A multi-lingual evaluation of the vAssist spoken dialog system: comparing Disco and RavenClaw

Javier Mikel Olaso, Pierrick Milhorat, Julia Himmelsbach, Jérôme Boudy, Gérard Chollet, Stephan Schlögl, Maria Inès Torres

A Multi-lingual Evaluation of the vAssist Spoken Dialog System. Comparing Disco and RavenClaw

Javier Mikel Olaso, Pierrick Milhorat, Julia Himmelsbach, Jérôme Boudy, Gérard Chollet, Stephan Schlögl and María Inés Torres

Abstract vAssist (Voice Controlled Assistive Care and Communication Services for the Home) is a European project for which several research institutes and companies have been working on the development of adapted spoken interfaces to support home care and communication services. This paper describes the spoken dialog system that has been built. Its natural language understanding module includes a novel reference resolver and it introduces a new hierarchical paradigm to model dialog tasks. The user-centered approach applied to the whole development process led to the setup of several experiment sessions with real users. Multilingual experiments carried out in Austria, France and Spain are described along with the analyses and results in terms of both system performance and user experience. An additional experimental comparison of the RavenClaw and Disco-LFF dialog managers built into the vAssist spoken dialog system highlighted similar performance and user acceptance.

Index Terms: spoken dialog systems, dialog management, real-user experiments

Javier Mikel Olaso, María Inés Torres
Universidad del País Vasco UPV/EHU, Spain, e-mail: javier-mikel.olaso@ehu.es,manes.torres@ehu.es

Pierrick Milhorat, Gérard Chollet
Télécom ParisTech, France e-mail: milhorat@telecom-paristech.fr,chollet@telecom-paristech.fr

Julia Himmelsbach
AIT Austrian Institute of Technology GmbH, Austria e-mail: julia.himmelsbach@ait.ac.at

Jérôme Boudy
Télécom SudParis, France e-mail: jerome.boudy@telecom-sudparis.eu

Stephan Schlögl
MCI Management Center Innsbruck, Austria e-mail: stephan.schloegl@mci.edu

1 Introduction

The vAssist project [1] aims at providing specific voice controlled home care and communication services for two target groups of older persons: seniors suffering from chronic diseases and persons suffering from (fine) motor skills impairments. The main goal is the development of simplified and adapted interface variants for tele-medical and communication applications using multilingual natural speech and voice interaction (and supportive graphical user interfaces where necessary) [2, 3]. The vAssist consortium consists of research institutes and companies from Austria, France and Italy. Toward the end of the project, the University of the Basque Country was included so as to expand the perimeter to Spanish speaking users.

2 Related Work

A Spoken Dialog System (SDS) is a system providing an interface to a service or an application through a dialog. An interaction qualifies as dialog when it exceeds one turn. It requires to keep track of the dialog state, including the history of past turns, in order to select the appropriate next step.

Those systems do not usually consist of a single component but comprise several specialized programs combined in order to recognize the speech, extract the relevant information in the transcriptions, act on back-end services, decide on the best next step, generate system responses and synthesize speech.

JUPITER [4] was one of the first SDSs released to the public. The phone-based weather information conversational interface has received over 30 000 calls between 1997 and 1999. Earlier, researchers from Philips implemented an automatic train timetable information desk for Germany [5]. More recently, Carnegie Mellon University provided Olympus [6], which has been used to build systems like the Let's Go! Bus Information System [7], leading to the biggest corpus of man-machine dialogs with real users publicly available today. Recent platforms for developing spoken dialogue systems include the Opendial toolkit [8] and the architecture developed by the University of Cambridge [9] for its startup VocalIQ.

ELIZA [10] is considered by many as the first dialog system. The core of the system was based on scripts, which associated a system's response by looking for a pattern in the input. Larsson and Traum argued that the state of the dialog, including its history, may be represented as the sum of the so far exchanged information [11]. An Information State (IS) designer defines the elements of the information relevant to a dialog, a set of update rules and an update strategy. An example-based dialog manager (DM) [12] constructs a request to a database from the annotated input Dialog Act (DA). The database stores examples seen in the interaction data so that the algorithm looks for the most similar entry and then executes the system's associated

action. On the other hand, plan-based DMs [13, 14] require a pre-programmed task model.

On the stochastic side, Markov Decision Processes (MDPs) represent a statistical decision framework to manage dialogs [15, 16]. Here, the dialog state space contains all the states the dialog may be in and the transitions dependent on the user inputs. The behavior of a DM based on MDPs is defined by a strategy which associates each state to an executable action. Statistical methods used for dialog management also include Stochastic Finite-State models [17, 18, 19] and SemiMDPs [20]. Finally the state-of-the-art POMDP extends the MDPs hiding the states which emit observations according to a probabilistic distribution [21, 22, 9]. This additional layer encodes the uncertainty about the Natural Language Understanding (NLU) and, in the case of SDSs, the Automatic Speech Recognition (ASR). Within a theoretical framework the proposal of a global statistical framework, allowing for optimization, is highlighted by POMDP [21]. However, practical POMDP-based DMs are currently limited in the number of variables and by the intractability of the computing power required to find an optimal strategy [23, 22, 9].

In vAssist the development context, along with the difficulty to collect 'real' training dialogs, favored the use of a deterministic control formalism. This was also motivated by the overall requirements of the system it had to be integrated with.

3 Main Goals and Contributions

This paper describes the vAssist SDS and presents the results of the final system evaluation. The vAssist DM system is based on an open and adaptative software architecture that allows for an easy configuration of the DMs according to the targeted scenario, user needs and context. In accordance with [24], the novelties of the vAssist SDS are the Semantic Unifier and Reference Resolver (SURR) defined in the natural understanding module and the Link-Form Filling (LFF) concept proposed to model the task (for both cf. Section 4.6). The vAssist prototype is based on the Disco plan-based DM and the LFF task model. For comparison purposes we have also integrated an alternative, plan-based DM, i.e. Ravenclaw (cf. Section 4.7).

The main contribution of this work is therefore a multilingual lab evaluation of the final vAssist assistive home care and communication service applications running on a smart-phone. Such was carried out with real users in Austria, France and Spain (cf. Section 5). As an additional contribution the evaluation has been carried out in terms of system performance and user experience (cf. Section 6). The final contribution of this work is the experimental comparison of the Disco-LFF DM and the Ravenclaw DM working within the same SDS architecture, dealing with the same task and language (i.e. Spanish), and interacting with the same users (also cf. Section 6).

4 System Description

The vAssist SDS extends the usual chained design. Components were split into modified sub-modules and new processes were integrated into a state-of-the-art workflow chain. Fig. 1 shows the resulting SDS architecture.

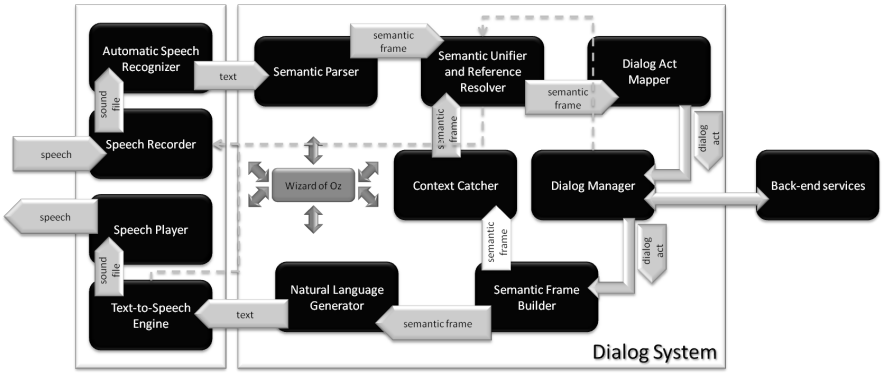

Fig. 1 Architecture of the platform

4.1 Speech Recognition

The system uses the Google Speech API where an HTTP POST request transmits the signal segment to be recognized. The API returns the n-best hypotheses (note: n is a parameter of the request), as well as the confidence score for the best one. An empty result is returned when the segment cannot be recognized with enough confidence, i.e. when it does not contain speech.

4.2 Natural Language Generation and Text-to-Speech

A simple but effective solution to produce natural language utterances conveying the DM's messages was targeted. Input messages are Semantic Frames (SFs). The engine is fed with a set of templates that consist of a title (identical to an SF's goal) associated with an utterance, and whose parts may be replaced by slot names or slot name-value pairs. The result is a natural language utterance to be synthesized or displayed on a screen.

MaryTTS [25], an open-source speech synthesis framework maintained by the Cluster of Excellence MMCI and the DFKI, is used for synthesis. It offers pre-built voice models for different languages as well as tools to create

and manipulate them. The MaryTTS module is a client connected to a generating server (hosted local or remote). A request containing the text to be synthesized with additional prosodic information is sent to the central server, which returns the speech stream. The text-to-speech module of the present platform is a basic client program embedded into an ActiveMQ wrapper.

4.3 Semantic Parsing

The semantic parser, which gets inputs from the ASR, associates semantic labels to text utterances (or parts of them). The most commonly used parsing techniques are based on context-free grammars or probabilistic context-free grammars, which are either hand-coded, based on the analysis of collected dialog data, or designed by experts.

Our semantic parser, integrates the algorithm proposed by [26], which is the application of the work from [27]. Instead of matching whole sentences with parse structures, the algorithm looks for patterns in chunks of the text-level utterance and in the temporary (i.e. currently assigned) SF. The module applies an ordered set of conditional rules, which is learned from data.

4.4 Semantic Unification and Resolution

The Semantic Unifier and Reference Resolver (SURR) holds a rather simplistic forest of nodes which is used to mine the dialog history, incorporate external information sources and add local turn context. It is the meeting point of the user's semantic frames, the external environment sensors and functions, the dialog history, and the links generated by the context catcher.

At its core the SURR embeds a forest structure. Trees consist of hierarchies of fully or partially defined SFs (some nodes are calls to external systems or services). When requested, the SURR may dynamically modify (remove/add) branches of the forest. The top node of a hierarchy defines the root.

The SURR algorithm tries to find a unique path from an input SF, i.e. from the parsed user input, to nodes of the forest, to a root node. Going up the trees, the algorithm applies the optional operations held on branches.

Reaching a root node equals the user input being contextualized [24]. In case the algorithm cannot find such a path, i.e. the SURR fails to produce a suitable SF (given the current context and available knowledge), a "NoMap" SF is generated to signal a 'non-understanding' to consecutive components.

4.5 Dialog Act Mapping

As a last stage of the NLU processing, the dialog act mapping is performed. Once an input has been parsed, external and local references have been resolved, and the semantic level has been unified, the ultimate step is to convert the SF into a DA. Following an input the mapper retrieves a set of available DAs. Then it looks for a unique match between the SF and the set of DAs.

4.6 Dialog Management Based on Disco

The core of the implemented DM is based on Disco [13], an open-source dialog management library, whose algorithm processes task hierarchy models. A dialog model is a constrained XML tree of tasks. The plan recognizer uses the recipes defined in the dialog models and this dialog state to select the best available plans for the tasks in the stack. Then the reasoning engine selects the most appropriate next step.

In an attempt to overcome the hurdles inherent to the specification of task models, the dialog modeling paradigm was shifted to a Linked-form-filling (LFF) one. Form-filling dialogs are based on structures containing sets of fields, which the user needs to provide a value for in order to trigger a terminal action. The order in which the DM asks for the values is not predefined. The user may define multiple field values within a single utterance/turn.

The LFF language offers to combine these properties with the ability to trigger actions at any point of the dialog and the inclusion of subforms. Furthermore, fields and subforms can be optional, i.e. either be ignored when unset or proposed to the user. Here, we use the unlimited depth of a task model to circle tasks while keeping a sequencing order; i.e. the link between two task nodes is a reference, hence a node can point to its 'parent' node.

The aim of the LFF language is to offer a somehow simpler design method to a powerful standard dialog modeling specification. Since it is also an XML based language we opted for XSLT to convert an LFF document into a compliant dialog model. A number of rules have been defined to create a well-formed LFF document. Doing this, the relative reduction in terms of code size and task hierarchy depth was 76% and 77%, respectively.

4.7 Dialog Management Based on RavenClaw

RavenClaw (part of the CMU Communicator system [28]) is a task-independent DM. It manages dialogues using a task tree and an agenda.

The task tree is basically a plan to achieve the overall dialog task. At runtime, the tree is traversed recursively from left to right and from top to

bottom. The execution of the dialog ends when the bottom-right node has been reached. During this process, loops and conditional control mechanisms may be added to the nodes in order to alter the normal exploration of the tree, allowing the definition of more complex dialog structures.

The second defining structure, the agenda, is an ordered list of agents that is used to dispatch inputs to appropriate agents in the task tree. It is recomputed for every turn and the current agent is placed on top of the stack. Inputs are matched to successive items on the agenda. When a match occurs the corresponding agent is activated with the matching concepts as inputs of the dialog. An agent may not consume all input concepts and thus remaining concepts are passed further down the agenda until agents can consume them.

In order to integrate RavenClaw in the architecture shown in Figure 1, the original Disco-LFF DM was substituted by a module responsible for translating the message format defined by RavenClaw to the message format defined by the Disco-based component and vice versa.

5 Task and Experimental Scenarios

Our goal was to empirically evaluate the operation of the developed voice-controlled application running on a smartphone under standardized conditions. For these experiments several scenarios were defined and implemented. This is the (translated) description of the scenarios as given to the participants:

- The Prescription Management enables to monitor medical prescriptions and individual intake times. To evaluate this scenario, participants were asked to add a new predefined prescription to the application database and to set a reminder for it (AP). The app requests information regarding name of medication, quantity, dosage form, frequency, and time of intake.
- The Health Report (HR) provides an overview of physiological data. Participants filled in predefined glycaemia and blood pressure data.
- The Sleep Report (SR) monitors sleep quality. The following data was provided by the users: the time he/she went to bed, the time he/she fell asleep, and their wake-up times. Participants also reported awake periods at night and the total number of hours slept. Finally, users were asked to rate their well-being on a six-point scale. Furthermore, the evaluation included setting a reminder to remember completing the sleep report (SRR).
- Fitness Data Management consists of reporting daily activities (FD) and setting reminders for the reports. Within the evaluation, participants were asked to enter a new report including the duration of their fitness activity.
- The Communication Services include sending messages (SM) and initiating phone calls (PC). Participants were asked to test both functions.

6 Experimental Evaluation

Two series of experiments were carried out: We evaluated the vAssist system including the Disco-LFF engine in three languages: French, German and Spanish. Further, we compared the RavenClaw and Disco-LFF DMs built into the vAssist system with Spanish users.

Sixteen users took part in the experiments in each of the trial sites. In France, 14 male and 2 female persons between 65 and 90 years (Mn=77.0) participated in the study. In Austria, 8 male and 8 female participants between 60 and 76 (Mn=68.0) years old took part. The Spanish trial site included 12 males and 4 females between 26 and 69 (Mn=39.6) years.

Users were first shown the smartphone application, followed with a short demonstration and some usage advices. The experimental scenarios were then carried out without any other introduction than the simple description of the goal. It was up to the user to figure out how to perform each task.

The system's performance was measured in terms of Task Completion (TC), i.e. success rate, and Average Dialog Length (ADL), i.e. efficiency. TC evaluates the success rate of the system in providing the user with the requested information, based on the total number of dialogues carried out and the number of successful dialogues achieved for a specific task. ADL is the average number of turns in a successful task.

For the subjective measures, a set of standardized questionnaires was applied. The standard Single Ease Questionnaire (SEQ) [29], the System Usability Scale (SUS) [30] and the Subjective Assessment of Speech System Interfaces (SASSI) [31] questionnaire were used to evaluate the vAssist system with the Disco-LFF DM. A custom set of questions was used to compare the Disco-LFF-based DM with the Ravenclaw-based DM. Results of the SEQ, SUS and SASSI are not given for Spanish, as for this language no localized mobile application interface was available.

6.1 System Performance

The first series of experiments was carried out in France, Austria and Spain, evaluating the vAssist system with the Disco-LFF DM. Table 1 shows the system performance evaluation in terms of TC and ADL values.

Table 1 reveals good TC rates, with the French version being the one generating the highest system performance and the Spanish version the one producing the lowest. Surprisingly, our results show that the vAssist system performance is not better for younger users (Spain: Mn=39.6 years) than for older ones (France: Mn=77 years). Language dependent modules, i.e. the ASR and, more importantly, the NLU, were more robust in French and German. Spanish results suffered from a less robust semantic parser and the missing mobile UI, leading to a higher number of turns to achieve the task goals.

	French		German		Spanish	
	TC	ADL	TC	ADL	TC	ADL
AP	93.33%	8.00	88.88%	8.18	84.00%	13.62
HR	100.00%	3.15	93.33%	3.78	100.00%	4.41
SR	91.66%	7.81	100.00%	7.25	100.00%	10.18
SRR	83.33%	3.40	100.00%	3.50	87.50%	5.78
FD	100.00%	3.00	66.66%	3.00	93.75%	4.53
SM	100.00%	3.86	100.00%	4.62	100.00%	6.21
PC	100.00%	1.92	100.00%	1.82	100.00%	2.00
Average	97.12%	4.44	95.18%	4.73	92.19%	6.21

Table 1 TC and ADL of the vAssist system using the Disco-LFF DM.

	Disco-LFF DM		RavenClaw DM	
	TC	ADL	TC	ADL
AP	84.00%	13.62	94.40%	15.64
HR	100.00%	4.41	100.00%	4.90
SR	100.00%	10.18	83.30%	11.90
SRR	87.50%	5.78	75.00%	6.08
FD	93.75%	4.53	92.80%	4.30
SM	100.00%	6.21	100.00%	6.64
PC	100.00%	2.00	100.00%	2.42
Average	92.19%	6.21	89.90%	6.60

Table 2 Comparing the Disco-LFF and RavenClaw DMs.

6.2 Task Easiness and Usability

Besides performance, the perceived task easiness is considered an important factor influencing user experiences [32]. This aspect was measured right after each task via the SEQ using a 7-point semantic differential (very difficult - very easy). The analysis revealed a sufficient ease of use for each task; i.e mean ratings for the Prescription Management and for sending a message were 4.94. Initiating a phone call and the Health Report were rated 5.06.

To obtain insights regarding the prototype's usability, learnability, and intuitivity, the SUS was used. SUS scores fall between 0 and 100; the higher the score the better. The values for Austria and France were 68 (sd = 17.2) and 70 (sd = 11.5), respectively. Hence, even though the perceived easiness of single tasks was good, the overall system experience could still be improved.

6.3 Speech Assessment

The SASSI questionnaire was employed to examine the interaction quality. The analysis provides developers with an assessment of the system along several axes such as easiness, friendliness, speed, etc.

Results indicate that both "Response Accuracy" (Austria: 4.27, France: 3.99) and "Speed" (Austria: 4.64, France: 4.19) were judged neutral. The analysis of the French sample reveals that "Likeability" (4.9) and "Cognitive Demand" (5.15) were fair. In contrast, the Austrian participants rated these factors as good (Likeability: 5.28, Cognitive Demand: 5.15). Hence, participants liked the system and were not overwhelmed by its cognitive demands.

6.4 Disco-LFF and RavenClaw DM Comparison

The second series of experiments was carried out in Spanish only. Note that both DMs were integrated in the same architecture (Figure 1), i.e. only the task planification and the agent execution differed. Each user carried out the scenarios defined in Section 5 with either of the DMs. Table 2 shows the system performance achieved by both systems in terms of TC and ADL, for each of the defined subscenarios. Both metrics show similar behavior for the Disco-LFF and the Ravenclaw DM. A Z-test comparing the average TC proportions and the ADL means showed no statistically significant difference between the two DMs (p-value = 0.05). A detailed scenario-based analysis showed, however, differences between TC values in the AP and the SR scenarios, which correspond to longer dialogs in terms of the ADL metric. A previous series of experiments has furthermore highlighted a certain lack of robustness exhibited by the language dependent modules of the Spanish vAssist version. This issue was more evident in longer dialogs (AP and SR).

As there was no mobile UI for the Spanish language, the user experience was evaluated trough a set of direct questions regarding the system efficiency, usability and user satisfaction. Task easiness received an average score of 3.00 for the Disco-LFF DM and 3.14 for the RavenClaw DM. The respective satisfaction scores were 3.57 and 3.43 and efficiency scored 3.28 and 3.14.

7 Conclusion

This paper had two objectives. First, we reported on the results of the final lab evaluation of the vAssist system, and second we compared the system's core DM implementation with a publicly deployed one.

Despite minimal differences between languages, the vAssist SDS performances proved to be sufficient for its target users, i.e. seniors suffering from chronic diseases and persons suffering from (fine) motor skills impairments.

The DM comparison showed similar performance and subjective experience for the system with the Disco-LFF DM and the one with RavenClaw, promoting the Disco-LFF as a valid alternative to existing DM approaches.

8 Acknowledgements

The presented research is conducted as part of the vAssist project (AAL-2010-3-106), which is partially funded by the European Ambient Assisted Living Joint Programme and the National Funding Agencies from Austria, France and Italy. It has also been partially supported by the Spanish Ministry of Science under grant TIN2014-54288-C4-4-R and by the Basque Government under grant IT685-13.

References

1. Gérard Chollet, Daniel R.S. Caon, Thierry Simonnet, and Jérôme Boudy, "vassist: Le majordome des personnes dépendantes.," *2e Conférence Internationale sur l'Accessibilité et les Systémes de Suppléance aux personnes en Handicap*, 2011.
2. Pierrick Milhorat, Stephan Schlögl, Gérard Chollet, and Jérôme Boudy, "Un systéme de dialogue vocal pour les seniors: Études et spécifications.," *Journées d'étude sur la TéléSanté*, 2013.
3. Stephan Schlögl, Pierrick Milhorat, and Gérard Chollet, "Designing, building and evaluating voice user interfaces for the home.," *Workshop on Methods for Studying Technology in the Home at the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI'13)*, 2013.
4. Victor Zue, Stephanie Seneff, James R. Glass, Joseph Polifroni, Christine Pao, Timothy J. Hazen, and Lee Hetherington, "JUPlTER: a telephone-based conversational interface for weather information," *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 1, pp. 85–96, 2000.
5. Harald Aust, Martin Oerder, Frank Seide, and Volker Steinbiss, "The Philips automatic train timetable information system," *Speech Communication*, vol. 17, no. 3-4, pp. 249–262, 1995.
6. Dan Bohus, Antoine Raux, Thomas K Harris, Maxine Eskenazi, and Alexander I. Rudnicky, "Olympus: an open-source framework for conversational spoken language interface research," in *Workshop on Bridging the Gap: Academic and Industrial Research in Dialog Technologies*, 2007, pp. 32–39.
7. Antoine Raux, Brian Langner, and Dan Bohus, "Let's go public! taking a spoken dialog system to the real world," in *InterSpeech*, 2005.
8. Pierre Lison, "A hybrid approach to dialogue management based on probabilistic rules," *Computer Speech & Language*, vol. 34, no. 1, pp. 232 – 255, 2015.
9. Steve Young, Milica Gašić, Blaise Thomson, and Jason D. Williams, "POMDP-based Statistical Spoken Dialog Systems: A review," *Proceedings of the IEEE*, vol. 101, no. 5, pp. 1160–1179, 2013.
10. Joseph Weizenbaum, "ELIZA- A computer program for the study of natural language communication between man and machine," *Communications of the ACM*, vol. 9, no. 1, pp. 36–45, 1966.
11. Staffan Larsson and David Traum, "Information state and dialogue management in the TRINDI Dialogue Move Engine Toolkit," *Natural language engineering*, vol. 6, pp. 323–340, 1998.
12. Cheongjae Lee, Sangkeun Jung, Jihyun Eun, Minwoo Jeong, and Gary Geunbae Lee, "A situation-based dialogue management using dialogue examples," in *International Conference on Acoustics, Speech and Signal Processing*, 2006, pp. 69–72.
13. Charles Rich, "Building task-based user interfaces with ANSI/CEA-2018," *IEEE Computer*, , no. 8, pp. 20–27, 2009.
14. Dan Bohus and Alexander I. Rudnicky, "The RavenClaw dialog management framework: Architecture and systems," *Computer Speech & Language*, vol. 23, no. 3, pp. 332–361, 2009.
15. Esther Levin, Roberto Pieraccini, and Wieland Eckert, "Using Markov Decision Process for Learning Dialogue Strategies," in *International Conference on Acoustics, Speech and Signal Processing*, 1998, pp. 201–204.
16. Steve Young, "Probabilistic Methods in SpokenDialogue Systems," *Philosophical Transactions of the Royal Society of London*, 2000.
17. David Griol, , Lluis Hurtado, Encarna Segarra, and Emilio Sanchis, "A statistical approach to spoken di- alog systems design and evaluation," *Speech Communication*, vol. 50, pp. 666–682, 2008.
18. Maria Inés Torres, "Stochastic Bi-Languages to model Dialogs," in *International Conference on Finite State Methods and Natural Language Processing*, 2013, pp. 9–17.

19. Fabrizio Ghigi and Maria Inés Torres, "Decision making strategies for finite state bi-automaton in dialog management.," in *International Workshop Series on Spoken Dialogue Systems Technology, IWSDS*, 2015, pp. 308–312.
20. Heriberto Cuayáhuitl, Steve Renals, Oliver Lemon, and Hiroshi Shimodaira, "Evaluation of a hierarchical reinforcement learning spoken dialogue system," *Computer, Speech and Language*, vol. 24, pp. 395–429, 2010.
21. Jason D. Williams and Steve Young, "Partially observable Markov decision processes for spoken dialog systems," *Computer Speech & Language*, vol. 21, no. 2, pp. 393–422, 2007.
22. Filip Jurčíček, Blaise Thomson, and Steve Young, "Reinforcement learning for parameter estimation in statistical spoken dialogue systems," *Computer Speech & Language*, vol. 26, no. 3, pp. 168–192, 2011.
23. Paul A. Crook, Brieuc Roblin, Hans-Wolfgang Loidl, and Oliver Lemon, "Parallel computing and practical constraints when applying the standard pomdp belief update formalism to spoken dialogue management," in *Proceedings of the Paralinguistic Information and its Integration in Spoken Dialogue Systems Workshop*, Ramn Lpez-Czar Delgado and Tetsunori Kobayashi, Eds., pp. 189–201. Springer New York, 2011.
24. Pierrick Milhorat, *An Open-source Framework for Supporting the Design and Implementation of Natural-language Spoken Dialog Systems*, Ph.D. thesis, Télécom ParisTech – 46, rue Barrault – 75013 Paris, 2015.
25. Marc Schröder and Jürgen Trouvain, "The German text-to-speech synthesis system MARY: A tool for research, development and teaching," *International Journal of Speech Technology*, vol. 6, no. 4, pp. 365–377, 2003.
26. Filip Jurčíček, François Mairesse, Milica Gašić, Simon Keizer, Blaise Thomson, Kai Yu, and Steve Young, "Transformation-based Learning for semantic parsing," in *InterSpeech*, 2009, pp. 2719–2722.
27. Eric Brill, "Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging," *Computational linguistics*, vol. 21, no. 4, pp. 543–565, 1995.
28. Dan Bohus and Alexander I. Rudnicky, "The ravenclaw dialog management framework: Architecture and systems," *Comput. Speech Lang.*, vol. 23, no. 3, pp. 332–361, july 2009.
29. Donna Tedesco and Tom Tullis, "A comparison of methods for eliciting post-task subjective ratings in usability testing," *Usability Professionals Association (UPA)*, vol. 2006, pp. 1–9, 2006.
30. John Brooke, "SUS-a quick and dirty usability scale," *Usability evaluation in industry*, vol. 189, no. 194, pp. 4–7, 1996.
31. Kate S. Hone and Robert Graham, "Towards a tool for the subjective assessment of speech system interfaces (SASSI)," *Natural Language Engineering*, vol. 6, pp. 287–303, 2000.
32. Viswanath Venkatesh and Hillol Bala, "Technology Acceptance Model 3 and a Research Agenda on Interventions," *Decision Sciences*, vol. 39, no. 2, pp. 273–315, 2008.