In-Car Multi-Domain Spoken Dialogs: A Wizard of Oz Study

Sven Reichel, Ute Ehrlich, André Berton
Speech Dialogue Systems
Daimler AG
Ulm, Germany
{sven.reichel, ute.ehrlich, andre.berton}@daimler.com

Michael Weber
Institute of Media Informatics
Ulm University
Germany
michael.weber@uni-ulm.de

Abstract

Mobile Internet access via smartphones puts demands on in-car infotainment systems, as more and more drivers like to access the Internet while driving. Spoken dialog systems support the user by less distracting interaction than visual/haptic-based dialog systems. To develop an intuitive and usable spoken dialog system, an extensive analysis of the interaction concept is necessary. We conducted a Wizard of Oz study to investigate how users will carry out tasks which involve multiple applications in a speech-only, user-initiative infotainment system while driving. Results show that users are not aware of different applications and use anaphoric expressions in task switches. Speaking styles vary and depend on type of task and dialog state. Users interact efficiently and provide multiple semantic concepts in one utterance. This sets high demands for future spoken dialog systems.

1 Introduction

The acceptance of smartphones is a success story. These devices allow people to access the Internet nearly anywhere at anytime. While driving, using a smartphone is prohibited in many countries as it distracts the driver. Regardless of this prohibition, people use their smartphone and cause severe injuries (National Highway Traffic Safety Administration (NHTSA), 2013). In order to reduce driver distraction, it is necessary to integrate the smartphone's functionality safely into in-car infotainment systems. Since hands and eyes are involved in driving, a natural and intuitive speech-based interface increases road safety (Maciej and Vollrath, 2009). There are already infotainment systems with Internet applications like e.g. weather, music streaming, gas prices, news, and restaurant search. However, not all of them can be controlled by natural speech.

In systems based on graphic and haptic modality, the functionality is often grouped into various applications. Among other things, this is due to the limited screen size. The user has to start an application and select the desired functionality. A natural speech interface does not require a fragmentation of functionalities into applications, as people can express complex commands by speech. In single-application tasks, such as calling someone, natural speech interfaces are established and proven. However, users often encounter complex tasks, which involve more than one application. For example, while hearing the news about a new music album, the driver might like to start listening to this album via Internet radio. Spoken language allows humans to express a request such as "Play this album" easily, since the meaning is clear. However, will drivers also use this kind of interaction while using an in-car **s**poken **d**ialog **s**ystem (SDS)? Or is the mental model of application interaction schema dominant in human-computer interaction? In a user experiment, we confront drivers with multi-domain tasks, to observe how they interact.

While interacting with an SDS, one crucial problem for users is to know which utterances the system is able to understand. People use different approaches to solve this problem, for example by reading the manual, using on-screen help, or relying on their experiences. In multi-domain dialog systems, utterances can be quite complex, thus remembering all utterances from the manual or displaying them on screen would not be possible. As a result, users have to rely on their experience in communications to know what to say. Thus, an advanced SDS needs to understand what a user would naturally say in this situation to execute a certain task.

Proceedings of the of the EACL 2014 Workshop on Dialogue in Motion (DM), pages 1–9,
Gothenburg, Sweden, April 26-30 2014.

In this paper, we present results from a **W**izard **o**f **O**z (WoZ) experiment on multi-domain interaction with an in-car SDS. The goal of this study is to build a corpus and analyze it according to application awareness, speaking styles, anaphoric references, and efficiency. Our results provide a detailed insight how drivers start multi-application tasks and switch between applications by speech. This will answer the question whether they are primed to application-based-interaction or use a natural approach known from human-human-communication. The results will be used to design grammars or language models for working prototypes, which establish a basis for real user tests. Furthermore, we provide guidelines for multi-domain SDSs.

The remainder of this paper is structured as follows: Section 2 provides an overview of other studies in this context. Section 3 describes the domain for the user experiment which is presented in Section 4. Data analysis methods are defined in Section 5. We present the results in Section 6 and discuss them in Section 7. Finally, we conclude and give guidelines for multi-domain SDSs in Section 8.

2 Related Work

Many studies exist which evaluate SDSs concerning performance, usability, and driver distraction (a good overview provides Ei-Wen Lo and Green (2013)). Usually, participants are asked to complete a task, while driving in a simulated environment or in real traffic. Geutner et al. (2002), for example, showed that a virtual co-driver contributes to ease of use with little distraction effects. In their WoZ experiment, natural language was preferred to command-and-control input. However, no in-depth analysis of user utterances is presented. Cheng et al. (2004) performed an analysis of natural user utterances. They observed that drivers, occupied in a driving task, use disfluent and distracted speech and react differently than by concentrating on the speech interaction task. None of the studies provide in-depth analysis of multi-domain tasks, as our work does.

Multi-domain SDS exist like e.g. SmartKom (Reithinger et al., 2003) or CHAT (Weng et al., 2007). They presented complex systems with many functionalities, however, they do not evaluate subtask switching from users' point of view. In CHAT, the implicit application switch was even disabled due to "extra burden on the system". Domain switches are analyzed in human-human communication as e.g. in Villing et al. (2008). However, people interact differently with a system than with a human. Even in human-computer communication, speaking styles differ depending on type of task, as (Hofmann et al., 2012) showed in a web-based user study. In order to develop an intuitive multi-application SDS, it is necessary to analyze how users interact in a driving situation by completing tasks across different domains.

3 User Tasks

In a user experiment it is crucial to set real tasks for users, since artificial tasks will be hard to remember and can reduce their attention. We analyzed current in-car infotainment systems with Internet access and derived eight multi-domain tasks from their functionality (see Table 1). The subtasks were classified according to Kellar et al. (2006)'s web information classification schema in information seeking (Inf), information exchange, and information maintenance. Since information maintenance is not a strong automotive use case, these tasks were grouped together with information exchange. We call them action subtasks (Act) as they initiate an action of the infotainment system (e.g. "turn on the radio").

No	App 1	App 2	App3
1	POI Search	Restaurant	Call
2	Knowledge	Ski Weather	Navigation
3	Weather	Hotel Search	Address book
4	Play Artist	News Search	Forward by eMail
5	Navigation	Restaurant	Save as Favorite
6	News Search	Play Artist	Share on Facebook
7	News Search	Knowledge	Convert Currency
8	Navigation	Gas Prices	Status Gas Tank

Table 1: Multi-application user tasks.

Since only few use cases involve more than three applications, every user task is a story of three subtasks. In task number 5 for example, a user has to start a subtask, which navigates him to Berlin. Then he would like to search an Italian restaurant at the destination. Finally, he adds the selected restaurant to his favorites. The focus is on task entry and on subtask switch, thus the subtasks require only two to four semantic concepts (like *Berlin* or *Italian restaurant*). One of these concepts is a reference to the previous subtask (like *at the destination* or *the selected restaurant*) to ensure a natural cross-application dialog flow. After the system's response for one subtask the user has to initiate the next subtask to complete his task.

4 User Experiment

Developing an SDS means specifying a grammar or training statistical language models for speech recognition. These steps precede any real user test. In system-initiated dialogs, with a few possible utterances, specifying a grammar is feasible. However, in strictly user-initiative dialogs with multiple applications, this is rather complicated. A WoZ study does not require to develop speech recognition and understanding as this is performed by a human. Analyzing the user utterances of a WoZ experiment provides a detailed view of how a user will interact with the SDS. This helps in designing spoken dialogs and specifying grammars and/or training language models for further evaluations (Fraser and Gilbert, 1991; Glass et al., 2000).

Interaction schemes of people vary among each other and depend on age, personality, experience, context, and many more. It is essential to conduct a user study with people who might use the SDS later on. A study by the NHTSA (National Highway Traffic Safety Administration (NHTSA), 2013) showed that in 2011 73% of the drivers involved in fatal crashes due to cell phone use, were less than 40 years old. For this reason, our study considers drivers between 18 and 40 years who are technically affine and are likely to buy a car equipped with an infotainment system with Internet access.

4.1 Experimental Set-Up

When designing a user interaction experiment, it is important that it takes place in a real environment. As driving on a real road is dangerous, we used a fixed-base driving simulator in a laboratory. In front of the car, a screen covers the driver's field of view (see Figure 1). Steering and pedal signals are picked from the car's CAN bus. It is important that the user assumes he is interacting with a computer as "human-human interactions are not the same as human-computer interactions" (Fraser and Gilbert, 1991). The wizard, a person in charge of the experiment, was located behind the car and mouse clicks or any other interaction of the wizard was not audible in the car. To ensure a consistent behavior of the wizard, we used SUEDE (Klemmer et al., 2000) to define the dialog, which also provides an interface for the wizard. SUEDE defines a dialog in a state machine, in which the system prompts are states and user inputs are edges between them. The content of system prompts was synthesized with NUANCE Vocalizer Expressive[1] version 1.2.1 (Voice: anna.full). During the experiment, after each user input the wizards clicks the corresponding edge and SUEDE plays the next prompt. All user utterances are recorded as audio files.

Figure 1: Experimental Set-Up

4.2 Experiment Design

Infotainment systems in cars are used while driving. This means the user cannot concentrate on the infotainment system only, but also has to focus on the road. According to multiple resource theory, the human's performance is reduced when human resources overlap (Wickens, 2008). In a dual-task scenario, like using the infotainment system while driving, multiple resources are allocated and may interfere. Considering this issue, we use a driving task to keep the participants occupied while they interact with the SDS. This allows us to observe user utterances in a stressful situation.

Infotainment systems in cars are often equipped with large displays providing visual and haptic interaction. These kinds of interaction compete for human resources which are needed for driving. This results in driver distraction, especially in demanding secondary tasks (Young and Regan, 2007). Furthermore, a visual interface can also influence the communication of users (e.g. they utter visual terms). As we intent to study how a user interacts naturally with a multi-domain SDS, we avoid priming effects by not using any visual interface.

4.2.1 Primary Task: Driving Simulator

One major requirement for the driving task is to keep the driver occupied at a constant level all the time. Otherwise, we would not be able to analyze user utterances on a fine-grained level.

[1] http://www.nuance.com/for-business/mobile-solutions/vocalizer-expressive/index.htm

Therefore, we used the **Con**tinuous **T**racking and **Re**action (ConTRe) task (Mahr et al., 2012) which allows controlled driving conditions. It consists of a steering and a reaction task, which require operating the steering wheel and pedals. In the steering task, a yellow cylinder moves unpredictable right and left at a constant distance from the driver and the driver must always steer towards it. This is similar to driving on a curved road. Sometimes a driver needs to react to sudden events to prevent an accident. For this a traffic light shows randomly red and green and requires the driver to push the throttle or brake pedal. The movement of the yellow cylinder and the appearance of the stop light can be controlled by manipulating control variables. The "hard driving setting" from Mahr et al. (2012) was used in this study.

4.2.2 Secondary Task: cross application tasks with speech interaction

As described in Section 3, a task consists of three subtasks and each subtask requires two to four semantic concepts. For a user it is possible to insert multiple concepts at once:

U: "Search an Italian restaurant at the destination"

or as single utterances in a dialog:

U: "Search an Italian restaurant"

S: "Where do you search an Italian restaurant?"

U: "At my destination"

For all possible combinations prompts were specified. SUEDE provides a GUI for the wizard to select which semantic concept a user input contains. Dependent on the selection, either another concept is requested or the answer is provided. Furthermore, a user input can optionally contain a verb expressing what the system should do. For example, if users say "Italian Restaurant" the reaction is the same as they would say "Search an Italian restaurant".

The user has basically two options to select or switch to an application. Either an explicit selection such as:

U: "Open restaurant application"

S: "Restaurant, what do you want?"

or an implicit selection such as:

U: "Search an Italian restaurant"

By using an explicit selection, users assume they have to set the context to a specific application. After that, they can use the functionality of this application. This is a common interaction schema for visual-based infotainment systems or smartphones, as they cluster their functionality into various applications. An implicit selection is rather like current personal assistants interact, as they do not cluster their functionality. Implicit selection facilitates the interaction for users since they can get an answer right away. After the user provided the necessary input for one subtask, the system responds for example:

S: "There is one Italian restaurant: Pizzeria San Marco."

Then the user needs to initiate an application switch to proceed with his task.

A system enabling user-initiated dialogs cannot always understand the user correctly. Especially in implicit selection, the language models increase, and thus recognition as well as understanding is error prone (Carstensen et al., 2010). Furthermore, the user could request a functionality which is not supported by the system. Therefore, error handling strategies need to be applied. In terms of miscommunication, it can be distinguished between misunderstanding and non-understanding (Skantze, 2007). In the experiment, two of our tasks do not support an implicit application switch, but require an explicit switch. So if users try to switch implicitly, the system will not understand their input in one task and will misinterpret it in the other task. A response to misunderstanding might look like:

U: "Search an Italian restaurant"

S: "In an Italian restaurant you can eat pizza"

A non-understanding informs the user and encourages him to try another request:

S: "Action unknown, please change your request"

These two responses are used until the user changes his strategy to explicit selection. If that does not happen, the task is aborted by the wizard if the user gets too frustrated. This enables us to analyze whether users will switch their strategy or not and how many turns it will take.

4.3 Procedure

The experiment starts with an initial questionnaire to create a profile of the participant, concerning age, experience with smartphones, infotainment systems and SDSs. Then participants are introduced to the driving task and they have time to practice till being experienced. After completing a baseline drive, they start to use the SDS. For each spoken dialog task users get a story describing in prose what they like to achieve with the system. To minimize priming effects, they have to remember their task and are not allowed to keep the description during the interaction. There

is no explanation or example of the SDS, apart from a start command for activation. After the start command, the system plays a beep and the user can say whatever he likes to achieve his task. The exploration phase consists of four tasks, in which users can switch applications implicitly and explicitly. Then they rate the usability of the system with the questionnaire: **S**ubjective **A**ssessment of **S**peech **S**ystem **I**nterfaces (SASSI) (Hone and Graham, 2000). In the second part of the experiment, four tasks with different interaction schemes for application switches are completed randomly: implicit & explicit switch possible, misunderstanding, non-understanding, and dialog-initiative change.

5 Dialog Data Analysis

All audio files of user utterances were transcribed and manually annotated by one person concerning the application selection/switch, speaking style, anaphoric references, and semantic concepts.

First of all, for each application entry and switch it was classified whether the participant used an implicit or explicit utterance. Additionally, the non-understanding and misunderstanding data sets were marked whether the dialog strategy was changed and how many dialog turns this took.

Since most of the user utterances were implicit ones (see Section 6.1), we classified them further into different speaking styles. In the data set of implicit utterances, five different speaking styles could be identified. Table 2 shows them with an example. The illocutionary speech act to search a hotel is always the same, but how users express their request varies. Keyword style and explicit demand is rather how we expect people to speak with machines, as these communication forms are short commands and might be regarded as impolite between humans. Kinder and gentler communications forms are implicit demands, Wh-questions, and Yes-No-Questions. This is how we would expect people to interact with each other.

Keyword Style	*"Restaurant search. Berlin"*
Implicit Demand	*"I'd like to search a restaurant in Berlin."*
Wh-Question	*"Which restaurants are in Berlin?"*
Yes-No-Question	*"Are there any restaurants in Berlin?"*
Explicit Demand	*"Search restaurants in Berlin"*

Table 2: Speaking styles of user utterances.

Two applications are always linked with a common semantic concept. The user has to refer to this concept which he can do in various ways with anaphoric expressions. The annotation of the data set is based on Fromkin et al. (2003) and shown in Table 3 (Examples are user utterances in response to the system prompt "Navigation to Berlin started"). In an elliptic anaphoric reference the concept is not spoken, but still understood because of context - also called gapping. Furthermore, pronominalization can be used as an anaphor. We distinguish between a pronoun or adverb anaphor and an anaphor with a definite noun phrase, since the later contains the type of semantic concept. Another way is simply to rephrase the semantic concept.

Elliptic	*"Search restaurants."*
Pronoun, Adverb	*"Search restaurants there."*
Definite Noun Phrase	*"Search restaurants in this city."*
Rephrase	*"Search restaurants in Berlin."*

Table 3: Anaphoric reference types.

6 Results

In the following, results on application awareness, speaking style, anaphoric expressions, efficiency, and usability are presented. We analyzed data from 31 participants (16m/15f), with average age of 26.65 (SD: 3.32). 26 people possess and use a smartphone on a regular basis and 25 of them are used to application-based interaction (18 people use 1-5 apps and 7 people use 6-10 apps each day). Their experience with SDS is little (6-Likert Scale, avg: 3.06, SD: 1.48) as well as the usage of SDSs (5-Likert Scale, avg: 2.04, SD: 1.16). We asked them how they usually approach a new system or app to learn its interaction schema and scope of operation. On the smartphone, all 31 of them try a new app without informing themselves how it is used. Concerning infotainment systems, trying is also the most used learning approach, even while driving (26 people). This means, people do not read a manual, but the system has to be naturally usable.

In total, we built a corpus of interactions with 5h 25min with 3h 08min of user speech. It contains 243 task entries and 444 subtask switches. Due to data loss 5 task entries could not be analyzed. Subtask switches were less than theoretically possible, because misunderstanding and non-understanding tasks were aborted by the wizard if the user did not change his strategy. Concerning the type of subtask, we analyzed 91 action and 152 information seeking subtasks for task entries, as well as

236 actions and 208 information seekings for task switches.

6.1 Application Awareness

The SDS was designed to be strictly user-initiative: after a beep users could say whatever they liked. We counted 4.9% of user utterances as explicit entries to start a task, which means users in general assume either the SDS is already in the right application context or it is not based on different applications. This is an interaction schema which would rather be used with a human communication partner. 1.1% explicit utterances in subtask switches reinforce this assumption. Utterances addressing more than one application could not be observed.

Furthermore, we analyzed whether users change their strategy from implicit to explicit subtask switch if the system does not react as expected. The implicit switch was prevented and the system answered as if a misunderstanding or a non-understanding has occurred. Table 4 shows results for the number of subtask switches (subt. sw.), number of successful strategy changes (succ.), and average number of user utterances (avg. UDT) till the strategy was changed. In total, only in 43.7% subtask switches users changed their strategy. The difference between non-understanding and misunderstanding was not significant (p=0.051), however, this might due to small sample size.

	subt. sw.	**succ.**	**avg. UDT**
non-underst.	42	15	2.93 (SD=1.91)
misunderst.	45	23	3.74 (SD=1.79)

Table 4: Dialog repair changes to explicit strategy.

In summary, only 6% of user utterances addressed the application explicitly and only 43.7% of users changed their strategy from implicit to explicit. These results reveal that most users are not aware of different applications or do not address applications differently in a speech-only infotainment system. They interact rather like with a human being or with a personal assistant than with a typical in-car SDS.

6.2 Speaking styles of implicit application selection

Even if people interact without being aware of different applications, they might speak to a system in another way than to a human. We analyzed the implicit user utterances according to different speaking styles (see Figure 2). Overall, explicit demand dominates with 37.07% for task entry and 42.42% for subtask switching. Keyword style is used in 16.16% for task entry and 9.29% for subtask switches. As mentioned, explicit demand and keyword style are rather used in human-computer interaction. Here, slightly more than half of the participants (entry: 53.23%; switch: 51.71%) use this kind of interaction. The other half interacts in kinder and gentler forms known from human-human communication.

Comparing task entry and subtask switch, differences could be found in keyword style, implicit demand, and Yes-No-Question. In the first contact with the system, users might be unsure what it is capable of, therefore, often keywords were used to find out how the system reacts. Additionally, the task description was formulated in implicit demand style, thus an unsure user might remember this sentence and use it. Concerning the Yes-No-Questions, they might be a reaction to the naturally formulated system prompts, thus the user adapts to a human-human-like communication style.

Finally, we compare information seeking subtasks with action subtasks. In action subtasks, implicit and explicit demand style dominate. This is reasonable, as people give commands in either form and expect a system reaction. Likewise, it was anticipated that question styles are used for information seeking. One interesting finding is that keyword style is more often used in information seeking. This could be due to priming effects of using search engines like Google[2], in which users only insert the terms they are interested in and Google provides the most likely answers.

In summary, speaking styles vary. Sometimes the system is considered as a human-like communication partner and sometimes users try to reach their goal as fast as possible by giving short commands. However, speaking styles depend on the type of subtask and dialog state.

6.3 Anaphoric Expressions

In a cross-application task, it is of interest how users refer to application-linking semantic concepts. Figure 3 shows which kind of anaphoric expressions were used in implicit utterances. Nearly half of the utterances (47.68%) contain a rephrase of the semantic concept and further 31.57% a def-

[2]www.google.de

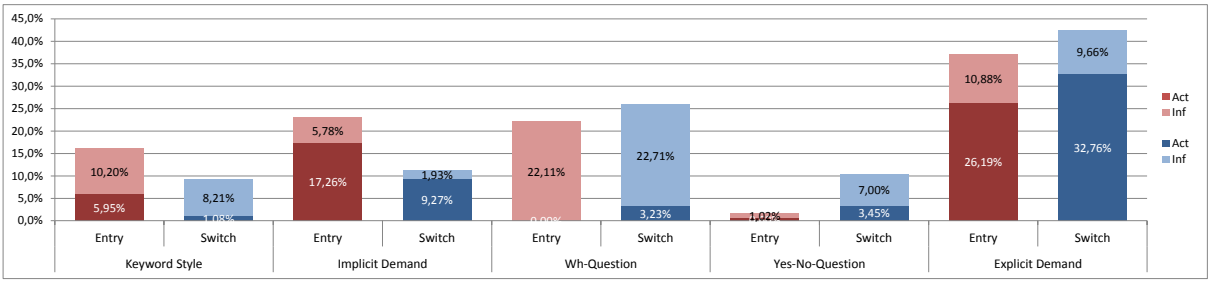

Figure 2: Speaking styles of implicit task entry and subtask switch distinguished by action (Act) and information seeking (Inf)

inite noun phrase. A rephrase utterance can be interpreted easily for an SDS, since there is no need to determine the right antecedent from dialog history. A definite noun phrase contains the semantic type of the antecedent and can be refered easily in a semantic annotated dialog history. However, a pronoun or elliptic anaphoric expression is harder to resolve, as the former only describes the syntactic form of the antecedent and the later does not contain any information of the antecedent. Sometimes, also humans are not able to resolve an anaphoric expression easily. Comparing information seeking and action subtasks, the only difference can be identified between definite noun phrases and rephrase. In information seeking subtasks, participants rephrased more often than using definite noun phrases.

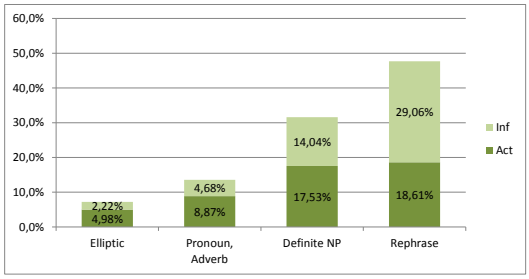

Figure 3: Anaphoric expressions used in implicit application switches.

6.4 Efficiency

Especially in the car it is essential to support short and efficient interactions. In this study, participants used on average 6.27 (SD: 2.62) words for one utterance. However, the word length of a user utterance is only one part which influences dialog length. The number of semantic concepts uttered is more important, as the more semantic concepts are spoken, the less system prompts are needed to request missing information. The semantic concepts of each user utterance were annotated and counted (avg: 2.77; SD: 0.73; min: 1; max: 6). They are set in relation to the maximum required semantic concepts (avg: 3.26; SD: 0.59; min: 2; max: 4) for the corresponding subtask. We divide the spoken concepts by the maximum concepts to calculate an efficiency score (avg: 0.86; SD: 0.22). This means 86% of user utterances contain all necessary semantic concepts to answer the request. Therefore, in-car SDS need to understand multiple semantic concepts in one utterance to keep a dialog short, such as the city, street and street number for a destination entry.

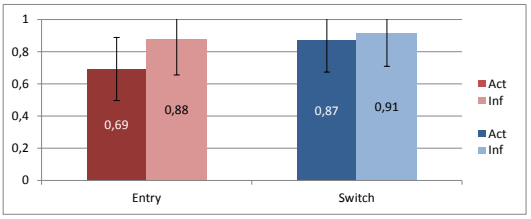

Figure 4: Efficiency scores of user utterances.

Figure 4 shows efficiency scores split into task entry and subtask switch as well as action and information seeking. In total, there is no significant difference between task entry and subtask switch concerning number of words, semantic concepts, or efficiency score. Comparing types of subtasks at task entry, the efficiency score for action subtasks (avg: 0.69; SD: 0.2) is significantly (p=0.0018) less than for information seeking subtasks (avg. 0.88; SD 0.22). Although, significantly (p=0.0003) more semantic concepts in actions were required (avg: 3.66; SD: 0.48) than in information seekings (avg: 3.2; SD: 0.4), users do not utter more semantic concepts. How many semantic concepts users can utter in one sentence while driving, needs to be addressed in the future.

6.5 Usability

Usability is a necessary condition in order to evaluate if people will use a system. The SASSI scores

provide valid evidence of a system's usability. Figure 5 shows results separated into the six dimensions System Response Accuracy (SRA), Likeability (Like), Cognitive Demand (Cog Dem), Annoyance (Ann), Habitability (Hab), and Speed. A 7-Likert scale was used and recoded to values [-3, ..., 3]. If a system is less annoying, its usability will be better. Thus, except of cognitive demand and habitability, the usability of our SDS is rated good. The low habitability score is due to the fact that we did not explain the SDS and after four tasks users are not completely accustomed to the system.

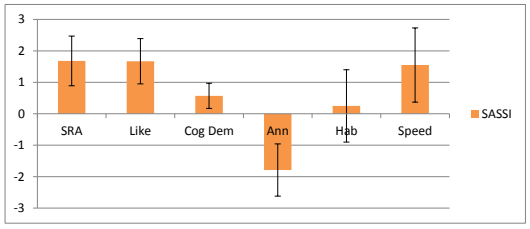

Figure 5: SASSI Usability scores.

7 Discussion and Further Research

The results show, that users are in general not aware of different applications in speech-only in-car SDSs and switch implicitly between different domains. This interaction schema is similar to human-human communication, but may differ if the user is primed through a visual representation. Concerning speaking styles, more than half of the participants used keyword style and explicit demand, which might be regarded impolite between humans. They are aware to communicate with a system lacking emotions. A user, who is not sure about the system's functions, will rather start with keywords and, after hearing natural formulated system prompts, is likely to adapt to natural speaking styles. A human-like prompt (instead of our beep) may ensure the user from the beginning. Obviously, speaking styles depend on type of task, thus question and keyword style is used for information seeking and demand style to initiate an action. More than 50% of the participants used anaphoric expressions, which have to be resolved within dialog context. This is comprehensible, as for people it is usually easier and more efficient to pronounce an anaphor than to pronounce the antecedent. For reaching their interaction goal fast and efficient, the participants used multiple semantic concepts in utterances. In total, 86% of user utterances contain all necessary information to answer the request. This results in less dialog turns and thus is fundamental for in-car systems. In addition, the usability is rated good, thus the system might be accepted by drivers.

Another crucial point for in-car systems is that they should distract the driver as little as possible. It can be assumed that without visual and haptic distractions, the driver would keep his focus on the road. However, cognitive demand also causes distraction. The moderate SASSI score for cognitive demand requires an objective test. Therefore, we will analyze multi-domain interactions with respect to mental pressure and driver performance for further research. So far, we have only considered multi-domain dialogs with one common semantic concept. By referring to multiple semantic concepts, drivers might use more anaphoric expressions or aggregate them with a general term, which needs to be address in further experiments.

8 Conclusions

This paper presents results on how young and technically affine people interact with in-car SDSs in performing multi-domain tasks. 31 participants completed all together 243 tasks (each with two application switches) while driving in a fixed-base driving simulator. In this experiment, a controlled WoZ setup was used instead of a real speech recognition system.

The results identify important guidelines for multi-domain SDSs. Since users are in general not aware of applications in speech-only dialog systems, implicit application switching is required. However, this should not replace explicit switching commands. Speaking styles vary and depend on type of task, and dialog state. Thus language models must therefore consider this issue. People rely on anaphora, which means an SDS must maintain a extensive dialog history across multiple applications to enable coreference resolution. It is further necessary that the SDS supports multiple semantic concepts in one utterance since it enables an efficient interaction and drivers use this. The SDS's usability was rated good by the participants. For further research, we will analyze multi-domain interaction with respect to driver performance and multiple semantic concept anaphora.

Acknowledgments

The work presented here was funded by GetHomeSafe (EU 7th Framework STREP 288667).

References

Kai-Uwe Carstensen, Christian Ebert, Cornelia Ebert, Susanne Jekat, Ralf Klabunde, and Hagen Langer. 2010. *Computerlinguistik und Sprachtechnologie*. Spektrum, Akad. Verl.

Hua Cheng, Harry Bratt, Rohit Mishra, Elizabeth Shriberg, Sandra Upson, Joyce Chen, Fuliang Weng, Stanley Peters, Lawrence Cavedon, and John Niekrasz. 2004. A wizard of oz framework for collecting spoken human-computer dialogs. In *Proc. of ICSLP-2000*.

Victor Ei-Wen Lo and Paul A. Green. 2013. Development and evaluation of automotive speech interfaces: Useful information from the human factors and the related literature. *Int. Journal of Vehicular Technology*, 2013:13.

Norman M. Fraser and G.Nigel Gilbert. 1991. Simulating speech systems. *Computer Speech & Language*, 5(1):81 – 99.

Victoria Fromkin, Robert Rodman, and Nina Hyams. 2003. *An Introduction to Language*. Rosenberg, Michael, 7 edition.

Petra Geutner, Frank Steffens, and Dietrich Manstetten. 2002. Design of the vico spoken dialogue system: Evaluation of user expectations by wizard-of-oz experiments. In *Proc. of the Int. Conf. on Language Resources and Evaluation*, volume 2.

James Glass, Joseph Polifroni, Stephanie Seneff, and Victor Zue. 2000. Data collection and performance evaluation of spoken dialogue systems: The mit experience. In *Proc. of 6th INT*.

Hansjörg Hofmann, Ute Ehrlich, André Berton, and Wolfgang Minker. 2012. Speech interaction with the internet - a user study. In *Intelligent Environments*, Guanajuato, Mexico.

Kate S Hone and Robert Graham. 2000. Towards a tool for the subjective assessment of speech system interfaces (sassi). *Natural Language Engineering*, 6(3&4):287–303.

Melanie Kellar, Carolyn Watters, and Michael Shepherd. 2006. A goal-based classification of web information tasks. In *In 69th Annual Meeting of the American Society for Information Science and Technology (ASIST)*.

Scott R. Klemmer, Anoop K. Sinha, Jack Chen, James A. Landay, Nadeem Aboobaker, and Annie Wang. 2000. Suede: a wizard of oz prototyping tool for speech user interfaces. In *Proc. of the 13th annual ACM symposium on User interface software and technology*, New York. ACM.

Jannette Maciej and Mark Vollrath. 2009. Comparison of manual vs. speech-based interaction with in-vehicle information systems. *Accident Analysis and Prevention*, 41(5):924 – 930.

Angela Mahr, Michael Feld, Mohammad Mehdi Moniri, and Rafael Math. 2012. The contre (continuous tracking and reaction) task: A flexible approach for assessing driver cognitive workload with high sensitivity. In Andrew L. Kun, Linda Ng Boyle, Bryan Reimer, and Andreas Riener, editors, *Adj. Proc. of the 4th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, Portsmouth. ACM.

National Highway Traffic Safety Administration (NHTSA). 2013. Distracted driving 2011. Technical report.

Norbert Reithinger, Jan Alexandersson, Tilman Becker, Anselm Blocher, Ralf Engel, Markus Löckelt, Jochen Müller, Norbert Pfleger, Peter Poller, Michael Streit, and Valentin Tschernomas. 2003. Smartkom: Adaptive and flexible multimodal access to multiple applications. In *Multimodal interfaces*, New York.

Gabriel Skantze. 2007. *Error Handling in Spoken Dialogue Systems*. Ph.D. thesis, KTH Computer Science and Communication.

Jessica Villing, Cecilia Holtelius, Staffan Larsson, Anders Lindstrm, Alexander Seward, and Nina berg. 2008. Interruption, resumption and domain switching in in-vehicle dialogue. In Bengt Nordstrm and Aarne Ranta, editors, *Advances in Natural Language Processing*, volume 5221 of *Lecture Notes in Computer Science*, pages 488–499. Springer Berlin Heidelberg.

Fuliang Weng, Baoshi Yan, Zhe Feng, Florin Ratiu, Madhuri Raya, Brian Lathrop, Annie Lien, Sebastian Varges, Rohit Mishra, Feng Lin, Matthew Purver, Harry Bratt, Yao Meng, Stanley Peters, Tobias Scheideck, Badri Raghunathan, and Zhaoxia Zhang. 2007. Chat to your destination. In *Proc. of 8th SIGdial Workshop on Discourse and Dialogue*.

Christopher D Wickens. 2008. Multiple resources and mental workload. In *Human factors*, volume 50, pages 449–55. USA.

Kristie Young and Michael Regan. 2007. Driver distraction: A review of the literature. *Distracted Driving*, pages 379–405.