CSE: U: Mixed-initiative Personal Assistant Agents

Joshua W. Buck
University of Dayton
jbuck1@udayton.edu
Advisors: Saverio Perugini and Tam V. Nguyen

ABSTRACT

Specification and implementation of flexible human-computer dialogs is challenging because of the complexity involved in rendering the dialog responsive to a vast number of varied paths through which users might desire to complete the dialog. To address this problem, we developed a toolkit for modeling and implementing task-based, mixed-initiative dialogs based on metaphors from lambda calculus. Our toolkit can automatically operationalize a dialog that involves multiple prompts and/or sub-dialogs, given a high-level dialog specification of it. The use of natural language with the resulting dialogs makes the flexibility in communicating user utterances commensurate with that in dialog completion paths—an aspect missing from commercial assistants like *Siri*. Our results demonstrate that the dialogs authored with our toolkit support the end user's completion of a human-computer dialog in a manner that is most natural to them—in a mixed-initiative fashion—that resembles human-human interaction.

Keywords

Bag of words model; function currying; functional programming; human-computer dialogs; interactive voice response systems; k-nearest-neighbor classifier; lambda calculus; mixed-initiative dialogs; mixed-initiative interaction; natural language processing; partial evaluation.

1. PROBLEM AND MOTIVATION

Human-computer dialogs, which are used to improve information access from smart phone apps, ATMs, and airport kiosks to intelligent tutoring/training, are woven into the fabric of our daily interactions with computer systems. The problem addressed through our research is the automatic construction of mixed-initiative, human-computer dialog systems (see Fig. 1). *Mixed-initiative interaction* is a flexible interaction strategy whereby the user and the system engage as equal participants in an activity and take turns exchanging initiative as the user progresses toward the satisfaction of a particular goal facilitated by her interaction with the system [14]. Since '[a]uthoring a dialogue is like writing a movie script with many different endings' [17], 'a central problem for mixed-initiative dialogue management is coping with utterances that fall outside of the expected sequence of the dialogue' [32] (e.g., see dialog in Fig. 2). Thus, '[d]eveloping a mixed-initiative dialog system is a complex task' [15] and '[c]reating an actual dialog system involves a very intensive programming effort' [13].

This problem is important since dialog has been established as an effective mechanism through which to achieve a rich form of human-computer interaction (e.g., dialog-based systems are now used in areas as critical as health care [23]). Being able to automatically create a dialog system in a new domain is important. We feel that i) a *mixed-initiative mode of interaction* driven by user utterances and ii) communicated through the use of *natural language* (see lower right hand cell of Table 1) is the key to the effectiveness and widespread adoption of personal assistant technologies. This extended abstract discusses a research project that addresses (i) and (ii).

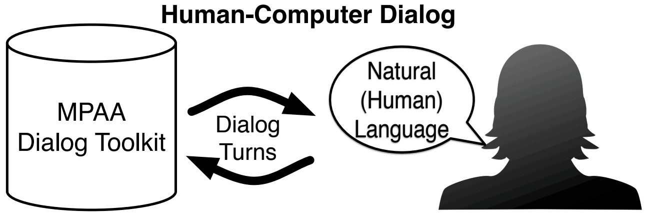


Figure 1: Use of natural language, mixed-initiative dialog.

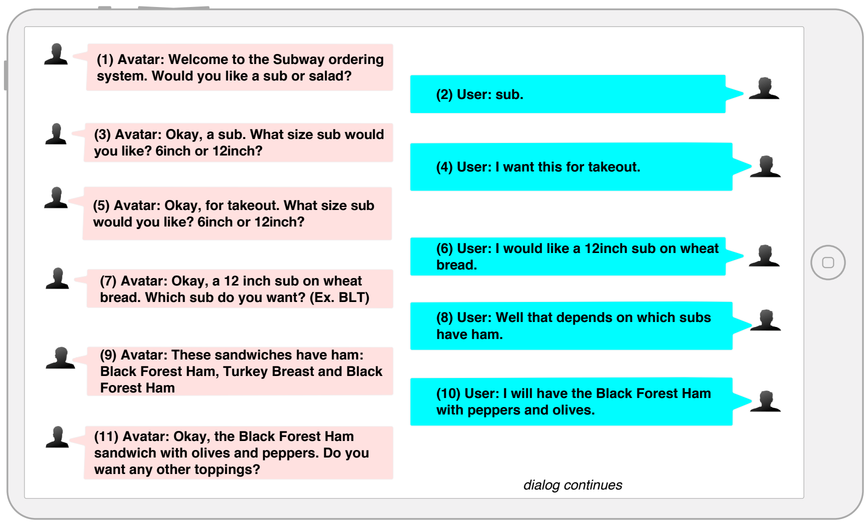
Degree of MII	no NL ⟵	*Degree of Natural Language* ···	⟶ complete NL
fixed	voice commands		*Stanford CoreNLP*[21]
↑			
⋮		*Apple's Siri*	
⋮			*NADIA* [2]
↓			
mixed	**our prior research** [5, 26, 27] ⇝		**our current research**[6]

Table 1: A design space for dialog-based systems.

2. BACKGROUND AND RELATED WORK

Our research lies in the area of automatic mixed-initiative, dialog system construction, with a particular focus on the dialog management component (i.e., knowing what to prompt for and/or accept next based on what has already been communicated and the current utterance) of a dialog-based system [18]. Dialog-based systems can be classified based on the degree of flexibility and natural language supported (see Table 1). The increasing popularity of personal assistant technologies, such as *Siri*, *Google Now*, *Cortana*, and *Alexa*, is driving and expanding progress toward the long-standing, albeit challenging, goal of applying artificial intelligence to build human-computer dialog systems capable of understanding natural language [19]. There are multiple research projects which seek to automate the implementation of flexible, dialog-based systems [15, 17] What sets our approach apart from these projects is our use of language-based concepts and operators, rather than task structures, to model dialog, which we discuss below.

Our work lies specifically in the *dialog management* area of dialog-based systems. The dialog management component plays a central role in the architecture of a traditional



(1) Avatar: Welcome to the Subway ordering system. Would you like a sub or salad?

(2) User: sub.

(3) Avatar: Okay, a sub. What size sub would you like? 6inch or 12inch?

(4) User: I want this for takeout.

(5) Avatar: Okay, for takeout. What size sub would you like? 6inch or 12inch?

(6) User: I would like a 12inch sub on wheat bread.

(7) Avatar: Okay, a 12 inch sub on wheat bread. Which sub do you want? (Ex. BLT)

(8) User: Well that depends on which subs have ham.

(9) Avatar: These sandwiches have ham: Black Forest Ham, Turkey Breast and Black Forest Ham

(10) User: I will have the Black Forest Ham with peppers and olives.

(11) Avatar: Okay, the Black Forest Ham sandwich with olives and peppers. Do you want any other toppings?

dialog continues

Motivation: Enabling Complex Dialog

- The Subway dialog above illustrates a human-computer mixed-initiative interaction that, due to the complexity and variability of the dialog, is not possible to realize with other dialog systems today. Trying to mix even 3 questions results in 8191 possible unique dialog interactions.
- Line (1) starts with a simple prompt for sub or salad and (2) shows the user responding directly to the prompt. This is the extent of flexibility (completely fixed) of most dialog systems today.
- Line (3) shows the system soliciting for the next item in a script, sandwich size, but in (4), the user responds to a different but forthcoming solicitation for takeout. This out-of-turn interaction is a form of mixed-initiative interaction (MII) where the user and the system engage as equal participants in dialog.
- In (5), the system again solicits for the unanswered sandwich size and in (6), the user responds with a size and specifies the type of bread desired, completing another forthcoming solicitation for bread type. This illustrates a form of MII where the user provides information for more than one solicitation in a single utterance.
- Line (7) Shows the system accepting the user information from (6), and asking which specialty sandwich the user wants. In line (8), rather than providing information to the system, the user seeks information from the system, i.e., which specialty sandwiches have ham.
- In (9), the system has successfully understood the user request for information and has provided the specialty sandwiches with ham. The user chooses one of the items with ham and also specifies the toppings peppers and olives.
- Line (11) shows the system accepting the user information and asking if the user wants any additional toppings.
- Our dialog toolkit supports all of these forms of mixed-initiative interaction.

Figure 2: Sample motivating, mixed-initiative dialog, built with our toolkit, running on a mobile phone.

dialog system, and is primarily concerned with controlling the flow of the dialog, while maintaining discourse history, sometimes referred to as system-action prediction, and coordinating with other (typically input/output) components of the system (e.g., automatic speech recognition, spoken language understanding, and presentation of results).

There are two main approaches to dialog management: task-based and data-driven. Our work combines the two: (i) our research targets task-based dialog systems whose goal is to support the user in satisfying clearly-defined goals by completing highly-structured tasks; and (ii) we use data-driven techniques (e.g., *bag-of-words* model and a *k-nearest-neighbor* classifier) to help support the users use of natural language to pursue these tasks (see Fig. 3—right). We focus on frameworks for (automatic) construction for their relatedness to our work [9, 17, 28].

The task-based approach involves modeling a collection of tasks to be supported by the system, using a modeling notation or language, and discerning how the user can be most effectively afforded (the desired) interaction flexibility in completing those tasks. Finite state automata (FSA), and other transition networks, context-free grammars (CFG), and events have been used as general task structures to model dialog [12].

Our approach factors the domain-dependent components of the system (e.g., the aspects of the dialog specific to the targeted domain) from the domain-independent components of the system (e.g., the dialog engine and mangement). Thus, our dialog engine acts as an interpreter, in the programming languages sense, for the given dialog specification. This approach provides a clean separation of the domain-dependent and -independent aspects (e.g., control logic and dialog flow) [1]. This approach is used in the *RavenClaw* dialog management framework [3, 4]. *RavenClaw* uses an agenda-based approach to task modeling [30, 31]. Our framework is an instantiation of this 'separation of task model and dialog engine' approach to dialog management (see Fig. 3—left).

3. APPROACH AND UNIQUENESS

Rather than agenda [31], rule-oriented [10], and the myriad of other task structures and task modeling approaches used for task-based dialog management, we use programming language theory. We designed a notation based on lambda calculus that serves as an authoring notation for specifying dialogs and also suggests implementation ideas. This distinguishes our model from other knowledge/task-based approaches which use hierarchical task/agenda models. Using program transformations [24], including partial evaluation [16], and language concepts, to specify dialogs and to intentionally model multiple paths through a dialog without extensionally hardcoding each into the control flow of the implementation, is a fundamentally different approach to dialog modeling, management, and implementation.

Our approach is unique in that it involves thinking of dialog as a function and using concepts from programming language theory, including function currying and partially evaluation, to automatically modify that function to achieve a mixed-initiative mode of interaction. 'As the user progresses through a dialog, we think of the steps that she takes as the evaluation of a function. Changing the evaluation method of the function (or transforming the function) then corresponds to different interaction policies [30] for the dialog (i.e., ways of mixing initiative). The overall idea is that different function evaluation strategies correspond to different interaction policies for the dialog (i.e., system initiated vs. mixed-initiative) or ways of mixing initiative' [5]. The structure of an expression in our dialog-authoring notation, and the language concepts used therein, provides a pattern for implementing the dialog. Based on this foundation, we built a dialog modeling and implementation toolkit, which is capable of automatically realizing a variety of mixed-initiative dialogs given only a single, high-level specification of each.

While prior research projects have approached engineering interactive computing systems from the perspective of (functional) programming languages [11, 20, 29], only few have sought to marry human-computer dialogs with concepts from programming languages [7, 25]. Due to the conceptual analogs between natural languages and programming languages, viewing human-computer dialog modeling, management, and implementation from the perspective of programming language theory suggests a natural, yet under-

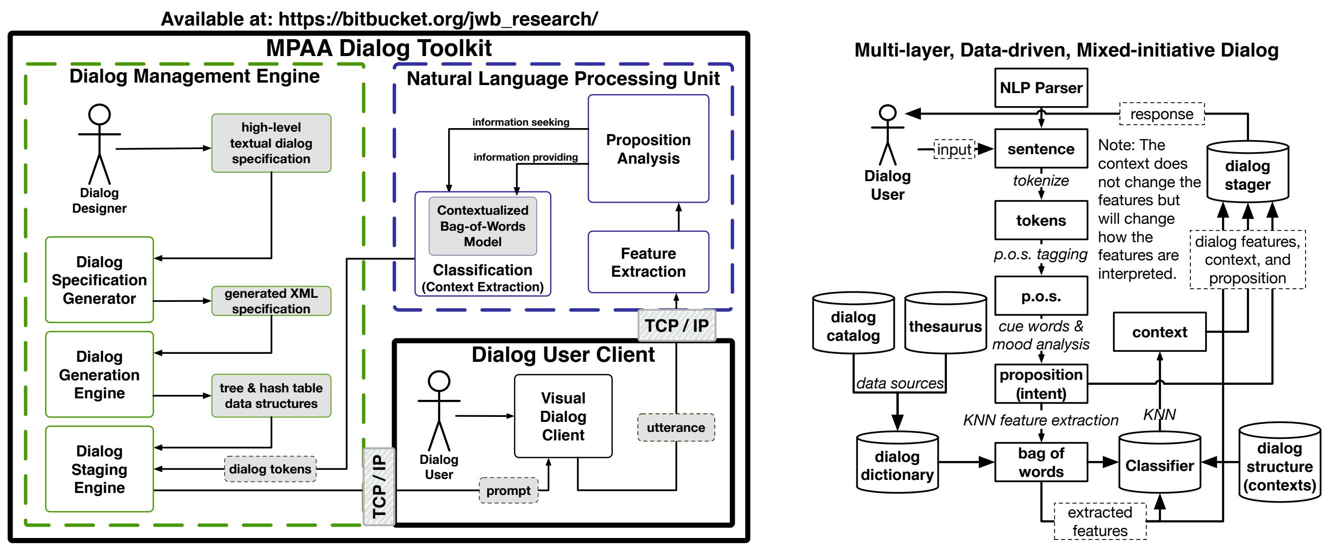

Available at: https://bitbucket.org/jwb_research/
MPAA Dialog Toolkit
Dialog Management Engine
Dialog Designer
high-level textual dialog specification
Dialog Specification Generator
generated XML specification
Dialog Generation Engine
tree & hash table data structures
Dialog Staging Engine
dialog tokens
TCP / IP
Natural Language Processing Unit
information seeking
information providing
Proposition Analysis
Contextualized Bag-of-Words Model
Classification (Context Extraction)
Feature Extraction
TCP / IP
Dialog User Client
Dialog User
Visual Dialog Client
utterance
prompt
Multi-layer, Data-driven, Mixed-initiative Dialog
NLP Parser
response
Dialog User
input
sentence
tokenize
tokens
p.o.s. tagging
p.o.s.
cue words & mood analysis
proposition (intent)
Note: The context does not change the features but will change how the features are interpreted.
dialog stager
dialog features, context, and proposition
context
dialog catalog
thesaurus
data sources
KNN feature extraction
KNN
dialog dictionary
bag of words
Classifier
dialog structure (contexts)
extracted features

Figure 3: (left) Architectural design of our natural language, mixed-initiative dialog construction toolkit and engine. (right) Detailed data-driven architectural design of our natural language processing unit on left.

explored, approach to dialog representation and reasoning. The concepts from programming languages are not just helpful metaphors for dialog specification, but also lend insight into operationalizing dialogs.

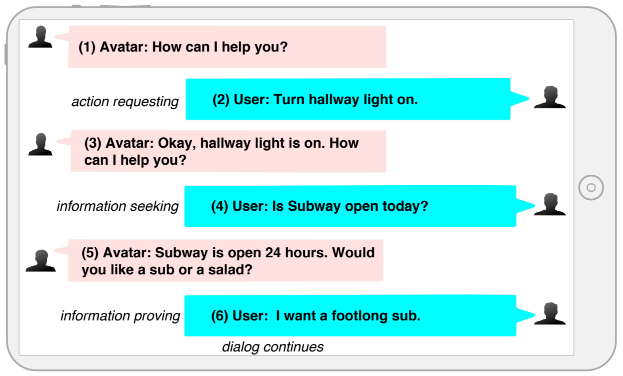
Dialog-based systems such as *Siri* support utterances communicated through natural language, but are limited to action-requesting, information-seeking (e.g., 'What is the weather forecast tomorrow?'), and information-providing utterances and, thus, only support a low degree of mixed-initiative interaction (see Fig. 4). We have enhanced our model for mixed-initiative dialog by using a *bag-of-words* model for a new dialog domain and a *k-nearest-neighbor* classifier to predict the context of a user utterance (i.e., map an unsolicited utterance to the dialog prompt to which it is a response) to improve the natural language and mixed-initiative capabilities of systems like *Siri* (see last row of Table 1). Fig. 3 (right) illustrates the design of the natural language processing unit in Fig. 3 (left). Fig. 5 demonstrates the natural language capabilities of our system. Our MPAA[1] dialog toolkit is available at https://bitbucket.org/jwb_research.

4. RESULTS AND CONTRIBUTIONS

4.1 Results

While '[c]reating an actual dialog system involves a very intensive programming effort' [13], our dialog authoring tool (see Fig. 3) is a contribution that simplifies that effort so that dialog designers can evaluate a variety of mixed-initiative, human-computer dialogs [5, 27]. Specifically, given q, the number of questions posed in a dialog, our system is capable of automatically implementing $2^{\sum_{p=1}^{q} p! \times S(q,\ p)} - 1$ dialog specifications (= 8,191 for $q = 3$)—i.e., the number of all subsets (minus the empty set) of all possible paths through a dialog involving q questions (or prompts). The expression $\sum_{p=1}^{q} p! \times S(q,\ p)$ describes the total number of paths possible through a dialog with q questions, where the *Stirling number* of a set of size m is $S(m, n) = |s(m, n)|$, and $s(m)$ is

[1]Mixed-initiative Personal Assistant Agents.


(1) Avatar: How can I help you?

action requesting (2) User: Turn hallway light on.

(3) Avatar: Okay, hallway light is on. How can I help you?

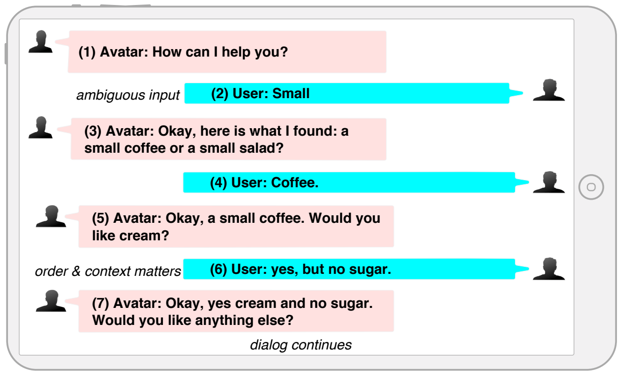
information seeking (4) User: Is Subway open today?

(5) Avatar: Subway is open 24 hours. Would you like a sub or a salad?

information proving (6) User: I want a footlong sub.

dialog continues

Figure 4: Illustration of the the low degree of natural language, mixed-initiative interaction (e.g., action-requesting, information-seeking, information-providing utterances) supported by systems such as *Siri*.


(1) Avatar: How can I help you?

ambiguous input (2) User: Small

(3) Avatar: Okay, here is what I found: a small coffee or a small salad?

(4) User: Coffee.

(5) Avatar: Okay, a small coffee. Would you like cream?

order & context matters (6) User: yes, but no sugar.

(7) Avatar: Okay, yes cream and no sugar. Would you like anything else?

dialog continues

Figure 5: Demonstration of the enhancements over personal assistants like *Siri* that our approach fosters.

the set of all partitions of a set of size m into non-empty subsets, where m is a positive integer. This corresponds to all possible permutations (i.e., orders) of all possible partitions

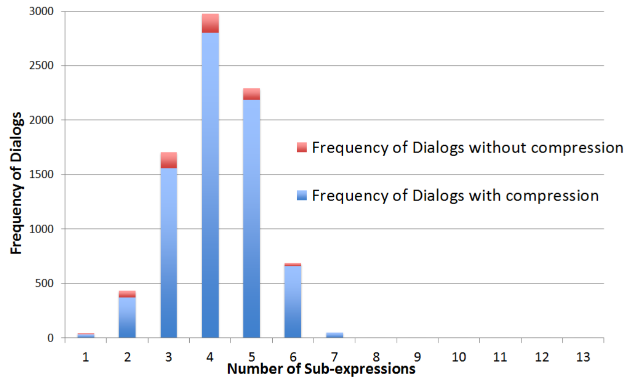


Figure 6: Histogram illustrating the frequency of of the 8,191 dialog specifications in $\mathcal{U}_3$ (y-axis) that can be represented in our notation with 1–13 sub-expressions (x-axis).

(i.e., combinations) of the prompts of the dialog. While the task structures for modeling dialog mentioned above, such as FSAs, CFGs, and events [12], are sound, and can be used to prove mathematical proprieties, tasks often need to be over-specified to model a rich and flexible form of human-computer interaction. Moreover, since dialogs can contain arbitrarily nested sub-dialogs, FSA are less effective as general discourse structures [10]. Similarly, CFGs might be appropriate if the evolution of a dialog was something known a priori [10].

Evaluating models for mixed-initiative dialog is itself an unsolved problem for a variety of reasons including the extremely limited nature of existing data and the ambiguity of the very definition of initiative [13]. One way to capture the efficacy of a model is to evaluate how well the model fits data. In the context of our model, this means evaluating the frequency of dialogs that can be captured by our notation and how well it captures each. Given any value for q, the number of questions per episode, every dialog in the space $\mathcal{U}_q$ can be specified using our dialog authoring notation. Since the specification expression of a dialog serves as a design pattern for implementing it, the number of sub-expressions in the specification is an evaluation metric for how well the notation captures the specification. A complete, mixed-initiative dialog can be captured by one expression: e.g., $\frac{PE^\star}{\text{size blend cream}}$. If we remove only one—≺(size blend cream)≻—of the thirteen episodes from this dialog, specifying it requires five sub-expressions: $\frac{SPE'}{\text{size blend cream}} \cup \frac{SPE}{\text{size blend cream}} \cup \frac{C}{\frac{I}{\text{size blend}}\text{cream}} \cup \frac{C}{\frac{I}{\text{blend cream}}\text{size}} \cup \frac{C}{\frac{I}{\text{size cream}}\text{blend}}$. We specified each of the 8,191 dialogs in $\mathcal{U}_3$ using our notation and computed the frequency that could be captured by 1, 2, ..., and 13 sub-expressions. Our results are shown in Fig. 6 (e.g., there are 46 dialogs that can be specified with one expression, and 2,977 that can be specified with four sub-expressions). For the details of our dialog-authoring notation (i.e., the fractional expressions above) based on functional programming language concepts, we refer the reader to [27].

Since there are no dialogs in $\mathcal{U}_3$ that require greater than seven sub-expressions to model, and there are dialogs in the space with greater than seven episodes (e.g., the maximum number of episodes in any one dialog is thirteen for q=3), the use of our notation provides a compressed dialog specification. However, what is not illustrated in Fig. 6 is the number of episodes in each dialog that can be represented with a particular number of sub-expressions or, in other words, the magnitude of the results given in Fig. 6. For instance, if all 46 dialogs that can be represented with only one sub-expression only contain one episode, then there is no compression. To measure the efficacy of the compression, we computed the frequency of dialogs which can be specified at the observed compression percentages. For instance, 533 dialogs of the 8,191 could not be compressed at all (i.e., there is a one-to-one relation between the number of episodes and the number of sub-expressions). However 1,197 dialogs can be compressed 33% (e.g., a dialog that involves nine episodes which can be specified with six sub-expressions), and 975 can be compressed 50%. Fig. 7 presents these compression results: over 20% of the dialogs (1,692/8,192) can be compressed 50% or more. While we cannot characterize the dialog specifications comprehensively beyond q=3 because it is not possible to enumerate and simulate [18, 22, 28] all of them, we can say intuitively that the results for $q > 3$ are better than q=3 because the opportunities for compression increase as the number of questions posed in an episode increases. Therefore, both the number of sub-expressions required to specify a dialog as well as the percentage of dialogs being compressed to a high degree increase.

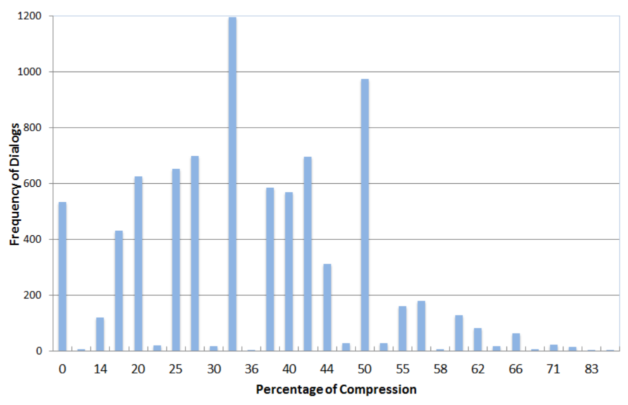


Figure 7: Histogram illustrating the frequency of the of the 8,191 dialog specifications in $\mathcal{U}_3$ (y-axis) that can be compressed to the observed percentages (x-axis) through use of our notation.

4.2 Contributions and Future Work

Dialog is essential in providing a rich form of human-computer interaction [8]. We summarize the contributions of our research as: we i) developed a language-based model for specifying and staging mixed-initiative, human-computer dialogs, ii) generalized and automated the activity of building a dialog system, and iii) evaluated its descriptive and staging capabilities by demonstrating that it can succinctly capture and stage a wide variety of dialogs, including those involving sub-dialogs. While "[c]reating an actual dialog system involves a very intensive programming effort" [13] and "complete automation in creating . . . dialog applications remains an extremely difficult problem" [9], given a specification of a dialog in our dialog authoring notation, from among a variety of mixed-initiative dialogs, our system automates the implementation of the dialog. Designers of task-based dialog systems can use our dialog authoring notation and staging engine as a dialog modeling and implementation toolkit to explore, prototype, and evaluate [17] a variety of unsolicited reporting, mixed-initiative dialogs.

While the use of simulation for evaluation of dialog systems is common [18, 22, 28], the application of our results will benefit from a formal usability evaluation. We intend to conduct studies with users to evaluate the interface through which users experience the human-computer dialog (i.e., Figs. 2 and 5) as well as the interface for task modeling used by dialog designers to specify the dialog as part of future work. Evaluating the interface through which dialog participants experience the dialog will help us discern whether mixed-initiative dialogs resulting from our language-based model have desirable qualities (i.e., How effective and efficient are they? Does mixed-initiative dialog help the user in an information-seeking activity and how, e.g., time-to-task completion, satisfaction? For which types of dialogs or tasks is mixed-initiative interaction most effective?). We desire "computational agents carrying out our dialog theory to produce conversations with desirable qualities" [13]. To this end, we plan to conduct a study similar to [9] and, in a more broad context, using the results of [33].

For future work, we envisage the incorporation of mixed-initiative personal assistants designed/implemented with our toolkit into airport kiosks, ATMs, and interactive, voice-responses systems, since the ubiquity of these platforms in a variety of service-oriented domains, such as education, health care, banking, and travel provide a fertile landscape for the use of our model for mixed-initiative interaction. We are also exploring the use of our model in a university course schedule application in an immersive, virtual environment. It will advance software development processes for virtual/cyberlearning environments, gaming, film, simulation, and telepresence, where MI dialog flexibility is also critical.

5. REFERENCES

[1] T. Ball, C. Colby, P. Danielsen, L. Jagadeesan, R. Jagadeesan, K. Läufer, P. Mataga, and K. Rehor. Sisl: Several interfaces, single logic. *International Journal of Speech Technology*, 3(2):93–108, 2000.

[2] M. Berg. *Modelling of Natural Dialogues in the Context of Speech-based Information and Control Systems*. PhD thesis, University of Kiel, 2014.

[3] D. Bohus and A. Rudnicky. RavenClaw: Dialog management using hierarchical task decomposition and an expectation agenda. In *INTERSPEECH*, 2003.

[4] D. Bohus and A. Rudnicky. The RavenClaw dialog management framework: Architecture and systems. *Computer Speech and Language*, 23(3):332–361, 2009.

[5] J. Buck and S. Perugini. A tool for staging mixed-initiative dialogs. In *Proceedings of (MAICS)*, pages 25–32, 2016.

[6] J. Buck and S. Perugini. Mixed-initiative personal assistants. In *Proceedings of ACM Technical Symposium on Computer Science Education*, pages 753–754, 2017.

[7] R. Capra, M. Narayan, S. Perugini, N. Ramakrishnan, and M. Pérez-Quiñones. The staging transformation approach to mixing initiative. In *IJCAI Workshop on Mixed-Initiative Intelligent Systems*, pages 23–29, 2003.

[8] A. Dix, J. Finlay, G. Abowd, and R. Beale. *Human-Computer Interaction*, chapter 16: Dialog Notations and Design. Prentice Hall, third edition, 2010.

[9] J. Feng, D. Hakkani-Tür, G. D. Fabbrizio, M. Gilbert, and M. Beutnagel. Webtalk: Towards automatically building spoken dialog systems through mining websites. In *Proceedings of ICASSP*, pages 573–576, 2006.

[10] R. Freedman. Using a reactive planner as the basis for a dialogue agent. In *Proceedings of FLAIRS*, pages 203–208, 2000.

[11] P. Graunke, R. Findler, S. Krishnamurthi, and M. Felleisen. Automatically restructuring programs for the web. In *Proceedings of ASE*, pages 211–222, 2001.

[12] M. Green. A survey of three dialogue models. *ACM Transactions on Graphics*, 5(3):244–275, 1986.

[13] C. Guinn. Evaluating mixed-initiative dialog. *IEEE Intelligent Systems*, 14(5):21–23, 1999.

[14] M. Hearst. Mixed-initiative interaction. *IEEE Intelligent Systems*, 14(5):14, 1999.

[15] J. Hochberg, N. Kambhatla, and S. Roukos. A flexible framework for developing mixed-initiative dialog systems. In *SIGDIAL Workshop on Discourse and Dialogue*, 2002.

[16] N. Jones. An introduction to partial evaluation. *ACM Computing Surveys*, 28(3):480–503, 1996.

[17] P. Jordan, M. Ringenberg, and B. Hall. Rapidly developing dialogue systems that support learning studies. In *Proceedings of ITS Workshop*, pages 1–8, 2006.

[18] C. Lee, S. Jung, K. Kim, D. Lee, and G. Lee. Recent approaches to dialog management for spoken dialog systems. *Journal of Computing Science and Engineering*, 4(1):1–22, 2010.

[19] P. Liang. Learning executable semantic parsers for natural language understanding. *Communications of the ACM*, 59(9):68–76, 2016.

[20] S. Malkov. Customizing a Functional Programming Language for Web Development. *Computer Languages, Systems and Structures*, 36(4):345–351, 2010.

[21] C. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. Bethard, and D. McClosky. The stanford corenlp natural language processing toolkit. In *ACL*, pages 55–60, 2014.

[22] T. Misu, K. Georgila, A. Leuski, and D. Traum. Reinforcement learning of question-answering dialogue policies for virtual museum guides. In *Proceedings of SIG Discourse and Dialogue*, pages 84–93, 2012.

[23] F. Morbini, E. Forbell, D. DeVault, K. Sagae, D. Traum, and A. Rizzo. A mixed-initiative conversational dialogue system for healthcare. In *Proceedings of ACL SIGDIAL Workshop on Discourse and Dialogue*, pages 137–139, 2012.

[24] H. Partsch and R. Steinbrüggen. Program transformation systems. *ACM Computing Surveys*, 15(3):199–236, 1983.

[25] M. Pérez-Quiñones. *Conversational Collaboration in User-initiated Interruption and Cancellation Requests*. Ph.D. thesis, The George Washington University, 1996.

[26] S. Perugini. Mining mixed-initiative dialogs. In *Proceedings of SMC*, pages 2287–2294, 2016.

[27] S. Perugini and J. Buck. A language-based model for specifying and staging mixed-initiative dialogs. In *Proceedings of the Eighth International ACM SIGCHI Symposium on Engineering Interactive Computing Systems (EICS)*, pages 204–216, 2016.

[28] J. Polifroni, G. Chung, and S. Seneff. Towards the automatic generation of mixed-initiative dialogue systems from web content. In *EUROSPEECH*, pages 193–196, 2003.

[29] C. Queinnec. The influence of browsers on evaluators or, continuations to program web servers. In *Proceedings of ICFP*, pages 23–33, 2000.

[30] A. Rudnicky, E. Thayer, P. Constantinides, C. Tchou, R. Stern, K. Lenzo, W. Xu, and A. Oh. Creating natural dialogs in the Carnegie Mellon Communicator system. In *Proceedings of EUROSPEECH*, 1999.

[31] A. Rudnicky and W. Xu. An agenda-based dialog management architecture for spoken language systems. *Proceedings of ASRU Workshop*, 13(4), 1999.

[32] D. Stallard. Dialogue management in the talk'n'travel system. In *Proceedings of ASRU Workshop*, 2001.

[33] M. Walker, L. Hirschman, and J. Aberdeen. Evaluation for DARPA communicator spoken dialogue systems. In *LREC*, 2000.