Chapter 7
User-Centred Spoken Dialogue Management

Florian Nothdurft, Stefan Ultes, and Wolfgang Minker

Abstract Adaptivity of intelligent environments to their surroundings provided by the ATRACO Spoken Dialogue Manager is only one means of adaptation. Recent work in Spoken Dialogue Systems focuses on the integration of user-centred adaptation means to alter the content, flow and structure of the ongoing dialogue. In this chapter, we introduce a general user-centred adaptation cycle, accompanied by two implemented adaptation approaches focusing respectively on short-term and long-term goals in human–computer interaction. After motivating the need for short-term and long-term goals to entail different adaptation mechanisms, we provide exemplary adaptation entities for each case with corresponding experiments and implementations. The short-term goal user satisfaction allows for detecting whether the user is not satisfied with the interaction and for triggering counter measures to improve the interaction. As a long-term goal, maintaining human–computer trust attempts to keep users still willing to use the system even if the interaction was confusing.

7.1 Introduction

Providing speech interfaces for intelligent environments requires all different kinds of adaptability of the corresponding Spoken Dialogue System (SDS). While the adaptivity described in previous chapters deals with a changing environment, e.g. adding or removing control devices during run-time, a recent research field has emerged: adapting the system behaviour to the user. Here, not only adaptivity to statically changing information, i.e. integrating user models into the interaction, is of interest. Adaptivity to dynamically changing information plays a key role in user-friendly speech interfaces influencing soft measures like user satisfaction or human–computer trust (HCT).

While the subjective user experience should clearly be one of the most important measures in a dialogue system, it might seem odd to take HCT into account as well. However, trust has been shown to be a crucial part in the interaction between

F. Nothdurft (✉) • S. Ultes • W. Minker
Institute Communications Engineering, Ulm University, Ulm 89081, Germany
e-mail: florian.nothdurft@uni-ulm.de; stefan.ultes@uni-ulm.de; wolfgang.minker@uni-ulm.de

S. Ultes et al. (eds.), *Next Generation Intelligent Environments*,
DOI 10.1007/978-3-319-23452-6_7

humans and machines. If the user does not trust the system and its actions, advice or instructions, this may lead to the user avoiding any further interaction with the system [28]. Those situations in particular where the user does not understand the system's behaviour or finds that the system behaves in an unexpected way are likely to impact negatively on the HCT relationship [23] and more generally on human–computer interaction (HCI).

Hence, it seems reasonable to incorporate adaptivity to user satisfaction and HCT into an SDS to foster a highly qualitative and trustworthy interaction. However, enabling an SDS to incorporate further dynamic information into the decision-making process is a twofold problem: the user information must be collected (e.g., estimated using statistical classifiers) and the course of the dialogue must be altered to match the extracted user information. Thus, a general model to dynamic user-adaptive dialogue management is described in Sect. 7.3, preceded by an introduction into the corresponding related work in the next section.

Finally, the implementation of several adaptation concepts is described. These concepts have been implemented within experiments with real and simulated users showing the general benefit of extending the speech interface by user-adaptivity capabilities. Roughly speaking, the adaptation approaches can be grouped into adaptation to dynamically changing information with short-term and long-term goals. While the former is aimed at the improvement of the current interaction step, the latter focuses on reaching a long-term goal, which does not necessarily result in an effective current interaction step.

7.2 Significant Related Work

The field of adaptive dialogue spans over different types of adaptation. While some systems adapt to their environment, the focus in this chapter lies on systems which are capable of adapting to the user and their characteristics. More specifically, an emphasis is put on adaptation to dynamically changing information, i.e. the dynamic adaptation to the user during the ongoing dialogue. In the following, we will present significant work on dynamic adaptation grouped by adaptation targeting a short-term or a long-term goal.

7.2.1 Short-Term Goal

For short-term goal adaptation, i.e. reaching the goal within the same interaction, very prominent work has been presented by Litman and Shimei [18]. They identify problematic situations in dialogues by analyzing the performance of the speech recognizer (ASR) and use this information to adapt the dialogue strategy. Each dialogue starts off with a user initiated strategy without confirmations. Depending on the ASR performance, a system-directed strategy with explicit confirmations may eventually be employed. Applied to TOOT [27], a system for getting information about train schedules, they achieved significant improvement in task success compared to a non-adaptive system.

Further work on user-adaptive dialogue with a short-term goal has been presented by Gnjatović and Rösner [6]. For solving the Tower-of-Hanoi puzzle with an SDS, they identify the emotional state of the user in order to recognize if the user is frustrated or discouraged. The dialogue is adapted by answering the questions "When to provide support to the user?", "What kind of support to provide?" and "How to provide support?" depending on the emotional state of the user. By that, the system is capable of providing well-adapted support for the user which helps to solve the task.

Nothdurft et al. [25] created a dialogue system which is adaptive to the user's knowledge with the short-term goal of increasing the user knowledge after the interaction has been performed. For the task of connecting a home cinema system, the multimodal system provides explanations on how to solve the task presenting text, spoken text, or pictures. The system makes assumptions about the user's knowledge by observing critical as well as successful events within the dialogue (e.g., failed tries, accomplished tasks). Based on the user's knowledge model, the system selects the appropriate explanation type and generates explanations so that the user can be expected to be capable of solving the upcoming task.

7.2.2 Long-Term Goal

Previous relevant work on adaptive dialogue systems with a long-term goal, i.e. maintaining a goal over more than one interaction, mostly involves trust in technical systems. Glass et al. [5] investigated factors that may change the level of trust users are willing to place in adaptive agents. Among these verified findings were statements like "provide the user with the information provenance for sources used by the system", "intelligently modulating the granularity of feedback based on context- and user-modeling" or "supply the user with access to information about the internal workings of the system". However, what is missing in Glass et al.'s work is the idea of rating the different methods to uphold HCT in general and the use of a complex HCT model.

Other related work was for example done by Lim et al. [16] on how different kinds of explanations can improve the intelligibility of context-aware intelligent systems. They concentrate on the effect of Why, Why-not, How-to and What-if explanations on trust and understanding system's actions or reactions. The results showed that Why and Why-not explanations were the best kind of explanation to increase the user's understanding of the system, though trust was only increased by providing Why explanations. Drawbacks of this study were that they only concentrated on understanding the system and trusting the system in general and did not consider that HCT is on the one hand not only influenced by the user's understanding of the system and on the other hand that if one component of trust is flawed, the HCT in general will be damaged [21].

7.3 User-Adaptive Dialogue Management

Realizing user-adaptive dialogue management represents the main contribution of this chapter. Here, two general adaptation types exist: adaptation to statically and dynamically changing information. For the latter, system behaviour is statically influenced, e.g. by user preferences stored in a user model. However, we focus on adaptation to dynamically changing information where the course of the ongoing dialogue is influenced dynamically by some adaptation entity (AE). For this, the general dialogue management concept has to be altered. Furthermore, for adapting the course of the dialogue to the user, two different types of adaptation have been identified:

Adaptation to dynamically changing information with Short-term Goal
This means adapting the dialogue to an AE derived from the ongoing dialogue and modifying the course of the dialogue to improve the AE for the current interaction, e.g. user satisfaction.

Adaptation to dynamically changing information with Long-term Goal
This means adapting the dialogue to an AE derived from the ongoing dialogue and modifying the ongoing dialogue to reach a long-term goal, e.g. establish HCT.

Both types have in common that an AE is derived from the interaction and used to influence the action selection. The difference is—for the first—the interaction is adapted to reflect an increase in the target AE directly for the same interaction. For instance, if the user is not satisfied with the interaction, the goal is to increase the satisfaction of the user for the same interaction. This is in contrast to a long-term goal. For example, if the long-term goal is to maintain HCT, it is not necessarily important that the current interaction is going well but that the users feel they can trust the system nonetheless.

For adapting the dialogue, several adaptation modes exist. A straight-forward mode is to use the AE to influence the selection of the next system action out of the pool of existing system actions (cf. [41, 47]). Thus, the dialogue strategy may depend on the AE. Here, all different kinds of dialogue strategy aspects are possible, e.g. the grounding strategy, the dialogue initiative or the prompt design. Of course, there are many more options.

Another way of adaptation mode is to add extra system actions only triggered by the adaptation mechanism. A help action or an error recovery strategy might be activated depending on the AE.

To implement any type of adaptation to dynamically changing information, the processing sequence of a SDS has to be extended. It may be viewed as a cyclic process—involving the human as one part. For the extension of the dialogue cycle to allow for this kind of adaptation, a new module has to be introduced (see Fig. 7.1). Without loss of generality, the cycle may be regarded to start with the system selecting the first system action. This can be seen as valid for all situations if the set of system actions also includes the action of only waiting for user input without

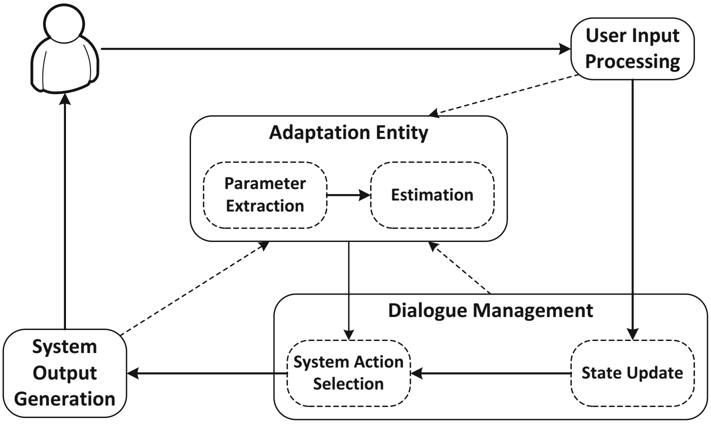

Fig. 7.1 The adaptive dialogue processing cycle. For adapting to additional user values, the modules Parameter Extraction and Value Estimation are integrated producing the estimation of the adaptation value

producing any output. Based on the selected system action, output is created and presented to the user. After the user turn, the created output of the user is processed as user input to the system. Usually, this involves automatic speech recognition and a semantic interpretation.

For enabling the dialogue system to react adaptively, the AE must be determined. As this has to be done without human intervention, often, an automatic estimation approach, e.g. statistical classification, is used. For this, parameters used as input to the estimator must be derived from the interaction taking into account information from all dialogue system components, i.e. speech recognition, semantic interpretation, dialogue management, and output generation. Based on these input parameters, the AE may then be determined and fed into the action selection module of the dialogue management. Based on the AE and the updated system state, the next system action is selected and the cycle starts anew.

Having this type of adaption to dynamically changing information also encompasses several issues. Adding an AE to the action selection may be regarded as an increase in dimensionality and complexity of the problem. This also results in an added uncertainty to the system which should be handled adequately, e.g. using Partially Observable Markov Decision Processes (POMDPs). Furthermore, having several adaptation modes, the question arises which mode is suitable, taking into account the type of adaptation to dynamically changing information as well as the dialogue situation. Finally, the concepts of dialogue management presented in this section represent a system where the only modifications of the system state are due to user input. However, in an intelligent environment, there are multiple entities which are able to modify the state. Hence, the mechanisms have to be integrated into such a framework. The following two sections will give more insight into the aforementioned issues for adaptation to dynamically changing information with short-term and with long-term goals.

7.4 Adaptation to Dynamically Changing Information with Short-Term Goal

Adapting the course of the ongoing dialogue to a dynamically derived adaptation entity with a short-term goal means that the goal should be reached within the same interaction. For this, adaptation entities which may be used for this are, for instance, the intoxication level [42], the emotional state [35, 45] or the user satisfaction [32, 33]. For the latter, if the system detects that the user is not sufficiently satisfied with the interaction, it may take measures to increase the user's satisfaction level. As user satisfaction is a domain-independent entity which may occur in almost all types of dialogue, this section focuses on adaptation to user satisfaction.

Today, there are multiple approaches and metrics which model user satisfaction. To be useful for adaptation to dynamically changing information, the user satisfaction metric must meet certain criteria [44]. The most important criterion is that it must be derivable automatically for each system-user exchange without human intervention. While many approaches exist to automatically determine user satisfaction on the exchange level [3, 7, 9, 10, 33], the Interaction Quality paradigm presented by Schmitt et al. [32] seems to be most suitable.

Consequently, in this section, an adaptation mechanism will be described adapting the course of the dialogue to the Interaction Quality (IQ) in order to increase the IQ of the ongoing dialogue. The Interaction Quality paradigm will be described in the following (Sect. 7.4.1). As adding an extra dimension to the action selection problem also results in adding more uncertainty, the dialogue system should be cast as a POMDP which are specially designed to handle uncertainty. Hence, theoretical aspects of mapping a POMDP to SDS and extending the ATRACO SDM presented in Sect. 6.5.2 to incorporate the POMDP will be described in Sect. 7.4.2. Finally, experiments incorporating quality-adaptivity using real users and a user simulator are also presented showing the viability of this approach in Sect. 7.4.3.

7.4.1 Interaction Quality

Interaction Quality (IQ) was originally proposed by Schmitt et al. [32] as an alternative and more objective measure of user satisfaction. For the authors, the main aspect of user satisfaction is that it is assigned by real users. However, this is impractical in many real world scenarios. Therefore, the usage of expert raters is proposed. Further studies have also shown that ratings applied by experts and users have a high correlation [46].

Furthermore, IQ fulfills all requirements identified by Ultes et al. [44] which are needed for a quality metric to be employable for dialogue adaptation:

- exchange level quality measurement
- automatically derivable features

- domain-independent features
- consistent labeling process
- reproducible labels
- unbiased labels
- sufficient estimation performance

The performance of a SDS may be evaluated either on the dialogue level or on the exchange level. As dialogue management is performed after each system-user exchange, dynamic adaption of the dialogue strategy to the dialogue performance requires exchange level performance measures. Therefore, dialogue-level approaches are of no use.

Features serving as input variables for a classification algorithm must be automatically derivable from the dialogue system modules. This is important because manually annotated features produce high costs and are also not available immediately during run-time in order to use them as additional input to the Dialogue Manager. Furthermore, for creating a *general* quality metric, features have to be domain-independent, i.e. not dependent on the task domain of the dialogue system.

Another important issue is the consistency of the labels. Labels applied by the users themselves are subject to large fluctuations among the different users [17]. As this results in inconsistent labels, which do not suffice for creating a generally valid quality model, ratings applied by expert raters yield more consistent labels. The experts are asked to estimate the user's satisfaction following previously established rating guidelines. Furthermore, expert labelers are also not prone to be influenced by certain aspects of the SDS, which are not of interest in this context, e.g. the character of the synthesized voice. Therefore, they create less biased labels.

Finally, the process of deriving the measure must perform adequately. Otherwise, the quality value is not reliable and hence no reasonable adaptation may be applied.

The IQ paradigm describes the Interaction Quality on a scale from five to one: 5 ("satisfied"), 4 ("slightly unsatisfied"), 3 ("unsatisfied"), 2 ("very unsatisfied"), and 1 ("extremely unsatisfied"). The paradigm is based on automatically deriving interaction parameters from the SDS and feeding these parameters into a statistical classification module which predicts the IQ level of the ongoing interaction at the current system-user-exchange. The interaction parameters are rendered on three levels (see Fig. 7.2): the exchange level the window level, and the dialogue level. The exchange level comprises parameters derived from the SDS modules Automatic Speech Recognizer, Spoken Language Understanding, and Dialogue Management directly. Parameters on the window and the dialogue level are sums, means, frequencies or counts of exchange level parameters. While dialogue level parameters are computed from all exchanges of the dialogue up to the current exchange, window level parameters are only computed from the last three exchanges.

These interaction parameters are used as input variables to a statistical classification module. The statistical model is trained based on annotated dialogues of the Lets Go Bus Information System in Pittsburgh, USA [30].[1] The annotated

[1] The Lets Go domain will be introduced in more detail in Sect. 7.4.3.2.

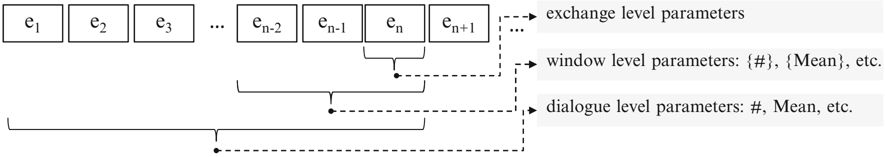
e1
e2
e3
...
en-2
en-1
en
en+1
...
exchange level parameters
window level parameters: {#}, {Mean}, etc.
dialogue level parameters: #, Mean, etc.

Fig. 7.2 The interaction parameters consist of three levels [34]: the exchange level containing information about the current exchange, the window level, containing information about the last three exchanges, and the dialogue level containing information about the complete dialogue up to the current exchange

data is packaged in the LEGOv2 corpus [34, 50]. Each of the 9638 exchanges (401 calls) has been annotated by three different raters resulting in a rating agreement of $\kappa = 0.52$. Furthermore, the raters had to follow labeling guidelines to enable a consistent labeling process [34]. Applying a Support Vector Machine [51] (SVM) for estimating the Interaction Quality achieved an unweighted average recall of 0.59 when including domain information [32] and 0.49 without domain information [39] both using only automatically derivable features. For the latter, Ultes et al. were able to improve the performance by applying an hierarchical approach introducing error correction [38] to a UAR of 0.53. While modeling IQ estimation as a sequential problem using regular Hidden Markov Models was not successful [43], the performance could be improved by applying a Hybrid Hidden Markov Model [39] achieving a UAR of 0.51.

7.4.2 Probabilistic Dialogue Management for Intelligent Environments

In order to enable the Dialogue Manager to deal with the added uncertainty inherent in adaptation to dynamically changing information when adding an automatic AE estimation module, a POMDP [12] is utilized which has been shown to work well with SPDs [54]. Formally, a POMDP consists of a set S of state variables, a set A of system actions, and a set O of all possible observations of the system. Furthermore, transition probabilities $P(s'|s, a)$ and observation probabilities $P(o'|s')$ are included. As the state of the underlying process cannot be determined exactly, a probability distribution over all possible states, called the *belief state* $b(s)$, is used instead. It is updated with the following equation:

$$b'(s') = p(o'|s') \cdot \sum_{s} P(s'|s, a) \cdot b(s) \ . \tag{7.1}$$

However, casting an SDS as a POMDP yields the problem that computing the probability distribution over all dialogue states is intractable. A promising methodology of handling multiple state hypotheses is the Hidden Information State

(HIS) introduced by Young et al. [55] which will be described in Sect. 7.4.2.1. To integrate the HIS approach into the ATRACO SDM (see Sect. 6.5.2), several alterations have to be made which are described in Sect. 7.4.2.2.

7.4.2.1 The Hidden Information State Approach

To cast an SDS as a POMDP within the Hidden Information State (HIS) approach proposed by Young et al. [55], the state s is decomposed into (u, g, h) representing user action u, user goal g, and dialogue history h as proposed by Williams and Young [54], who also introduce reasonable independence assumptions. The user goal space is further partitioned into equivalence classes, or partitions p, according to the possible values a slot can take. Introducing further simplification, one slot in the partition may only take all values or one single value, or it may exclude a set of values. For user input, the partitions are first split and the probability mass of the originating partition is distributed to the resulting partitions. In the second phase, the belief $b(s)$ of state $s = (u, p, h)$ is updated according to equation

$$b'(u',p',h') = k \cdot P(o'|u')P(u'|p',a)\sum_{h} P(h'|u',p',h,a)\sum_{u} P(p'|p)b(u,p,h) \;, \tag{7.2}$$

where o' is the current observation and a the last system action. $P(p'|p)$ denotes the probability of partition p' originating from partition p or, in other words, the fraction of probability mass which is transferred from p to p' if p is split into p' and $p - p'$.

According to Williams [53], the splitting probability

$$P(p'|p) = \frac{b_0(p')}{b_0(p)} \tag{7.3}$$

is computed as the ratio of the prior probability of the new partition $b_0(p')$ to the prior probability of the originating partition $b_0(p)$.

In order to make optimization of the policy $\pi(b)$ that determines the next system action more tractable, the resulting belief is transformed to a summary belief point $\hat{b}$ containing only information about the two most likely partitions. Based on this, a summary system action $\hat{a}$ is selected according to the trained policy $\pi(\hat{b})$. This summary system action then has to be refined using heuristics. The resulting action a is then executed by the system.

For better illustration of the partitioning approach, an example in a flight booking domain is shown in Fig. 7.3. Initially, there is only one partition containing all values for each slot, namely `Origin` and `Destination`. If new user input arrives, the partition is split according to the slot the user input belongs to. In this example, the n-best list belongs to the slot `Destination` and contains two entries. Therefore, the root partition is split and two new partitions are created, each one containing one of the two values in the slot `Destination`. The range

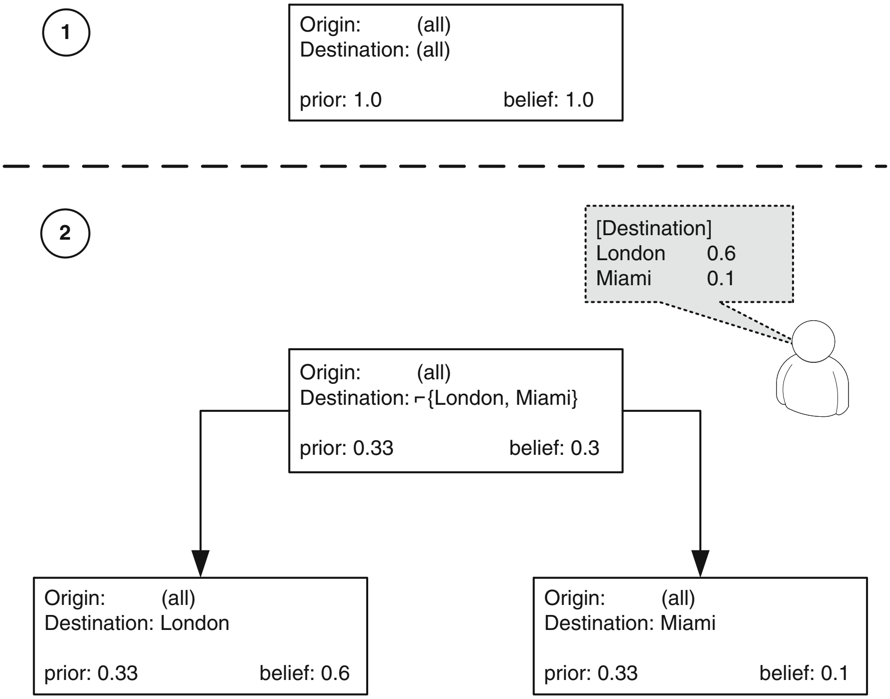

Fig. 7.3 An example of partition splitting [40] with three possible destinations: “London”, “Miami”, and “Paris”. First, there is only one partition subsuming all values for the two slots. After splitting the partition on the user input, two new partitions are created each representing all goals containing “London” or “Miami”, respectively, as destination, while the original partition excludes both values

of slot-values the original partition is representing has been reduced to exclude the two destinations provided by the user. Following that, the new belief values are determined. In order to select the next system action, the summary belief is computed. Based on this, the policy is applied and the resulting system action is refined and executed.

7.4.2.2 ATRACO Spoken Dialogue Management and the Hidden Information State Approach

For modeling spoken dialogue interaction between the user and the IE, the ATRACO SDM has been developed. As it is based on the Information State (IS) approach [13], applying the Hidden Information State approach—an extension of the IS—for introducing probabilistic dialogue into the ATRACO SDM seems natural. The resulting system contains the capability of having different operation modes concerning the number of state hypotheses and the policy. A more detailed description may be found in [40].

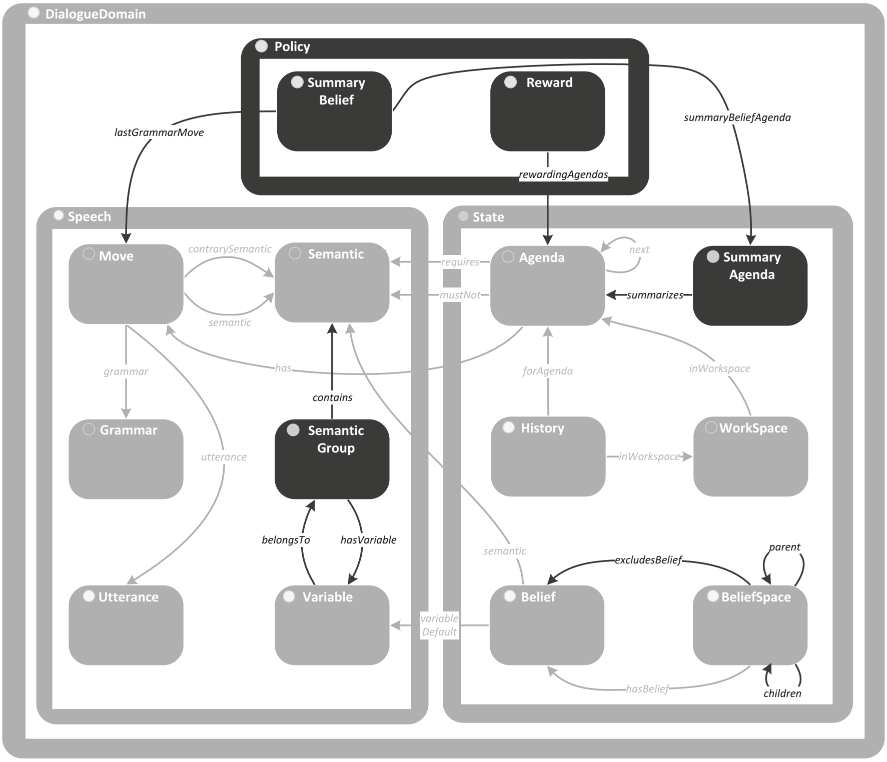

Fig. 7.4 A scheme of the Extended Spoken Dialogue Ontology (SDO) [40] based on Heinroth et al. [8]. Concepts belonging to the original SDO are *light grey* while concepts and relations introduced for the HIS implementation are *dark grey*. The static dialogue description is shown on the *left side* of the picture within the *Speech* class while the concepts belonging to the dynamic *State* of the system are shown on the *right side*. Additionally, concepts belonging to the *Policy* are shown at the *top*

The core of the ATRACO SDM, which has been designed following the *model-view-presenter* design pattern [29] allowing for a strict separation of data management, dialogue logic and dialogue interface, depicts the Domain Model. This SDO is internally structured—using the Web Ontology Language (OWL) [1]—into a static and a dynamic part. To integrate HIS functionality into the model, some relations and classes have to be altered and added. The new ontology is shown in Fig. 7.4 with all additional classes and relations colored in dark grey.

The main difference between the original SDO—implementing the IS approach—and the HIS approach is the state model. While for the IS, only one single state exists (represented by the *BeliefSpace* class), the HIS includes a tree of states associated with a probability. To introduce this hierarchical structure of the HIS partition state into the SDO, the relations *parent* and *children* have been added.

As each HIS partition either represents all values (for each slot), exactly one value, or excludes a set of slot values this has also be modeled within the SDO. For the latter, the relation *excludesBelief* is added representing all slot values, which are excluded by this partition. A slot taking one specific value, on the other hand, is represented by the already existing *hasBelief* relation. The slot concept itself is introduced by the *SemanticGroup* class subsuming all values (*Semantics* or *Variables*) of the given slot.

To realize the application of automatically trained policies based on summary space belief points [56], the general class *Policy* is added to the *DialogueDomain*. It contains the *SummaryBelief* class representing the summary belief point $\hat{b}$ which is related to the new class *SummaryAgenda* representing the summary action $\hat{a}$ (summarizing "actual" system actions *Agendas*). To enable automatic optimization of policies, a reward function has to be defined. It is realized by the *Reward* class defining a reward value for all agendas which are connected by the *rewardingAgendas*.

Basing the ATRACO SDM on the HIS approach also entails a probability model $P(o|u)$ modeling the probability of the user input. It is usually approximated with $P(o|u) \approx P(u|o)$ and thus modeled using the confidence scores or the n-best list input. As the view of the ATRACO SDM is based on automatically creating VoiceXML [26] documents for each turn, the created documents have to be altered to add n-best list functionality. Fortunately, VoiceXML provides mechanisms for using n-best list with confidences inherently which only have to be added and activated.

Spoken dialogue management, in general, has two major tasks: updating the internal state representation and, based on this updated state, selecting the next system action. For the ATRACO SDM, this is handled within the presenter which hence contains the dialogue logic. In order to incorporate HIS functionality, i.e. handling multiple state hypotheses and applying an optimized ontology, the AT&T Statistical Dialog Toolkit (ASDT) [52] is integrated into the SDM. Thus, only certain probability models have to be designed within the ATRACO SDM: the partition splitting model $P(p'|p)$, the user model $P(u'|p',a)$, and the history model $P(h'|u',p',h,a)$.

While the latter two are modeled rather simply by checking if the user input matches the current partition and history state, the partition splitting is more complex. First, the decision if the partition is split or solely updated has to be made. The latter happens if the user action does not represent new slot information but only a confirmation. The partition is split if the user action contains new slot information.

While the new ATRACO SDM offers HIS functionality, the original control modes and system state representations are still usable. Hence, the new ATRACO SDM incorporating the HIS approach offers multiple operation modes:

- Rule-Based Control + Single State Hypothesis
- Rule-Based Control + Multiple State Hypotheses
- Trained Policy-Control + One State Hypothesis
- Trained Policy-Control + Multiple State Hypotheses

The original ATRACO SDM conveys rule-based control with a single state hypothesis, i.e. the dialogue state is modeled within the ontology class *Beliefspace*. By extending the SDM with the HIS approach, both the capability of training an optimized policy automatically as well as handling multiple parallel state hypotheses, i.e. the HIS partitions, is introduced.

7.4.3 Experiments

To evaluate the impact of adding IQ-adaptivity to the dialogue, two studies have been conducted using the ATRACO SDM presented in the previous section. First, a pilot user study employing IQ-adaptation techniques adapting the grounding strategy in a limited train booking domain has been conducted [48]. The general aim of the study was to gain an initial insight into the capabilities of IQ-adaptive dialogue. A second study with a simulated user has been conducted in a more complex domain adapting the initiative [49]. The design of both studies along with the results will be described in the following.

7.4.3.1 Pilot User Study in the Train-Booking Domain

To gain an initial insight into the capabilities and opportunities of IQ-adaptive dialogue, a study within a simple train booking dialogue with real users was conducted. Depending on the current IQ value, the grounding strategy was adapted, i.e. each time the system requests a confirmation about a certain slot value from the user, the IQ value is used to decide whether the system uses an explicit or implicit confirmation prompt. In the following, the design and setup of the study will be presented before giving details about the results.

Design and Setup

For conducting a pilot study for IQ-adaptive dialogue, the grounding strategy was selected as it is an easily adaptable concept which occurs in almost every dialogue. A dialogue in the train booking domain was created asking the user for information about the origin, the destination, the day of the week, and the time of travel. The user could choose out of 22 cities which were used as origin and destination alike. Furthermore, the time of travel was restricted to every full hour (1 pm, 2 pm, 3 pm, etc.). Three different dialogues were created: one only applying explicit confirmation (all-explicit), one applying only implicit confirmation (all-implicit), and one adapting the confirmation type to the current IQ value (adapted). Besides these differences, the dialogues were the same. The complete dialogue was system initiated and the course of the dialogue was predetermined, i.e. the order of information the user was asked to provide was given. As only two different options for adapting the dialogue exist, i.e. either selecting implicit or explicit confirmation, the IQ value has been limited to only two values: two representing a satisfied user and one representing an unsatisfied user. If the user was recognized as being satisfied with the dialogue (high IQ value), slot values were confirmed implicitly while explicit confirmation was applied for unsatisfied users (low IQ value). In the end of the dialogue, the user was provided with a dummy message stating that the reservation has been made.

The IQ estimation module was based on the LibSVM implementation [2] using a linear kernel.

Before the experiment, each participant was presented with a sheet of paper stating all options they could say during the dialogue. This also included a list of all cities. Furthermore, each user participated in three runs of the dialogue—one for each type of confirmation strategy. During the experiment, the order of these dialogues has been alternated to get an equal distribution over all combinations so that learning effects are taken account of. However, the user was not aware of the different dialogue types. After each dialogue, the participants were asked to fill out a questionnaire based on the SASSI questionnaire [11] to evaluate their overall experience with the dialogue. Each item was rated on a seven-point scale.

In total, there were 24 participants (eight female, 16 male) creating 72 dialogues with an average number of turns of 33.58. The participants, who were students from multiple disciplines, were between 19 and 38 years old with an average age of 26.42.

Experimental Results

The results for all questions from the questionnaires are depicted in Table 7.1. Each row shows the average score for one of the three different strategies. It is a well-known fact that, for simple tasks like this, an all-implicit strategy is usually preferred over an all-explicit strategy (cf. [4]). Hence, as expected, the all-implicit strategy performed best outperforming the all-explicit strategy clearly: it achieved a better score for almost all questions. The difference is even significant for 16 out of 25 values ($\alpha < 0.05$ applying the Mann–Whitney U test [20]). Comparing the all-explicit to the adapted strategy gives a similar impression: The scores for almost all questions are better for the adapted strategy. However, this is not as significant having only seven significant different values. More revealing is the conclusion drawn from comparing the all-implicit with the adapted strategy. While the all-implicit strategy again governs the scores, almost all results are not significantly different. Hence and in contrast to the expectations, the adapted strategy did not perform significantly worse despite the dialogue being very simple.

This result is underpinned by looking at the users' overall satisfaction score with the dialogue as an emphasis was put on the question which strategy people liked best. A bar graph showing the average outcome of the user ratings grouped by the respective dialogue strategy is depicted in Fig. 7.5. While the adapted strategy resulted in 45.6 % explicit and 54.4 % implicit confirmations, it is very interesting that it was not rated significantly different compared to the all-implicit strategy.

Table 7.1 The average results of the user questionnaires

	Fun with interaction	System reaction	Hard to lose track	Natural communication	User is in control	Overall impression	User felt calm	**user felt tense**	Dialogue length appropriate	User knew what to say	High level of concentration necessary	System understands user input	Functionality	Interaction was pleasant
All-implicit	5.3^{e}	5.9^{e}	6.2	4.8^{e}	3.0	5.4^{e}	5.0	$6.3^{a,e}$	5.4^{e}	6.0	2.7	5.9^{e}	5.5	5.6^{e}
All-explicit	$3.7^{a,i}$	4.8^{i}	5.8	3.5^{i}	2.7	4.3^{i}	4.5	5.0^{i}	$3.5^{a,i}$	6.1	3.3	4.2^{i}	4.6	4.3^{i}
Adapted	4.8^{e}	5.3	6.1	4.5	3.0	5.0	4.8	5.4^{i}	5.0^{e}	6.2	3.2	5.2	5.3	5.0

	Interaction was not exhausting	Interaction was not boring	Interaction was diversified	Interaction was not frustrating	System made no errors	Error due to user behaviour	System is friendly	System is easy to use	System is flexible	System is understandable	Reaction time	Would use system again	Human operator preferred	Overall satisfaction
All-implicit	5.6^{e}	4.7^{e}	3.9^{e}	5.9^{e}	5.9^{e}	3.3	5.8	6.3^{e}	4.1	5.8	5.0^{e}	5.2	4.7	5.5^{e}
All-explicit	4.3^{i}	$3.3^{a,i}$	$3.0^{a,i}$	$4.4^{a,i}$	4.5^{i}	2.1	5.8	5.6^{i}	3.8	5.5	$3.9^{a,i}$	4.2	5.1	$4.4^{a,i}$
Adapted	5.3	4.3^{e}	3.9^{e}	5.5^{e}	5.3	2.8	5.6	6.0	4.2	5.8	4.9^{e}	5.0	4.7	5.3^{e}

Each question could be answered by a seven-point scale being translated to scores from one to seven. Significant differences are marked with a, e, and i marking significance with the adaptive, explicit, and implicit strategy respectively. (Please note: the original questionnaire was in German)

That is even although the ASR component made almost no errors (due to the limited number of options). Moreover, calculating Spearman's Rho [37] shows significant correlation ($\alpha < 0.01$) with $\rho = 0.6$ between the users' overall satisfaction of the all-implicit and adapted strategy. Additionally, the dialogue length, which is one main indicator for user satisfaction in simple dialogues like this, is significantly higher for the adapted strategy compared to the all-implicit strategy.

In other words, although the task was quite simple, there was no difference between the all-implicit and adapted strategies encouraging the hope that for more complex dialogues, quality-adaption will perform best.

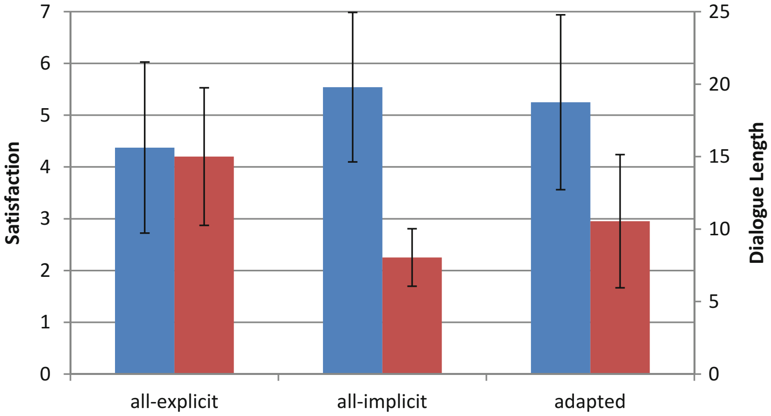

Fig. 7.5 The overall satisfaction with the dialogue (*left bar*, left *y*-axis) and the average dialogue length in number of turns (*right bar*, right *y*-axis) according to questionnaire evaluation. Satisfaction for implicit and adapted do not differ significantly while all other differences are significant

7.4.3.2 User Simulator Study in the Bus Schedule Information Domain

For a second experiment with a system providing bus schedule information, the dialogue initiative was adapted. Conventional dialogue initiative categories are *user initiative*, *system initiative* and *mixed initiative* [22]. As there are different interpretations of what these initiative categories mean, we stick to the understanding of initiative as used by Litman and Pan [18]: the initiative influences the openness of the system question and the set of allowed user responses. The latter is realized by defining which slot values provided by the user are processed by the system and which are discarded. Hence, for *user initiative*, the system asks an open question allowing the user to respond with information for any slot. For *mixed initiative*, the system poses a question directly addressing a slot. However, the user may still provide information for any slot. This is in contrast to the *system initiative*, where the user may only respond with the slot addressed by the system. For instance, if the system asks for the arrival place and the user responds with a destination place, this information may either be used (*mixed initiative*) or discarded (*system initiative*).

Design and Setup

In order to evaluate the dialogue strategies, the adaptive ATRACO SDM is used interacting with a user simulator having rule-based control with a single state hypothesis. For creating dialogues, the Lets' Go Domain is chosen as it represents a domain of suitable complexity. The Let's Go Bus Information System [30] is a live system in Pittsburgh, USA providing bus schedule information to the user. It consists of four slots: bus number, departure place, arrival place, and travel time.

However, the bus number is not mandatory. The original system contains more than 300,000 arrival or departure places, respectively. The Let's Go User Simulator (LGUS) by Lee and Eskenazi [14] is used for evaluation to replace the need for human evaluators.

The IQ estimation module was based on the LibSVM implementation [2] using a linear kernel. The trained model achieves an accuracy of 54.1 % on the training data using tenfold cross-validation. All exchanges of the LEGO corpus have been used for training.

For evaluation, a total of 5000 simulated dialogues for each strategy have been created. In accordance to Raux et al. [30], short dialogues (less than 5 exchanges[2]) which are considered "not [to] be genuine attempts at using the system" are excluded from all statistics in this paper.

Three objective metrics are used to evaluate the dialogue performance: the average dialogue length (ADL), the dialogue completion rate (DCR) and task success rate (TSR). The ADL is modeled by the average number of exchanges per completed dialogue. A dialogue is regarded as being completed if the system provides a result—whether correct or not—to the user. Hence, DCR represents the ratio of dialogues for which the system was able to provide a result, i.e. provide schedule information:

$$\mathrm{DCR} = \frac{\#\mathrm{completed}}{\#\mathrm{all}} .$$

TSR is the ratio of completed dialogues where the user goal matches the information the system acquired during the interaction:

$$\mathrm{TSR} = \frac{\#\mathrm{correctResult}}{\#\mathrm{completed}} .$$

Here, only destination place, arrival place, and travel time are considered as the bus number is not a mandatory slot and hence not necessary for providing information to the user. Furthermore, the average IQ value (AIQ) is calculated for each strategy based on the IQ values of the last exchanges of each dialogue.

Experimental Results

Figure 7.6 shows the ration of complete, incomplete, and omitted dialogues for each strategy with respect to the total 5000 dialogues. As can be seen, about the same ratio of dialogues is omitted due to being too short. The DCR clearly varies more strongly for the five strategies.

[2] The minimum number of exchanges to successfully complete the dialogue is 5.

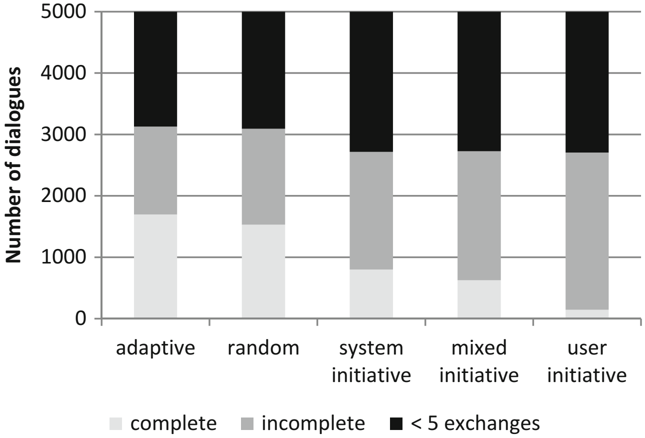

Fig. 7.6 The ratio of omitted dialogues due to their length (<5 exchanges), the completed dialogues (complete), and the dialogues which have been aborted by the user (incomplete) with respect to the dialogue strategy. While the amount of short dialogues is similar for each strategy, the number of completed dialogues varies strongly

The results for DCR, TSR, ADL, and AIQ are presented in Fig. 7.7. TSR is almost the same for all strategies, meaning that, if a dialogue completes, the system almost always found the correct user goal. Hence, TSR is not further regarded. DCR, ADL and AIQ on the other hand vary strongly. They strongly correlate with a Pearson's correlation of $\rho = -0.953$ (level of significance $\alpha < 0.05$) for DCR and ADL, $\rho = 0.960$ ($\alpha < 0.01$) for DCR and AIQ, and $\rho = -0.997$ ($\alpha < 0.01$) for ADL and AIQ.

Comparing the performance of the adaptive strategy to the three non-adaptive strategy clearly shows that the adaptive strategy performs significantly best for all metrics achieving a DCR of 54.27 % (which is comparable to the rate achieved on the training data of LGUS (cf. [14]).

Furthermore, the adaptive strategy has a significant higher average IQ (AIQ) value calculated from the IQ value for the whole dialogues, i.e. the IQ value of the last system-user-exchange, than all other non-adaptive strategies.

7.4.4 Conclusion

Using the short-term goal Interaction Quality for adapting to dynamically changing information seems to be a promising approach to increase the overall dialogue performance for both user experience and objective metrics. Adapting the grounding strategy as well as the dialogue initiative in a rule-based setting are both reasonable measures. Moreover, casting the dialogue system as a POMDP—resulting in an

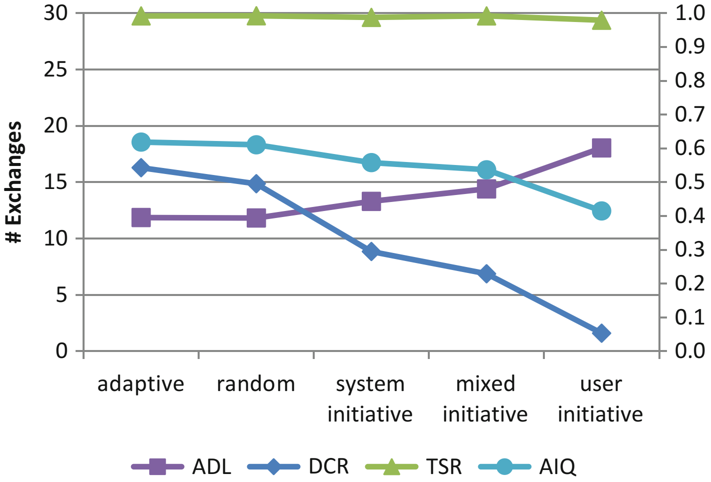

Fig. 7.7 The average dialogue length (ADL), task success rate (TSR), the dialogue completion rate (DCR), and the average Interaction Quality (AIQ) for all for dialogue strategies[3]. With decreasing DCR, also AIQ decreases and ADL increases. (AIQ values are normalized to the interval [0–1])

extension of the ATRACO SDM—allows not only for an improvement in dialogue performance but also for a better handling of the added uncertainty inherent in IQ estimation.

7.5 Adaptation to Dynamically Changing Information with Long-Term Goal

While the adaptation to IQ is focused on achieving short-term goals, the HCT relationship between human and dialogue system represents a long-term goal. A user's HCT model can not be measured directly during run-time, but only by means of a questionnaire. What can be assessed are though the dialogue history and the user's affective state (at least a hypothesis on that can be used). This means, that symptoms like user frustration or confusion, which may indicate the possibility of a decreasing HCT relationship, have to recognized and then the resulting change in HCT must be estimated. Now the question might arise why we should bother with modeling HCT in the first place instead of reacting only to affective user states?

[3] All results for DCR and TSR are significantly different (chi-squared test). Significant differences in ADL (unpaired *t*-test) and AIQ (Mann–Whitney *U* test) with the respective strategy to the right are on the level of $\alpha < 0.01$ for system initiative (ADL) and mixed initiative (ADL) and on the level of $\alpha < 0.05$ for adaptive (AIQ), random (ADL, AIQ), and mixed initiative (AIQ). All other comparisons between non-neighbours are significant with $\alpha < 0.01$.

The most important difference is that human–computer trust is a model which evolves long-term, but affective states model dynamic events. Though we want to use dynamic events as well to determine system behaviour, we also want to develop a healthy HCT relationship between human and computer. This way, despite, for example, being confused, the user might still be willing to continue interacting with a dialogue system.

7.5.1 Human–Computer Trust

Trust has shown to be a crucial part of the interaction between human and machines. If the user does not trust the system and its actions, advice or instructions, the interaction with the machine may change up to a complete abortion of future interaction [28]. Situations, where the system's actions do not match the user's expectations, are likely to have a negative impact on the HCT relationship [23]. Those situations occur as a consequence of incongruent models of the system: During the interaction the user builds a mental model of the system and its underlying processes that determine system actions and output. However, if this perceived mental model and the actual system model do not match, the HCT relationship may be influenced negatively [23].

Mayer et al. [21] define trust in human–human interaction to be "the extent to which one party is willing to depend on somebody or something, in a given situation with a feeling of relative security, even though negative consequences are possible". For HCI, trust can be defined as "the attitude that an agent will help achieve an individual's goals in a situation characterized by uncertainty and vulnerability" [15]. Machines that serve as intelligent assistants with the purpose of helping the user, in complex as well as in critical situations, seem to be very dependent on an intact HCT relationship. However, trust is multi-dimensional and consists of several components. For human relationships, Mayer et al. defined three levels that build trust: ability, integrity, and benevolence. The same holds for HCI, where HCT is a composite of several components. For human–computer trust Madsen and Gregor [19] constructed a hierarchical model (see Fig. 7.8) resulting in five basic components of trust, which can be divided into two general categories, namely cognitive-based and affect-based ones. In short-term HCI, cognition-based HCT components seem to be more important because it will be easier to influence those. Perceived understandability can be seen in the sense that the human supervisor or observer can form a mental model and predict future system behaviour. Perceived reliability in the usual sense of repeated, consistent functioning. Furthermore, technical competence in the sense that the system is perceived to perform the tasks accurately and correctly based on the input information. In this context it is important to mention, that as Mayer already stated, the components of trust are separable, yet related to one another. All components must be perceived highly for the trustee to be deemed trustworthy. If any of the components does not fulfill this requirement, the overall trustworthiness can suffer [19]. Hence, a dialogue system

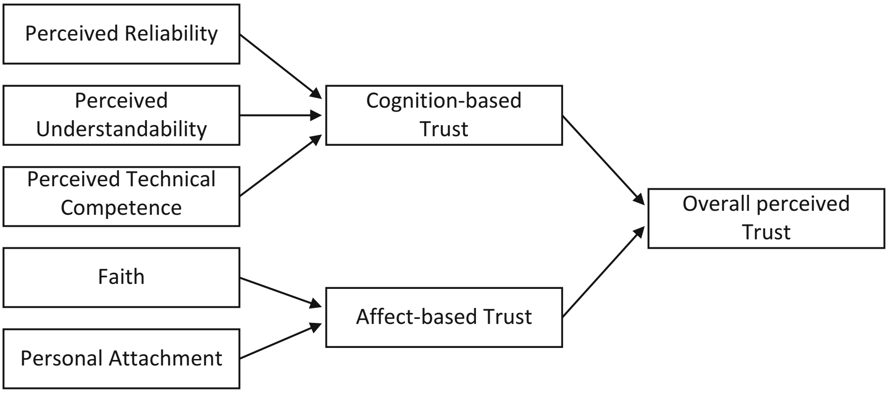

Fig. 7.8 Human–computer trust model: personal attachment and faith are the components of affect-based trust. Perceived understandability, technical competence and reliability for cognition-based trust

should not only adapt to the estimated general trust, but to the single components. In the following we will introduce our approach of adapting the dialogue to single HCT components to foster the long-term relationship between user and system.

7.5.2 Integrating HCT Adaptation

While the adaptation of the task-oriented dialogue flow using a probabilistic model described previously is a good way to foster a high Interaction Quality, it still has its drawbacks. The integration of other *Adaptation Entities* like single HCT components (e.g., the user's perceived understandability of the system, the perceived system's technical competence, or the system's reliability) will result in a highly complex dialogue structure. Dialogue moves might only be suitable for a certain combination of AE values and, therefore, several permutations of AE values might be required, resulting in a highly complex dialogue flow. However, the most important dialogue strategy for coping with HCT issues is to provide explanations. Explanations are, however, related to the task-oriented dialogue at hand, not directly connected with them. This means that the flow of a task-oriented dialogue is not altered by including or augmenting the ongoing task-oriented dialogue with additional explanations. Therefore, we developed a dialogue system which incorporates a dedicated decision-making component dealing with domain-independent situations in HCI, which are also independent of the task-oriented dialogue. For example, the decrease of the user's perceived understandability of system can be estimated when observing user confusion. This type of situation is not domain-dependent and may be therefore handled by a domain-independent probabilistic decision model. Though the situation and the resulting decision may be domain-independent, it is important to note that the resulting dialogue strategy does not necessarily have to be domain-independent. In our example of observed user confusion, the dialogue strategy is to provide some explanation corresponding to the ongoing dialogue. Though the decision to provide explanations is domain-independent, the explanation itself should, at least in the optimal case, integrate domain knowledge.

In a nutshell, our goal was to integrate a dedicated probabilistic decision-making component, modeling domain-independent characteristics or adaptation entities, which can be plugged into existent dialogue systems. Hence, as the task-oriented dialogue should remain untouched, the kind of adaptation strategy has to be domain-independent and not conflicting but supplementary to the planned dialogue. The task of the dedicated component described here is to estimate the user's human–computer trust model and to augment the IQ-adaptive dialogue. Therefore, we perform an adaptation to dynamically changing information with the long-term goal of HCT adaptation.

7.5.2.1 Probabilistic HCT Model

A probabilistic model of the HCT relationship between user and dialogue system is used to determine strategies that lead in the long run to a trustworthy HCI. The AE, which are in this case the cognition-based components of HCT (i. e. perceived reliability, perceived understandability and perceived technical competence) are estimated by the observation of affective user states along with the dialogue history. This is described using a POMDP (cf. Sect. 7.4.2) and formalized in the Relational Dynamic Influence Diagram Language (RDDL) [31]. RDDL is a uniform language that allows an efficient description of POMDPs by representing its constituents (actions, observations, belief state) with variables. Figure 7.9 shows a simplified

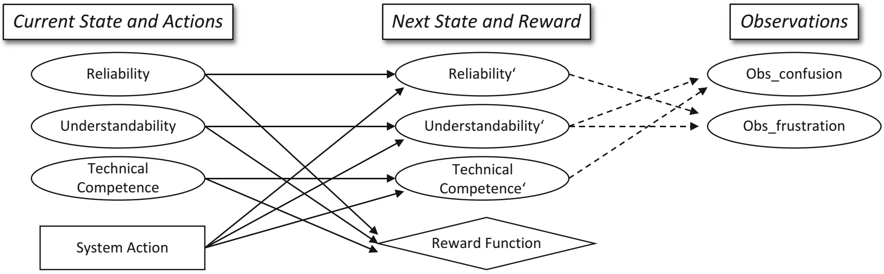

Fig. 7.9 This simplified figure of the POMDP model incorporates exemplary observations of the affective states *confusion* and *frustration* which influence the current state. These observations combined with the cognition-based components of trust (perceived reliability, perceived understandability and perceived technical competence), which are also part of the system state, and the current system action determine whether the next system action should be for example a transparency explanation

model of the in RDDL defined POMDP model. The system actions A are the dialogues presented to the user. These are the different goals of explanations (justification, transparency, conceptualization, relevance and learning). The POMDP model is a probabilistic representation of the domain, which determines when and how to augment the dialogue with explanations at run-time. Now, the quest is to define the reward function $R(s, a)$ in a way that it leads to an optimal flow of actions. I. e. the system should receive a penalty when the dimensions of HCT do not remain intact, and actions should incur a cost so that the system only executes them when the human–computer trust is endangered. For example, following the conducted experiment, the reward is defined in a way that providing transparency explanations is beneficial for increasing the state variables perceived understandability and reliability, though it also inflicts a cost for providing extra information dialogues.

The POMDP is then used by a planner [24, 36] to search for a policy that determines the system's behaviour. This policy is, e. g. represented as a decision tree that recommends the most suitable action based on the system's previous actions and observations. For example, a policy for a POMDP that models HCI with respect to HCT, can thus represent a decision tree which represents a guideline for a dialogue flow that ensures an intact HCT relationship.

7.5.2.2 Dialogue Augmentation Process

Integrating the probabilistic HCT model is done by plugging the component into the existent pipeline depicted earlier, by integrating the component in the system selection (see Fig. 7.10). The HCI is started using the task-oriented dialogue approach. The POMDP checks during the ongoing dialogue whether the user's trust or components of it are endangered. If this is the case, the proposed explanation has to be integrated into the ongoing task-oriented dialogue. Hence, the POMDP is used only for the augmentation of the task-oriented part of the dialogue with explanations and serves two purposes. First, it proposes the integration of domain-independent dialogue strategies into the task-oriented dialogue. Second, it selects what kind of explanation has to be selected or generated.

Though we know that explanations can help in keeping a system trustworthy (cf. Sect. 5.2), the question remains what kind of explanation is the best for which situation in HCI. Since the components of HCT are impaired differently in different situations, the system reaction to those situations should be directed as well. Hence, we conducted an experiment to test which explanations work best to deal with impairments of specific HCT components, to be able to generate directed explanation dialogue strategies.

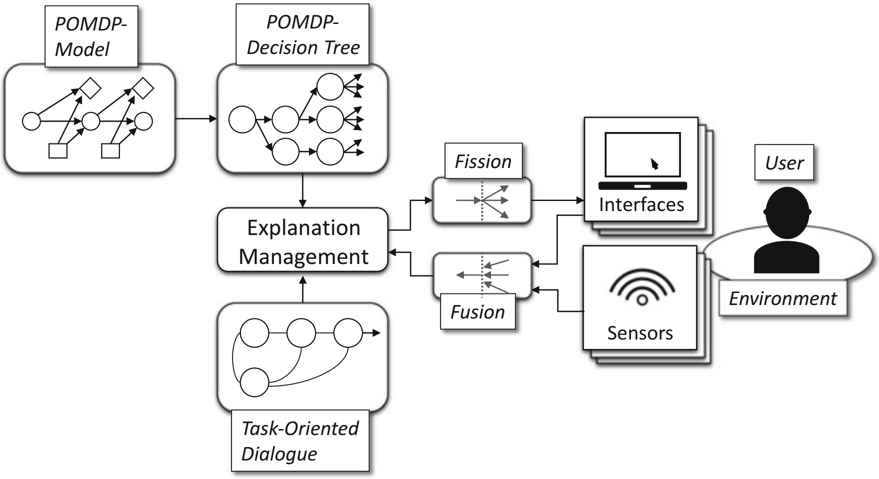

Fig. 7.10 This figure shows the architecture used for the augmentation of the task-oriented dialogue with domain-independent dialogue strategies

7.5.3 Experiments

The experiment was a web-based study inducing events to create unclear or not anticipated situations and to compare the effects of different explanations on the components of HCT. For our experiment, we concentrated on justification and transparency explanations. Justifications are the most obvious goal an explanation can pursue. The main idea of this explanation is to provide support for and increase confidence in given system advice or actions. The goal of transparency is to increase the user's understanding of how the system works. It may help to change the user's perception of the system from a black-box to a system the user can comprehend (i. e. a white box). Thereby, the user can build a mental model of the system and its underlying reasoning processes. Therefore, our hypothesis was that transparency explanations will perform best to recover the user's perceived understandability. The user's perceived reliability, measuring the impression of consistent functioning, was also expected to be recovered best by transparency explanations, because they explain how the system works, and thus explain why the system reacted inconsistently.

Design and Setup

The main objective for the participants was to organize four parties for friends or relatives in a web-based environment. They had to use the browser at home or the university to organize, for example, the music, select the type and amount of food or order drinks. The first two rounds were meant to go smoothly and were supposed to get the subject used to the system and in this way build a mental model of it. After the first two rounds, a HCT questionnaire [19] was presented to the user. As expected the users had built a relationship with the system by gaining an understanding of the systems processes. The next two rounds were meant to influence the HCT-relationship negative with incomprehensible, unexpected external events. These unexpected and incongruous system events in terms of the user's mental model would pro-actively influence the decisions and solutions the user could make to solve the task. Without warning, the user was overruled by the system and either simply informed by this change or was presented an additional justification or transparency explanation.

Results

One hundred and thirty-nine starting participants were distributed among the three test groups (no explanation, transparency, justifications). Ninety eight accomplished round 2, reaching the point when the external events were induced and 59 participants completed the experiment. The first main result was that 47 % from the group receiving no explanations quit during the critical rounds 3 and 4. However, if explanations were presented only 33 % (justifications) and 35 % (transparency) did quit. This means that even though the participants would encounter negative consequences of losing the reward money, they dropped out of the experiment. Therefore, we can state that the use of explanations in incomprehensible and unexpected situations can help to keep the HCI running.

The main results from the HCT questionnaires can be seen in Fig. 7.11. The data states that providing no explanations in rounds three and four resulted in a decrease in several components of trust. Therefore, we can conclude that the external events did indeed result in our planned negative change in trust. *Perceived understandability* diminished on average over the people questioned by 1.2 on a Likert scale with a range from 1 to 5 when providing no explanation at all

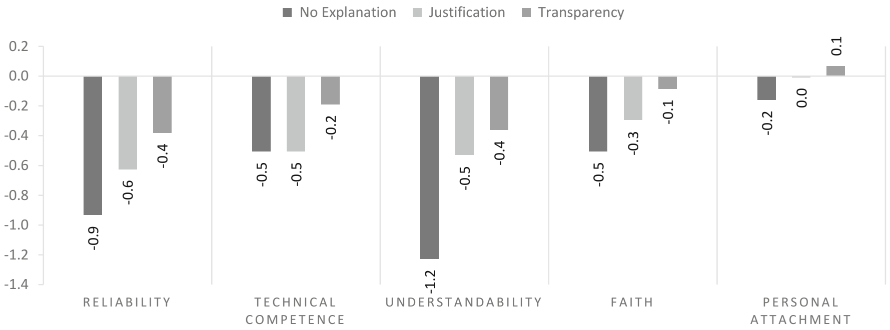

Fig. 7.11 This figure shows the changes of HCT components from round 2 to round 4. The scale was a 5 point Likert scale with e. g., 1 the system being not understandable at all and 5 the opposite

compared to only 0.4 when providing *transparency* explanations [no explanation vs. transparency $t(34) = -3.557, p = 0.001$], and on average by 0.5 with *justifications* [no explanation vs. justifications $t(36) = -2.023, p = 0.045$]. Omitting explanations resulted in an average decrease of 0.9 for the *perceived reliability*, with transparency explanations in a decrease of 0.4 and for *justifications* in a decrease of 0.6 [no explanation vs. transparency $t(34) = -2.55, p = 0.015$].

Discussion

These results support our hypothesis that transparency explanations can help to reduce the adverse effects of trust loss regarding the user's perceived understandability and reliability of the system in incomprehensible and unexpected situations. Particularly for the perceived understandability, meaning the prediction of future outcomes, transparency explanations fulfill their purpose in a good way. Additionally, they seem to help with the perception of a reliable, consistent system. Though justifications do not perform best in any situation, they are still helpful to keep an intact HCT relationship. Justifications have the significant advantage that, compared to transparency explanations, they do not require reasoning about system processes, but can be predefined. Hence, despite being less efficient in keeping HCT, they might still be a valuable option if the system's reasoning capabilities about inner processes are not available in the dialogue system.

In our dialogue system for transparency explanations, this includes the imparting of domain-dependent knowledge (i. e. reasoning about system processes and functionalities) to foster a deeper understanding of the system. For the other explanations predefined content suitable for the present domain can be selected and presented to the user. For example, if the user is frustrated during an instruction, a justification explanation for the present dialogue may be chosen from the domain, without the need to generate it at run-time.

In general the results show that it is worthwhile to augment ongoing dialogues with explanations to maintain HCT. However, HCT can only be estimated by the use of additional factors like affective user states. Hence, to estimate impairment in the HCT relationship, we need to incorporate observations prone to uncertainty, resulting in the necessity to model HCT in a probabilistic way.

Apart from adaptation to HCT, the present approach can be used for any domain-independent adaptation entity with a long-term goal. Probably it is even more advantageous for domain-independent AE which do not require the integration of domain-dependent content in the dialogue strategy.

7.6 Conclusion

While conventional non-adaptive Spoken Dialogue Systems are rigid and inflexible regarding the user's needs, introducing user-centred dynamic adaptation mechanisms into the spoken dialogue results in an improvement in user experience.

For this, a general model of adaptive dialogue management has been presented. This approach yields the possibility to adapt not only to short-term but also to long-term goals of adaptation to dynamically changing information. For distinguishing between long-term and short-term goals, two example implementations and corresponding experiments have been presented showing the general benefit of this approach. Furthermore, POMDP structures have been introduced into the dialogue management to handle additional uncertainty introduced by the entities of the adaptation mechanism.

References

1. Antoniou, G., van Harmelen, F.: Web ontology language: owl. In: Staab, S. (ed.) Handbook on Ontologies in Information Systems, pp. 76–92. Springer, Berlin (2003)
2. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines. ACM Trans. Intell. Syst. Technol. **2**, 27:1–27:27 (2011). Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm
3. Engelbrecht, K.P., Gödde, F., Hartard, F., Ketabdar, H., Möller, S.: Modeling user satisfaction with hidden markov model. In: SIGDIAL '09: Proceedings of the SIGDIAL 2009 Conference, pp. 170–177. Association for Computational Linguistics, Morristown, NJ, USA (2009)
4. Fraser, N.M.: The sundial speech understanding and dialogue project: results and implications for translation. In: Aslib Proceedings, vol. 46, pp. 141–148. MCB UP Ltd (1994)
5. Glass, A., McGuinness, D.L., Wolverton, M.: Toward establishing trust in adaptive agents. In: IUI '08: Proceedings of the 13th International Conference on Intelligent User Interfaces, pp. 227–236. ACM, NY, USA (2008)
6. Gnjatović, M., Rösner, D.: Adaptive dialogue management in the nimitek prototype system. In: PIT '08: Proceedings of the 4th IEEE Tutorial and Research Workshop on Perception and Interactive Technologies for Speech-Based Systems, pp. 14–25. Springer, Berlin (2008). doi:10.1007/978-3-540-69369-7_3
7. Hara, S., Kitaoka, N., Takeda, K.: Estimation method of user satisfaction using n-gram-based dialog history model for spoken dialog system. In: Calzolari, N., Choukri, K., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., Rosner, M., Tapias, D. (eds.) Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10). European Language Resources Association (ELRA), Valletta, Malta (2010)
8. Heinroth, T., Denich, D.: Spoken interaction within the computed world: evaluation of a multitasking adaption spoken dialogue system. In: 35th Annual IEEE Computer Software and Applications Conference (COMPSAC), Munich, Germany, pp. 134–143. IEEE (2011)
9. Higashinaka, R., Minami, Y., Dohsaka, K., Meguro, T.: Issues in predicting user satisfaction transitions in dialogues: Individual differences, evaluation criteria, and prediction models. In: Lee, G., Mariani, J., Minker, W., Nakamura, S. (eds.) Spoken Dialogue Systems for Ambient Environments. Lecture Notes in Computer Science, vol. 6392, pp. 48–60. Springer, Berlin (2010). doi:10.1007/978-3-642-16202-2_5
10. Higashinaka, R., Minami, Y., Dohsaka, K., Meguro, T.: Modeling user satisfaction transitions in dialogues from overall ratings. In: Proceedings of the SIGDIAL 2010 Conference, pp. 18–27. Association for Computational Linguistics, Tokyo, Japan (2010)
11. Hone, K.S., Graham, R.: Towards a tool for the subjective assessment of speech system interfaces (sassi). Nat. Lang. Eng. **6**(3–4), 287–303 (2000). doi:10.1017/s1351324900002497
12. Kaelbling, L.P., Littman, M.L., Cassandra, A.R.: Planning and acting in partially observable stochastic domains. Artif. Intell. **101**(1–2), 99–134 (1998)

13. Larsson, S., Traum, D.: Information state and dialogue management in the TRINDI dialogue move engine toolkit. Nat. Lang. Eng. **6**, 323–340 (2000)
14. Lee, S., Eskenazi, M.: An unsupervised approach to user simulation: toward self-improving dialog systems. In: Proceedings of the 13th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pp. 50–59. Association for Computational Linguistics (2012)
15. Lee, J.D., See, K.A.: Trust in automation: designing for appropriate reliance. Hum. Factors J. Hum. Factors Ergon. Soc. **46**(1), 50–80 (2004)
16. Lim, B.Y., Dey, A.K., Avrahami, D.: Why and why not explanations improve the intelligibility of context-aware intelligent systems. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '09, pp. 2119–2128. ACM, NY (2009)
17. Lindgaard, G., Dudek, C.: What is this evasive beast we call user satisfaction? Interact. Comput. **15**(3), 429–452 (2003)
18. Litman, D., Pan, S.: Designing and evaluating an adaptive spoken dialogue system. User Model. User-Adap. Inter. **12**(2–3), 111–137 (2002). doi:10.1023/a:1015036910358
19. Madsen, M., Gregor, S.: Measuring human-computer trust. In: Proceedings of the 11th Australasian Conference on Information Systems, pp. 6–8 (2000)
20. Mann, H.B., Whitney, D.R.: On a test of whether one of two random variables is stochastically larger than the other. Ann. Math. Stat. **18**(1), 50–60 (1947)
21. Mayer, R.C., Davis, J.H., Schoorman, F.D.: An integrative model of organizational trust. Acad. Manag. Rev. **20**(3), 709–734 (1995)
22. McTear, M.F.: Spoken Dialogue Technology – Toward the Conversational User Interface. Springer, London (2004)
23. Muir, B.M.: Trust in automation: part i. theoretical issues in the study of trust and human intervention in automated systems. In: Ergonomics, pp. 1905–1922 (1992)
24. Müller, F., Späth, C., Geier, T., Biundo, S.: Exploiting expert knowledge in factored POMDPs. In: Proceedings of the 20th European Conference on Artificial Intelligence (ECAI 2012), pp. 606–611 (2012)
25. Nothdurft, F., Honold, F., Kurzok, P.: Using explanations for runtime dialogue adaptation. In: Proceedings of the 14th ACM International Conference on Multimodal Interaction, pp. 63–64. ACM, New York (2012)
26. Oshry, M., Auburn, R., Baggia, P., Bodell, M., Burke, D., Burnett, D., Candell, E., Carter, J., Mcglashan, S., Lee, A., Porter, B., Rehor, K.: Voice extensible markup language (voicexml) version 2.1. Technical Report, W3C – Voice Browser Working Group (2007)
27. Pan, A.S., Litman, D.J., Pan, S.: Empirically evaluating an adaptable spoken dialogue system diane j. litman. In: Proceedings of the 7th International Conference on User Modeling, pp. 55–64 (1999)
28. Parasuraman, R., Riley, V.: Humans and automation: use, misuse, disuse, abuse. Hum. Factors J. Hum. Factors Ergon. Soc. **39**(2), 230–253 (1997)
29. Potel, M.: MVP: model-view-presenter the taligent programming model for C++ and java. Technical Report, Taligent Inc (1996). http://www.wildcrest.com/Potel/Portfolio/mvp.pdf
30. Raux, A., Bohus, D., Langner, B., Black, A.W., Eskenazi, M.: Doing research on a deployed spoken dialogue system: one year of let's go! experience. In: Proceedings of the International Conference on Speech and Language Processing (ICSLP) (2006)
31. Sanner, S.: Relational dynamic influence diagram language (rddl): language description (2010). Http://users.cecs.anu.edu.au/ ssanner/IPPC2011/RDDL.pdf
32. Schmitt, A., Schatz, B., Minker, W.: Modeling and predicting quality in spoken human-computer interaction. In: Proceedings of the SIGDIAL 2011 Conference, pp. 173–184. Association for Computational Linguistics, Portland, Oregon, USA (2011)
33. Schmitt, A., Schatz, B., Minker, W.: A statistical approach for estimating user satisfaction in spoken human-machine interaction. In: Proceedings of the IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT), pp. 1–6. IEEE, Amman, Jordan (2011)
34. Schmitt, A., Ultes, S., Minker, W.: A parameterized and annotated spoken dialog corpus of the cmu let's go bus information system. In: International Conference on Language Resources and Evaluation (LREC), pp. 3369–337 (2012)

35. Sidorov, M., Ultes, S., Schmitt, A.: Emotions are a personal thing: Towards speaker-adaptive emotion recognition. In: IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), pp. 4836–4840 (2014)
36. Silver, D., Veness, J.: Monte-carlo planning in large POMDPs. In: NIPS, pp. 2164–2172 (2010)
37. Spearman, C.E.: The proof and measurement of association between two things. Am. J. Psychol. **15**, 88–103 (1904)
38. Ultes, S., Minker, W.: Improving interaction quality recognition using error correction. In: Proceedings of the 14th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pp. 122–126. Association for Computational Linguistics (2013). http://www.aclweb.org/anthology/W/W13/W13-4018
39. Ultes, S., Minker, W.: Interaction quality estimation in spoken dialogue systems using hybrid-hmms. In: Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL), pp. 208–217. Association for Computational Linguistics (2014). http://www.aclweb.org/anthology/W14-4328
40. Ultes, S., Minker, W.: Managing adaptive spoken dialogue for intelligent environments. J. Ambient Intell. Smart Environ. **6**(5), 523–539 (2014). doi:10.3233/ais-140275
41. Ultes, S., Heinroth, T., Schmitt, A., Minker, W.: A theoretical framework for a user-centered spoken dialog manager. In: Proceedings of the Paralinguistic Information and its Integration in Spoken Dialogue Systems Workshop, pp. 241–246. Springer, New York (2011)
42. Ultes, S., Schmitt, A., Minker, W.: Attention, sobriety checkpoint! can humans determine by means of voice, if someone is drunk... and can automatic classifiers compete? In: Proceedings of the 12th Annual Conference of the International Speech Communication Association (INTERSPEECH 2011), pp. 3221–3224 (2011)
43. Ultes, S., ElChabb, R., Minker, W.: Application and evaluation of a conditioned hidden markov model for estimating interaction quality of spoken dialogue systems. In: Mariani, J., Devillers, L., Garnier-Rizet, M., Rosset, S. (eds.) Proceedings of the 4th International Workshop on Spoken Language Dialog System (IWSDS), pp. 141–150. Springer, New York (2012)
44. Ultes, S., Schmitt, A., Minker, W.: Towards quality-adaptive spoken dialogue management. In: NAACL-HLT Workshop on Future directions and needs in the Spoken Dialog Community: Tools and Data (SDCTD 2012), pp. 49–52. Association for Computational Linguistics, Montréal, Canada (2012). http://www.aclweb.org/anthology/W12-1819
45. Ultes, S., ElChabb, R., Schmitt, A., Minker, W.: Jachmm: a java-based conditioned hidden markov model library. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2013, Vancouver, BC, Canada, pp. 3213–3217. IEEE (2013)
46. Ultes, S., Schmitt, A., Minker, W.: On quality ratings for spoken dialogue systems – experts vs. users. In: Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 569–578. Association for Computational Linguistics (2013)
47. Ultes, S., Dikme, H., Minker, W.: Dialogue management for user-centered adaptive dialogue. In: Proceedings of the 5th International Workshop On Spoken Dialogue Systems (IWSDS) (2014). http://www.uni-ulm.de/fileadmin/website_uni_ulm/allgemein/2014_iwsds/iwsds2014_lp_ultes.pdf
48. Ultes, S., Dikme, H., Minker, W.: First insight into quality-adaptive dialogue. In: International Conference on Language Resources and Evaluation (LREC), pp. 246–251 (2014)
49. Ultes, S., Kraus, M., Schmitt, A., Minker, W.: On objective performance measures of quality-adaptive spoken dialogue—and their correlation with interaction quality. In: Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (2015, submitted)
50. Ultes, S., Platero Sánchez, M.J., Schmitt, A., Minker, W.: Analysis of an extended interaction quality corpus. In: Proceedings of the 6th International Workshop On Spoken Dialogue Systems (IWSDS) (2015). Accepted for publication
51. Vapnik, V.N.: The Nature of Statistical Learning Theory. Springer, New York (1995)
52. Williams, J.D.: At&t statistical dialog toolkit (2010). http://www2.research.att.com/sw/tools/asdt/

53. Williams, J.D.: Incremental partition recombination for efficient tracking of multiple dialog states. In: Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on, Dallas, Texas, pp. 5382–5385. IEEE (2010)
54. Williams, J., Young, S.: Partially observable markov decision processes for spoken dialog systems. Comput. Speech Lang. **21**(2), 231–422 (2007)
55. Young, S., Schatzmann, J., Weilhammer, K., Ye, H.: The hidden information state approach to dialog management. In: IEEE International Conference on Acoustics, Speech and Signal Processing, 2007 (ICASSP 2007), vol. 4 (2007)
56. Young, S.J., Gačić, M., Keizer, S., Mairesse, F., Schatzmann, J., Thomson, B., Yu, K.: The hidden information state model: a practical framework for POMDP-based spoken dialogue management. Comput. Speech Lang. **24**(2), 150–174 (2010)