of spoken dialog agents outside science fiction, presumably becoming the most prominent spoken application.

Other developments in this direction can be expected in the future, as the recent release of the Amazon *Echo*, [Inc] a spoken cloud-based assistance device designed for the use at home. Even the entertainment industry is starting on advanced commercial applications; Toy manufacturer *Mattel* announced the speaking Barbie Doll toy '*Hello Barbie*' for fall 2015. [Sar15]

2.2. Persuasive Dialog Systems

For a long time, spoken systems deployed in practice mostly constituted simple command and query applications. Such spoken systems are more user utilities than active helpers, enabling users to access common services and programs through a more interesting interface. The recent years has seen a rise of systems who act more conversationally, rendering the conversation not only useful but enjoyable as well. Still, these systems usually take a passive role in the conversation, reacting and responding to the current user query, without any long term agenda.

This leaves a large potential untapped, as there might be more advantages specific to spoken dialog over other interfaces. Generally, people will act differently if they know they are conversing with a machine. Yet when the robot avatar of the Japanese professor Hiroshi Ishiguro (as described by Sakamoto et al. [Sak+07]) was created for remote student consultations, test subjects were found to attribute the robot a human presence. The dialog itself was of course human-to-human. Nevertheless this serves to demonstrate humans' readiness to regard human-like machines as persons. Speech is a very natural human interaction, so potential can be seen of advanced SDS being attributed with the character and personality as well.

As humans are social creatures, they are influenced in their behavior and opinions by other persons⁵. This could be used by applications to influence a user in a desired direction. If the

³[Tat+03] compares their own Internet-based conversational system with seven others.

⁴As discussed in [Aro11a].

⁵Exemplary, Schwimmer [Sch04] identified positive reinforcement and social support as successful concepts in adolescent weight management programs.

user is able to bond with the agent, the application could act as a helpful companion, actively supporting the user in reaching his goals.

Systems with manipulation purpose where the environment to be manipulated is the user himself respective his opinions and plans are called *persuasive systems* for their goal of persuading the user to a certain opinion or action. The application discussed in this thesis is part of this group, aiming at helping the user to help himself. Still, existing concern about the possible misuse of such influencing systems for nefarious purposes deserve their due consideration as well. [Pur+11, p.427 - 431]

In recent years, the development and research of systems with their own agenda has greatly risen in interest. Ranging from goals congruent, [Hir+13; MRC07] independent [Hol+06] or even contrary to the user goals. [Tsu+15] This leads to a need to balance user and system goal satisfaction, as a tool without benefit for the user might not be used voluntarily, and as such will not be able to achieve its own goal.

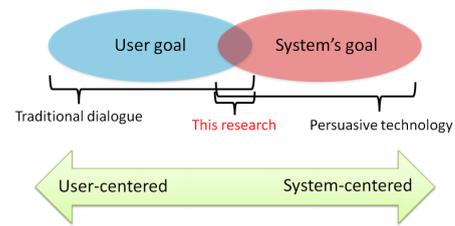


Figure 2.1.: Research striving for the balance between user and system goals (taken from work by Hiraoka et al. [Hir+13, p.1])

2.3. Personal Health Care Systems

The age of the Internet of free information is also the age of a certain do-it-yourself-mentality. Being in charge of ones own health and well-being and improving one's body has been a concern of many people⁶. Thus blogs, websites and videochannels about health information and self-improvement prosper on the Internet, attracting crowds of followers⁷.

The creation of programs aiming at assisting users in this objective through a diversity of tasks is a logical next step: Documenting health facts (nutrition intake, symptoms, miles run) as an aid to self-monitoring, providing diagnoses, etc. Figures of app stores serve to show just how wide spread this subject is. Estimates name 87,000 fitness- and health-apps as well as 55,000 medical tools in the standard online stores for smartphone apps. Some of these are even issued by health providers such as physicians and insurances, to facilitate their services. [Krü15]

One example to note is the app *My Diet Coach*, [LTD] employing many contents in this research like exercise tracking, reminders and motivational phrases to motivate a user long term. The notable differences are the lack of a spoken interface and the way motivational sentences are used. The app does not react to the user behavior though, instead the user has to actively request to be shown a few motivational phrases per “panic button”. As such, getting motivated by the program is a barely interactive self-service feature. Plus, speech is not utilized as a communication medium.

Another interesting approach is the environmental persuasion system Mathew [Mat05] describes, aiming at little everyday behavioral changes (like taking the stairs instead of the elevator in a subway station) to increase health. Instead of through users active usage decision, it can interact through changing the environment. By trying to catch a passerby's attention in a playful way, putting a nudging message towards desirable behavior into the peoples' view and then rewarding the better action again through entertaining feedback, it can reinforce small habit changes in a large group of people passing by.

⁶Bish et al. [Bis+05] found 46% of American women and 33% of American men were trying to loose weight in the year 2000.

⁷Lauckner and Hsieh [LH13] summarizes data sources for the year 2011. They show amongst other things that the Health Network WebMD, a leading source of health information, had a 29% increase on both unique users and total page views from a year earlier. [Web11] Project and Foundation [PF11] as well states “Eight in ten Internet users look online for health information”.

Of course, this trend was not missed by spoken dialog developers, naming healthcare as “an important new domain for spoken dialogue applications.” [JM09] Graph based spoken systems were developed in the middle of the last decade, providing telephone access to patient data collection for screening purposes. [Bla+04; LL05]

In [Pur+11], a group of persuasive system designers describe a persuasive agent, providing automatic data measurement, spoken persuasive conversations based on the user's behavior and the implementation of peer pressure to motivate people to embrace a healthier way of living---including everything this thesis suggests, and surpassing it by taking things even further. It is fictional though, an invention how systems may look in the future, combining many of the previously mentioned techniques and approaches. It serves the whole purpose of sparking a critical discussion in this field of research concerning the difficult balance of such systems which impose on the individual's will, and the thin boundary of ethics.

To summarize, it has been shown that a multitude of systems in the domain of health exist, trying to improve a user's life, marking the high relevance of this topic.

Yet the described systems only combined either relatively simple data tracking with a spoken interface, or were pure tracking applications, basing motivation mostly on the user getting rewards (e.g. scores, achievements to share socially, additional gimmicks for the application...) for his tracked progress. That being the case, the attempt to combine the easy access and more personal experience of a spoken dialog interface with data tracking and motivational approaches seems justified.

3. Introduction to the Health Assistant System

The system developed in this work is a computer program with a spoken dialog interface, aiding the user in managing his own personal health through keeping track of physician and exercise sessions in a calendar and attempting motivation to increase exercise frequency.

3.1. Application Scenario

The main goal of this research is toward a spoken dialog service that can actively assist and encourage a user in reaching self set health goals. This is attempted by bringing the user plans he set himself to mind and validating whether they were realized, encouraging him to commit to previously made plans or at least undertake compensating activities soon.

To achieve this, the program possesses the following features:

- The basis is formed by a scheduler service as it keeps track of elapsed time, e.g. concerning appointments with physicians, sports activities the user resolved to do, etc. It deploys a checking routine to verify whether the user kept his appointments and plans.
- Based thereupon a persuasive routine is build, reacting to the user's compliance to his schedule.
- On top of that, if the user lets things slide, these routine attempts to convince the user to take appropriate steps to increase his health by issuing positive encouragements for training.

So why were these specific features chosen?

Achieving a healthy lifestyle requires regularity of health measures. Committing to an extreme high impact exercise on occasion has it merits too, but to positively effect a persons health status long term, establishing a regular workout routine is essential. In addition, regular medical checkups are important to pick up on possible health problems early on, especially for people with health conditions requiring periodical screening. Previous research has proven that reminding patients helps to achieve higher compliance rates to health behavior plans. [MNR89; Les+10]

As Bish et al. [Bis+05] states, many people are supporting the notion of working towards a better health and try to loose weight, yet the percentage of overweight people is increasing steadily in most developed countries¹. So why does the wide spread willingness to improve one's own health not merit results in health statistics? Well, a good intent is admirable, anyhow in addition the

¹Summaries of the new WHO study "projections in adults to 2030", presented on the 22nd European Congress on Obesity show devastating predictions for overweight and obesity rate in many European countries. An example

so-called *volition*, commitment and long term motivation² for action is needed. This is the reason for the persuasive encouragement approach, as Mazzotta, Rosis, and Carofiglio [MRC07] showed how positive emotional motivation increased the persuasive ability in the described problem of motivating healthier eating habits.

3.2. Applied Dialog Management Principles

From the dialog design choices illustrated in 1.2, this system was designed with the following properties:

The ensuing dialog is of a mixed initiative, the dialog control is shared. Depending on data input in past sessions, the system might start introducing the topic of date notifications and sport motivation. If not, it will wait for the user to issue a command. Entering a command handling routine the system will proactively seize initiative again to fill the slots of its information frame. Nonetheless the user can still issue new commands at any time or offer additional information.

So called barge-in---the user's ability to interrupt the computer's speech---is partly available³. Recognizing user utterances is a mix of rule-based and keyword spotting policy, trying to match defined rules with spotted keywords⁴.

Additionally, to accommodate for the occurrence of misrecognition or intended nonsensical input, the system is designed to allow a return to previous states from almost all parts of the dialog. Wrong input may very well lead to a wrong internal program flow, but the basic user commands are available from anywhere, enabling the user to easily switch to the desired task regardless of previous misunderstandings. An erroneous program flow will result in unfitting system prompts however. Before data manipulation or exiting the system requires a confirmation, in a further effort to lessen misrecognition consequences.

3.3. Usage Environment

The following assumptions about the usage environment were made for the system design:

This service is designed for execution as a stand alone system on a Personal Computer in a private home. Little hardware limitations result from this (as opposed to the usage on a portable device), although there are still limits on the computational power of a standard PC that have to be heeded. Being designed for usage at home, a generally low background noise level can be assumed, posing no extra difficulties for the speech recognition accuracy. A single user is assumed as multi-user support is not required for research purposes. Yet adding this feature lateron would require only minimal change by way of storing multiple different calenders (one for each user) and handling them within the dialog system.

The typical user this systems aims at can be of nearly every age⁵. Though an average user may or may not have prior experience using spoken applications (e.g. on smartphones), this is not assumed and the system should in general be usable without it, aiming at a dialog natural enough to understand typical sentences uttered by novice users. Only a short briefing on what the program is for, and what tasks it can carry out should be given in the beginning. If the user wishes for instructions on what to say during usage later, he can fall back on the built-in help functionality.

Furthermore, a regular---ideally daily---usage of the system is recommended. Giving the user a long list of information in a spoken manner is unfavorable as explained previously in section 1.2, so it is best to not have too many reminders in queue when the system is started.

is the UK:

“Three in every four men and two in every three women in the UK will be overweight by 2030” [Cha15]

²Previous research like Moorman and Matulich [MM93] already discussed the influence of motivation on health behavior.

³This is a default behavior from the used framework described in 5.1 and could not be changed by the designer.

⁴As with the above, this is part of the functionality of the framework.

⁵As long as there are no influences to the ability for speech (→Toddler) or hearing (→Elderly).

4. Features

4.1. Start Up Routine and General System Behavior

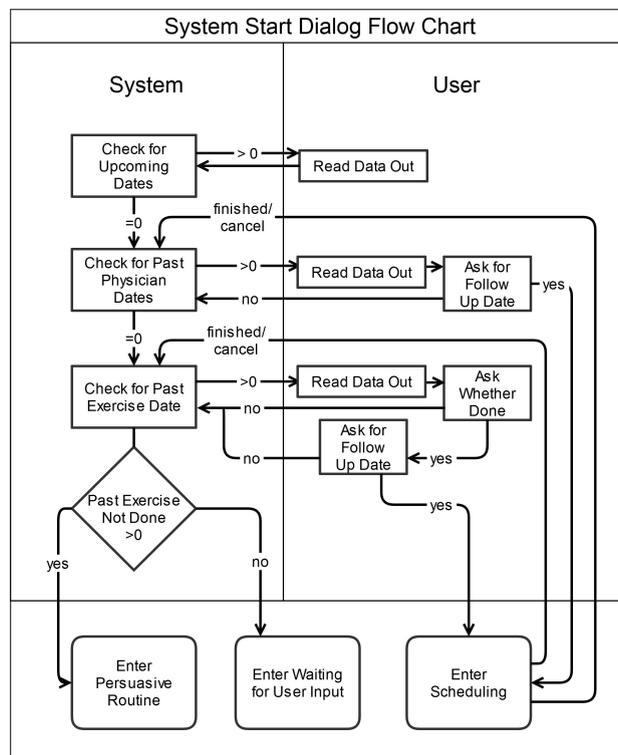


Figure 4.1.: The dialog flow on system start.

Let's have a look on how the system is realized from a user's point of view.

When the program is started, there is a start-up routine as displayed in figure 4.1. First, a welcome message is played. Meanwhile, the system accesses the calendar to retrieve upcoming dates. If some appointments due soon are found, they will be played back one by one. Once there are no more upcoming notifications to be read out, the system starts looking at schedules that lie in the past, but have not been discussed with the user yet.

For past physician appointments, the system will ask whether a follow-up checkup should be

stored, and then resumes reading out the next entries. In the case of a training plan however, the application prompts the user to confirm carrying it out. For a confirmation, the user will receive praise and asked for a follow up just like with the other appointments. A negative answer will be remembered by the system, but the output process will continue with the next appointment for the time being.

Once there are no more date notifications, depending on whether all training plans were carried out or not, the system will enter a routine to try to persuade the user to do sports, or just ask the user what else it can do for him, waiting silently for input afterward.

If the user has agreed to a follow up plan during any occasion, this current flow is suspended and the scheduler becomes active. Once a new date is stored or the user expresses his wish to cancel the storing process, notifications are picked of where they were left.

During this whole routine, as well at any other given time during system usage, the user can interrupt the system by requesting it to carry out another task for him. As with the date storage before, the active routine is suspended until the new task is completed or aborted, then resumed where it was left off.

The constantly available requests are: *Store date*, *retrieve date* (both explained in 4.2 below), *help*, *repeat*, *return*, *suspend* and *exit*.

Help: Tells the user about system usage and available commands.

Repeat: Causes the last relevant prompt to be repeated to the user, in case he forgot or did not understand properly

Return: Aborts execution of the current task and either returns to a task execution suspended before or, in the absence of such, confines itself to waiting until the user requests another function.

Suspend: Stops listening to all user input except for commands to return and continue task execution, so that the user might focus on something else for a while without the system responding to all noises made. That being the case, this poses as single exemption in that the commands mentioned here are not available while suspended.

Exit: Quits program execution. To avoid accidental system shut down due to misrecognition¹, a prompt for confirmation is issued before finally exiting.

4.2. Scheduler Interface

For scheduling a follow up, adding a new date in general or retrieving information about the calendar, the system launches a scheduling dialog.

This is merely a calendar interface with limited functionality though, because engineering a fully developed calendar application is well beyond the scope of this thesis. Most widely-used calendar programs have been developed for many years by experienced programmers. It would be neither constructive nor feasible to attempt to recreate this in the scope of a Bachelor thesis like this one. Little benefit would be seen in it, as so many advanced programs already exist. The focus on this thesis is on dialog management, both in general and with a persuasive component, and the calendar functionality is only a means to this end, providing a background and data to be acted upon.

That being said, let's have a look at the calendar interface. It is designed to guide the dialog in accordance to the appointment frame. This needs to be completed with information about date and time of the appointment and what type of plan it is².

When requesting the *storage* of a date, the user can offer any of the above described information right away, or the agent will ask for any missing and necessary information. In the end the appointment is read back and confirmation asked. If the user answers negatively, he will be

¹“Dentist” and “end” are rather similar words to a recognizer...

²Types include diverse physicians, workout or miscellaneous (for types not especially handled)

instructed to just repeat the information that was gotten wrong. Yet this instruction is only for the user's sake. Information given twice anywhere throughout the process (not just at the end), is assumed to be a correction of a faulty input and replaces the previous information.

For date *retrieval* the system only requires the user to limit the results at little. So it is enough if the user either states a date or a type of appointment he is looking for. Matching dates will be read back without further confirmation of the search filter. If some input was misunderstood, the user is able to notice this when the dates are read out to him and is able to just issue a corrected query.

4.3. Persuasive Advising on Physical Exercising

As mentioned in section 4.1, one part of the system's strife to motivate the user is already done when praising him for done exercises.

The other part are the persuasion attempts, triggered when plans for exercise were not kept. In order to not annoy users pointlessly, the system will not react to every single instance of foregone plans, instead it will remember their existence and only react after all past schedules have been discussed. Then it talks with the user about having missed workouts and tries to make him consider his goal again, by giving a little motivational speech.

Consecutively it will attempt to convince the user to exercise in that very instance, asking whether he has not got any time. In case of a positive answer, the user receives positive feedback, and then is asked whether he would also like to schedule another session for the future and according to the answer will switch to the scheduling of a date.

Declines the user to exercise at that given time though, the system will make another suggestion to schedule a new session soon. If this fails again, the user is asked whether he wants to decide a new appointment for a different, freely chosen date. Following a repeated rejection, the system will respect the users decision with a short cautionary notice to not loose sight of his health goals, and not annoy him with further inquiries. After this the persuasion routine is ended and the systems awaits commands again.

The motivational phrases and praise given are randomly selected from a set of phrases, to keep repetitiveness limited.

4.4. Dialog Framework

To be able to focus on the dialog management aspects of creating a conversational agent, the usage of a framework offering basic features was suggested from the beginning.

The general idea was to be unburdened by details of *ASR* and *TTS* synthesis and reduce *NLU* and generation *NLG* complexity through an API handling the details of integration with the *ASR* and *TTS* components. One more requirement was to have the code open source.

Two seemingly adequate frameworks were found: *Jindigo* and *Ravenclaw/Olympus*:

Ravenclaw is a dialog manager developed by Dan Bohus and Alexander Rudnicky at the *Carnegie Mellon University* for integrated use in the Olympus Dialog framework, using mostly C++. [BR09] From the website [BR] at first glance it seemed appealing, yet broken download links on the site and the newest date found being 2009 did not leave a good second impression. The biggest problem however was the limitation to Windows only development. The main part of this research was to be done at the *Nara Institute of Science and Technology*, where the available workspace was equipped with Linux based operating systems. So Ravenclaw was discarded as possible choice.

Jindigo was the second possible choice. It is "a Java-based open source framework for implementing and experimenting with incremental dialogue systems", [Ska10] developed by Gabriel Skantze at the *Department of Speech Music and Hearing, KTH*. The website stated, that Jindigo is at

an early stage of development³ and scarcely documented, [Ska] but Download links worked, the demonstration programs seemed to demonstrate the full functionality for Dialog management and---most importantly---it is platform independent.

Thus it was decided to use Jindigo as a framework for the dialog management⁴.

³The note at the top of the web page, stating that development is discontinued, was not in place yet. Furthermore, the other dialog framework system IrisTK introduced there [SA12] was inadequate for the same reason as Ravenclaw: Only development under Windows is supported.

⁴A more thorough description of Jindigo follows in 5.1

5. Implementation

5.1. Working with the Jindigo Framework

Things are not self-explanatory when starting on the journey to develop a Jindigo program. So the workings of Jindigo as perceived during the practical part of this thesis shall be explained in this section. Especially since that knowledge is partly essential for the execution of this thesis's system¹.

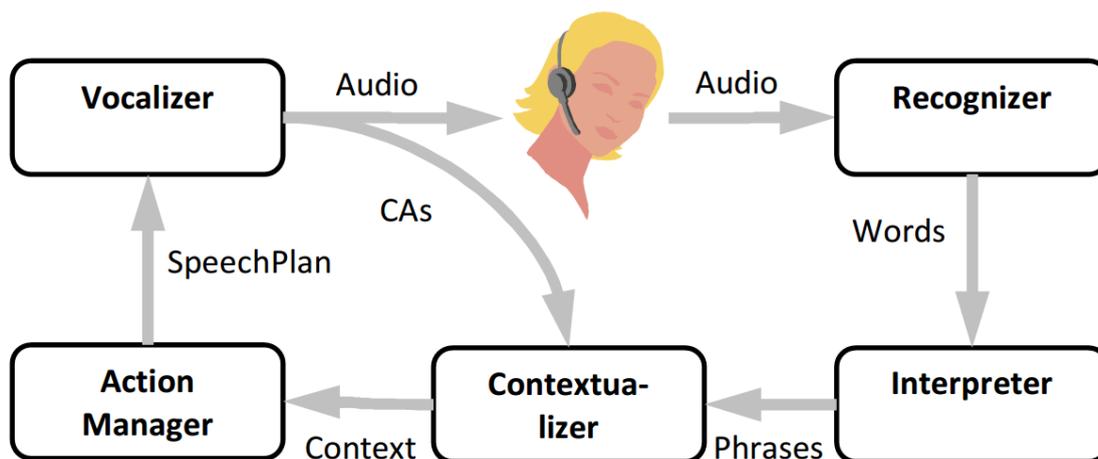


Figure 5.1.: The Jindigo layout as depicted by Skantze [Ska10, p.3]

So what does Jindigo do? Jindigo is a framework developed in Java, making it platform independent. Its creator Skantze [Ska10] describes it as incremental instead of turn based, which is closer to the way humans talk and enables the system to adapt to long sentences with the topic changing.

To achieve this, Jindigo uses multiple threads so all modules of the speech processing can work simultaneously. In processing, Jindigo utilizes a *CMU Sphinx 4 recognizer* [Lam+03] adapted to act as a Jindigo module for ASR. ASR output is transformed into concepts by an *interpreter* module, based on rules specified in a grammar XML file (as seen in appendix section A.1.3).

A *contextualizer* module enables a reinterpretation of concepts depending on discourse history (specifically what questions were asked). These concepts are given to an *action manager* module,

¹It seems justified, as the time spend analyzing and debugging Jindigo for this work is not negligible.

where the dialog management takes place. This is where most work is done when developing a spoken system with Jindigo.

The created response is then passed on to the *vocalizer* module where the audio output is generated, utilizing *MaryTTS* [ST03] for *TTS*. The dependencies can be seen in figure 5.1.

To be able to use programs based on Jindigo, a *MaryTTS* version 4.1 server (available from Lab and Phonetics at Saarland University [LP]) needs to be running. Trying to run the example programs from the Jindigo website does not run out of the box though. This is because in the Jindigo libraries the output voice “dfki-prudence-hsmm” is defined and *MaryTTS* does not necessarily come with it pre-installed. Thus the voice needs to be installed via the Mary component installer packaged in the Mary Download. If using Jindigo from source instead of the library, this voice can also be changed² arbitrarily to any other voice installable with *MaryTTS*.

After these adjustments are made, the example programs provided on the Jindigo website should be working.

5.2. Natural Language Understanding and Generation

When working with the Jindigo framework, most of *NLU* and *NLG* are already taken care of. Most of the work left to do is to design what words and which phrases are to be used for responding to and understanding utterances.

To handle the dialog, language needs to be reduced to concepts of the topic, on which the system can operate. These topics need to be defined. This can either be done in a special hierarchical XML file listing the concepts and their structure (\rightarrow child concepts), or via a Java class.

This system deployed the Java class approach, in which many inner stub classes inheriting from the Jindigo class *Concept* are defined. For concepts being used as frames with the purpose of being filled with information, the concept class should not extend *Concept* directly, but extend the subclass *Form* instead. Doing this in Java allows to define methods for concepts as well, rendering more complex tasks possible.

So what are concepts when they are mostly empty classes? Concepts are not necessarily empty, arguments and values can be defined for them. Action management can decide on actions based on which subclass of *Concept* is handed there and then work on the included data. As an example from this work, an appointment can be a concept class, having as argument other concepts classes representing time, date and appointment type which in turn save the specific values either directly (type only stores a string with the type name) or further down the argument tree (time is constituted of hours and minutes).

After the known concepts are defined, a Jindigo application needs three different kinds of XML files.

The first is used for system *configurations* and needs to be called to configure a newly created Jindigo system (presumably in the main method) by defining the modules and their communication channels and speech grammars. It can mostly be taken from the example application. Required changes are the registration of the systems dialog management class as a module and specification of the path to two other XML files: the in- and output grammars.

The input grammar is comprised of two parts. In the lexicon words the system should be able to recognize have to be defined. The defined words can then be grouped and assigned labels to make them referenceable in grammar rules. If applicable, a concept with or without values can be assigned to these groups as well³.

These concepts can then be used in the rules that are the other part of the input grammar. In the rules, text and lexicon entries as well as other rules can be combined to utterance structures the

²In `net.jindigo.module.vocalizer.MaryVocalizer.java`

³See appendix A.1.2 for an example.

system will be able to handle. Again, concepts can---and for the most part should---be assigned here. The concepts created through matching an input to a rule serve as the input the dialog manager receives and acts upon.

Output is also done by the means of a grammar XML file. Rules for the output sentences are defined similar to the input rules---through combining text, rules and lexicon entries.

Concepts can also be assigned here, although with a slightly different purpose. To output dynamic answers depending on the dialog management, concepts can be handed as rule input. These concepts must have additional functionality though. First, a verbalizer class implementing the `Verbalizer` interface must be created. In it the way to output the concept is defined in the `verbalize()` method⁴. Then the class of the output concept needs to implement the interface `Verbalizable`, requiring it to override the method `verbalize()`. Within this method it should call the verbalizer class's `verbalize()` parsing itself as input⁵.

If a certain answer is to be requested from the user to be put into the right context, a request concept can be defined as output, specifying the requested concept as answer parameter.

In terms of the system layout seen in section 1.2, the input grammar as semantic grammar acts as the developer's access point to *NLU*, while the output grammar coupled with verbalizable concepts is the means Jindigo provides for *NLG* control. Except for these grammar definitions, a developer does not have to concern himself with the how-to of *NLU* and *NLG*, as Jindigo is taking it off his hands.

5.3. Dialog Management

For implementing the dialog management, Jindigo provides an “action manager” class, which already does some preprocessing of the concepts given by the Interpreter (e.g. unpacking, resolving context) and can be subclassed to override the methods responsible for handling input and/or responding. Other modules could also be designed and registered in the XML file with the system configurations and set up for inter-thread communication for this, but this was not deemed to have any real merits over the other approach.

5.3.1. State Handling

For this work, the module class `ActionManager` of the Jindigo framework was extended for the implementation of the dialog management.

Depending on the current state/context of the dialog, the system has to handle the same input differently. For example a simple “yes” affects the system flow in various ways, depending on which system question precedes it⁶. For this reason the decision was taken to define logically separated dialog states. Through implementing a respond method, the current state could then decide on the course of action more context sensitive than through the context provided by Jindigo.

As a further benefit, the resulting code is better structured and the representation is clearer. The `ActionManager` subclass was reduced to the functionality of a state handler (and called: `StateHandler`). On resume of a state, the past conversation history is available again, but on switching to another state a fresh context is guaranteed to avoid carrying over unrelated information. Conversation history comes in the form of linking the current concept to past concepts of the same type to permit information given in one turn to still be available in the next. It should be noted, that the existence of a relevant context is only applicable for store, retrieve and

⁴A working approach is to define `toString()` methods for the concepts and then wrap them into the `verbalize` datastructure.

⁵Actually, another way to verbalize concepts is defined in Jindigo. But as it does not even work in the provided travel example application due to a grammar not being assigned correctly, it will not be explained here.

⁶Some examples include: the termination of the system, an app being stored, a dialog for storing a new app started.

persuasion functions, as the others have no state variables, but are “static states”⁷.

However, as illustrated in section 4.1, some commands should be made available from anywhere in the program flow. This induces the creation of an abstract superclass for all states (`AbstractState`), ensuring a uniform interface for state manipulation by the state handler and avoiding redundant code. Thus, all system-wide utterances are handled within the `respond` method of `AbstractState`, enabling the subclasses to access them per `super.respond()`.

Through the defined public methods there, the handler is able to register itself as handler for the states (which is necessary if the states are to request a change of state). They provide a state entry message played as a feedback to the user about the current system state after a switch. They also enable the handler to retrieve the last concept to which the user responded as context for storage on a state change.

There are also methods access the states as singletons. With this the handler is able to ensure the existence of only one instance of each state in his next states list. This is done on the assumption that returning to an “older” context would confuse the user more than benefit, because after a user requests a new date storage, he most likely lost interest in (or even forgot) a previously attempt to store plans.

State changes are generally initiated by the states themselves, after reaching a changing point in the dialog or encountering a state change generating system-wide command. They can request the handler to suspend them and activate a requested state, or they can just finish their own execution and yield, causing the handler to switch to the next state in the list. Some states will request the state handler to run a new instance of themselves if they want to process a list of concepts (e.g. all found appointments after a search or all past plans not discussed yet) recursively. Calling them-self with a new, partly copied concept is also the only way to reset a previously given argument without loosing any other information. Recursive calls lead to the removal of the calling state from the waiting states, too.

5.4. Scheduler Functionality

The scheduling functionality is represented by two state classes, one for the date storage (`StorageState`) and one for date look up (`RetrievalState`). This is to enable a user to suspend storing a date to look up whether or not he is free on a given date and then return to storing, with all previously given information still intact.

Both states work on an appointment frame with the values *begin*, *DateTime* and *type* (`AppType`), a concept where the value is a string representing the type the user specifies. The difference rather lies in the way the frame is handled. Through conditions on what information is already in the frame a `StorageState` instance will prompt a user for remaining information until its frame is filled completely. For the `RetrievalState` on the other hand either date or type information are sufficient to execute a search, while the time of day is completely ignored⁸. Only one prompt is issued to retrieve this data.

For permanent storing, the system maintains two data sets of appointments. In both the appointments are sorted with their *DateTime* as their sorting key⁹. Newly stored appointments are entered into a data set used for active appointments.

On each system start the program checks for scheduled entries that have crossed over into the past since the last visit. These are handled as described in section 4.1. After speaking with the user about the scheduled plan(s), the appointment is transferred to the other list (the list for past

⁷This is not completely true, as the exit state has a variable for confirming the exit request. If it is set the system is either shut down or the user returns to the previous dialog, rendering a save unnecessary.

⁸Because it was deemed unlikely that a user would have an interest in all appointments he has on a certain time of day regardless of on what days they are.

⁹Which enables a check whether the next date in the active appointment list has passed since the last system use to be completed in $\mathcal{O}(1)$

appointments). As of the current system version, this data is never accessed to be retrieved and processed again. Yet it can be used for debugging and it has the potential to implement persuasive strategies based even more strongly on user behavior in future development (see chapter 7).

5.4.1. Grammar Design

For such a superficially limited domain as talking about dates, creating a robust recognition grammar is quite complex. Schillo [Sch96] was surprised by “the variety in the time and date category: there are 97 different ways of combining dates and times and this after numbers, weekdays and ordinals have been replaced by tags.”

As mentioned earlier, the system introduced here was not designed to become a large scale calendar application, so it was never attempted to create quite as much diversity. Still, the lion's share of all defined grammar rules consists of scheduling rules. Many rules are required for number recognition alone, because of the many available combinations. Especially when paired with ordinal numbers as when talking about the day of the month.

The selection of rules to create was initially derived merely from the authors mind and afterward supplemented by means of the corpora of the wizard of oz experiment by Schillo [Sch96] and skipping¹⁰ through the *verbmobil* text corpora on scheduling dates. [HKT95] Later they were adapted further according to the results of the scheduler's preliminary user evaluation summarized in section 5.4.2.2.

The definitions for types of appointments proved to be rather futile, so only a few example types were created. The reasons are limitations of Jindigo (which are discussed in more detail in section 5.6). Namely the impossibility to directly access what a user says to store it as a type, because every utterance is mapped to words named in the grammars. This could be circumvented with (the tedious work of) entering many different types manually. But this too is limited by Jindigo due to the enormous RAM cost of every single rule in the grammars.

5.4.2. Preliminary User Evaluation of the Scheduler

In order to get an early feedback to the general user acceptance of the program, a small scope user evaluation was held with an interface-only version of the program's scheduler component. This provided an additional benefit of first practical experiences in conducting user studies, in preparation for the final evaluation that took place after completing the implementation. The information retrieval and storage dialogs were tested with the goals of checking the coverage of the designed grammar the system output acceptability and the general user satisfaction.

5.4.2.1. Scenario and Realization

Held was the study with five international members from three different countries of the *Augmented Human Communication Laboratory* at the *Nara Institute of Science and Technology*, Japan. The program being in English, it was also the language of the study, yet none of the participants was a native English speaker.

To avoid resulting problems with the different speaker's accents and also because of the high level of background noises in a room full of people, system input was given in written form instead of the system's normal spoken input. This in turn resulted in the problems described in section 5.6. It also seemed to influence the subjects choice of words.

In a first pretest of the evaluation, it quickly showed that the vocabulary of the instructions, especially written ones, strongly set the wording used by the participant. So to avoid undue influence on the way requests were phrased, the decision was made to give the user no explicitly written instructions¹¹.

¹⁰Judging from the size, a thorough exploration of used expressions might as well take multiple weeks.

¹¹In a first test run of the set up, the participant used the exact wording on the task description.

This was achieved by formulating the user task verbally and giving information the participant needed to reference in form of a graphical representation¹². This representation consisted of a calendar with a marked date and the icon of a clock set to a specific time, as can be seen in 5.2 The instructions were a freely spoken version of:

You need to go to the dentist. You already asked for an appointment and he gave you three possible dates. Now you have to check in your schedule on which of them you have no plans. Once you found when you are free, please try to store the date in you calendar. Whenever you have trouble, you can ask the system for help.

If necessary parts of these instructions were repeated during the execution of the task, especially the hint at the availability of a help function.

During the evaluation execution the conductress sat beside the subject, taking notes and trying to limit the number of erroneous inputs due to spelling mistakes. In the meanwhile a logger recorded quantitative measures of the program run. To get a qualitative assessment, subjects were asked for comments and to give a score (ranging from 1–bad to 5–good) on how much they liked the system.

5.4.2.2. Results

On average, the five test participants took eleven minutes to complete the set tasks each, with an average of 31.2 user utterances (For an overview, refer to table 5.3). With giving the system only one information per utterance, the task could be achieved with merely ten user utterances. Considering the possibility of giving multiple information at once, the absolute number of necessary utterances is reduced to four. This means that a dialog without any misrecognition and instances of wrong input (such as misspelling and not supported commands) should lie between four and ten utterances.

Considering this, over thirty utterances can in no way be interpreted as a good result. The high percentage of utterances stemmed from multiple sources, pictured in figure 5.4. The most important one had neither to do with the system nor the participants, but the conductress. Without prior evaluation experience, the task description given turned out to be unclear and confusing. This way users performed or repeated actions unnecessarily, instead of working only at the task at hand.

Culling all instances of repeated input, the average user utterances drop to 21.8 per user. Of these three quarters (15.8) were recognized and processed correctly. Other problems were related to the input method. The average user misspelled five utterances.

On top of that Jindigo shows a strange barge-in behavior when a typer is used as input. It theoretically allows for it, but when the user types before the system message is completely played, the output is halted shortly, then the same output is given again¹³. Users were told to let the system finish before typing, so this did not happen often (0.4 times per user). Users also tried to issue commands the system generally does not support. To this the system has of course no useful answer, accounting for an average of four more utterances being handled wrong.



Figure 5.2.: The user task in the evaluation scenario. Stick-figure pictures taken from Munroe [Mun].

¹²An idea taken from Schillo [Sch96].

¹³When using the recognizer barge-in usually is enabled, but rarely the system still reacts as described for no apparent reason.

Evaluation Criteria	Average Value
Usability score [1-5]	3.7
# User utterances	31.2
Dialog duration	11m 3s
# Help calls	2
Correct system answers	55.13%
Corr. ans. to valid input	77.48%

Figure 5.3.: An overview of the quantitative results.

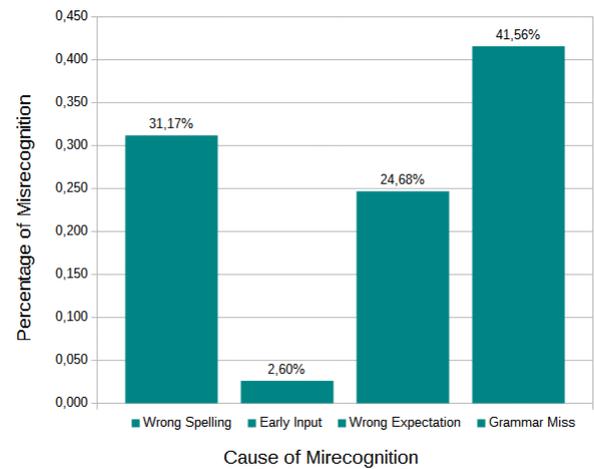


Figure 5.4.: Reason utterance misrecognition in percent.

All the above mentioned were noted and conclusions for future studies drawn, but of real interest were mostly the grammar misses, which means things the system should have recognized but did not. From these, the system could be improved directly through complementing the grammar. An average of 6.4 misrecognized utterances stemmed entirely from grammar shortcomings.

The overall task completion rate was 80%, although again uncompleted tasks were mostly caused by imprecise communication of the tasks at hand. What helped keeping the completion from going even lower, was the help function, which was called an average two times per user and proving essential for good task completion.

Despite all its shortcomings (and reaching an average score of 3.7), the system was still acceptable to the users.

As a consequence of the explained findings, the grammar was expanded, and notes for improvements of the scenario setup were taken for future use.

5.5. Persuasion for Exercising

To devise a strategy, pedagogical sources were examined. This stemmed from the idea that the field of education is the largest scale real life application. Arguably, there is no other situation where so many people in the world are to be convinced to work motivated and improve themselves on their own accord¹⁴.

The result of these studies postulate that motivation should aim to increase *coherence*, [Mül13] the feeling to gain something, *self-determination*¹⁵, the notion of being in charge of oneself, *self-efficacy*, [Ban97] the knowledge of one's personal abilities and *volition*, [Pel13] the commitment to see plans through.

For this system, the following conclusions were drawn:

1. Motivational phrases said to the user should bring the user's attention to his own benefits regularly.
2. The user should always feel in control, the system is not supposed to act too patronizing or penetrant.
3. The user's own ability to reach his goals should be strengthened through positive feedback.

¹⁴Whether or not this is achieved at all times is a different matter.

¹⁵"Resistance sometimes arises when people perceive that their freedom of choice is being threatened; they tend to react by asserting their liberty." [And12]

4. The system should check whether the user sticks to his plans and support him on making up for past negligences.

Previous research (e.g. Mazzotta, Rosis, and Carofiglio [MRC07]) has established positive and emotional strategies to be perceived more effective in persuading. Positive should be self-explanatory, emotional means appealing to feelings rather than sticking to facts. The emotional “Just imagine how great you will feel after you achieved ...” is more likely to pluck a string on the conversational partner as stating rationally: “Studies show doing ... improves the health condition ... by X percent.”. Yet both these sentences work better than talking about negative effects on unwanted behavior. So phrases motivating the user should aim for a positive and personal tone.

From the above, the principle on which the persuasion aspect is based is positive reinforcement. The user shall have the impression of the system empowering him to reach his goal.

It shall assist him to achieve his long term goals, not to define goals for him. If the program is too annoying and insisting, users might not use it. Therefore the aim is to be a friendly reminder, not a “pest”. “The most important aspect of incremental persuasion is that there are only suggestions, no obligations”, as can be read in work by Mathew [Mat05]. Unlike the scheduler, the motivational dialog is graph based. With each turn the user declines the proposals, the system adapts its behavior, suggesting less ambitious actions with each new step. A randomized order of this would not make sense, and opposed to the form filling used in the scheduler dialogs, there is no real information to be collected from the user, except his answer to the program's attempts to convince him.

This builds the basis for the systems behavior (as described in section 4.3) of checking with the user past appointments, not reprimanding the user for failed ambitions, but reassuring him of his ability to do better in the future and for the praise given for accomplished goals.

With all this in mind, the task was set to find sentences applying to those findings. For this, another study was conducted.

5.5.1. Motivational Sentence Study

Searching supportive health communities on the Internet, a base of sentences was collected. These were assessed to remove copies, trivial phrases (“Well done”) and correct the orthography. Afterward they were classified into three categories; *praise* for positive behavior, *reprove* for failure to enact the targeted behavior, and *motivation* as a general approach to lift the spirit and stay on track with one's goals. As a result, 13 sentences of each of the three types were available to be used in the planned persuasive study.

Next on the agenda was the design of the situation, in which the sentences should be used. For praising phrases this was easy, they are in and off themselves of a motivating nature and of course should be used as feedback on carrying out the desired behavior. As mentioned above, negative statements do not have high persuasive power themselves, so the reprovng sentences were coupled with the motivational sentences. Being general statements aimed at keeping a positive outlook and goals in mind, the motivational sentences were deemed to be usable in all situations. Coupling reprove with motivation was done as not commenting negative behavior at all seemed odd and so the motivation was hoped to smooth over the sting from being criticized.

For the questionnaire, five different scenarios were designed. In the first, third and fifth scenario the user stated he did not do his exercise or that he did not want to do any in the future, to which the system with reprove followed directly with a motivational sentence. The second scenario was the only scenario where the user responded that he had done his exercise, and is rewarded by appraisal. In the fourth scenario the user just could not find any time for exercise, so he neither met reprove nor praise. The system just gave him a motivational phrase to raise his willingness to do something. Then, in all five scenarios a resulting persuasive suggestion was added in the end.

In the questionnaire five answer phrases or phrase combinations were picked randomly from

the corresponding categories. For each question/scenario the evaluation subjects were asked to give all five system output options a score from 1 to 5, 5 being really motivating, 1 being not motivating at all. This was done in order to get both a ranking amongst the sentences and a score for each individual phrase. With the ranking, only a number of the best ranking phrases could be picked for future use. The scoring served the purpose to get an additional idea of the absolute quality of the phrases.

The final questionnaire consisted of 25 questions in total, repeating the scenarios with combined system responses seven times and the ones with a single response two times each. This imbalance was set to compensate for the many more possible combinations when two phrases were given. In total five questionnaires with different system answers were distributed out to 22 people for evaluation, to achieve greater coverage.

5.5.2. Results and Impact on the Implementation

A total of twelve filled questionnaires was returned. For all five send out question variations, at least one exemplar was included.

The most important finding of the study concerns the reproving phrases. In all three situations including reprimands (scenario 1, 3 & 5) a clear tendency for worse results as can be seen in figure 5.5.

“Fantastic job! Keep up the good work!” (Praise) and “You might not be where you want to be right now, but you are getting there.” (Motivation) were rated the most motivating phrases.

“Way to show perseverance!”, “What an incredible accomplishment!” and “Doesn't it feel good to stay motivated to succeed?” had the lowest scores for praise. “Discipline is simply the ability to discern between what we want most, and what we want right now.”, “Whether you think you can or whether you think you can't, you're right!” and “But remember, nothing beats a failure but a try!” were rated the least motivating. As can also be seen in figure 5.5, generally the acceptance of the supportive phrases was good. With the average score in the questions including praise (scenario 2) being 3.34 out of 5 and the motivation giving questions (scenario 4) with an average score of 3.2. For a list of all phrases and their resulting scores, see appendix B.

Through comments received together with the answers, participants complained multiple times over the utterances being too long and preferring shorter once. Two people also stated how they generally preferred specific suggestions over general statements like “You know this is bad for your health!”. This is in accord with the findings of the theoretical research done previously.

Combining the bad acceptance of the system showing reprove and the preference for shorter output, the foremost idea for improvement was to abolish the negative sentences completely. Like this the system should respond positively to kept plans, while missed exercises only elicit a focus on the future, delivering a motivating phrase combined with a suggestion to make up for the occurred slip.

As a further measure, a probability depending on their ranking was applied to all phrases. Utilizing this probability, the system shall give out highly motivating phrases more often. Phrases with lower scores, except the three lowest, are kept and frequently delivered as well, to preserve a degree of variance in the responses.

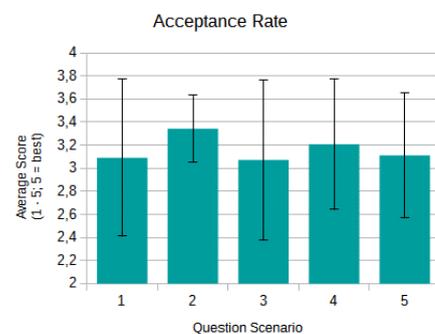


Figure 5.5.: Average scores for the different scenarios, showing a correlation between reproof and lower scores.

5.6. Encountered Framework Limitations

During various stages, implementation was felt to be hindered by Jindigo features, lack thereof, or, in one case, a system bug.

Finding the source and exploring ways to circumvent the limits where possible was very time-consuming. This is not too surprising though, as the website itself warns about Jindigo being in an early stage of development and, more recently, the complete discontinuation of it. [Ska] For that reason, this section is *not* meant to air complaints about things not going the authors way, but to state finding and maybe account for one or another suboptimal implementation choice or limitation of this health system.

Earlier in section 5.1 the problem with Jindigo not running out of the box and the solution found for it were already discussed. Further problems stemmed from the same source, the discontinuation of development. It left some features only partly implemented and the API hardly commented at all. Jindigo was found to be unfeasible to use without studying the source code and the examples thoroughly.

Generally error handling in Jindigo is very non-descriptive. Many errors are defined to be thrown throughout the code, yet most of them lacking an error message or description. Yet in most places errors were thrown, the code was merely surrounded by an empty catch block printing the stack trace, with the program continuing execution unperturbed. Some thrown errors were even found to limit the usage of concepts for conditional program flow based on set concepts.

Also once an argument in a concept is set, it could still be changed, but neither removed nor set to null. This was hindering grounding implementation, as the logical value for confirming whether an utterance was correct. Once set to false, it would repeat the same system prompt requesting the user to correct the data endlessly. As workaround a new concept needs to be created, and all values with the exception of the cumbersome one have to be copied manually to this new concept. This new concept could then be used for processing in stead of the original one.

Another feature that would have been useful for context sensitive grounding was the only partly implemented confidence scores for the recognition. A field for it exists in the class for communicative acts and a value is set. All the same, examining its value determined that it was always 1.0 for the interpretation which was chosen as best hypotheses, rendering the score useless.

Looking at the provided examples to try to understand how a feature works, sometimes was not very helpful either. Trying to incorporate feedback of acquired information through verbalizing it, the verbalizing in the provided examples did work only partly. Previously, it was already explained in section 5.2 how verbalizing can be implemented using an alternate method. Yet even with this a curious problem remained: When the output the verbalized concept ended on a number (an Integer, more specifically) and no further text was added after it in the rule, the output stopped at the beginning of the verbalized concept. No logical reason could be found for this. In the end all such response rules had to be rephrased in a way ensuring text as the last element (sometimes producing a bit unusual sentences in the process).

A major field for inconvenience was the way grammars act. For example, in XML files the corresponding Java class of the domain is referenced. With this class information, all concepts in the file could be instantiated. It is possible for one grammar to include another. When working with the built-in grammars defined in the Jindigo framework (e.g. the `number_grammar`), the domains could still be resolved without an issue. Was the same applied to a self-created grammar, somehow the domain was not imported together with the grammar file and as resulted in an inability to create the corresponding concepts. This is not a major problem, but resulted in all concepts having to be defined in the same class, resulting in an unsightly huge class.

A lot more severe was the lack of a (known) way to deactivate currently unused grammars. On first glance this may seem as a minor issue. But it resulted in a rather severe limitation in the grammars used. The cause is that each rule the Sphinx module has to handle seems to come with

a major cost of memory space. So grammars with good coverage are likely to produce an out of memory error on system start and render the used machine very unresponsive until the program finally terminates.

As a result grammars have to be limited in size, making it more difficult to reach good coverage of diverse utterances. Combine this with an apparent ASR policy to match all utterances with the grammar by hook or by crook, and it makes for interesting recognitions. It matches every sound with the best fitting lexicon entry if it is only loud enough, even if the “best fitting” entry is not fitting at all and confidence should be low. On tests even a grammar with only one word was constantly “recognizing” this one word on every audio input.

With these effects combined, it is unclear whether a larger or a smaller grammar are beneficial for recognition. With a large grammar, there will potentially be more words with similar sounds, might causing more misclassifications. Whereas in a small grammar there is less confusion between the defined words, but relatively more words will be linked to a concept. In addition, if random noise or unknown utterances are matched to concepts¹⁶, the dialog manager reacts to those inputs and the system is more likely to change states erratically.

Making the creation of grammars even more challenging is the apparent lack of wild cards to be added to a rule. When the speech recognition is used, it also matches not so closely related phrases, resulting in a hybrid of rule based and keyword spotting recognition. This actually lessens the effect of missing wild cards. Still, when switching to written input, this makes for very low recognition rates. Jindigo offering the possibility to easily switch to written input is great for testing, as the program loading time is reduced to nearly instantaneous. Also testing can be done without disturbing everyone working in the vicinity.

The drawback lies in the fact that the written input has to match exactly what the recognizer would have returned, not what the user said. All meaningless filling words need to be defined as optionally occurring (bloating the definitions files and slowing down the startup in speech recognition mode even further). This in turn limits the recognition rate significantly compared to speech input. (This is because the Sphinx module matches any input not exactly defined by the grammar to the closest sounding one, thus in practice rendering filling words (e.g. at, the, my,...) and similar-sounding expressions (plan vs. plans) irrelevant to grammar definition.)

Furthermore, all written input needs to be lower case, orthographically correct and with numbers written out. Due to environmental circumstances¹⁷, written input had to be used during the conducted user studies, making the execution and the resulting analysis tedious and ambiguous.

Another comparatively minor issue with grammar entries is the bad handling of one grammar entry being substring to other grammar entry. For example, defining both “i have an appointment on...” and “do i have an appointment on...” causes only the one being the suffix to be recognized. This doesn't sound like a severe issue but it can cause concept misrecognition--in this case an appointment lookup is misinterpreted as a request for storing a new appointment date. As this is a systematic misinterpretation, this has the potential to severely negatively influence the user perception of any program built based on Jindigo.

¹⁶Laughing sounds were constantly recognized as “pause”.

¹⁷Usually taking place in a room full of people, partly with speakers with strong accents, partly with bad quality of the available microphone.

6. User Evaluation

A limited case study was conducted after implementation to verify the acceptability and identify specific lacks of the program. For this purpose, a catalog of questions was developed from previous works with adjustments to the underlying scenario. Eventually, advantages and disadvantages were evaluated and suggestions on further improvements devised.

6.1. Evaluation Targets

As human dialog is diverse, different agents can cope with this free speech only in a varying degree. So there is a set of common descriptors aiming to measure the common goals, which are speech naturalness and flexibility on one side, robustness and good coverage of the input on the other.

General quality of the system needs to be evaluated as is discussed in literature like [JM09, ch.6]. One of the first things to consider is what our evaluation targets are to decide which data should be evaluated [GMM00, p.123]

Namely, we want to obtain reliable information about the *usability*, *efficiency* and *effectiveness*. With *usability* we are talking about the quality of a system from a user's point of view. Did he like it, was it easy to use, did he know what to do? To obtain these knowledge, the common approach is to just ask the user. After running a test scenario with the program, a questionnaire is handed to the user to express his opinion about the system. See the section 6.2 for the design principles and details of the questionnaire used in this work. For *efficiency*, the ease of use, we do not have to rely on a questionnaire as much, because many objective measures can be obtained in an automatized fashion. In our study we logged data to explore the efficiency on the basis of the following measures: dialog length, number of system and user utterances, number of concepts recognized, calculated from these two the average concepts per user utterance [Eng12], the error rate (not or misunderstood utterances), and the number of help function calls. *Effectiveness* then considers the result of the interaction. This is expressed through the task and subtask completion rates and the concept accuracy, the rate of correctly recognized concepts [McT04].

Additionally to these general measures, of higher interest to this specific application is the user acceptance and interest in the persuasive component. How does the persuasion of this system fare? Would users appreciate this kind of motivating agent in their everyday lives? Are there system-specific negative properties preventing higher persuasion success? As the system was developed combining a scheduler with the added persuasion routines, both parts should be looked upon. Thus the evaluation scenario was divided into two parts. The questionnaire was designed accordingly to ask questions concerning both components, then subsequent requesting opinions to the overall impressions while using the system. For the user to be able to judge the two as-

pects independently, he was asked to pause the execution after the first of part was completed and answer the specific questions. This way, his memory would still be fresh and his impression of one part not influenced by the other.

6.2. Questionnaire

Gathering data for judging the usability of the system was done through a questionnaire given to users during the execution of the study. Please see C.0.6 for the complete question sheet.

For the questionnaire, the open-source tool LimeSurvey [TS12] was used, as it offers customizable design and statistical preprocessing of given answers, including diagrams.

First of all, to account for the influence of user related factors, a certain amount of demographic information is collected. Age and gender are assessed as standard practice, while the participant's major of studies and previous experiences with similar systems is requested to process whether familiarity (stemming from usage or even research experiences) with dialog systems changes user satisfaction and task performance success. This is deemed useful as evaluation subjects are from the authors professional and personal environment, thus being over-representative of computer scientists in general, and researchers in related fields in particular.

[]General Quality: How much do you agree with the following statements? *

Please choose the appropriate response for each item:

	1	2	3	4	5
The system is accurate	<input type="radio"/>				
The system is useful	<input type="radio"/>				
I felt confident using the system	<input type="radio"/>				
The interaction with the system is not repetitive	<input type="radio"/>				
I always knew what to say to the system	<input type="radio"/>				
The interaction with the system is fast	<input type="radio"/>				

(1 = don't agree at all; 5 = completely agree)

Figure 6.1.: The quality assessment questions based on SASSI [HG00].

As already mentioned, the evaluation was divided into the persuasion and the scheduling part. Both had the questions assessing general system quality in common. The persuasion part however was extended by additional questions specific to the persuasion goals. All questions concerning the quality were implemented in a 5-point Likert scale in a manner ensuring 5 being the best and 1 the worst score¹ as seen in figure 6.1. The common approach when questioning quality was to first request a general score, followed by scores for different quality parameters. These were taken from SASSI [HG00] by selecting a sentence with high relevance score from each quality categories described². Only one question for each category was asked to keep the questionnaire short, so answers would not be affected by a sinking mood of a user working his way through pages of questions. For the persuasion two additional types of questions were asked: Questions to measure the acceptance of the persuasion, both as a general idea and as seen in the system, and opposite statements checking whether the user found a system trying to persuade him uncomfortable or the generated persuasive attempts unfitting (see figure 6.2).

After the two component parts the user should think back again to the whole performance and give an overall system feedback, consisting of another general score and two questions about the

¹with one exception consisting of three questions about persuasion, because no suitable substitutes to formulate the same statement in a positive way was found. Analyzing the results, the values of this questions were subtracted from the value 6, reverting the scale to match the others.

²These quality categories are: Response Accuracy, Likeability, Cognitive Demand, Annoyance, Habitability and Speed.

[] Persuasion Quality: How much do you agree with the following statements? *

Please choose the appropriate response for each item:

	1	2	3	4	5
The system suggestions were helpful	<input type="radio"/>				
Using the system was motivating	<input type="radio"/>				
I like the general idea of a motivational system for my personal use	<input type="radio"/>				

(1 = don't agree at all; 5 = completely agree)

[] Persuasion Flaws: How much do you agree with the following statements? *

Please choose the appropriate response for each item:

	1	2	3	4	5
The phrases to motivate me felt out of place or unfitting	<input type="radio"/>				
The system's manner of speech was rude	<input type="radio"/>				
I felt patronized by the system	<input type="radio"/>				

(1 = don't agree at all; 5 = completely agree)

Figure 6.2.: The extra questions to evaluate the persuasion.

user's interest in using this kind of system. One question was specifically asking about the given version. The second aimed to get an opinion on the idea of a spoken health assistant with persuasive features in general regardless of faults the user might find in this specific implementation (shown in 6.3). In addition, a field for expected features the system did not offer provided. This way both input on future improvements and a possible correlation between missing features and a low score could be obtained. To enable a user to give differentiated feedback, a free text field for comments was added to the persuasive, the scheduling and the overall quality pages.

[] How much do you agree with the following statements? *

Please choose the appropriate response for each item:

	1	2	3	4	5
I would use this system in everyday life	<input type="radio"/>				
Generally a system of this kind could be useful	<input type="radio"/>				

(0 = don't agree at all; 5 = completely agree)

Figure 6.3.: Questions aimed at the difference between the given system and the concept of a persuasive health program detached from potential lacks in the presented program.

6.3. Scenario

As mentioned previously, users were asked to complete two scenarios aimed at the two main components of the system. Based on experiences made in the preliminary evaluation of section 5.4.2, instructions on these scenarios were given only in spoken form using German instead of English, to avoid influencing the user's wording of utterances. As the testing was done supervised, clarifications and reminders of them could be given again if necessary. This way, they developer could also behold people's immediate reaction to the system. Because of trouble with a unusual low quality signal from the microphone combined with a room full of people, the input had to be given in a written form (in 5.6 considerations of the effects of using written input is disclosed).

Having gained an impression of the different way users formulate written input from spoken one in 5.4.2, this time a twist of the Wizard of Oz method was deployed. Instead of substituting the system responses with human-written ones, the spoken input to the system was substituted with written input. Meaning the test subject addressed the study conductress, who in turn typed first the exact user utterance into the system for the sake of documentation and a shortened version

of what was assumed the system would have recognized per keyword spotting afterward. To this the system did formulate its response to the user. This way the user could still use speech as a medium, not limiting his wordings through typing.

The study was conducted starting with a request to enter the demographic information into the questionnaire. Next a spoken introduction to the system was given, informing the user about features and the system help, also dismantling unfitting expectations towards the features³. User's were also encouraged to try rephrasing sentences the system did not understand.

Then the instructions for the first task were given. The user was asked to imagine being interested in regular exercise and thus having scheduled training sessions. On system start, the training and physician sessions lying now in the past were read out to the user and handled as described in section 4.1. For the ensuing persuasive dialog, the user was encouraged to answer freely and see how the system reacted. When an appointment storage dialog was entered in response to the participant's actions, the storage was canceled to focus on the persuasion, as interaction with the scheduler was tested later. Once every feedback on past appointments and the ensuing persuasion attempts were finished the system entered a passive listening state. This was the signal for the user to fill in the questionnaire part related to system persuasion.

Before the second task of the study, an introduction to the calendar and the tasks at hand was given. This time there were two subtasks to complete. In the first scenario, the participant got a birthday invitation and was tasked to check his calendar to inform the inviting person whether he could participate. The task was accomplished when the information was retrieved correctly. For the second task, the wording was kept in a more open style, to see how the user interacted with the calendar on his own. Accordingly the user was told about having an urge go see a dentist. So he was requested to take care that in the end a dentist appointment was in his schedule. Not mentioned to him was a previously scheduled dentist session already in the calendar. The task was seen as fulfilled, when the user *knew* of an appointment being in his calendar, either through finding the existing one, or though storing a new one (optionally checking whether he was free beforehand)⁴.

Lastly, the remainder of the questionnaire concerning the scheduler and the overall system was filled in.

6.4. Results and Discussion

In total, nine participants executed the study. The statistical results can be regarded in appendix section C.0.8. All of them were able to complete the set tasks without difficulties. The average age of participants was very uniform, with a mean of 24.5 years. The male female ratio was split very evenly with four female and five male participants. Also evenly split was the extend of expert knowledge, with five participants involved in related research and four inexperienced users. The rate of people stating to use dialog systems regularly was comparatively low at one third. Most participants were computer scientists, with two exceptions: an electrical engineering and a biology student. This is statistically not significant enough to show any difference in the results, especially since every single person makes a noticeable difference with only nine persons in total. With the demographics, a correlation in answers was found for participants answering to experience in fields related to dialog and having knowledge of artificial intelligence. It can not be called surprising however, as the fields are connected. Another correlation seeming interesting on first glance was a correlation of female participants and low experience details. Looking beyond the information given in the questionnaire, non of the female participants were from the working environment of the author, while some of their male counterparts were.

³The system can not make suggestions for free dates in the calendar. Also, the calendar was not enabled to handle relative dates, they have to be stated explicitly.

⁴Afterward the user was encouraged to shut down the system, where a surprising number tried saying "shut up" for amusement. The program did react, but not by terminating itself. Instead it took the user literally and paused the system, thus *shutting up* instead of *exiting*.

Looking at the quantitative data collected during the study, it must be noted that regrettably it holds not much value. This is due to the unexpected switch to written input. As a result, only very few conclusion about the efficiency of the system can be drawn.

The problems with written input already illustrated in section 5.4.2.2 and 5.6 result in a large amount of “wrong misrecognitions”, as typing has to be very precise. This inflates measures as the average dialog length and number of utterances unduly. As only explicitly written statements are given to the system, participants would sometimes pause and ask questions or comment on things, again increasing the overall dialog time that was measured. Additionally, the incremental feature of Jindigo combined with written input is responsible to increasing the number of recognized concepts, as it will start recognize once a word is finished typing, multiplying the amount of concepts recognized if the user gave more than one information in one utterance. Considering all the above, the only quantitative measure with any merit is the number of help function calls. In total, one third of the users requested the system help. Compared to the previous evaluation, this was greatly reduced. This may be due to the input setup, as in the previous evaluation users had to enter recognized phrases exactly, while in this study the conductress entered a system readable command when she felt sure the utterance included the keywords the system would recognize. As hardly any conclusions can be drawn about effectiveness, another study needs to be conducted if reliable information is desired.

Effectiveness can be accessed on a better basis. All users were able to complete the tasks and subtasks given, allowing for a completion rate of 100%. The concept accuracy however can not be evaluated, as the written input maps keywords directly to concepts so that each correctly spelled input is automatically a concept, if the user utterance was not completely out of context (e.g. responding “yes” when no question has been asked).

Under these circumstances, turning to the usability results is the most fruitful. Of course, the opinion of the system was likely affected by the lack of speech input, too. Nonetheless data on many different aspects was acquired and can be analyzed. An example of typical dialog produced during the study can be found in appendix section C.0.7. The dialog stems from one of the participants, however the log has been cleaned of duplicates and the incremental word for word processing of the system to increase readability. For the complete statistical results given through the survey tool, please refer to appendix section C.0.8.

Let's first regard the individual scores. The scores given for the components was 3.11 for the persuasion, 3.33 for the scheduler and the overall score was 3.11, being identical to the persuasion score. The average score calculated from the *SASSI* questions was generally a bit higher: The persuasion got 3.26 and the scheduler rated at 3.46. A speculation is that the inconveniences stemming from the written input were not represented in the detailed quality questions, resulting in less reduction of the scores because of it.

Looking at the difference in scores for the scheduler and the persuasion, the scheduler had a better acceptance rate. The persuasion specific scores combined however were a high 3.87. If this is combined with the normal score, the total surpasses the scheduler component's rate. Thus it can be hypothesized that the interaction with the persuasive component was very limited and thus did not fare well as a general system, but the main part, the persuasive capabilities, were rated high (“Using the system was motivating.” on its own received 3.78). This can be considered a good result, as the prototype was only to test persuasion success.

The general idea of a persuasive health adviser when removed from the specific implementation got a score as high as 4.33, showing the high interest in the subject. The average value users gave when asked whether they would use this system in everyday life showed little satisfaction with a low average of only 3. However, this is not surprising as the system was designed as a prototype and the limits in functionality are known. Participants who had first hand knowledge about related fields generally tended to give lower scores (Persuasion: 3; Scheduler: 3.2; Overall: 3), presumably because they knew where limitations were likely found. Users regularly using

dialog systems gave the lowest scores for the scheduler component (2.67), but tended to rate the overall score higher (3.33). A possible explanation lies in the comparison. In contrast to a commercial spoken interface for smartphones, the provided scheduler was indeed modest. Yet regular users of dialog interfaces are likely to have a better opinion of dialog systems in general. An interesting aspect was the perceived correlation between the overall score and the persuasion-only score. In figure 6.4, the individual scores are portrayed separately for different demographics. It can be seen that the score for persuasion and the complete system are identical for the answers of all participants, and if separated into the genders. The scores with only computer scientists are still close together, but not the same.

A possible explanation could derive from the fact that the persuasion part is unusual and received good ratings for the persuasion itself, resulting in a disproportional impact to the total impression users had of the system.

Another interesting finding regarding the persuasion is that user voted it as not being rude or overbearing. Comments given even included “system is not pushing enough”. This is contrary to what was hypothesized, allowing thoughts about a more forward persuasion strategy. It could be due to the limit interaction though, because on a short test run it is difficult to annoy the user fast enough for this to have a major impact on the respective scores. In a longer usage study, it should be examined whether a more aggressive behavior would become annoying on a longer usage period. An other possibility is that users opting to use a program of this kind might prefer a stronger approach. This could be accommodated by a setting to enable users to choose between suggestive and assertive modes, depending on their preferences or circumstances⁵.

A large amount of user feedback was given in the form of comments. Most comments were addressing the quality (or lack thereof) of the *TTS* voice. It was described as chippy, difficult to understand and with a to unsteady pitch⁶. For future development, the integration into an other framework seems to be an important step to improve system quality.

The next most common commentary was about missing features of the calendar⁷.

So summarize, users missed features concerning the calendar and did not like the voice of the system. System components got moderate scores, still acceptable but with much room for future improvement. The general interest in the persuasive feature was very high, and when narrowing down the view to persuasion specific aspects, scores were comparatively high. This seems to strengthen the results obtained by the sentence evaluation talked about in section 5.5.2. Now the theoretical knowledge of good motivational approaches and ways to verbally express support should be taken to an improved prototype with more solid system functions to build on. Subsequent, working on more interactive persuasion targets should be continued.

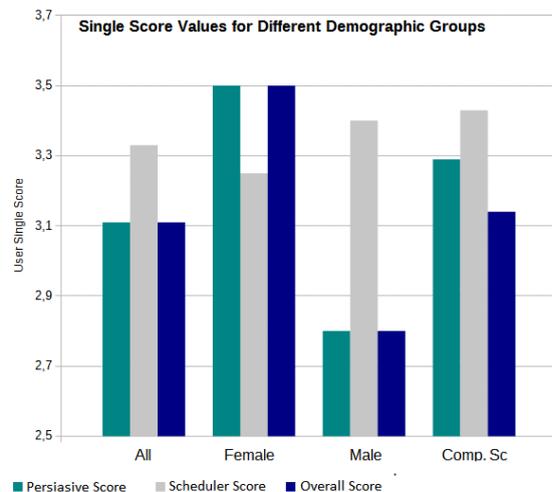


Figure 6.4.: A comparison of the component specific and overall score for different demographics.

⁵A participant pointed out that reducing the persuasion would be useful in times of stress or injury.

⁶Although one participant described it as ‘cuddly’.

⁷The limitations had been explained to the users beforehand, so they were well aware of them.

7. Future Works

This section shall contemplate into which directions further research could be taken, and how the system devised in this work could be improved. Going far into the future and crossing the border to wishful thinking, ideally we would desire an application tracking various health related properties (calorie intake, weight changes, blood pressure, etc.), perform a basic medical diagnosis on this data and offer suggestion for better health behavior suited to the user. With this, the system would be enabled to convince the user of their potential gain referring to the stored user history to personalize the persuasion attempts. All of this in an opt-in manner for the user, ensuring him full control over data stored.

However, thinking more pragmatic, greater merit can be found in focusing on the near and intermediate future.

Stemming from the problems with the evaluation explained above in 6.4, a second execution of the user evaluation, taking extra care to ensure a suitable environment would be beneficial to access missing measurements and possibly adjust future development accordingly.

As has been illustrated extensively in 5.6, there are many limits to the further extension of the program in its current state. Especially for all improvements aiming for a more solid scheduler base (which is a dire user need according to the evaluation results), abandoning Jindigo as framework seems unavoidable¹.

Especially the limits to grammar size set severe boundaries on fulfilling the many user expectations for features². As there is no use in reinventing the wheel, finding an existing calendar application that could be expanded by the speech interface seems a more fruitful approach for future improvement than creation a new solid calendar program.

Although for a focus on improving the persuasion logic, with no further demands toward the scheduler abilities, these two improvements are not such an imminent need.

A first step to improve the persuasion is the creation of a more extensive base from which sentences can be picked to avoid repetition. These could receive online updates of their priority with a learning algorithm rewarding user acceptance after the specific phrase was issued. Another step into the right direction is the creation of a user model. With information stored in there, suggestions better fitting the user's need could be produced. For this frequency of exercise and

¹Alternatively, a major rewrite of parts of its core might suffice.

²The requests for a function to suggest free time slots in the calendar is disregarded here. Granted, it would be a desirable feature for a scheduler. But it is not deemed beneficial towards the goal of increased persuasion performance.

past commitment rates could be employed. User specific settings also enable the user to specify which grade of encouragement and persistence he would like to receive, potentially allowing to suit a wider variety of different user tastes.

Bibliography

- [And12] Stephen R. Andrew. *Motivational Interviewing: A Guided Dialogue*. 2012. URL: <http://www.hetimaine.org/Literature/AGuidedDialogue.pdf> (cit. on p. 23).
- [Aro11a] Jacob Aron. „How innovative is Apple's new voice assistant, Siri?“ In: *New Scientist* 212(2836) (2011), pp. 24–25. URL: <http://nlp.hivefire.com/articles/share/26301/> (cit. on p. 8).
- [Aro11b] Jacob Aron. *Software tricks people into thinking it is human*. New Scientist Article on cleverbot passing the turing test. accessed: March 2015. 2011. URL: <http://www.newscientist.com/article/dn20865-software-tricks-people-into-thinking-it-is-human.html?DCMP=OTC-rss&nsref=online-news> (cit. on p. 8).
- [Aus+95] Harald Aust, Martin Oerder, Frank Seide, and Volker Steinbiss. „The Philips Automatic Train Timetable Information System“. In: *Speech Communication* 17 (1995). URL: <http://research.microsoft.com/apps/pubs/default.aspx?id=121945> (cit. on p. 4).
- [Ban97] A. Bandura. *Self-Efficacy: The Exercise of Control*. Worth Publishers, 1997. ISBN: 9780716728504. URL: http://books.google.de/books?id=eJ-PN9g%5C_o-EC (cit. on p. 23).
- [Bin39] Eando Binder. „I, Robot (short story)“. In: *Amazing Stories* (1939) (cit. on p. 1).
- [Bis+05] Connie L. Bish, Heidi Michels Blanck, Mary K. Serdula, Michele Marcus, Harold W. Kohl, and Laura Kettel Khan. „Diet and Physical Activity Behaviors among Americans Trying to Lose Weight: 2000 Behavioral Risk Factor Surveillance System“. In: *Obesity Research* 13.3 (2005), pp. 596–607. ISSN: 1550-8528. URL: <http://dx.doi.org/10.1038/oby.2005.64> (cit. on pp. 9, 11).
- [Bla+04] L. A. Black, M. McTear, N. Black, R. Harper, and M. Lemon. „DIL-log by design: an integrated tele-care communication infrastructure.“ In: *Conference proceedings : ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Conference* 5 (2004), pp. 3290–3293. ISSN: 1557-170X. URL: <http://dx.doi.org/10.1109/iembs.2004.1403925> (cit. on p. 10).
- [BR] Dan Bohus and Alexander I. Rudnicky. *The RavenClaw Dialog Management Architecture*. Official Website of the Ravenclaw Management Architecture. accessed: 2014. URL: <http://www.cs.cmu.edu/d%CC%83bohus/ravenclaw-olympus/research.html> (cit. on p. 15).
- [BR09] Dan Bohus and Alexander I. Rudnicky. „The RavenClaw dialog management framework: Architecture and systems“. In: *Comput. Speech Lang.* 23.3 (2009), pp. 332–361. URL: <http://dx.doi.org/10.1016/j.csl.2008.10.001> (cit. on p. 15).
- [Car] Rollo Carpenter. *Cleverbot.com - a clever bot - speak to an AI with some Actual Intelligence? A web application that uses an artificial intelligence algorithm to have conversations with humans*. URL: <http://www.cleverbot.com/> (cit. on p. 8).

- [Cha15] The Independent Charlie Cooper. *Three in every four British men will be overweight by 2030, says World Health Organisation*. Online article of the Independent summarizing a WHO study on rising obesity in Europe. accessed: May 2015. 2015. URL: <http://www.independent.co.uk/life-style/health-and-families/health-news/three-in-every-four-british-men-will-be-obese-by-2030-says-the-world-health-organisation-10227099.html> (cit. on p. 12).
- [Dic] Oxford English Dictionary. *Oxford English Dictionary, the definitive record of the English language*. Official Website of Oxford English Dictionary with online service. accessed: February 2015. URL: <http://www.oed.com/> (cit. on p. 2).
- [Eng12] Klaus-Peter Engelbrecht. *Estimating Spoken Dialog System Quality with User Models*. Springer Publishing Company, Incorporated, 2012. ISBN: 3642315909, 9783642315909 (cit. on p. 29).
- [GMM00] D. Gibbon, I. Mertins, and R.K. Moore. *Handbook of Multimodal and Spoken Dialogue Systems: Resources, Terminology and Product Evaluation*. The Springer International Series in Engineering and Computer Science. Springer US, 2000. ISBN: 9780792379041. URL: <http://books.google.co.uk/books?id=Ntb0T7gfIn8C> (cit. on pp. 2, 29).
- [HG00] Kate S. Hone and Robert Graham. „Towards a Tool for the Subjective Assessment of Speech System Interfaces (SASSI)“. In: *Nat. Lang. Eng.* 6.3-4 (2000), pp. 287–303. ISSN: 1351-3249. URL: <http://dx.doi.org/10.1017/S1351324900002497> (cit. on p. 30).
- [Hir+13] Takuya Hiraoka, Yuki Yamauchi, Graham Neubig, Sakriani Sakti, Tomoki Toda, and Satoshi Nakamura. „Dialogue management for leading the conversation in persuasive dialogue systems.“ In: *ASRU. IEEE, 2013*, pp. 114–119. URL: <http://dblp.uni-trier.de/db/conf/asru/asru2013.html#HiraokaYNSTN13> (cit. on p. 9).
- [HKT95] Wolfgang J. Hess, Klaus J. Kohler, and Hans-Günther Tillmann. „The Phondat-verbmobil speech corpus.“ In: *EUROSPEECH. ISCA, 1995*. URL: <http://dblp.uni-trier.de/db/conf/interspeech/eurospeech1995.html#HessKT95> (cit. on p. 21).
- [Hol+06] Hartwig Holzapfel, Thomas Schaaf, Hazim Kemal Ekenel, Christoph Schaa, and Alex Waibel. „A Robot Learns to Know People - First Contacts of a Robot“. In: *KI 2006: Advances in Artificial Intelligence, 29th Annual German Conference on AI, KI 2006, Bremen, Germany, June 14-17, 2006, Proceedings*. 2006, pp. 302–316. URL: http://dx.doi.org/10.1007/978-3-540-69912-5_23 (cit. on pp. 8 sq.).
- [Inc] Amazon.com Inc. *Introducing Amazon Echo*. Official Website of the voice command device Amazon Echo. URL: <http://www.amazon.com/oc/echo/> (cit. on p. 8).
- [JM09] Kristiina Jokinen and Michael F. McTear. *Spoken Dialogue Systems*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers, 2009. DOI: 10.2200/S00204ED1V01Y200910HLT005. URL: <http://dx.doi.org/10.2200/S00204ED1V01Y200910HLT005> (cit. on pp. 4 sq., 10, 29).
- [Krü15] Alfred Krüger. *Experten warnen: Gesundheits-Apps - und ihre Nebenwirkungen - heute-Nachrichten*. online article from the ZDF news portal heute.de. language: German; accessed: April 2015. 2015. URL: <http://www.heute.de/apple-watch-gesundheits-apps-37945462.html> (cit. on p. 9).
- [Kud+96] Ikuo Kudo, Takao Nakama, Tomoko Watanabe, and Reiko Kameyama. „Data collection of Japanese dialects and its influence into speech recognition.“ In: *ICSLP. ISCA, 1996*. URL: <http://dblp.uni-trier.de/db/conf/interspeech/icslp1996.html%5C#KudoNWK96> (cit. on p. 5).

- [Lam+03] Paul Lamere, Philip Kwok, Evandro Gouvea, Bhiksha Raj, Rita Singh, William Walker, Manfred Warmuth, and Peter Wolf. „The CMU SPHINX-4 speech recognition system“. In: *IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP 2003), Hong Kong 1 (2003)*, pp. 2–5 (cit. on p. 17).
- [Les+10] Richard T. Lester, Paul Ritvo, Edward J. Mills, Antony Kariri, Sarah Karanja, Michael H. Chung, William Jack, James Habyarimana, Mohsen Sadatsafavi, Mehdi Najafzadeh, Carlo A Marra, Benson Estambale, Elizabeth Ngugi, T. Blake Ball, Lehana Thabane, Lawrence J. Gelmon, Joshua Kimani, Marta Ackers, and Francis A Plummer. „Effects of a mobile phone short message service on antiretroviral treatment adherence in Kenya (WelTel Kenya1): a randomised trial“. In: *The Lancet* 376 (2010), pp. 1838–1845 (cit. on p. 11).
- [LF05] Diane Litman and Kate Forbes-riley. „Speech recognition performance and learning in spoken dialogue tutoring“. In: *In Proceedings of the 9th European Conference on Speech Communication and Technology (Interspeech/Eurospeech. 2005)*, pp. 161–164 (cit. on p. 7).
- [LH13] Carolyn Lauckner and Gary Hsieh. „The Presentation of Health-related Search Results and Its Impact on Negative Emotional Outcomes“. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. CHI '13. Paris, France: ACM, 2013*, pp. 333–342. ISBN: 978-1-4503-1899-0. URL: <http://doi.acm.org/10.1145/2470654.2470702> (cit. on p. 9).
- [LL05] Esther Levin and Alex Levin. „Spoken Dialog System for Real-Time Data Capture“. In: *in Proc. Interspeech2005. 2005* (cit. on pp. 4, 10).
- [LP] DFKI’s Language Technology Lab and Institute of Phonetics at Saarland University. *The MARY Text-to-Speech System*. Official Website of the Text-to-Speech Synthesis Tool MaryTTS. accessed: 2014. URL: <http://mary.dfki.de/> (cit. on p. 18).
- [LT68] J. C. R. Licklider and Robert W. Taylor. „The Computer as a Communication Device“. In: *Science and Technology* 76 (1968), pp. 21–31 (cit. on p. 2).
- [LTD] InspiredApps (A.L) LTD. *My Diet Coach - Weight loss motivation for women & calorie counter*. Official Website of the smartphone application My Diet Coach. accessed: May 2015. URL: <http://www.mydietcoachapp.com/> (cit. on p. 9).
- [Mat05] Anijo Punnen Mathew. „Using the Environment As an Interactive Interface to Motivate Positive Behavior Change in a Subway Station“. In: *CHI '05 Extended Abstracts on Human Factors in Computing Systems. CHI EA '05. Portland, OR, USA: ACM, 2005*, pp. 1637–1640. ISBN: 1-59593-002-7. URL: <http://doi.acm.org/10.1145/1056808.1056985> (cit. on pp. 9, 24).
- [McT04] M. McTear. *Spoken Dialogue Technology: Towards the Conversational User Interface*. Springer London, 2004. ISBN: 9781852336721. URL: <http://books.google.de/books?id=0kNmwa30o5IC> (cit. on pp. 2 sq., 5, 8, 29).
- [MM93] Christine Moorman and Erika Matulich. „A Model of Consumers' Preventive Health Behaviors: The Role of Health Motivation and Health Ability“. In: *Journal of Consumer Research* 20.2 (1993), pp. 208–28. URL: <http://EconPapers.repec.org/RePEc:ucp:jconrs:v:20:y:1993:i:2:p:208-28> (cit. on p. 12).
- [MNR89] I. McDowell, C. Newell, and W. Rosser. „A randomized trial of computerized reminders for blood pressure screening in primary care.“ In: *Med Care* 27.3 (1989), pp. 297–305. ISSN: 0025-7079. URL: <http://view.ncbi.nlm.nih.gov/pubmed/2494397> (cit. on p. 11).
- [MRC07] I. Mazzotta, F. de Rosis, and V. Carofiglio. „Portia: A User-Adapted Persuasion System in the Healthy-Eating Domain“. In: *Intelligent Systems, IEEE* 22.6 (2007), pp. 42–51. URL: <http://dx.doi.org/10.1109/mis.2007.115> (cit. on pp. 9, 12, 24).

- [Mül13] Andreas Müller. *Die Schule schwänzt das Lernen*. Hep verlag, 2013. ISBN: 3039055135, 978-3039055135 (cit. on p. 23).
- [Mun] Randall Munroe. *A webcomic of romance, sarcasm, math, and language*. Webcomics licensed under Creative Commons Attribution-NonCommercial 2.5 License. URL: <https://xkcd.com/> (cit. on p. 22).
- [Pel13] Waldemar Pelz. *Von der Motivation zur Volition*. Research report as discussion paper. 2013. URL: https://homepages.thm.de/~hg10086/diskussionspapiere/volition_motivation.pdf (cit. on p. 23).
- [PF11] The Pew Internet Project and California HealthCare Foundation. *Health Topics*. Study on health related information look-ups on the internet. accessed: April 2015. 2011. URL: <http://www.pewinternet.org/2011/02/01/health-topics-2/> (cit. on p. 9).
- [Pur+11] Stephen Purpura, Victoria Schwanda, Kaiton Williams, William Stubler, and Phoebe Sengers. „Fit4Life: The Design of a Persuasive Technology Promoting Healthy Behavior and Ideal Weight“. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '11. Vancouver, BC, Canada: ACM, 2011, pp. 423–432. ISBN: 978-1-4503-0228-9. URL: <http://doi.acm.org/10.1145/1978942.1979003> (cit. on pp. 9 sq.).
- [Rau+03] Antoine Raux, Brian Langner, Alan W Black, and Maxine Eskenazi. „LET’S GO: Improving spoken dialog systems for the elderly and non-native“. In: *in Eurospeech03*. 2003 (cit. on p. 7).
- [Raz+12] Agha Ali Raza, Mansoor Pervaiz, Christina Milo, Samia Razaq, Guy Alster, Jahanzeb Sherwani, Umar Saif, and Roni Rosenfeld. „Viral Entertainment As a Vehicle for Disseminating Speech-based Services to Low-literate Users“. In: *Proceedings of the Fifth International Conference on Information and Communication Technologies and Development*. ICTD '12. Atlanta, Georgia, USA: ACM, 2012, pp. 350–359. ISBN: 978-1-4503-1045-1. URL: <http://doi.acm.org/10.1145/2160673.2160715> (cit. on p. 7).
- [Rea14] University of Reading. *Turing Test success marks milestone in computing history*. Press release of the University of Reading the first program passing the turing test. accessed: March 2015. 2014. URL: <https://www.reading.ac.uk/news-and-events/releases/PR583836.aspx> (cit. on p. 8).
- [SA12] Gabriel Skantze and Samer Al Moubayed. „IrisTK: A Statechart-based Toolkit for Multi-party Face-to-face Interaction“. In: *Proceedings of the 14th ACM International Conference on Multimodal Interaction*. ICMI '12. Santa Monica, California, USA: ACM, 2012, pp. 69–76. ISBN: 978-1-4503-1467-1. URL: <http://doi.acm.org/10.1145/2388676.2388698> (cit. on pp. 5, 16).
- [Sak+07] Daisuke Sakamoto, Takayuki Kanda, Tetsuo Ono, Hiroshi Ishiguro, and Norihiro Hagita. „Android As a Telecommunication Medium with a Human-like Presence“. In: *Proceedings of the ACM/IEEE International Conference on Human-robot Interaction*. HRI '07. Arlington, Virginia, USA: ACM, 2007, pp. 193–200. ISBN: 978-1-59593-617-2. URL: <http://doi.acm.org/10.1145/1228716.1228743> (cit. on p. 8).
- [Sar15] Washington Post Sarah Halzack. *Privacy advocates try to keep ‘creepy,’ ‘eavesdropping’ Hello Barbie from hitting shelves*. Online article of the Washington Post about the introduction of the toy prototype Hello Barbie by Mattel. accessed: March 2015. 2015. URL: <http://www.washingtonpost.com/blogs/the-switch/wp/2015/03/11/privacy-advocates-try-to-keep-creepy-eavesdropping-hello-barbie-from-hitting-shelves/> (cit. on p. 8).

- [Sch04] Jeffrey B Schwimmer. „Managing overweight in older children and adolescents“. In: *Pediatric annals* 33.1 (2004), pp. 39–44. ISSN: 0090-4481. DOI: 10.3928/0090-4481-20040101-11. URL: <http://dx.doi.org/10.3928/0090-4481-20040101-11> (cit. on p. 8).
- [Sch96] Michael Schillo. „Working while Driving: Corpus based language modelling of a natural English Voice-User Interface to the in-car Personal Assistant“. MA thesis. University of Leeds, 1996 (cit. on pp. 7, 21 sq.).
- [Ska] Gabriel Skantze. *Jindigo - Java-based Incremental Dialog Framework*. Official Website of the Jindigo Framework. accessed: 2014. URL: <http://jindigo.net/> (cit. on pp. 16, 26).
- [Ska10] Gabriel Skantze. *Jindigo: a Java-based Framework for Incremental Dialogue Systems*. Tech. rep. Department of Speech Music and Hearing, KTH, 2010 (cit. on pp. 15, 17).
- [SM13] Alexander Schmitt and Wolfgang Minker. *Towards Adaptive Spoken Dialog Systems*. Boston (USA): Springer, 2013. URL: <http://www.springer.com/engineering/signals/book/978-1-4614-4592-0> (cit. on p. 5).
- [ST03] Marc Schröder and J. Trouvain. „The German Text-to-Speech Synthesis System MARY: A Tool for Research, Development and Teaching“. In: *International Journal of Speech Technology* 6 (2003), pp. 365–377 (cit. on p. 18).
- [Tat+03] Gábor Tatai, Annamária Csordás, Árpád Kiss, Attila Szaló, and László Laufer. „Happy Chatbot, Happy User“. English. In: *Intelligent Virtual Agents*. Ed. by Thomas Rist, Ruth S. Aylett, Daniel Ballin, and Jeff Rickel. Vol. 2792. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2003, pp. 5–12. ISBN: 978-3-540-20003-1. URL: http://dx.doi.org/10.1007/978-3-540-39396-2_2 (cit. on p. 8).
- [TS12] LimeSurvey Project Team and Carsten Schmitz. *LimeSurvey: An Open Source survey tool*. LimeSurvey: An Open Source survey tool. 2012. URL: <http://www.limesurvey.org> (cit. on p. 30).
- [Tsu+15] Yuiko Tsunomori, Graham Neubig, Sakriani Sakti, Tomoki Toda, and Satoshi Nakamura. „An Analysis Towards Dialogue-based Deception Detection“. In: *IWSDS2015*. 2015. URL: http://www.uni-ulm.de/fileadmin/website_uni_ulm/allgemein/2015_iwsds/iwsds2015_submission_17.pdf (cit. on p. 9).
- [Tur50] A. M. Turing. „Computing Machinery and Intelligence“. In: *Mind* 59 (1950). One of the most influential papers in the history of the cognitive sciences: <http://cogsci.umn.edu/millennium/final.html>, pp. 433–460. URL: <http://cogprints.org/499/> (cit. on p. 7).
- [Web11] WebMD. *WebMD Announces Fourth Quarter and Year End Financial Results*. WebMD website summarizing fourth quarter and year end financial results. accessed: April 2015. 2011. URL: <http://investor.shareholder.com/wbmd/releasedetail.cfm?ReleaseID=65117&CompanyID=WBMD> (cit. on p. 9).

Appendix

A. Code Examples

A.1. Grammar and Configuration Example XML Files

A.1.1. XML File with Program Configurations

```
1      <components>
2          <component name="InputGrammar" class="net.jindigo.domain.grammar.
3              ↳ InputGrammar">
4              <params>
5                  <param name="include">grammar.xml</param>
6              </params>
7          </component>
8          <component name="OutputGrammar" class="net.jindigo.domain.grammar.
9              ↳ OutputGrammar">
10             <params>
11                 <param name="include">responses.xml</param>
12             </params>
13         </component>
14
15     <module name="Recognizer" class="net.jindigo.module.sphinx.Sphinx">
16         <params>
17             <param name="configuration">sphinx.xml</param>
18             <param name="grammar">InputGrammar</param>
19             <param name="grammarTransformer">jsgfGrammarTransformer</param>
20             <param name="sphinx.frontend">micFrontEnd</param>
21             <param name="sphinx.utteranceSegmenter">speechDetector</param>
22             <param name="sphinx.acousticModel">wsj.acousticModel</param>
23             <param name="sphinx.dictionary">wsj.dictionary</param>
24         </params>
25         <output>
26             <channel target="Interpreter" class="net.jindigo.module.recognizer.Word
27                 ↳ "/>
28             <channel target="Vocalizer" class="net.jindigo.module.recognizer.
29                 ↳ SpeechEvent"/>
30         </output>
31     </module>
32
33     <module name="Interpreter" class="net.jindigo.module.interpreter.
34         ↳ Interpreter">
35         <params>
36             <param name="grammar">InputGrammar</param>
37         </params>
38         <output>
39             <channel target="Contextualizer" class="net.jindigo.module.
40                 ↳ interpreter.Phrase"/>
41         </output>
42     </module>
43
44     <module name="Contextualizer" class="net.jindigo.module.contextualizer.
45         ↳ Contextualizer">
46         <output>
47             <channel target="ActionManagment" class="net.jindigo.module.
48                 ↳ contextualizer.Context"/>
49         </output>
50     </module>
```

```

40         <channel target="Vocalizer" class="net.jindigo.module.
           ↪ contextualizer.Context"/>
41     </output>
42 </module>
43 <module name="ActionManagment" class="dialogManager.StateHandler">
44     <params>
45         <param name="outputGrammar">OutputGrammar</param>
46     </params>
47     <output>
48         <channel target="Vocalizer" class="net.jindigo.module.vocalizer
           ↪ .SpeechPlan"/>
49     </output>
50 </module>
51 <module name="Vocalizer" class="net.jindigo.module.vocalizer.
           ↪ MaryVocalizer">
52     <params>
53         <param name="host">127.0.0.1</param>
54         <param name="port">59125</param>
55     </params>
56     <output>
57         <channel target="Contextualizer" class="net.jindigo.domain.
           ↪ Concept"/>
58     </output>
59 </module>
60 </components>

```

A.1.2. Input Grammar for General Program Functions

```

1     <!--
2     *****
3     *** Recognition Rules ***
4     *****
5     -->
6     <rules use="input">
7
8         <rule name="programControls">
9             <form>
10                <example>quit</example>
11                <example>back</example>
12                <example>repeat</example>
13                <example>i don't understand</example>
14                <entryref name="control" link="command"/>
15            </form>
16
17            <sem>
18                <copy link="command"/>
19            </sem>
20        </rule>
21
22
23        <rule name="yesNo">
24            <form>
25                <entryref name="yes" link="yesNo"/>
26            </form>
27            <form>
28                <entryref name="no" link="yesNo"/>
29            </form>
30            <sem>
31                <cref link="yesNo"/>
32            </sem>
33        </rule>
34
35    </rules>
36
37    <!--

```

```

38 *****
39 *** Lexicon ***
40 *****
41 -->
42 <lexicon top="false">
43     <!-- *** Menu and control expressions ***-->
44
45     <!-- Access help -->
46     <entry name="control">
47         <form>help</form>
48         <form>i don't understand</form>
49         <form>>manual</form>
50         <form>explain</form>
51         <form>explanation</form>
52         <form>What should i do</form>
53         <form>how to</form>
54         <sem>
55             <concept name="Help"/>
56         </sem>
57     </entry>
58
59
60     <!-- End Program -->
61     <entry name="control">
62         <form>exit</form>
63         <form>quit</form>
64         <form>shut down</form>
65         <form>end</form>
66         <form>that's all</form>
67         <form>i'm done</form>
68         <sem>
69             <concept name="Exit"/>
70         </sem>
71     </entry>
72
73     <!-- Return -->
74     <entry name="control">
75         <form>return</form>
76         <form>back</form>
77         <form>previous</form>
78         <form>cancel</form>
79         <form>abort</form>
80         <sem>
81             <concept name="Return"/>
82         </sem>
83     </entry>
84
85     <!-- Repeat -->
86     <entry name="control">
87         <form>repeat</form>
88         <form>once more</form>
89         <form>again</form>
90         <form>what did you say</form>
91         <form>what was that</form>
92         <sem>
93             <concept name="Repeat"/>
94         </sem>
95     </entry>
96
97
98     <!-- Suspend -->
99     <entry name="control">
100         <form>suspend</form>
101

```

```

102     <form>pause</form>
103     <form>wait</form>
104     <form>stop</form>
105     <form>be quiet</form>
106     <form>shut up</form>
107     <form>freeze</form>
108     <form>silence</form>
109     <form>be silent</form>
110     <form>one minute</form>
111     <form>one moment</form>
112     <form>sleep</form>
113     <sem>
114         <concept name="Suspend"/>
115     </sem>
116 </entry>
117
118 <!-- Resume -->
119 <entry name="control">
120     <form>resume</form>
121     <form>continue</form>
122     <form>i'm back</form>
123     <form>let's start</form>
124     <form>wake up</form>
125     <sem>
126         <concept name="Resume"/>
127     </sem>
128 </entry>
129
130
131 <!-- Yes -->
132 <entry name="yes">
133     <form>yes</form>
134     <form>yeah</form>
135     <form>OK</form>
136     <form>all right</form>
137     <form>good</form>
138     <form>sure thing</form>
139     <form>by all means</form>
140     <form>yep</form>
141     <form>okey dokey</form>
142     <form>>true</form>
143     <form>right</form>
144     <form>correct</form>
145     <form>fine</form>
146     <form>that's what i said</form>
147     <form>sure</form>
148     <form>affirmative</form>
149     <form>positive</form>
150     <form>exactly</form>
151     <form>precisely</form>
152     <form>i agree</form>
153     <form>agreed</form>
154     <sem>
155         <concept name="YesNo" value="true"/>
156     </sem>
157 </entry>
158
159 <!-- No -->
160 <entry name="no">
161     <form>no</form>
162     <form>wrong</form>
163     <form>nope</form>
164     <form>nay</form>
165     <form>not really</form>

```

```

166     <form>not at all</form>
167     <form>that's wrong</form>
168     <form>>false</form>
169     <form>negative</form>
170     <form>don't</form>
171     <sem>
172         <concept name="YesNo" value="false"/>
173     </sem>
174 </entry>
175
176 <!-- *** Others ***-->
177
178 <!-- Determiner -->
179 <entry name="determiner">
180     <form>the</form>
181     <form>this</form>
182     <form>a</form>
183     <form>one</form>
184     <form>my</form>
185     <form>some</form>
186     <form>any</form>
187 </entry>
188
189 </lexicon>
190 </grammar>

```

A.1.3. Output Grammar for Persuasion Specific Responses

```

1     <!-- ***** -->
2     <!-- *** Persuasive Rules *** -->
3     <!-- ***** -->
4
5     <rule name="feedback">
6         <form>
7             <optional><verbalize link="app"/></optional>
8         </form>
9         <sem use="input" match="lax">
10            <cref link="app"/>
11        </sem>
12    </rule>
13
14    <rule name="appPastNotification">
15        <form>
16            <text>I have a reminder for you,</text>
17            <ruleref name="feedback" link="feedback"/>
18            <text>was scheduled .</text>
19        </form>
20        <sem use="input">
21            <cref link="feedback"/>
22        </sem>
23    </rule>
24
25    <rule name="PersuasionReEntrance">
26        <form>
27            <text>we were discussing ,</text>
28            <ruleref name="feedback" link="feedback"/>
29            <text>you had in the past .</text>
30        </form>
31        <sem use="input">
32            <cref link="feedback"/>
33        </sem>
34    </rule>
35
36    <rule name="notAllExercisesDone">
37        <form>

```

```

38     <text> seems like you did not do all the exercise you set for yourself
      ↪ .</text>
39   </form>
40 </rule>
41
42 <rule name="noFollowUp">
43   <form>
44     <text>as you wish.</text>
45     <text>just take care you don't loose sight of your health goals .</text>
      ↪ >
46   </form>
47 </rule>
48
49 <rule name="pastAppFollowUp">
50   <form>
51     <text>would you like to schedule a follow up date .</text>
52   </form>
53   <sem use="output">
54     <concept name="net.jindigo.domain.builtin.RequestWH">
55       <arg name="struct">
56         <concept name="PersuasiveCase">
57           <arg name="followUp">
58             <concept name="YesNo" link="1"/>
59           </arg>
60         </concept>
61       </arg>
62       <arg name="requested">
63         <cref link="1"/>
64       </arg>
65     </concept>
66   </sem>
67 </rule>
68
69
70 <rule name="doneQuestion">
71   <form>
72     <text> So how was it , did you do it ?</text>
73   </form>
74   <sem use="output">
75     <concept name="net.jindigo.domain.builtin.RequestWH">
76       <arg name="struct">
77         <concept name="ExercisePersuasionCase">
78           <arg name="done">
79             <concept name="YesNo" link="1"/>
80           </arg>
81         </concept>
82       </arg>
83       <arg name="requested">
84         <cref link="1"/>
85       </arg>
86     </concept>
87   </sem>
88 </rule>
89
90 <rule name="nowQuestion">
91   <form>
92     <text> if you have a moment, why don't you exercise right now .</text>
93     <text> even if it is only for ten minutes .</text>
94     <text>will you do it .</text>
95   </form>
96   <sem use="output">
97     <concept name="net.jindigo.domain.builtin.RequestWH">
98       <arg name="struct">
99         <concept name="ExercisePersuasionCase">

```

```

100         <arg name="now">
101             <concept name="YesNo" link="1"/>
102         </arg>
103     </concept>
104 </arg>
105     <arg name="requested">
106         <cref link="1"/>
107     </arg>
108 </concept>
109 </sem>
110 </rule>
111
112 <rule name="tomorrowQuestion">
113     <form>
114         <text> maybe you can find some time tomorrow . How is it .</text>
115     </form>
116     <sem use="output">
117         <concept name="net.jindigo.domain.builtin.RequestWH">
118             <arg name="struct">
119                 <concept name="ExercisePersuasionCase">
120                     <arg name="tomorrow">
121                         <concept name="YesNo" link="1"/>
122                     </arg>
123                 </concept>
124             </arg>
125             <arg name="requested">
126                 <cref link="1"/>
127             </arg>
128         </concept>
129     </sem>
130 </rule>
131
132 <rule name="otherDateQuestion">
133     <form>
134         <text> would you like to schedule an exercise for an other time . </
135         ↔ text>
136     </form>
137     <sem use="output">
138         <concept name="net.jindigo.domain.builtin.RequestWH">
139             <arg name="struct">
140                 <concept name="ExercisePersuasionCase">
141                     <arg name="followUp">
142                         <concept name="YesNo" link="1"/>
143                     </arg>
144                 </concept>
145             </arg>
146             <arg name="requested">
147                 <cref link="1"/>
148             </arg>
149         </concept>
150     </sem>
151 </rule>

```

Score	Motivation Phrase
2,38	Discipline is simply the ability to discern between what we want most, and what we want right now.
2,50	Whether you think you can or whether you think you can't, you're right!
2,67	But remember, nothing beats a failure but a try!
2,90	It is not more than you can handle. It is merely more than you thought you could.
3,00	You may have to fight a battle more than once to win it.
3,07	Start off with what you can do, so you're not discouraged when you do something you can't do. You'll work your way up in no time.
3,10	In one month you can be saying, look at these results, I'm glad I kept pushing, or, oh no, I'd be so much further if I didn't let it all go. Which do you want to say?
3,22	Regret will not change your fitness, only action will!
3,29	One setback will not defeat you. One setback will not define you. You are strong. You will succeed.
3,50	Start where you are. Use what you have. Do what you can.
3,67	So come on. You can do this. Start small. Don't give up when you get frustrated. Just keep going.
4,33	Just keep at it!
4,50	You might not be where you want to be right now, but you are getting there.

Figure B.2.: The score obtained for different motivational phrases. The three sentences marked red were removed from the program because of their bad score.

Score	Praise Phrase
2,50	Way to show perseverance!
2,60	What an incredible accomplishment!
2,70	Doesn't it feel good to stay motivated to succeed?
2,89	You have done a great thing for your health
3,00	Keeping to your plans helps you reach your goals.
3,00	Great, I am very proud of you.
3,00	Once you get some momentum going it gets easier to stay going in the right direction.
3,38	Awesome, it is all about persistence
3,62	Keep that Spark burning, and you will get to your goal!
3,70	Well done, that is awesome
3,75	Sounds like you are well on the way to getting to your goals!
3,92	Keep up the hard work. You're doing great.
4,50	Fantastic job! Keep up the good work!

Figure B.3.: The score obtained for different praising phrases. The three sentences marked red were removed from the program because of their bad score.

B. Persuasive Phrases

B.0.4. Motivational Phrases

B.0.5. Appraisal Phrases

C. Evaluation Files

C.0.6. Final Evaluation Questionnaire

Persuasive Personal Health Aid

Hello and thank you for participating in this user evaluation of this persuasive personal health managing aid.

There are 18 questions in this survey

Demographic Values

First a few general informations about you.

[]Your age: *

Only an integer value may be entered in this field.

Please write your answer here:

[]Your gender: *

Please choose **only one** of the following:

- Female
- Male

[]Your field of study: *

Please choose **only one** of the following:

- Computer Science
- Electrical Engineering
- Physics
- Mathematics
- Industrial Engineering (Wlmg)
- Biology
- Other

This list is only meant suggestive, not exhaustive. If your field of study isn't in the predefined examples please write it in the 'other' field.

[]Please rate how much previous experiences you have with dialog systems by selecting how much you agree with the following statements. *

Please choose the appropriate response for each item:

	1	2	3	4	5
I can judge the abilities of modern artificial intelligence systems.	<input type="radio"/>				
I use one or more dialog systems (e.g. Siri on iPhone) on a regular basis.	<input type="radio"/>				
I have worked in or researched the development of dialog systems or related fields	<input type="radio"/>				

(1 = don't agree at all; 5 = completely agree)

Persuasion Quality

Please complete the second test scenario. Afterwards please rate how you liked the interaction with the system.

[] What is your opinion of the tested functionality? Please give a score. *

Please choose **only one** of the following:

- 1
 2
 3
 4
 5

(1 = bad; 5 = good)

[] General Quality: How much do you agree with the following statements? *

Please choose the appropriate response for each item:

	1	2	3	4	5
The system is accurate	<input type="radio"/>				
The system is useful	<input type="radio"/>				
I felt confident using the system	<input type="radio"/>				
The interaction with the system is not repetitive	<input type="radio"/>				
I always knew what to say to the system	<input type="radio"/>				
The interaction with the system is fast	<input type="radio"/>				

(1 = don't agree at all; 5 = completely agree)

[]

Now to a few questions especially about system persuasion.

[] Persuasion Quality: How much do you agree with the following statements? *

Please choose the appropriate response for each item:

	1	2	3	4	5
The system suggestions were helpful	<input type="radio"/>				
Using the system was motivating	<input type="radio"/>				
I like the general idea of a motivational system for my personal use	<input type="radio"/>				

(1 = don't agree at all; 5 = completely agree)

[] Persuasion Flaws: How much do you agree with the following statements? *

Please choose the appropriate response for each item:

	1	2	3	4	5
The phrases to motivate me felt out of place or unfitting	<input type="radio"/>				
The system's manner of speech was rude	<input type="radio"/>				
I felt patronized by the system	<input type="radio"/>				

(1 = don't agree at all; 5 = completely agree)

Do you have any further comments about usage experience?

Please write your answer here:

--

Scheduler Quality

Please complete the first test scenario. Afterwards please rate how you liked the interaction with the system.

What is your opinion of the tested functionality? Please give a score. *Please choose **only one** of the following:

- 1
 2
 3
 4
 5

(1 = bad; 5 = good)

General Quality: How much do you agree with the following statements? *

Please choose the appropriate response for each item:

	1	2	3	4	5
The system is accurate	<input type="radio"/>				
The system is useful	<input type="radio"/>				
I felt confident using the system	<input type="radio"/>				
The interaction with the system is not repetitive	<input type="radio"/>				
I always knew what to say to the system	<input type="radio"/>				
The interaction with the system is fast	<input type="radio"/>				

(1 = don't agree at all; 5 = completely agree)

Do you have any further comments about usage experience?

Please write your answer here:

--

Overall Quality

[]

What is your opinion of the system as a whole? Please give a score. *

Please choose **only one** of the following:

- 1
 2
 3
 4
 5

(0 = bad; 5 = good)

[] **How much do you agree with the following statements? ***

Please choose the appropriate response for each item:

- | | 1 | 2 | 3 | 4 | 5 |
|---|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| I would use this system in everyday life | <input type="radio"/> |
| Generally a system of this kind could be useful | <input type="radio"/> |

(0 = don't agree at all; 5 = completely agree)

[]

Where there features you expected that the system didn't offer? *

Please choose **only one** of the following:

- Yes
 No

[] **What kind of features?**

Only answer this question if the following conditions are met:

Answer was 'Yes' at question '16 [OverallMissingFeatur]' (Where there features you expected that the system didn't offer?)

Please write your answer here:

[] Do you have any further comments about usage of the system as a whole or about this Evaluation?

Please write your answer here:

Thank you very much for completing this survey.
06.05.2015 – 20:19

Submit your survey.
Thank you for completing this survey.

C.0.7. Final Evaluation Dialog Example

```

1  System:      "hello , i am your talking schedule ., i have a reminder for
   ↪ you , dermatologist session on the 30th of april at 11 30 was scheduled
   ↪ ., would you like to schedule a follow up date ."
2  User:        - no
3  System:      "i have a reminder for you , workout session on the 25th of
   ↪ april at 5 o'clock was scheduled ., so how was it , did you do it ?"
4  User:        - no
5  System:      "so how was it , did you do it ?"
6  User:        - no
7  System:      "seems like you did not do all the exercise you set for
   ↪ yourself ., yet you may have to fight a battle more than once to win it
   ↪ ., if you have a moment , why don't you exercise right now . even if it
   ↪ is only for ten minutes . will you do it ?"
8  User:        - yes
9  System:      "way to show perseverance !, would you like to schedule a
   ↪ follow up date ."
10 User:        - no
11 System:      "is there anything i can do for you"
12 User:        - any appointments on may the thirtieth
13 System:      "alright . let's see what's in your calendar , limit the search
   ↪ by specifying the type or date ."
14 User:        - may the thirtieth
15 System:      "workout session on the 30th of may at 9 20 found, is there
   ↪ anything i can do for you"
16 User:        - retrieve date
17 System:      "alright . let's see what's in your calendar , limit the search
   ↪ by specifying the type or date ."
18 User:        - may fourteenth
19 System:      "there are no matching appointments ., is there anything i can
   ↪ do for you"
20 User:        - schedule dentist appointment on may the fourteenth
21 System:      "accessing your calendar , at what time"
22 User:        - at nine o'clock
23 System:      "what kind of appointment is it"
24 User:        - a dentist appointment
25 System:      "good , about to store dentist session on the 14th of may at 9
   ↪ o'clock , is that correct ,"
26 System:      "understood . appointment stored ., is there anything i can do
   ↪ for you"
27 User:        - shut down
28 System:      "are you sure you want to end me"
29 User:        - yes
30 System:      "shutting me down . until next time . bye"

```

C.0.8. Final Evaluation Statistical Results



Quick statistics
Survey 189194 'Persuasive Personal Health Aid'

Results

Survey 189194

Number of records in this query:	9
Total records in survey:	9
Percentage of total:	100.00%



Quick statistics
Survey 189194 'Persuasive Personal Health Aid'

Field summary for Age

Your age:

Calculation	Result
Count	9
Sum	221.0000000000
Standard deviation	3.1
Average	24.56
Minimum	21.0000000000
1st quartile (Q1)	22
2nd quartile (Median)	24
3rd quartile (Q3)	25.5
Maximum	32.0000000000

Null values are ignored in calculations
Q1 and Q3 calculated using minitab method



Quick statistics
Survey 189194 'Persuasive Personal Health Aid'

Field summary for Gender

Your gender:

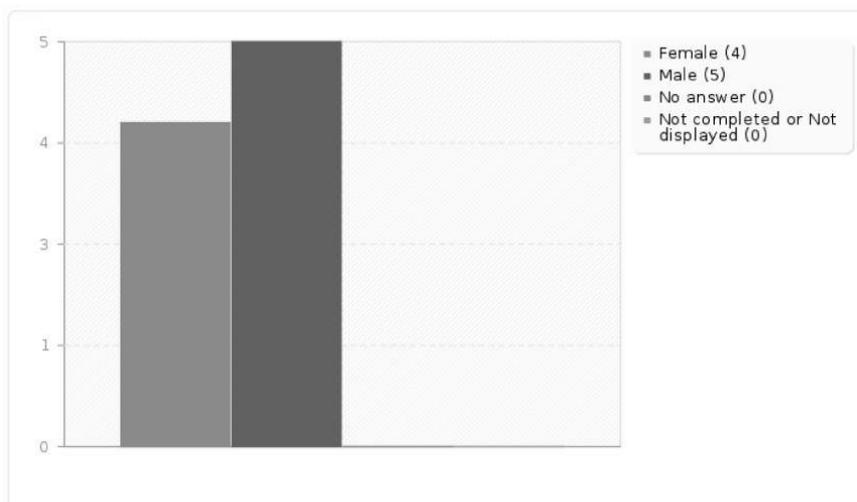
Answer	Count	Percentage
Female (F)	4	44,44%
Male (M)	5	55,56%
No answer	0	0,00%
Not completed or Not displayed	0	0,00%



Quick statistics
Survey 189194 'Persuasive Personal Health Aid'

Field summary for Gender

Your gender:





Quick statistics
Survey 189194 'Persuasive Personal Health Aid'

Field summary for Mayor

Your field of study:

Answer	Count	Percentage
Computer Science (CS)	7	77.78%
Electrical Engineering (EE)	1	11.11%
Physics (Ph)	0	0.00%
Mathematics (Math)	0	0.00%
Industrial Engineering (WIng) (WIng)	0	0.00%
Biology (Bio)	1	11.11%
Other	0	0.00%
No answer	0	0.00%
Not completed or Not displayed	0	0.00%

ID	Response
----	----------



Quick statistics
Survey 189194 'Persuasive Personal Health Aid'

Field summary for PreviousExperience(AI)

Please rate how much previous experiences you have with dialog systems by selecting how much you agree with the following statements. [I can judge the abilities of modern artificial intelligence systems.]

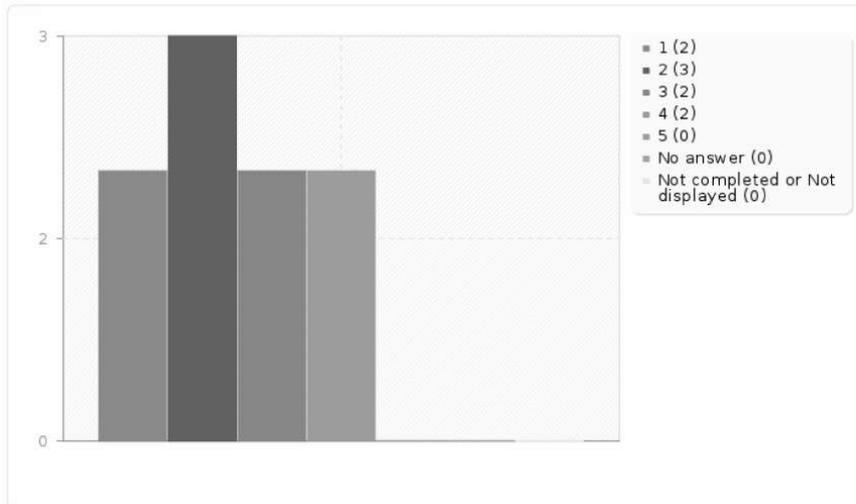
Answer	Count	Percentage	Sum
1 (1)	2	22.22%	55.56%
2 (2)	3	33.33%	
3 (3)	2	22.22%	22.22%
4 (4)	2	22.22%	
5 (5)	0	0.00%	22.22%
No answer	0	0.00%	
Not completed or Not displayed	0	0.00%	
Arithmetic mean	2.44		
Standard deviation	1.13		
Sum (Answers)	9	100.00%	100.00%
Number of cases	9	100.00%	



Quick statistics
Survey 189194 'Persuasive Personal Health Aid'

Field summary for PreviousExperience(AI)

Please rate how much previous experiences you have with dialog systems by selecting how much you agree with the following statements. [I can judge the abilities of modern artificial intelligence systems.]



Quick statistics
Survey 189194 'Persuasive Personal Health Aid'

Field summary for PreviousExperience(Use)

Please rate how much previous experiences you have with dialog systems by selecting how much you agree with the following statements. [I use one or more dialog systems (e.g. Siri on iPhone) on a regular basis.]

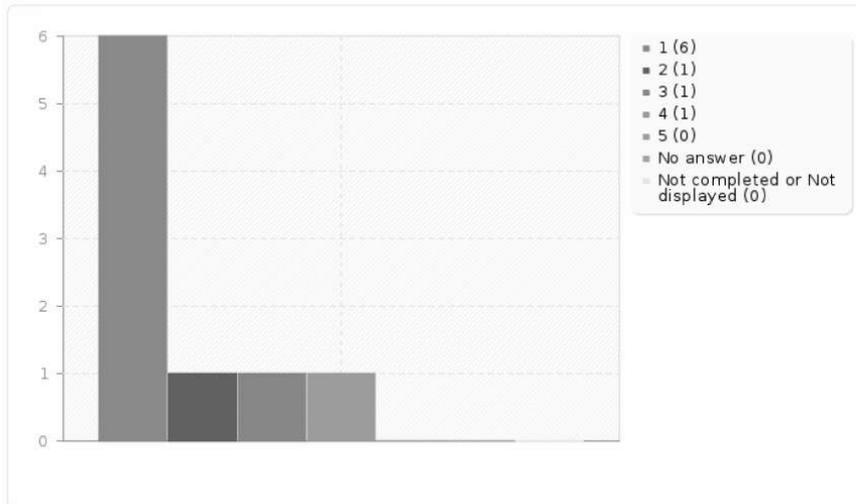
Answer	Count	Percentage	Sum
1 (1)	6	66.67%	77.78%
2 (2)	1	11.11%	
3 (3)	1	11.11%	11.11%
4 (4)	1	11.11%	
5 (5)	0	0.00%	11.11%
No answer	0	0.00%	
Not completed or Not displayed	0	0.00%	
Arithmetic mean	1.67		
Standard deviation	1.12		
Sum (Answers)	9	100.00%	100.00%
Number of cases	9	100.00%	



Quick statistics
Survey 189194 'Persuasive Personal Health Aid'

Field summary for PreviousExperience(Use)

Please rate how much previous experiences you have with dialog systems by selecting how much you agree with the following statements. [I use one or more dialog systems (e.g. Siri on iPhone) on a regular basis.]



Quick statistics
Survey 189194 'Persuasive Personal Health Aid'

Field summary for PreviousExperience(Work)

Please rate how much previous experiences you have with dialog systems by selecting how much you agree with the following statements. [I have worked in or researched the development of dialog systems or related fields]

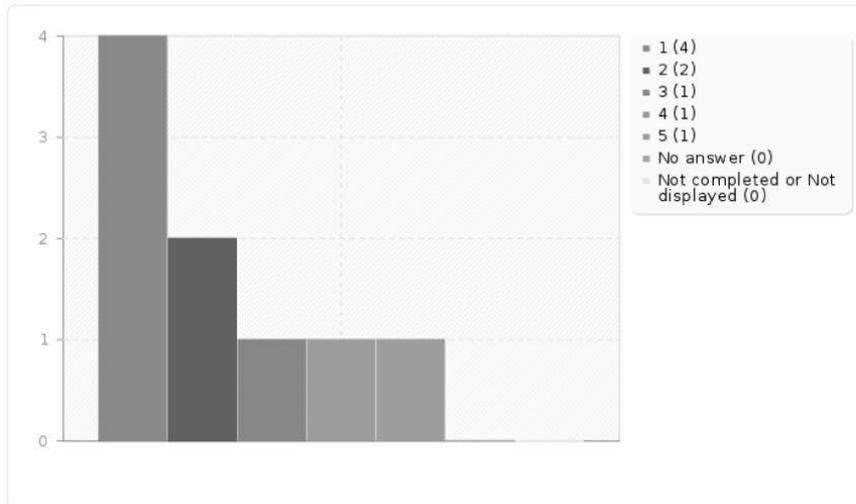
Answer	Count	Percentage	Sum
1 (1)	4	44.44%	66.67%
2 (2)	2	22.22%	
3 (3)	1	11.11%	11.11%
4 (4)	1	11.11%	
5 (5)	1	11.11%	22.22%
No answer	0	0.00%	
Not completed or Not displayed	0	0.00%	
Arithmetic mean	2.22		
Standard deviation	1.48		
Sum (Answers)	9	100.00%	100.00%
Number of cases	9	100.00%	



Quick statistics
Survey 189194 'Persuasive Personal Health Aid'

Field summary for PreviousExperience(Work)

Please rate how much previous experiences you have with dialog systems by selecting how much you agree with the following statements. [I have worked in or researched the development of dialog systems or related fields]



Quick statistics
Survey 189194 'Persuasive Personal Health Aid'

Field summary for PersuasiveScore

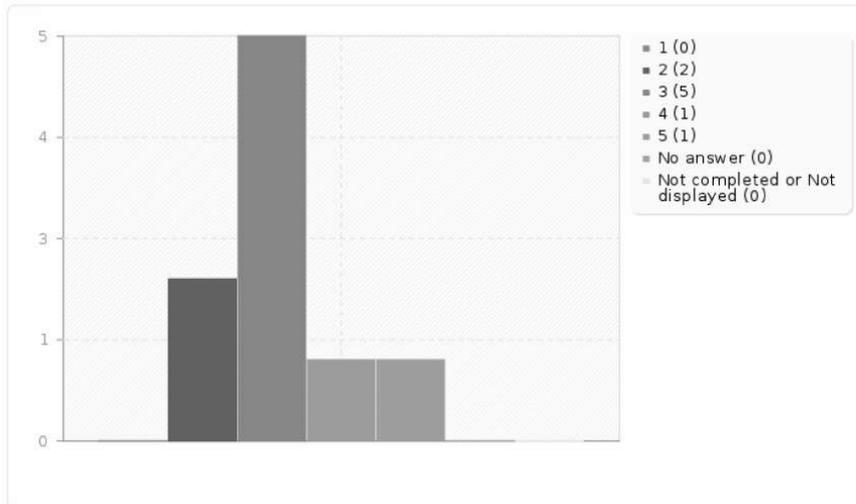
What is your opinion of the tested functionality? Please give a score.

Answer	Count	Percentage	Sum
1 (1)	0	0.00%	22.22%
2 (2)	2	22.22%	
3 (3)	5	55.56%	55.56%
4 (4)	1	11.11%	
5 (5)	1	11.11%	22.22%
No answer	0	0.00%	
Not completed or Not displayed	0	0.00%	
Arithmetic mean	3.11		
Standard deviation	0.93		
Sum (Answers)	9	100.00%	100.00%
Number of cases	9	100.00%	



Field summary for PersuasiveScore

What is your opinion of the tested functionality? Please give a score.



Field summary for PersuasiveSASSI(ResponseAccuracy)

General Quality: How much do you agree with the following statements? [The system is accurate]

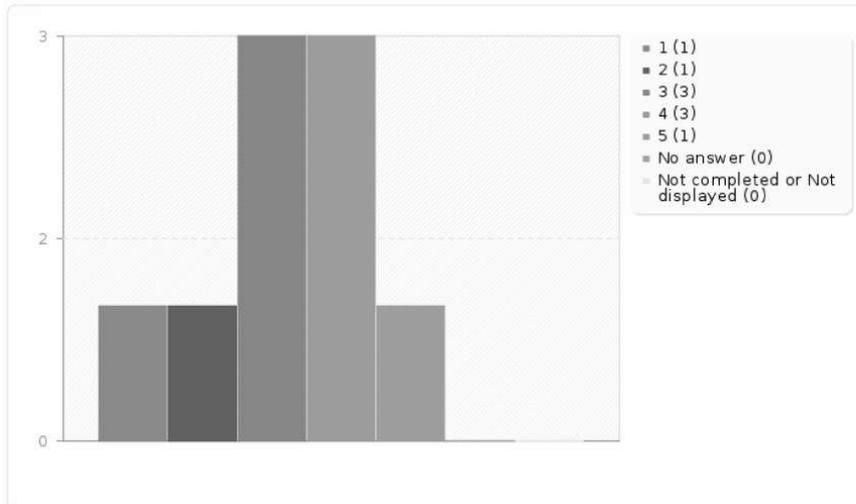
Answer	Count	Percentage	Sum
1 (1)	1	11.11%	22.22%
2 (2)	1	11.11%	
3 (3)	3	33.33%	33.33%
4 (4)	3	33.33%	
5 (5)	1	11.11%	44.44%
No answer	0	0.00%	
Not completed or Not displayed	0	0.00%	
Arithmetic mean	3.22		
Standard deviation	1.2		
Sum (Answers)	9	100.00%	100.00%
Number of cases	9	100.00%	



Quick statistics
Survey 189194 'Persuasive Personal Health Aid'

Field summary for PersuasiveSASSI(ResponseAccuracy)

General Quality: How much do you agree with the following statements? [The system is accurate]



Quick statistics
Survey 189194 'Persuasive Personal Health Aid'

Field summary for PersuasiveSASSI(Likeability)

General Quality: How much do you agree with the following statements? [The system is useful]

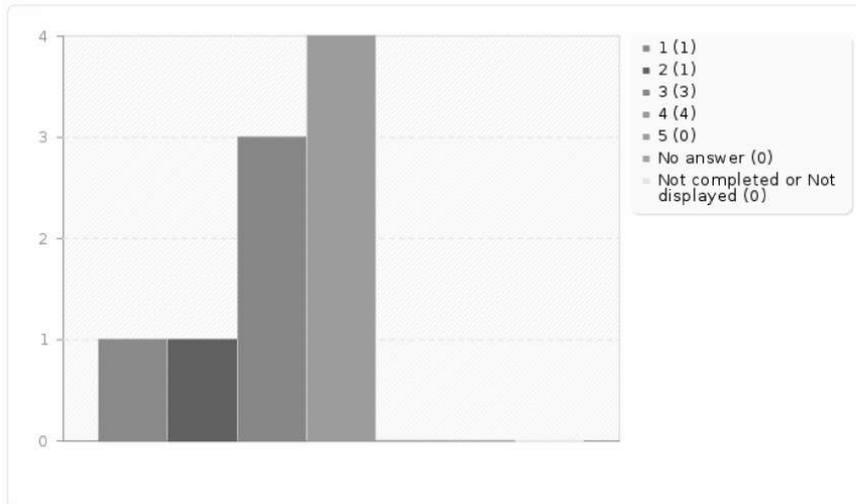
Answer	Count	Percentage	Sum
1 (1)	1	11.11%	22.22%
2 (2)	1	11.11%	
3 (3)	3	33.33%	33.33%
4 (4)	4	44.44%	
5 (5)	0	0.00%	44.44%
No answer	0	0.00%	
Not completed or Not displayed	0	0.00%	
Arithmetic mean	3.11		
Standard deviation	1.05		
Sum (Answers)	9	100.00%	100.00%
Number of cases	9	100.00%	



Quick statistics
Survey 189194 'Persuasive Personal Health Aid'

Field summary for PersuasiveSASSI(Likeability)

General Quality: How much do you agree with the following statements? [The system is useful]



Quick statistics
Survey 189194 'Persuasive Personal Health Aid'

Field summary for PersuasiveSASSI(CognitiveDemand)

General Quality: How much do you agree with the following statements? [I felt confident using the system]

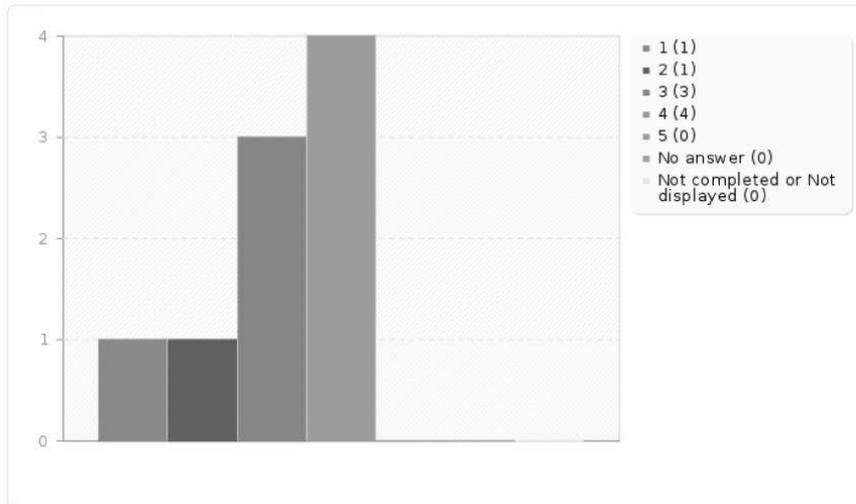
Answer	Count	Percentage	Sum
1 (1)	1	11.11%	22.22%
2 (2)	1	11.11%	
3 (3)	3	33.33%	33.33%
4 (4)	4	44.44%	
5 (5)	0	0.00%	44.44%
No answer	0	0.00%	
Not completed or Not displayed	0	0.00%	
Arithmetic mean	3.11		
Standard deviation	1.05		
Sum (Answers)	9	100.00%	100.00%
Number of cases	9	100.00%	



Quick statistics
Survey 189194 'Persuasive Personal Health Aid'

Field summary for PersuasiveSASSI(CognitiveDemand)

General Quality: How much do you agree with the following statements? [I felt confident using the system]



Quick statistics
Survey 189194 'Persuasive Personal Health Aid'

Field summary for PersuasiveSASSI(Annoyance)

General Quality: How much do you agree with the following statements? [The interaction with the system is not repetitive]

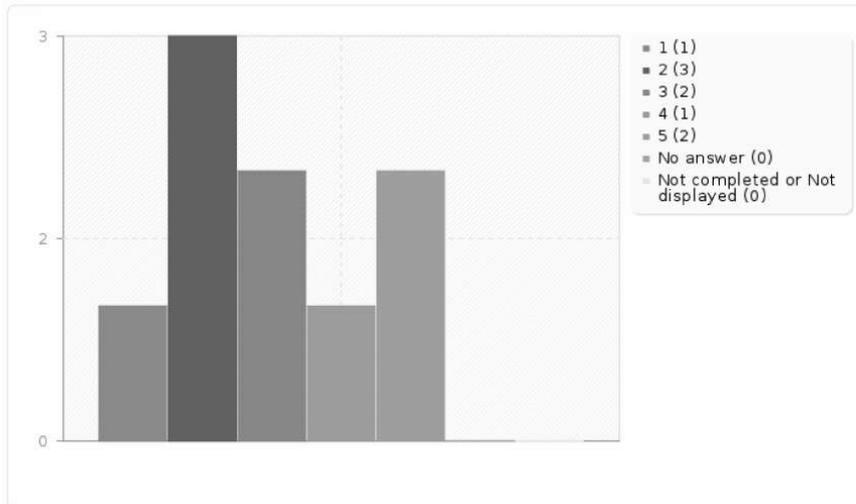
Answer	Count	Percentage	Sum
1 (1)	1	11.11%	44.44%
2 (2)	3	33.33%	
3 (3)	2	22.22%	22.22%
4 (4)	1	11.11%	
5 (5)	2	22.22%	33.33%
No answer	0	0.00%	
Not completed or Not displayed	0	0.00%	
Arithmetic mean	3		
Standard deviation	1.41		
Sum (Answers)	9	100.00%	100.00%
Number of cases	9	100.00%	



Quick statistics
Survey 189194 'Persuasive Personal Health Aid'

Field summary for PersuasiveSASSI(Annoyance)

General Quality: How much do you agree with the following statements? [The interaction with the system is not repetitive]



Quick statistics
Survey 189194 'Persuasive Personal Health Aid'

Field summary for PersuasiveSASSI(Habitability)

General Quality: How much do you agree with the following statements? [I always knew what to say to the system]

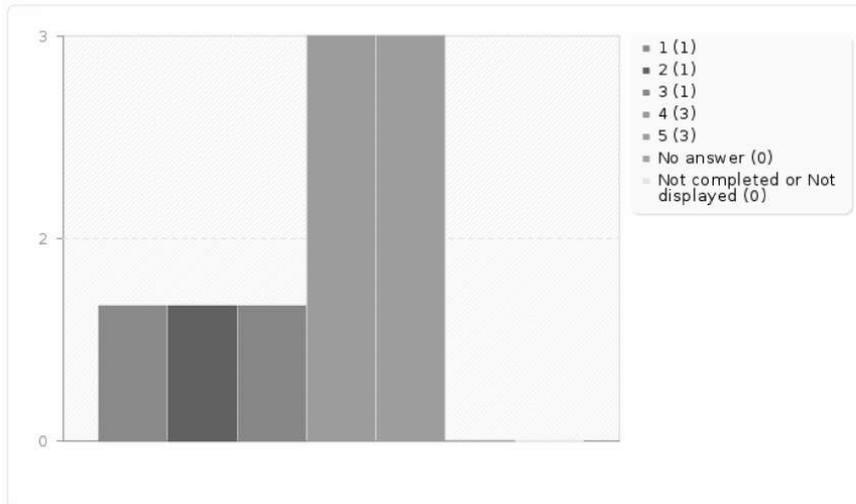
Answer	Count	Percentage	Sum
1 (1)	1	11.11%	22.22%
2 (2)	1	11.11%	
3 (3)	1	11.11%	11.11%
4 (4)	3	33.33%	
5 (5)	3	33.33%	66.67%
No answer	0	0.00%	
Not completed or Not displayed	0	0.00%	
Arithmetic mean	3,67		
Standard deviation	1,41		
Sum (Answers)	9	100.00%	100.00%
Number of cases	9	100.00%	



Quick statistics
Survey 189194 'Persuasive Personal Health Aid'

Field summary for PersuasiveSASSI(Habitability)

General Quality: How much do you agree with the following statements? [I always knew what to say to the system]



Quick statistics
Survey 189194 'Persuasive Personal Health Aid'

Field summary for PersuasiveSASSI(Speed)

General Quality: How much do you agree with the following statements? [The interaction with the system is fast]

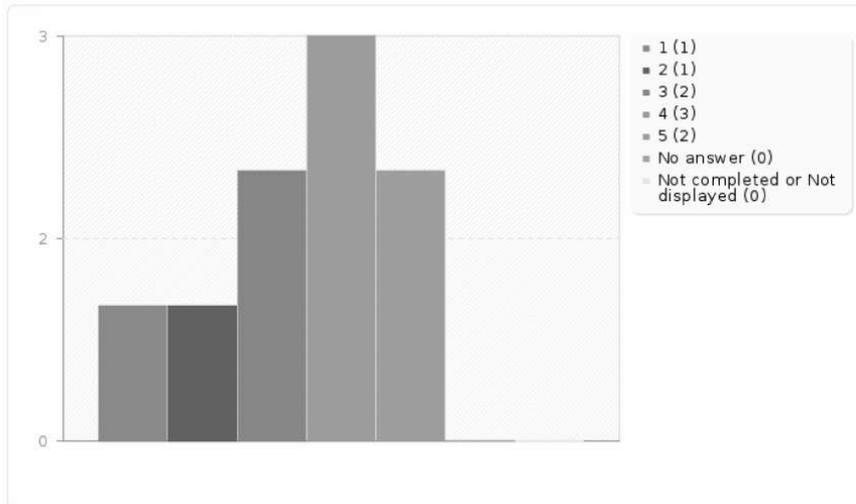
Answer	Count	Percentage	Sum
1 (1)	1	11.11%	22.22%
2 (2)	1	11.11%	
3 (3)	2	22.22%	22.22%
4 (4)	3	33.33%	
5 (5)	2	22.22%	55.56%
No answer	0	0.00%	
Not completed or Not displayed	0	0.00%	
Arithmetic mean	3,44		
Standard deviation	1,33		
Sum (Answers)	9	100.00%	100.00%
Number of cases	9	100.00%	



Quick statistics
Survey 189194 'Persuasive Personal Health Aid'

Field summary for PersuasiveSASSI(Speed)

General Quality: How much do you agree with the following statements? [The interaction with the system is fast]



Quick statistics
Survey 189194 'Persuasive Personal Health Aid'

Field summary for PersuasiveSpecific(PS003)

Persuasion Quality: How much do you agree with the following statements? [The system suggestions were helpful]

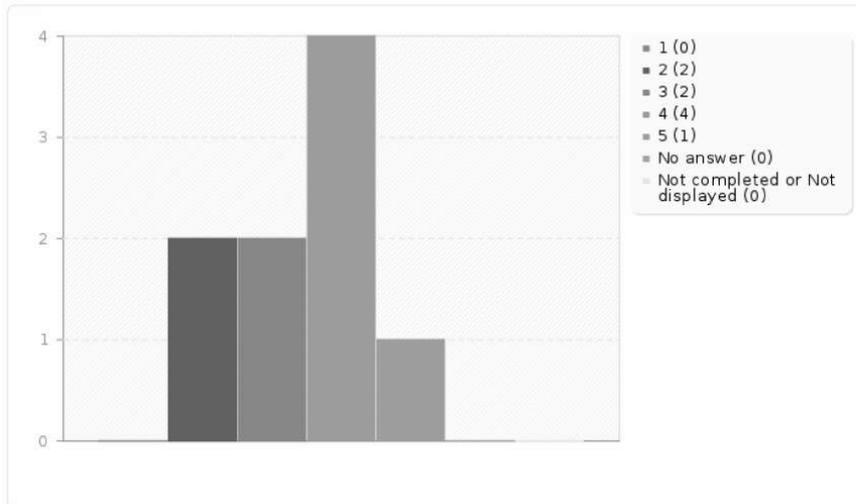
Answer	Count	Percentage	Sum
1 (1)	0	0,00%	22,22%
2 (2)	2	22,22%	
3 (3)	2	22,22%	22,22%
4 (4)	4	44,44%	
5 (5)	1	11,11%	55,56%
No answer	0	0,00%	
Not completed or Not displayed	0	0,00%	
Arithmetic mean	3,44		
Standard deviation	1,01		
Sum (Answers)	9	100,00%	100,00%
Number of cases	9	100,00%	



Quick statistics
Survey 189194 'Persuasive Personal Health Aid'

Field summary for PersuasiveSpecific(PS003)

Persuasion Quality: How much do you agree with the following statements? [The system suggestions were helpful]



Quick statistics
Survey 189194 'Persuasive Personal Health Aid'

Field summary for PersuasiveSpecific(PS001)

Persuasion Quality: How much do you agree with the following statements? [Using the system was motivating]

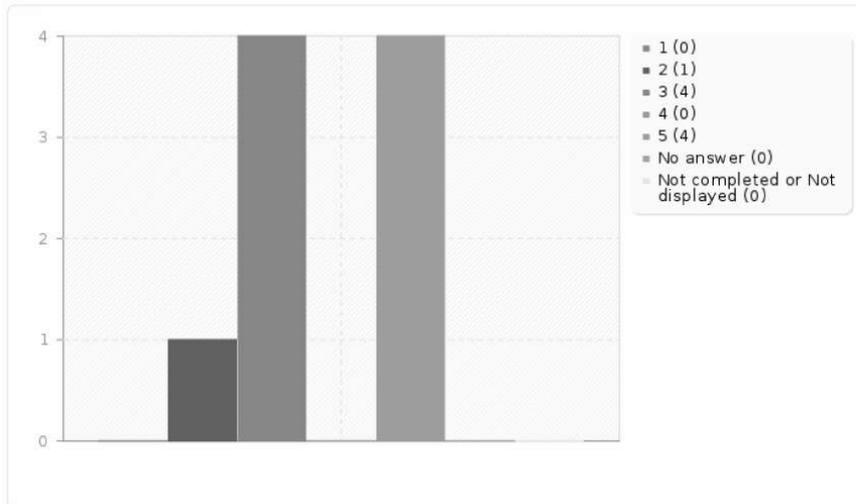
Answer	Count	Percentage	Sum
1 (1)	0	0.00%	11.11%
2 (2)	1	11.11%	
3 (3)	4	44.44%	44.44%
4 (4)	0	0.00%	
5 (5)	4	44.44%	44.44%
No answer	0	0.00%	
Not completed or Not displayed	0	0.00%	
Arithmetic mean	3.78		
Standard deviation	1.2		
Sum (Answers)	9	100.00%	100.00%
Number of cases	9	100.00%	



Quick statistics
Survey 189194 'Persuasive Personal Health Aid'

Field summary for PersuasiveSpecific(PS001)

Persuasion Quality: How much do you agree with the following statements? [Using the system was motivating]



Quick statistics
Survey 189194 'Persuasive Personal Health Aid'

Field summary for PersuasiveSpecific(PS002)

Persuasion Quality: How much do you agree with the following statements? [I like the general idea of a motivational system for my personal use]

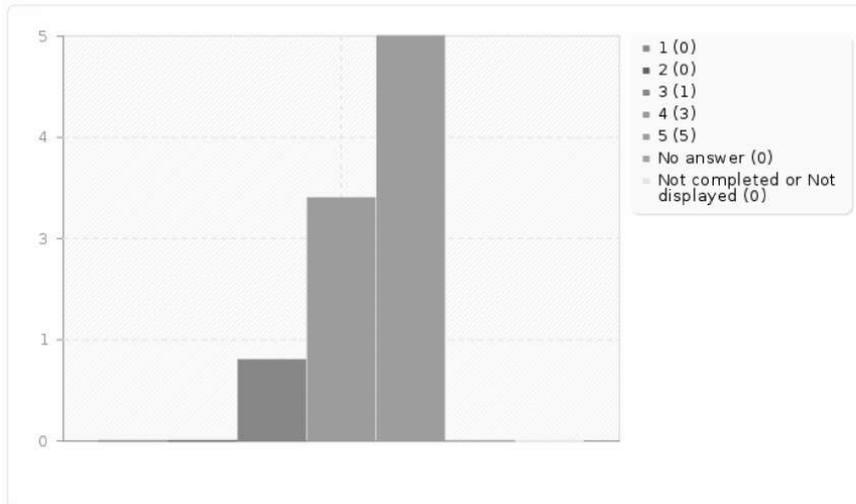
Answer	Count	Percentage	Sum
1 (1)	0	0.00%	0.00%
2 (2)	0	0.00%	
3 (3)	1	11.11%	11.11%
4 (4)	3	33.33%	
5 (5)	5	55.56%	88.89%
No answer	0	0.00%	
Not completed or Not displayed	0	0.00%	
Arithmetic mean	4.44		
Standard deviation	0.73		
Sum (Answers)	9	100.00%	100.00%
Number of cases	9	100.00%	



Quick statistics
Survey 189194 'Persuasive Personal Health Aid'

Field summary for PersuasiveSpecific(PS002)

Persuasion Quality: How much do you agree with the following statements? [I like the general idea of a motivational system for my personal use]



Quick statistics
Survey 189194 'Persuasive Personal Health Aid'

Field summary for PersuasiveNegative(PN003)

Persuasion Flaws: How much do you agree with the following statements? [The phrases to motivate me felt out of place or unfitting]

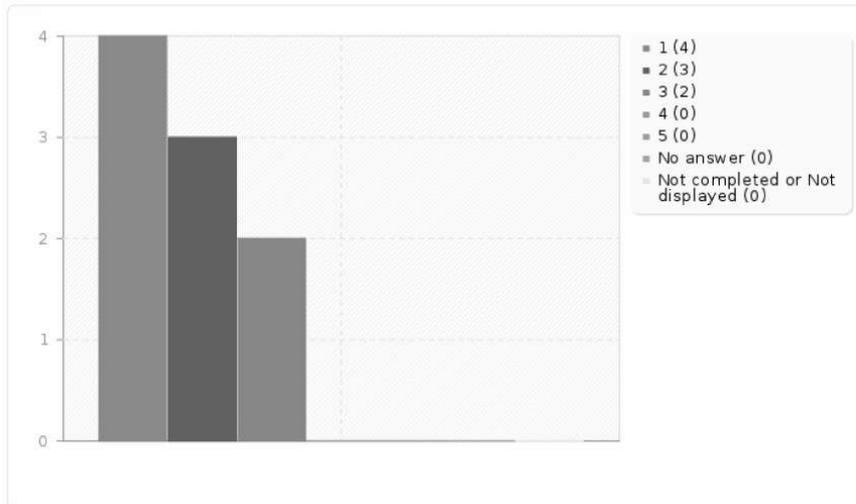
Answer	Count	Percentage	Sum
1 (1)	4	44.44%	77.78%
2 (2)	3	33.33%	
3 (3)	2	22.22%	22.22%
4 (4)	0	0.00%	
5 (5)	0	0.00%	0.00%
No answer	0	0.00%	
Not completed or Not displayed	0	0.00%	
Arithmetic mean	1.78		
Standard deviation	0.83		
Sum (Answers)	9	100.00%	100.00%
Number of cases	9	100.00%	



Quick statistics
Survey 189194 'Persuasive Personal Health Aid'

Field summary for PersuasiveNegative(PN003)

Persuasion Flaws: How much do you agree with the following statements? [The phrases to motivate me felt out of place or unfitting]



Quick statistics
Survey 189194 'Persuasive Personal Health Aid'

Field summary for PersuasiveNegative(PN002)

Persuasion Flaws: How much do you agree with the following statements? [The system's manner of speech was rude]

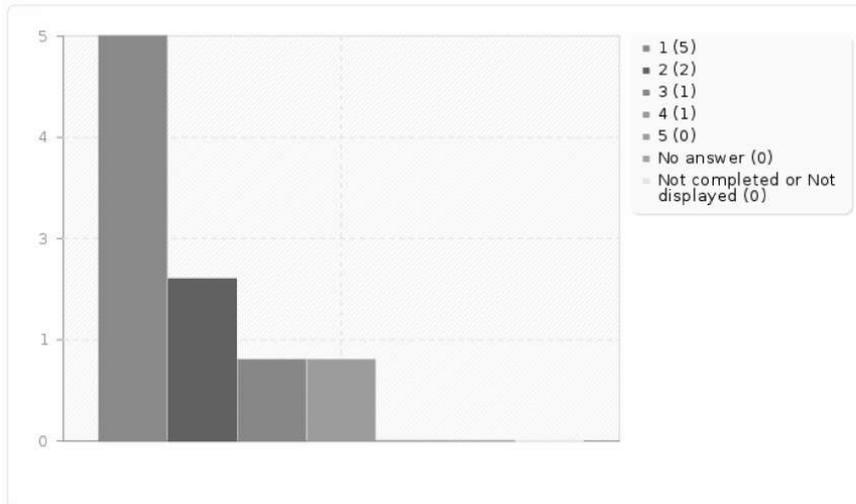
Answer	Count	Percentage	Sum
1 (1)	5	55.56%	77.78%
2 (2)	2	22.22%	
3 (3)	1	11.11%	11.11%
4 (4)	1	11.11%	11.11%
5 (5)	0	0.00%	11.11%
No answer	0	0.00%	
Not completed or Not displayed	0	0.00%	
Arithmetic mean	1.78		
Standard deviation	1.09		
Sum (Answers)	9	100.00%	100.00%
Number of cases	9	100.00%	



Quick statistics
Survey 189194 'Persuasive Personal Health Aid'

Field summary for PersuasiveNegative(PN002)

Persuasion Flaws: How much do you agree with the following statements? [The system's manner of speech was rude]



Quick statistics
Survey 189194 'Persuasive Personal Health Aid'

Field summary for PersuasiveNegative(PN001)

Persuasion Flaws: How much do you agree with the following statements? [I felt patronized by the system]

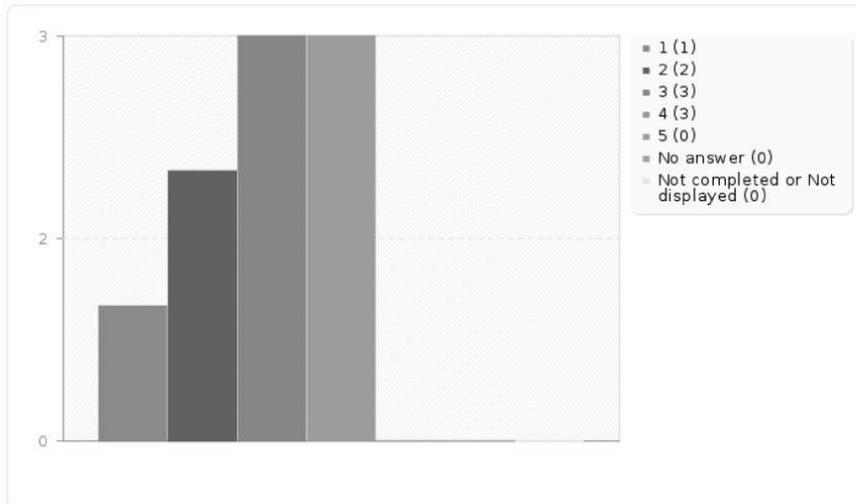
Answer	Count	Percentage	Sum
1 (1)	1	11.11%	33.33%
2 (2)	2	22.22%	
3 (3)	3	33.33%	33.33%
4 (4)	3	33.33%	
5 (5)	0	0.00%	33.33%
No answer	0	0.00%	
Not completed or Not displayed	0	0.00%	
Arithmetic mean	2.89		
Standard deviation	1.05		
Sum (Answers)	9	100.00%	100.00%
Number of cases	9	100.00%	



Quick statistics
Survey 189194 'Persuasive Personal Health Aid'

Field summary for PersuasiveNegative(PN001)

Persuasion Flaws: How much do you agree with the following statements? [I felt patronized by the system]



Quick statistics
Survey 189194 'Persuasive Personal Health Aid'

Field summary for PersuasiveFreeText

Do you have any further comments about usage experience?

Answer	Count	Percentage
Answer	5	55.56%
No answer	4	44.44%
Not completed or Not displayed	0	0.00%

ID	Response
2	Speech recognition was terrible. Verbal GUI navigation was well done.
4	it forgot what it was asking
6	too small vocabulary
6	synthesizer is horrible, overlaps, not natural
10	system is not pushing enough
11	Needs a "maybe"



Quick statistics
Survey 189194 'Persuasive Personal Health Aid'

Field summary for SchedulerScore

What is your opinion of the tested functionality? Please give a score.

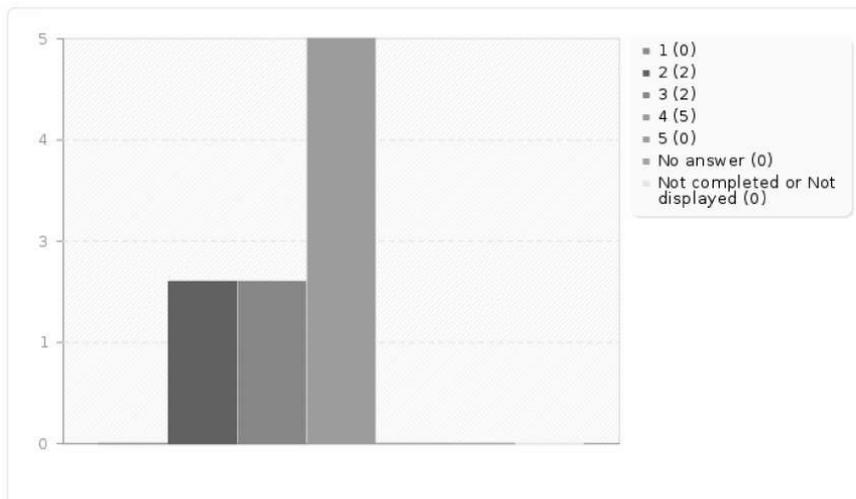
Answer	Count	Percentage	Sum
1 (1)	0	0.00%	22.22%
2 (2)	2	22.22%	22.22%
3 (3)	2	22.22%	22.22%
4 (4)	5	55.56%	55.56%
5 (5)	0	0.00%	55.56%
No answer	0	0.00%	
Not completed or Not displayed	0	0.00%	
Arithmetic mean	3,33		
Standard deviation	0,87		
Sum (Answers)	9	100.00%	100.00%
Number of cases	9	100.00%	



Quick statistics
Survey 189194 'Persuasive Personal Health Aid'

Field summary for SchedulerScore

What is your opinion of the tested functionality? Please give a score.





Field summary for SchedulerSASSI(ResponseAccuracy)

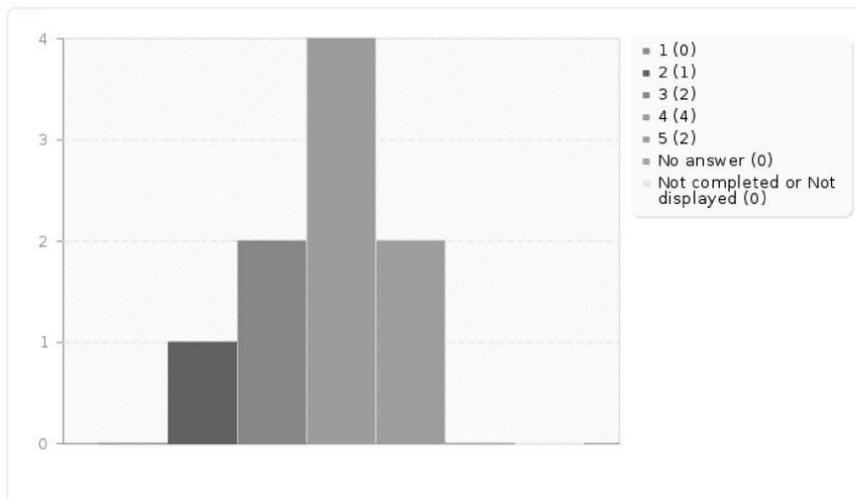
General Quality: How much do you agree with the following statements? [The system is accurate]

Answer	Count	Percentage	Sum
1 (1)	0	0.00%	11.11%
2 (2)	1	11.11%	
3 (3)	2	22.22%	22.22%
4 (4)	4	44.44%	
5 (5)	2	22.22%	66.67%
No answer	0	0.00%	
Not completed or Not displayed	0	0.00%	
Arithmetic mean	3.78		
Standard deviation	0.97		
Sum (Answers)	9	100.00%	100.00%
Number of cases	9	100.00%	



Field summary for SchedulerSASSI(ResponseAccuracy)

General Quality: How much do you agree with the following statements? [The system is accurate]





Quick statistics
Survey 189194 'Persuasive Personal Health Aid'

Field summary for SchedulerSASSI(Likeability)

General Quality: How much do you agree with the following statements? [The system is useful]

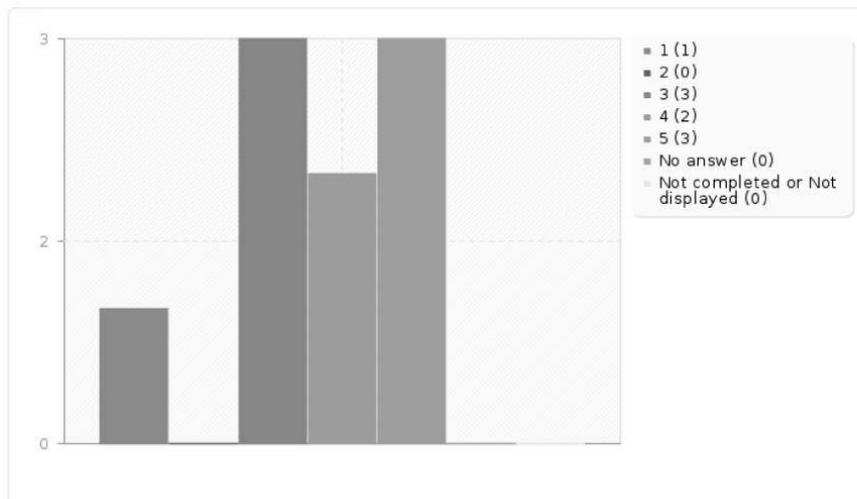
Answer	Count	Percentage	Sum
1 (1)	1	11.11%	11.11%
2 (2)	0	0.00%	
3 (3)	3	33.33%	33.33%
4 (4)	2	22.22%	
5 (5)	3	33.33%	55.56%
No answer	0	0.00%	
Not completed or Not displayed	0	0.00%	
Arithmetic mean	3.67		
Standard deviation	1.32		
Sum (Answers)	9	100.00%	100.00%
Number of cases	9	100.00%	



Quick statistics
Survey 189194 'Persuasive Personal Health Aid'

Field summary for SchedulerSASSI(Likeability)

General Quality: How much do you agree with the following statements? [The system is useful]





Field summary for SchedulerSASSI(CognitiveDemand)

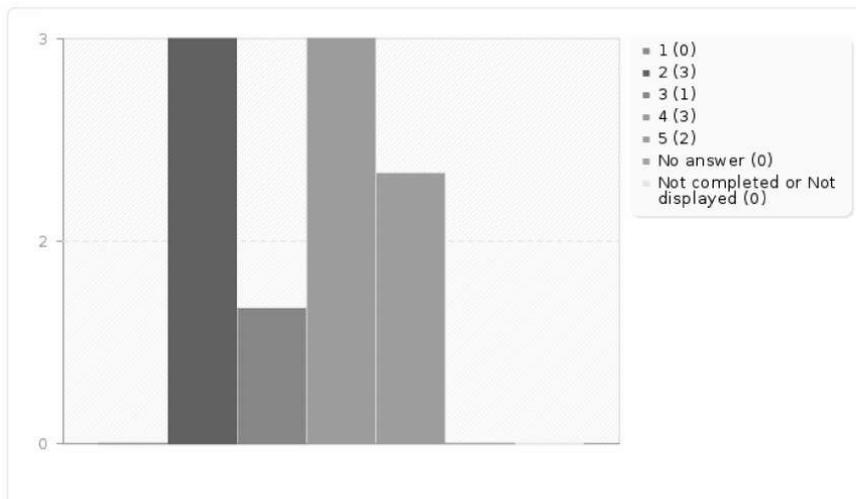
General Quality: How much do you agree with the following statements? [I felt confident using the system]

Answer	Count	Percentage	Sum
1 (1)	0	0.00%	33.33%
2 (2)	3	33.33%	
3 (3)	1	11.11%	11.11%
4 (4)	3	33.33%	
5 (5)	2	22.22%	55.56%
No answer	0	0.00%	
Not completed or Not displayed	0	0.00%	
Arithmetic mean	3,44		
Standard deviation	1,24		
Sum (Answers)	9	100.00%	100.00%
Number of cases	9	100.00%	



Field summary for SchedulerSASSI(CognitiveDemand)

General Quality: How much do you agree with the following statements? [I felt confident using the system]





Quick statistics
Survey 189194 'Persuasive Personal Health Aid'

Field summary for SchedulerSASSI(Annoyance)

General Quality: How much do you agree with the following statements? [The interaction with the system is not repetitive]

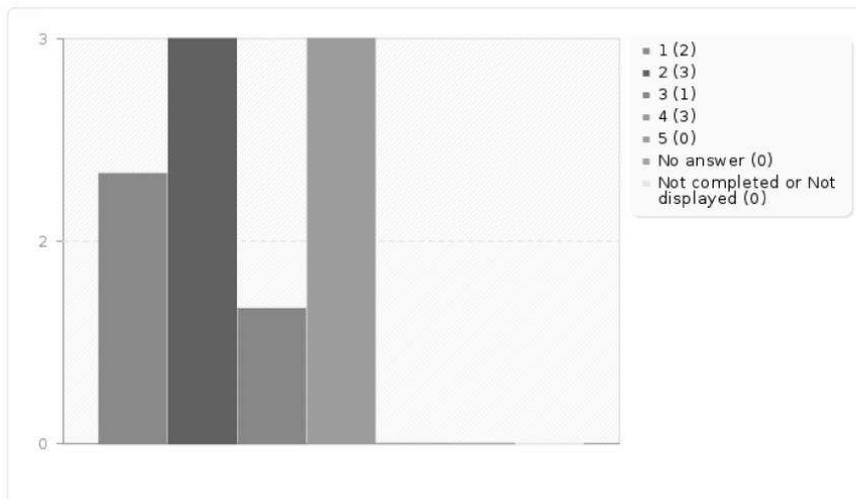
Answer	Count	Percentage	Sum
1 (1)	2	22.22%	55.56%
2 (2)	3	33.33%	
3 (3)	1	11.11%	11.11%
4 (4)	3	33.33%	
5 (5)	0	0.00%	33.33%
No answer	0	0.00%	
Not completed or Not displayed	0	0.00%	
Arithmetic mean	2,56		
Standard deviation	1,24		
Sum (Answers)	9	100.00%	100.00%
Number of cases	9	100.00%	



Quick statistics
Survey 189194 'Persuasive Personal Health Aid'

Field summary for SchedulerSASSI(Annoyance)

General Quality: How much do you agree with the following statements? [The interaction with the system is not repetitive]





Quick statistics
Survey 189194 'Persuasive Personal Health Aid'

Field summary for SchedulerSASSI(Habitability)

General Quality: How much do you agree with the following statements? [I always knew what to say to the system]

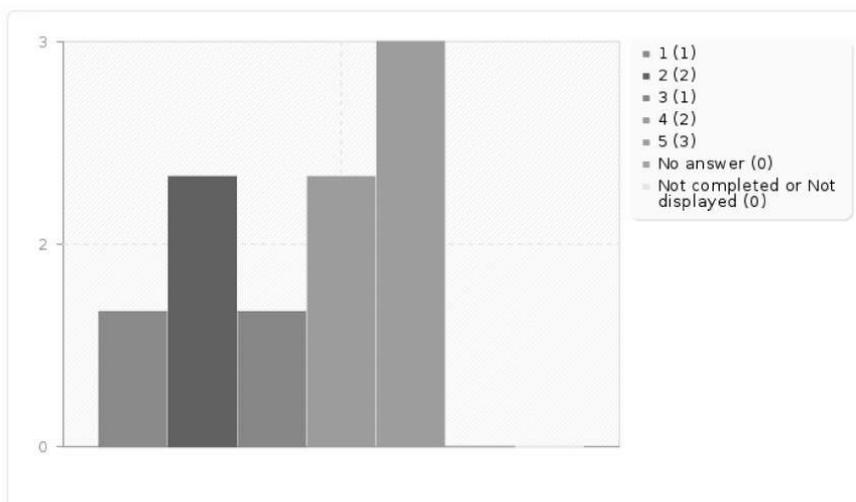
Answer	Count	Percentage	Sum
1 (1)	1	11.11%	33.33%
2 (2)	2	22.22%	
3 (3)	1	11.11%	11.11%
4 (4)	2	22.22%	
5 (5)	3	33.33%	55.56%
No answer	0	0.00%	
Not completed or Not displayed	0	0.00%	
Arithmetic mean	3,44		
Standard deviation	1,51		
Sum (Answers)	9	100.00%	100.00%
Number of cases	9	100.00%	



Quick statistics
Survey 189194 'Persuasive Personal Health Aid'

Field summary for SchedulerSASSI(Habitability)

General Quality: How much do you agree with the following statements? [I always knew what to say to the system]





Quick statistics
Survey 189194 'Persuasive Personal Health Aid'

Field summary for SchedulerSASSI(Speed)

General Quality: How much do you agree with the following statements? [The interaction with the system is fast]

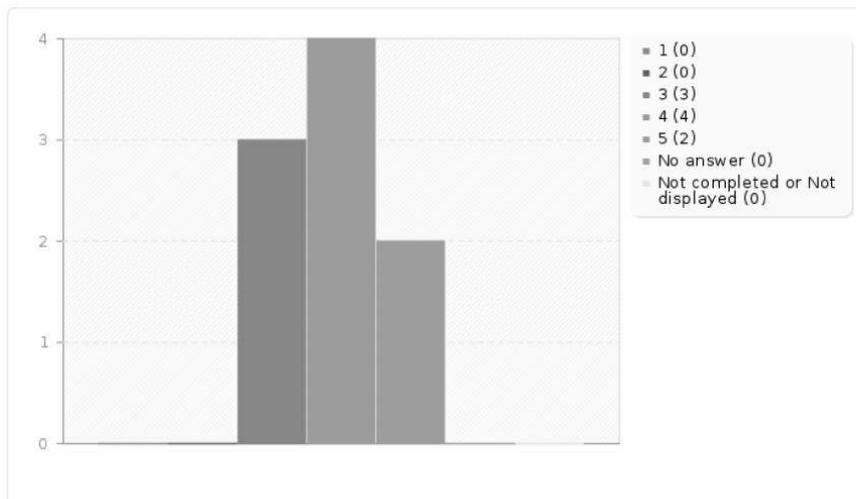
Answer	Count	Percentage	Sum
1 (1)	0	0.00%	0.00%
2 (2)	0	0.00%	
3 (3)	3	33.33%	33.33%
4 (4)	4	44.44%	
5 (5)	2	22.22%	66.67%
No answer	0	0.00%	
Not completed or Not displayed	0	0.00%	
Arithmetic mean	3.89		
Standard deviation	0.78		
Sum (Answers)	9	100.00%	100.00%
Number of cases	9	100.00%	



Quick statistics
Survey 189194 'Persuasive Personal Health Aid'

Field summary for SchedulerSASSI(Speed)

General Quality: How much do you agree with the following statements? [The interaction with the system is fast]





Quick statistics
Survey 189194 'Persuasive Personal Health Aid'

Field summary for SchedulerFreeText

Do you have any further comments about usage experience?

Answer	Count	Percentage
Answer	1	11.11%
No answer	8	88.89%
Not completed or Not displayed	0	0.00%

ID	Response
8	it would be useful to have the time from 0 to 24 or with pm-am



Quick statistics
Survey 189194 'Persuasive Personal Health Aid'

Field summary for OverallScore

What is your opinion of the system as a whole? Please give a score.

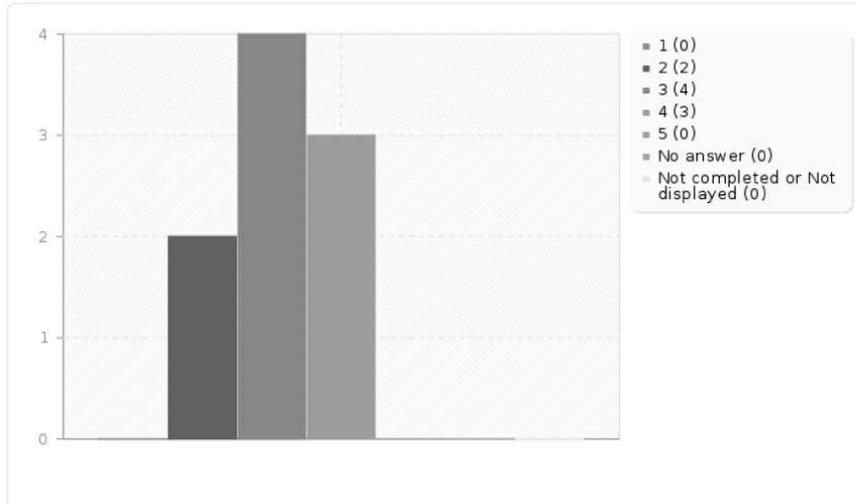
Answer	Count	Percentage	Sum
1 (1)	0	0.00%	22.22%
2 (2)	2	22.22%	
3 (3)	4	44.44%	44.44%
4 (4)	3	33.33%	
5 (5)	0	0.00%	33.33%
No answer	0	0.00%	
Not completed or Not displayed	0	0.00%	
Arithmetic mean	3.11		
Standard deviation	0.78		
Sum (Answers)	9	100.00%	100.00%
Number of cases	9	100.00%	



Quick statistics
Survey 189194 'Persuasive Personal Health Aid'

Field summary for OverallScore

What is your opinion of the system as a whole? Please give a score.



Quick statistics
Survey 189194 'Persuasive Personal Health Aid'

Field summary for OverallUsage(OU01)

How much do you agree with the following statements? [I would use this system in everyday life]

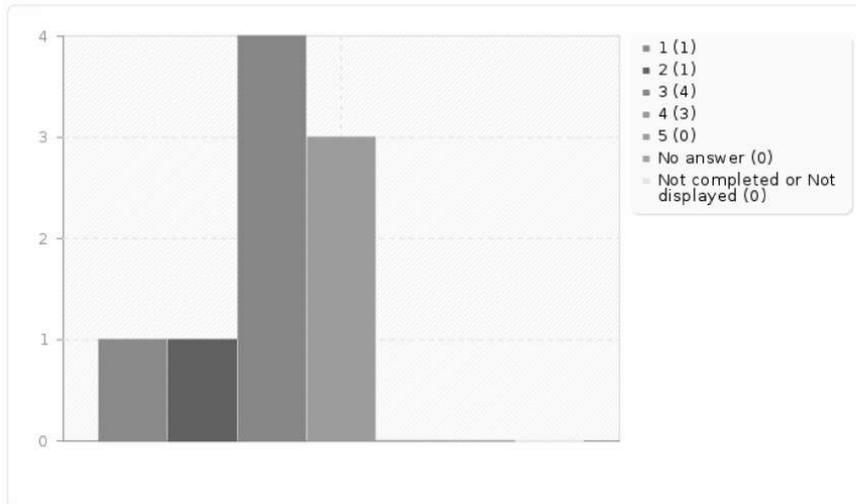
Answer	Count	Percentage	Sum
1 (1)	1	11.11%	22.22%
2 (2)	1	11.11%	
3 (3)	4	44.44%	44.44%
4 (4)	3	33.33%	
5 (5)	0	0.00%	33.33%
No answer	0	0.00%	
Not completed or Not displayed	0	0.00%	
Arithmetic mean	3		
Standard deviation	1		
Sum (Answers)	9	100.00%	100.00%
Number of cases	9	100.00%	



Quick statistics
Survey 189194 'Persuasive Personal Health Aid'

Field summary for OverallUsage(OU01)

How much do you agree with the following statements? [I would use this system in everyday life]



Quick statistics
Survey 189194 'Persuasive Personal Health Aid'

Field summary for OverallUsage(OU02)

How much do you agree with the following statements? [Generally a system of this kind could be useful]

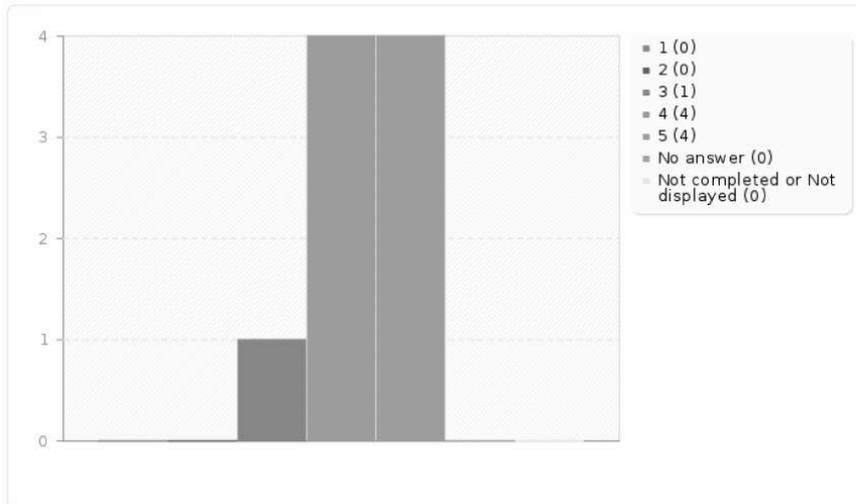
Answer	Count	Percentage	Sum
1 (1)	0	0.00%	0.00%
2 (2)	0	0.00%	
3 (3)	1	11.11%	11.11%
4 (4)	4	44.44%	
5 (5)	4	44.44%	88.89%
No answer	0	0.00%	
Not completed or Not displayed	0	0.00%	
Arithmetic mean	4,33		
Standard deviation	0,71		
Sum (Answers)	9	100.00%	100.00%
Number of cases	9	100.00%	



Quick statistics
Survey 189194 'Persuasive Personal Health Aid'

Field summary for OverallUsage(OU02)

How much do you agree with the following statements? [Generally a system of this kind could be useful]



Quick statistics
Survey 189194 'Persuasive Personal Health Aid'

Field summary for OverallMissingFeatur

Where there features you expected that the system didn't offer?

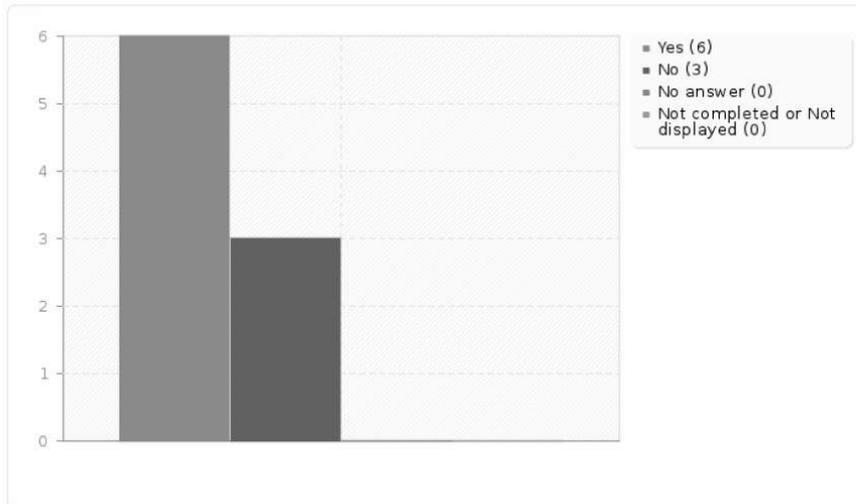
Answer	Count	Percentage
Yes (Y)	6	66.67%
No (N)	3	33.33%
No answer	0	0.00%
Not completed or Not displayed	0	0.00%



Quick statistics
Survey 189194 'Persuasive Personal Health Aid'

Field summary for OverallMissingFeatur

Where there features you expected that the system didn't offer?



Quick statistics
Survey 189194 'Persuasive Personal Health Aid'

Field summary for OverallMissingInput

What kind of features?

Answer	Count	Percentage
Answer	6	66.67%
No answer	0	0.00%
Not completed or Not displayed	3	33.33%

ID	Response
2	The system didn't allow to remotely order beer.
4	relative dates it should understand more sentence variants
5	relative dates ;) more pressure while motivating me
8	-suggest sport excersices -am/pm time support -next week etc.
9	If the relative time schedule can work
11	-Variety of other possibilites -tell me when i'm free -pospone apointments



Quick statistics
Survey 189194 'Persuasive Personal Health Aid'

Field summary for OverallComments

Do you have any further comments about usage of the system as a whole or about this Evaluation?

Answer	Count	Percentage
Answer	4	44,44%
No answer	5	55,56%
Not completed or Not displayed	0	0,00%

ID	Response
2	Plz use a speech recognition engine the next time that is actually able to understand the words that I speak more often than one out of three times.
9	I felt like the pause between 2 sentences from system is too short, sometimes heard overlapping.
10	sound output is not natural
11	-difficult to understand because of the changing pitch