2017

CSE: U: Mixed-initiative Personal Assistant Agents

Joshua W. Buck  
*University of Dayton, jbuck1@udayton.edu*

Saverio Perugini  
*University of Dayton, sperugini1@udayton.edu*

Tam Nguyen  
*University of Dayton, tnguyen1@udayton.edu*

Follow this and additional works at: [https://ecommons.udayton.edu/cps_fac_pub](https://ecommons.udayton.edu/cps_fac_pub)  
Part of the Graphics and Human Computer Interfaces Commons, and the Other Computer Sciences Commons

eCommons Citation


This Conference Paper is brought to you for free and open access by the Department of Computer Science at eCommons. It has been accepted for inclusion in Computer Science Faculty Publications by an authorized administrator of eCommons. For more information, please contact frice1@udayton.edu, mschlangen1@udayton.edu.
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 work lies specifically 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. 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. There are multiple research projects which seek to automate the implementation of flexible, dialog-based systems. What sets our research apart from these projects is our use of language-based concepts and operators, rather than task structures, to model dialog, which we discuss below.

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. Since authoring a dialogue is like writing a movie script with many different endings, a central problem for mixed-initiative dialogue management is coping with utterances that fall outside of the expected sequence of the dialogue. Thus, developing a mixed-initiative dialog system is a complex task and creating an actual dialog system involves a very intensive programming effort.

This problem is important since dialog has been established as an effective mechanism through which to achieve a rich form of human-computer interaction. 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 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).

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. 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. There are multiple research projects which seek to automate the implementation of flexible, dialog-based systems. 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
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 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 understanding, and presentation of results).

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 approach from other knowledge/task-based approaches which use hierarchical task/agency 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], only few have sought to marry human-computer dialogs with concepts from programming languages [4] [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-
4. RESULTS AND CONTRIBUTIONS

4.1 Results

While creating 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 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 $\sum_{k=1}^{q} k! \times S(q, k)$ 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 $\sum_{k=1}^{m} k! \times S(m, k)$.

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. 5(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 dialog toolkit is available at https://bitbucket.org/jwb_research/.

Note: The context does not change the features but will change how the features are interpreted.

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.

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.
(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 $U_q$ 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., $\text{size blend cream}$ $\cup$ $\text{size blend cream}$ $\cup$ $\text{color}$ $\cup$ $\text{size}$ $\cup$ $\text{color}$ $\cup$ $\text{size blend cream}$ $\cup$ $\text{size blend cream}$. If we remove only one—e.g., the thirteen episodes from this dialog, specifying it requires five sub-expressions: $\text{size blend cream}$ $\cup$ $\text{size blend cream}$ $\cup$ $\text{color}$ $\cup$ $\text{size}$ $\cup$ $\text{color}$ $\cup$ $\text{size blend cream}$. We specified each of the 8,191 dialogs in $U_q$ 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 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 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 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.

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 “creating 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 a variety of unsolicited reporting, mixed-initiative dialogs.

While the use of simulation for evaluation of dialog systems is common, 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