

User-created Forms as an Effective Method of Human-agent Communication

John Zimmerman^{1,3}, Kathryn Rivard², Ian Hargraves³, Anthony Tomasic², Ken Mohnkern¹

¹HCI Institute | ²Institute for Software Research | ³School of Design

Carnegie Mellon University

{john.zimmerman, krivard, ian.hargraves, tomasic, mohnkern}@cmu.edu

ABSTRACT

A key challenge for mixed-initiative systems is to create a shared understanding of the task between human and agent. To address this challenge, we created a mixed-initiative interface called Mixer to aid administrators with automating tedious information-retrieval tasks. Users initiate communication with the agent by constructing a form, creating a structure to hold the information they require and to show context in order to interpret this information. They then populate the form with the desired results, demonstrating to the agent the steps required to retrieve the information. This method of form creation explicitly defines the shared understanding between human and agent. An evaluation of the interface shows that administrators can effectively create forms to communicate with the agent, that they are likely to accept this technology in their work environment, and that the agent's help can significantly reduce the time they spend on repeated information-retrieval tasks.

Author Keywords

agents, interaction design, mixed initiative.

ACM Classification Keywords

H.5.2 User Interfaces: interaction design

INTRODUCTION

Advances in machine learning (ML) have led to remarkable applications that significantly improve people's lives. Examples include spam filters for email, recommenders that help find good content, speech interfaces trained on many voices, and even interruptability detectors that make devices more socially appropriate [11]. Most applications of ML focus on increasing the capabilities of the computer; however, systems that employ a mixed-initiative approach seek harmonious intersections of machine and human

intelligence, where collaboration through interaction allows problems to be addressed that neither human nor machine could complete independently with the same effectiveness. In general, these systems strive to make people (end-users) more capable by providing them greater access to computational power.

Taking a research-through-design approach [24], we began our project by conducting observational research with office workers, looking for opportunities for ML to improve work performance. We observed that most administrators regularly engaged in information-retrieval tasks, pulling and organizing information from several sources to effectively respond to an email. For example, we observed an administrator who must contact a student's instructors when a student goes to the hospital. This task involves gathering enough information to identify the individual student, retrieving his/her schedule of classes, retrieving the names of the instructors for each class, and looking up each instructor's email address individually. In completing this task the administrator must interact with, and link information from, several different IT systems. Administrators did not think the information retrieval was difficult, but they did consider it very tedious. The repetitive and procedural structure of the tasks makes them ripe for automation via ML.

To address this opportunity we created Mixer, an interface that allows end-users to train agents to retrieve information. Once trained, the agents learn to recognize the opportunity to retrieve information in incoming email requests, anticipatorily retrieve the relevant information, and augment the message with the information needed to complete the request (Figure 1). Mixer addresses a perennial problem of mixed-initiative interaction—creating a shared understanding of the task between users and agents—through a novel, form-building interface. Instead of describing the procedure, users create a form that describes the outcome they want and the context for understanding and evaluating the outcome. By populating the form with data, users implicitly demonstrate to the agent the steps required to produce the outcome. By allowing end-users to select the tasks they wish to have automated and define the scope of that automation, Mixer allows end-users to take ownership of their own problems [10].

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CHI 2009, April 4–9, 2009, Boston, Massachusetts, USA.

Copyright 2009 ACM 978-1-60558-246-7/09/04...\$5.00

To test our form-building concept, we developed a working prototype of the Mixer interface (without the agent) and conducted an evaluation with real administrators. Results show that: (i) administrators can conceive of and create forms that effectively communicate the information they want to have retrieved, (ii) administrators would likely use Mixer to automate the tedious information-retrieval tasks they regularly perform, and (iii) Mixer significantly reduces the amount of time needed to complete repeated information-retrieval tasks.

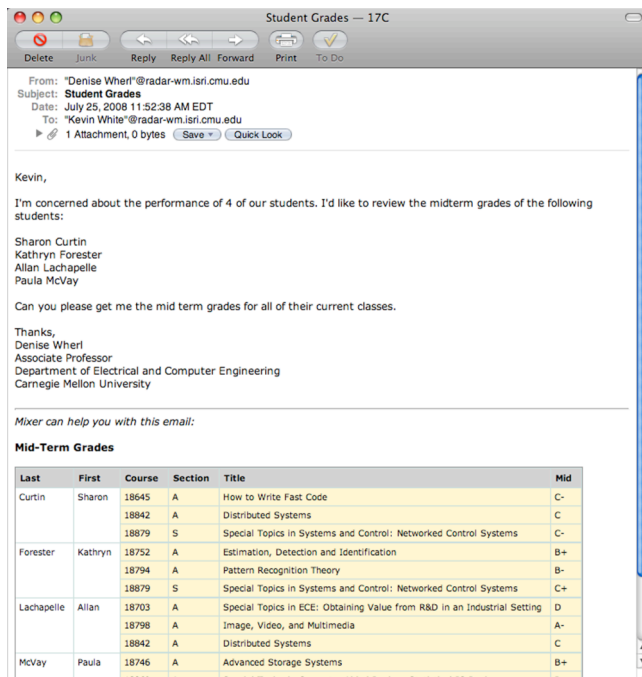


Figure 1 New email message with the information the Mixer agent has anticipatorily retrieved at the bottom.

RELATED WORK

Our research provides a novel integration of inventions and discoveries from across a variety of domains including collaborative interaction with mixed-initiative assistants and programming by demonstration.

Mixed-initiative interaction

Mixed-initiative research focuses on advancing methods for collaboration between computer agents and people where each party has its own knowledge, ways of reasoning, and abilities to understand and act in order to advance toward a common goal [13, 15]. Many issues remain to be answered, including several interrelated needs with respect to interaction between agent assistants and users [21]:

- *Awareness*: knowledge of problem and goal must be shared between human and agent.
- *Task*: roles and responsibilities must be shared between human and agent.
- *Communication*: both human and agent must be able to express knowledge and needs.

Mixed-initiative researchers have challenged the community to invent new strategies for establishing common representations of joint action [15, 1, 5]. Instead of literal dialog, Mixer presents a form as the shared knowledge and goal state. Malone saw the potential of semi-structured forms as a means of expressing human practice and intention in a manner that was amenable to agent assistance [18]. Malone's work focuses on structuring email conversations so that agents can assist in the coordination of human activities. Our previous mixed-initiative system VIO advanced Malone by focusing on how developer-constructed forms create a context where users can understand the actions an agent has taken, and quickly identify and repair errors the agent makes [25]. Mixer, in contrast, utilizes user-built forms that support the integration of information from free-text email requests and from structured IT systems. The forms focus on the outcome, an explicit goal structure—a goal that originates in human-human communication and ends in information retrieved from various information systems. The form approach in Mixer allows it to leverage the repair advances made with VIO.

The task issue concerns the division of action between humans and agents. Some systems make a strong division between human work and agent work; e.g., LookOut [14]. Systems such as MAPGEN [4] allow the degree of responsibility between human and agent to vary. Mixer strongly divides the actions of the human and agent, giving them clear roles based on their abilities.

The communication issue arises from the tension between the system's need for rich input to improve learning and the user's need to focus attention on the task at hand. Mixer's interaction revolves around the activities of form description, information demonstration, and repair of agent errors and informational ambiguity. In the design and discussion sections, form construction is described as *natural* because it mirrors workers' current activity of creating their own shortcuts to avoid information retrieval from IT systems, offsetting well-known weaknesses in information systems [7, 20].

Programming by demonstration

Programming-by-demonstration systems range from simple recording of macros in spreadsheets to systems that generalize the demonstrated actions of users across several applications [16]. In general, their focus has been to help end-users automate repeated, direct-manipulation tasks.

One of the key challenges for these systems has been scope. Toolkit developers attempt to make it easy for users to get a system to do something ("lower the threshold") and also increase the number of things they can do with the system ("raise the roof") [19]. However, in Nardi's [20] analysis of end-user programming, she observed much greater success for task-specific systems. Mixer is not a general end-user programming system since it is (i) very task specific and (ii) the task is not presented as programming or as the

articulation of the process. However, the challenge with our system, as with programming-by-demonstration systems, is to find a way for users to transition from specifying a plan (communication of what they want the system to do) to the agent taking the correct actions from the plan [9]. Mixer explicitly engages the user in resolving ambiguities that arise in this transition.

Programming-by-demonstration systems also face several practical challenges such as the user's communication of the termination condition for the agent, and the agent's ability to detect and remove training noise caused by users performing unrelated or unnecessary tasks during the demonstration [20]. Mixer avoids the termination problem by automatically providing a single "loop" over the input data that terminates when the input data is exhausted. To eliminate noise caused by unnecessary user actions, Mixer uses the user's demonstration of the first input row to construct a program and uses the remaining input rows to *test* the constructed program. PLOW [1] also uses a single example model, but relies on the user to describe additional required information, such as conditionals. MIXER uses additional input rows to determine conditionals.

Previous research has examined the use of forms as a target for communication between the user and the system. Query-by-Example and Office-by-Example [26] permit a user to assemble a form on a canvas. However, both systems require users to specify variables in forms, effectively requiring a user to *program* the query that generated the completed form.

CoScripter uses natural language processing ("sloppy programming"), programming by demonstration, and a script-debugging interface to enable users to define scripts. These scripts are then shared through a wiki. CoScripter and Mixer share similar task objectives of supporting workers in their interactions with IT systems. CoScripter [17] focuses on the recording, sharing, and replay of procedural knowledge. The common representation of CoScripter is a script. Mixer focuses on the recording, sharing, and replay of queries in an appropriate context. The common representation of Mixer is a table. As such, CoScripter provides more context for users to understand the generated program. Mixer hides this process somewhat from the user, keeping the focus on the desired outcome and not on the procedure for achieving this outcome.

Tuchinda et al [23] describe another system that uses a table as a common task representation between human and agent. In this system, users discover queries and query results by iteratively providing sample keywords to columns in a table. Mixer focuses less on discovery and more on providing a natural way for a user to express what they want, thereby demonstrating a known retrieval procedure. Tuchinda et al.'s system demonstrates an agent can effectively process information entered into a table, and Mixer advances this by demonstrating that users can create

effective forms and that administrators recognize the value of this for automating their own information retrieval tasks.

Finally, while Mixer is currently only an interface, we have carefully designed Mixer to interoperate with an existing programming-by-example system [12]. MIXER will use a simple induction-based learning system. For the moment, a more complex multiple-hypothesis system (e.g, [6]) is not required for the tasks that we observed in our contextual inquiry. We are currently developing the connection between the two systems.

INTERFACE DESIGN

The Mixer interaction design emerged from a user-centered design process involving user research, exploration of many design concepts, and rapid prototyping on the form-based interaction. Initially we performed contextual inquiries (CI) observing office workers [22]. These inquiries took a match-making approach [3] where we specifically looked for opportunities to apply ML technology to aid office workers. During the CIs we observed many administrators spending a great deal of time retrieving data from several sources in order to respond to emails. Administrators had great familiarity with their jobs and their information systems, so they did not find the retrieval tasks to be difficult; however, they did view the work as very tedious. The more challenging work came once they had retrieved the information, when they needed to make a judgment and initiate actions based on the retrieved information.

We observed that many administrators create their own tools. For example, at the beginning of each semester, one administrator created a spreadsheet containing all students enrolled in each class for which she was responsible. When a request arrived that required use of this information, instead of checking the school's IT system, she used her local copy. All of the shortcuts were either kept digitally on the user's local computer or were posted as hard-copy near the work area. One shortcoming of the shortcuts was that they lost sync with the latest information in the IT system.

Based on our findings and on our study of the related literature in mixed-initiative interaction and end-user programming, we engaged in a process of design exploration, focusing on how administrators could communicate what information they wanted an agent to retrieve. Using a scenario-driven approach drawn from the examples we had observed, we evolved a design where an administrator creates a form, then trains the agent through demonstration. To reduce the risk of making an interface users would not understand, we conducted paper prototyping/participatory design sessions where participants performed tasks similar to the tasks we had observed administrators complete. We gave them paper form-building elements and asked them to assemble forms and demonstrate how to retrieve the information needed to complete the form. Figure 2 shows a form created by one of the participants.

Following the paper session, we conducted a similar session where participants completed the same tasks using Microsoft Excel to design the form and demonstrate the information retrieval. From both sessions we learned the following:

1. People could conceive of the information-retrieval task in terms of a small set of related forms (tables, lists).
2. For some tasks it was difficult to create forms that return exactly the information needed to fulfill the request, but forms that return a superset of that information were considered quite valuable.
3. People tended to create forms with a single table, not with multiple tables.

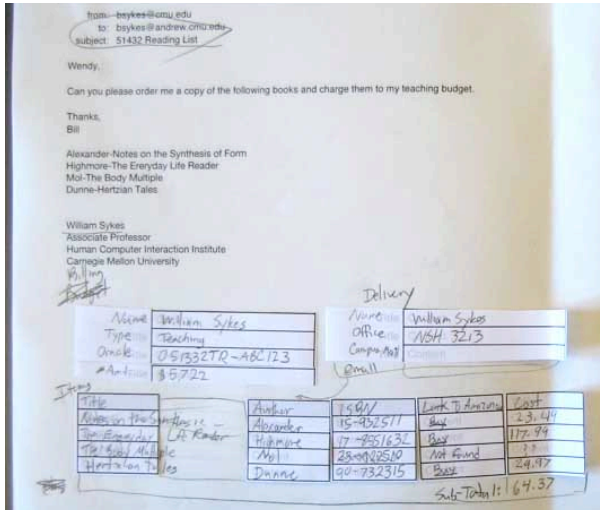


Figure 2 Example of a user-created paper form to automate looking up textbook prices at the campus bookstore and at an online bookstore for all of the books needed for a specific class.

In our review of the prototyping sessions, we recognized that the agent could provide help with filling the form once users had demonstrated a single row. This insight along with the results of the prototyping sessions lead to the current Mixer interaction design.

Interaction Design

Interaction begins when users receive an incoming email that requires an information-retrieval task they want to automate. Clicking a link embedded in the email launches Mixer’s form-building interface (Figure 3a), which displays the email on the left and tools for form building and demonstrating on the right.

The form-building interface opens with a new form of one row and column. To add columns users drag the right edge of the table to the right. Dragging to the left removes columns. Only one row is available to receive demonstrated values. Users can title the form, add labels to column headers, and demonstrate information retrieval by visiting various data sources and copying and pasting the specific data they want into specific cells of the form. The agent observes this interaction.

At any time in the design and demonstration of the one-row table, users may click the “Fill In Table” button (Figure 3a). The agent then attempts to add rows to complete the form following the procedure used by the user in the first row. For cells where the information to be displayed is ambiguous, a link labeled “*n* possibilities” is displayed. To resolve this ambiguity, users click on the link, causing the *Resolver* (Figure 3c) to appear. This tool allows users to examine the multiple available results and specify whether they want the form to show a single item, a subset of the possible items, or all of the items. In the example in Figure 3c, the agent does not know which midterm grades the user wants displayed. The user could select a single item (indicating they want grades from one class), multiple classes (e.g. in a specific department), or all of the classes in the list. When the user is done constructing and resolving any ambiguity, they save the form, preparing Mixer to automate this task.

When a new email arrives that the agent classifies as being associated with a Mixer form, the agent automatically appends the appropriate form to the bottom of the incoming email message and fills it in with the information it has retrieved (Figure 1) [25].

The interaction design supports a piecemeal approach to specifying what a user wants. When the first triggering email arrives, users can create a form specific to the task. As similar tasks arrive, users can easily augment the form by adding or removing columns. These changes are then saved and reflected in future instances of the form. This allows users to address exceptions as they occur and not have to imagine every case as they design the initial form. It allows the form to evolve without forcing the user to start over with each minor variation in a task.

Model of Learning System

Figure 4 details how the interaction design will connect to the learning system. Users begin to construct a new form with the build/demo/repair interface. When they complete the first row and click the “Fill In Table” button, Mixer will pass the demonstrated data to the Learner, which will create a new task model, and, based on this model, attempt to fill in the remainder of the form. Each time the user changes a value in the form (makes a repair), resolves ambiguity with the Resolver, or modifies the structure of the form, the Learner will update the task model and the form filler will update the table. When a new email comes in, the selector agent will compare it to previous triggering messages. It then will choose whether to select a message type or to ignore the message. If it selects a message type, the agent will be able to automatically retrieve all the information, mark areas that are ambiguous, and augment the incoming message with a table at the bottom. If the agent misclassifies the message, the user will be able to correctly classify the message, via the right mouse button, both causing the correct information to be retrieved and providing an update to the learner in the selector agent.

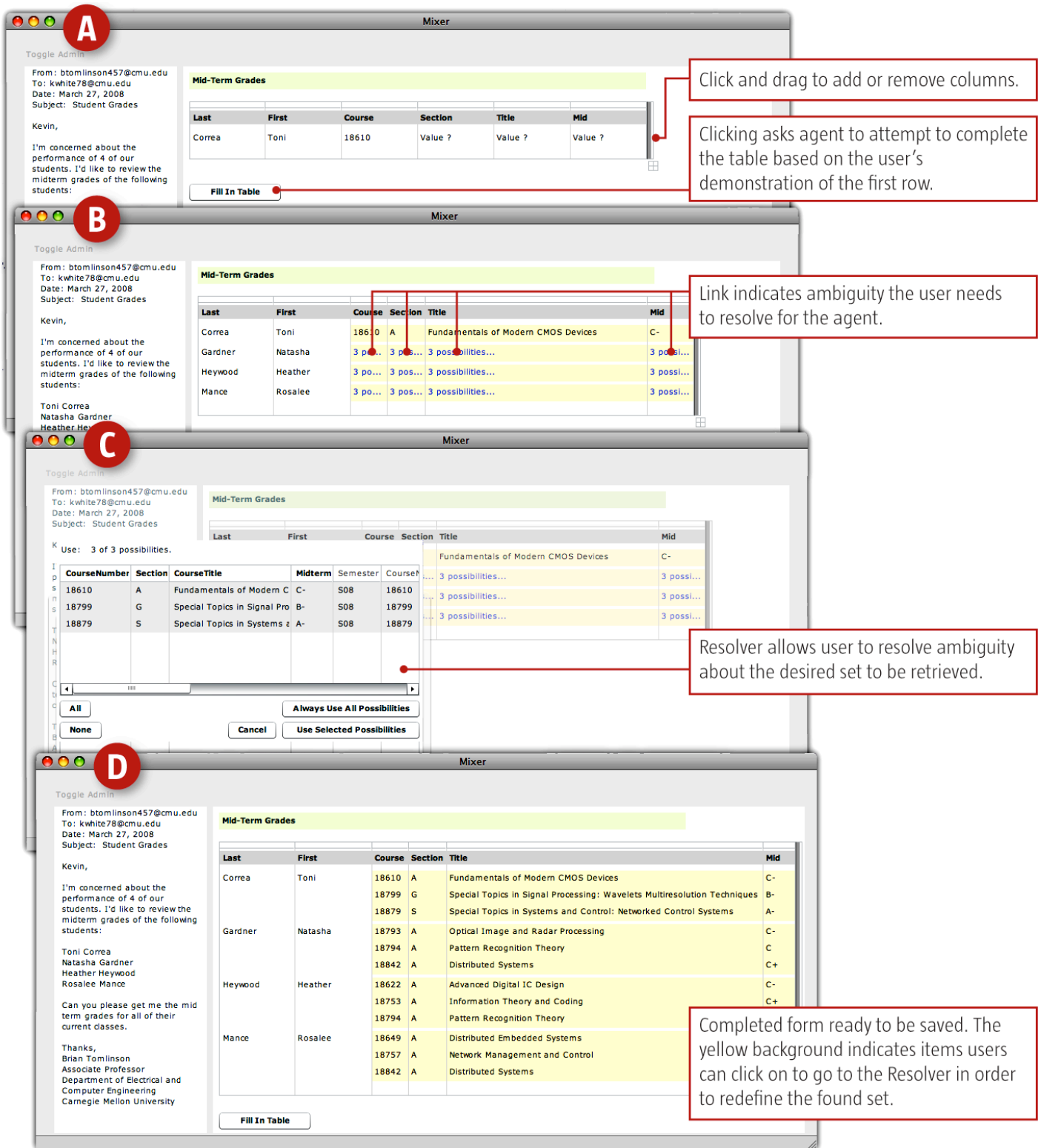


Figure 3 Screen shots of the prototype Mixer interaction from our evaluation. (A) shows form creation. (B) shows the agent's response to clicking on the "Fill In Table" button. (C) shows how the *Resolver* allows users to specify what they want from an ambiguous found set. (D) shows a completed demonstration waiting to be saved.

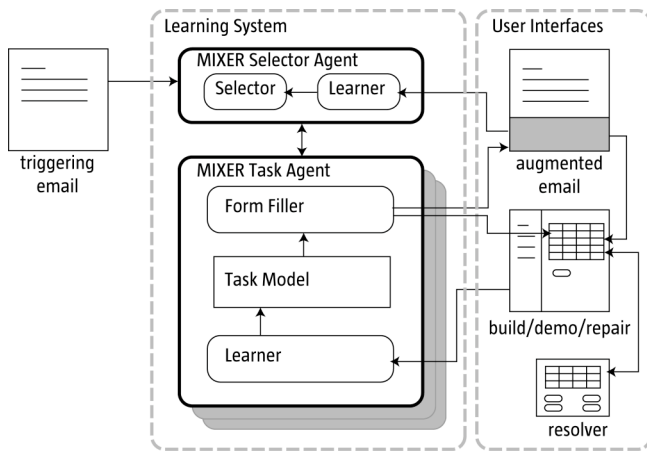


Figure 4 Mixer system functional architecture.

EVALUATION

In order to reduce the risk of developing an agent that users cannot effectively use, we evaluated a prototype of our interface. We used a Wizard-of-Oz approach, where a researcher monitored the participant’s behavior and then simulated the agent’s response. We had the following three hypotheses:

H1: Table-based forms are an effective method of communication between human and agent for information-retrieval tasks:

- Administrators can conceive of and express information demands through designing and demonstrating a tabular form.
- Administrators can make sense of, and work with, information retrieved in collaboration with an agent and presented in tabular form.

H2: Administrators will recognize the benefit of automated information retrieval and would be interested in using this interface for their work.

H3: Mixer will decrease the amount of time needed to complete repeated, information-retrieval tasks.

We selected two *task types* from our contextual inquiries that were typical of the kinds of tasks we had seen performed. Task type “Grades” asked administrators to

retrieve mid-term grades for all classes taken by a given set of students. This required repeated lookups of the same information from many records in a single information source. Task type “Sick” asked administrators to contact all of a student’s instructors. For this task administrators had to retrieve and relate information from more than one information source. In this case, it involved accessing our university’s Student Information System (SIS) to get the student’s course roster with the instructor’s name for each course, and accessing the campus directory to get instructors’ email addresses. For the evaluation, both tasks were significantly shortened to allow participants to complete all work within 90 minutes. For “grades” we reduced the number of students to 4 and the number of classes a student takes to 3. For “sick” we reduced the number of classes a student takes to 4.

We used a within-subjects design. Participants used two tools, completing 4 tasks using Mixer (Mixer) and 4 tasks using Microsoft Excel (Manual). Participants completed both task types (grades & sick) twice with each tool in order to have a First Transaction (FT) and Second Transaction (ST) experience. We used two data sets (D1 and D2) so participants could complete the same tasks with each tool without needing to look up the exact same data; likewise we used different data for the First and Second Transactions of the same task. We counterbalanced for tool order and for datasets, resulting in four conditions (Table 1). Table 2 shows a script view of how a participant experienced the evaluation.

Conditions	Mixer first	Manual first
Dataset 1 (D1) first	A	C
Dataset 2 (D2) first	B	D

Table 1. Four conditions ensured proper counterbalancing between tools.

Participants began each task by viewing an email message. For Mixer FT, participants used the Mixer interface to create a form, demonstrate the retrieval, and resolve ambiguity using the Resolver (Figure 3). For Mixer ST, participants viewed an augmented email (Figure 1) with the requested information in the structure the participant

Cond.	Intro	First Tool				Demo second tool	Second tool				Post-study questionnaire
		First Transaction (FT)		Second Transaction (ST)			First Transaction (FT)		Second Transaction (ST)		
		Grades	Sick	Grades	Sick		Grades	Sick	Grades	Sick	
A:	Demo first tool	Mixer, D1	Mixer, D1	Mixer, D1	Mixer, D1	Demo second tool	Manual, D2	Manual, D2	Manual, D2	Manual, D2	
B:		Mixer, D2	Mixer, D2	Mixer, D2	Mixer, D2		Manual, D1	Manual, D1	Manual, D1	Manual, D1	
C:		Manual, D1	Manual, D1	Manual, D1	Manual, D1		Mixer, D2	Mixer, D2	Mixer, D2	Mixer, D2	
D:		Manual, D2	Manual, D2	Manual, D2	Manual, D2		Mixer, D1	Mixer, D1	Mixer, D1	Mixer, D1	

Table 2. In each condition subjects did four Grade-Report tasks and four Sick-Student tasks: two each with Mixer, and two manually.

created during the first transaction of this task (Mixer FT). When completing Mixer ST tasks related to task type “sick,” the email had an “*n* possibilities” link. To complete this task, participants needed to use the Mixer interface to resolve ambiguity about instructor email addresses for two of the courses.

For Manual, participants used Microsoft Excel as a scratchpad for storing the data they looked up. They copied and pasted information they needed from both the email request and from the information systems into a blank spreadsheet. Participants worked with this spreadsheet until they felt they had all the information required for the task.

Both tools required participants to access our university’s student information system (SIS). In order to address privacy and security concerns, we created a web-based simulation of the system using fictional data.

The evaluation had the following structure (see Table 2):

1. Introduction and demonstration of the first tool.
2. Completion of 4 tasks using the first tool, completing a short evaluation after each task.
3. Demonstration of the second tool.
4. Completion of 4 tasks using the second tool, completing a short evaluation after each task.
5. Completion of post-study questionnaire.
6. Exit interview

Participants beginning the study with the Manual tool were introduced to the simulated version of SIS before proceeding with the Manual tasks. Before commencing Mixer tasks, participants were introduced both to the simulated SIS and given a demo of Mixer. Participants were then asked to create a Mixer form for the demonstration task. This was the participants’ only training in creating Mixer forms, and it typically took less than 10 minutes. Participants received no training on how to deal with Mixer ST tasks (Mixer-augmented emails). Participants were instructed to press an “I’m Done” button on the desktop when they felt that they had successfully completed each task.

Following completion of each of the 8 tasks, participants were asked to respond to a set of 5 statements:

1. The task was tedious to complete.
2. The task was difficult to complete.
3. I’m confident I completed the task well.
4. I knew the right things to do.
5. I completed the task quickly.

Responses were made on a 5-point Likert scale (1 = “strongly agree,” 3 = “neutral,” 5 = “strongly disagree”).

Following completion of all 8 tasks, participants answered a post-study questionnaire containing the TAM (Technology Acceptance Model) [8] instrument. TAM measures a new technology’s perceived usefulness and perceived ease of use. Previous research [9] shows a strong relationship between these two perceptions and eventual system use. The instrument was administered twice: once evaluating

perceptions of creating a Mixer form (Mixer FT) and once evaluating the augmented email based on a previously completed form (Mixer ST). Responses to TAM were made on a 7-point Likert scale (1 = “extremely unlikely,” 4 = “neither,” 7 = “extremely likely”).

The session concluded with a conversational interview addressing the participant’s experience and probing if they could think of tasks in their daily work that Mixer could assist them with.

Participants’ actions as well as audio from the session were recorded using Camtasia. We recorded a start and end time for each task, from which the task time was computed in seconds. Start time was defined as the first action taken by the participant after the task introduction, and end time as the participant’s pressing of the “I’m Done” button.

We conducted evaluations with administrators from our university who all had experience with the university’s IT systems including SIS. 17 participants took part in the evaluation; however, 5 were excluded from the study because they could not complete all 8 tasks within the 90-minute deadline. The excluded participants came from three of the four conditions: one from A, two from B, and two from D. 12 participants completed the study within the 90-minute limit, resulting in 3 participants for each condition.

Each evaluation had a proctor who instructed the participant, administered the questions, and conducted the interview. In addition, a researcher played the role of the agent, observing the participant’s work during the Mixer FT (form building), using a display-mirroring application, to prepare the results of the “Fill In Table” button, as well as to create the augmented email for the Mixer ST.

FINDINGS

Ability to Conceive Forms

Of the 12 participants who completed the evaluation on time, three were able to complete an effective form on their first try while the other nine clicked the “Fill In Table” button multiple times, indicating an exploratory approach to learning what discovering what Mixer can do. Two of the five excluded participants could not conceive of an effective form during their first Mixer task; however, they were successful on subsequent Mixer tasks; successfully constructing a form (Mixer FT) and successfully responding to a form appended to an email (Mixer ST).

Participants took two approaches to create tables describing a one-to-many relationship in the data (such as a single student with several classes). The Mixer interface only allows participants to create and demonstrate the first row, and then use the Resolver to clarify any ambiguity. The two excluded participants who could not complete their first Mixer tasks created a column for each of the many items in that first row. For example, one created a separate column for each class a single student took. Additionally, one of the 9 “exploratory” participants began building the form using

this technique; however, after viewing the agent’s response to this layout, they were able to understand the constraint on the interaction and modify their form in order to complete the task successfully. The other participants all created a single example of the one-to-many relationship (such as listing a single course), which is the behavior the interaction design currently requires.

Acceptance

To understand how acceptable the Mixer interface is to administrative users, we gathered evidence based on the following:

1. Perception of tediousness and confidence
2. Responses to TAM
3. Recognition that Mixer can automate work participants regularly perform

This analysis is limited to the 12 participants who completed the evaluation within the 90 minutes.

After performing each task, participants rated task tediousness, task difficulty, their confidence, and their perception of speediness. The measures for tediousness and confidence returned significant differences in perceptions according to a two-sample t-test (Table 3). Perception of tediousness was significant: $F(1,81) = 190, p < .001$. The average tedious rating of Mixer was 1.8 versus Manual at 4.2. Looking more closely at FT and ST (form creation versus a completed form appended in email), subjects rated Mixer ST, where the agent has retrieved the information and appended it to the incoming email request, as the least tedious: $F(1,81) = 4.14, p = .045$.

Participants rated confidence in Mixer and Manual about the same: $F(1,81) = 2.94, p = .090$. Participants were slightly less confident in their performance on their Mixer FT tasks than on their Manual FT tasks: $F(1,81) = 4.72, p = .033$.

Task	N	Mixer		Manual		p
		Avg	SD	Avg	SD	
Tedious FT*	24	2.1	0.992	4.2	0.637	<0.0001
Tedious ST*	24	1.4	0.717	4.2	1.050	<0.0001
Tedious All*	48	1.8	0.928	4.2	0.859	<0.0001
Confid. FT*	24	3.7	1.13	4.4	0.970	<0.05
Confid. ST	24	4.3	0.917	4.3	0.944	0.9882
Confid. All	48	4	1.072	4.3	0.949	0.1267

Table 3. Two-sample t-test analysis of tediousness and confidence ratings.

TAM responses were deemed reliable with Cronbach alpha scores greater than 0.854 (Table 4). Indices for FT Perceived Usefulness, FT Perceived Ease of Use, ST Perceived Usefulness, and ST Perceived Ease of Use were computed by averaging the participants’ responses.

Eleven of the 12 participants were asked if they could think of tasks in their work that Mixer could help them with (one session had run out of time). Participants offered a total of 40 tasks Mixer could perform for them. The majority of tasks involved data within a single IT system that required

viewing multiple records and screens to retrieve—similar to task type grades. Mixer would be able to directly complete 21 of these tasks. For the remaining 19, Mixer would be able to retrieve a set of information, within which the precise information could be found through filtering, analysis, and/or aggregation using a tool such as a spreadsheet.

TAM Set	Avg rating (of 7)	Cronbach's alpha
FT Perceived Usefulness	6	0.9174
FT Perceived Ease of Use	6.25	0.9319
ST Perceived Usefulness	6.05	0.8541
ST Perceived Ease of Use	6.43	0.8733

Table 4. Technology Acceptance Model measures for creating forms (FT) and addressing forms created by the agent (ST).

Task completion times

Table 5 shows the time participants took to complete tasks in seconds. The effect of tool on task time was significant: $F(1,81) = 55.28, p < .001$. The interaction between tool and transaction was also significant: $F(1,81) = 28.85, p < .001$. Two-sample t-tests were used to more precisely determine how task time differed between the different task/tool combinations. No significant difference in task time was measured between Mixer FT (form building) and either Manual FT or Manual ST ($p > 0.1$). Mixer ST (viewing email) took 74% less time than Manual ST ($p < 0.0001$). The average Mixer task time across both first (FT) and second (ST) tasks was 39% faster than the average Manual task time ($p < 0.0001$).

Task	N	Mixer		Manual		p
		Mean	SD	Mean	SD	
FT grade*	12	298	108	407	106	<0.05
FT sick	12	347	112	311	84	0.7644
FT All	24	322	110	359	106	0.4667
ST grade*	12	70	41	330	89	<0.0001
ST sic*k	12	91	75	285	77	<0.0001
ST All*	24	81	60	307	85	<0.0001
All FT & ST*	48	202	150	333	98	<0.0001

Table 5. Tukey post-hoc analysis of speed in seconds.

There were no significant effects on task time from the interaction between tool and tool order: $F(1,82) = 0.00, p = 0.948$, or from the interaction between tool and dataset: $F(1,72) = 0.02, p = 0.898$.

For FT there was a significant interaction between tool and task type: $F(1,33) = 7.57, p = 0.010$. For the grades task type only, Mixer FT was 27% faster than Manual FT. For ST, where Mixer was always significantly faster than Manual, there were no significant effects on task time from the interaction between tool and task type (grades or sick): $F(1,33) = 2.96, p = 0.095$.

DISCUSSION

Our interaction design hinges on the idea of using a user-created form to establish a shared sense of the problem

between human and agent. This concept comes from our observation that users already build lists as a shortcut for retrieving information, and we wanted to extend this *natural behavior* in order to significantly increase people's abilities to take control of their computational systems. However, there is a fairly large conceptual leap between making static tools and instructing an agent to automate information-retrieval tasks. Results from our evaluation demonstrate that almost all administrators can make this conceptual leap and can successfully create forms to automate the tasks they regularly perform, and they can successfully do this with almost no training; considerably less training than they have had on the information systems they depend upon.

One challenge still remains, envisioning the display of one-to-many relationships. A few participants created forms that displayed data in columns, and these columns should have instead been new rows. Our current design forces users to enter a single row of information and then ask the agent to add additional rows. This strict requirement may limit user's ability to envision the entire table. Users cannot experiment with different layouts to discover the relationships between columns and rows. Most users adopted a productive trial-and-error approach in which they adjusted their use of the form they were creating to the information returned by the agent. We see two potential ways of addressing this challenge. First, we could increase the flexibility of the interaction, allowing users to create multiple rows before asking the agent to proceed. Second, we could have the agent specifically look for instances where users express a one-to-many relationship through the creation of separate columns, and then have the agent automatically modify the layout by reformatting the table to clarify how it prefers to see one-to-many relationships structured.

While the administrators who evaluated our prototype interface could build forms, it is a more difficult task to evaluate if they would actually use them in practice. However, our three instruments (between-task questions, TAM, and the interview about tasks) all point to a strong likelihood of acceptance.

The between-task questions show that participants immediately recognized that our mixed-initiative system could meet its intention of significantly reducing tediousness. This perception should help address the general frustration administrators shared during our contextual inquiries about the information systems they depend upon every day. The rating of confidence does raise some concern. In general, participants felt confident doing the task manually or with Mixer; however, for Mixer ST, where they viewed the email augmented with a pre-filled form, they expressed less confidence. We see two explanations for this result. First, participants may have been confused by the question. For Mixer ST, the agent has done the work and appended the results to the email. Participants may not have known how to rate their confidence for a task they felt they had not actually

performed. And in fact, during the evaluation, many users took several seconds when they encountered the augmented email, wondering what the researchers expected them to do since the work appeared to be done. Second, this lack of confidence could be an indication that they do not completely trust that the agent has performed the work correctly. This finding warrants additional research.

TAM produced very high ratings for Mixer for both *ease of use* and *usefulness*. These results held true of ratings for both creating new forms (Mixer FT) and responding to forms the agent appended to emails (Mixer ST). Scores of 6 or above on a seven-point-scale give us confidence that administrators would likely use Mixer for their work.

The interviews reveal that participants easily recognize after a very brief encounter how they could leverage the Mixer system in their own work. Repeatedly, we heard participants state that if they had had Mixer a day or two before, it would have helped with a particularly tedious information retrieval task. Participants seemed to recognize that Mixer is good at retrieving a small set of data that they could then process to produce the precise information they need for a task. In addition, some participants recognized that Mixer would also aid with repeated tasks that do not have an email trigger, but that have seasonal triggers such as the end or the beginning of a fiscal year.

While perceptions of a system are important in getting users to adopt it, evidence of improved performance helps motivate businesses to make a change. The evaluation shows that Mixer can very significantly reduce the time it takes to complete information-retrieval tasks, even with tasks we purposely shortened to fit within a time-constrained evaluation. Our evaluation looked at tasks performed twice; once where the user must create the form and demonstrates the retrieval, and once where the agent does the work and appends a completed form to an email. However, for tasks that are repeated more than once, the 74% time reduction by Mixer during ST (viewing email) tasks would apply to *every subsequent repetition of the task*. The tasks used in the study were deliberately kept short so that they would fit within the study session. If the grade report task gathered grades for 20 students in a program, each of whom was taking six classes, we expect that Mixer FT (form building) would perform well compared to gathering this information manually. Users spend most of the time in the first transaction laying out columns and demonstrating the first row. Subsequent rows, even if they require disambiguating, are relatively quick to process.

CONCLUSION

As interaction designers begin to approach machine learning as a new material to bring to the design of computational systems that improve the quality of people's lives, mixed-initiative interaction provides a way of framing the application of machine learning as increasing the abilities of people and not just of computers. One of the main challenges mixed-initiative systems face is in creating

a shared understanding of the problem and of the roles between the user (human) and the agent (computer).

Mixer advances mixed-initiative interaction through a novel form-construction communication method that allows users to declare the outcome they want while implicitly demonstrating how the agent should perform the task. Mixer specifically allows administrators to automate repetitive information retrieval tasks they find to be tedious to perform. Our evaluation of the interaction shows form building to be an effective method for people to communicate with the agent. The evaluation also reveals a strong likelihood that administrators would use Mixer if it were available to them, and that it would save them time in their work. The interaction presented in Mixer represents a transition in how office workers engage in computing. Instead of forcing workers to rely on their ability to adapt to the design of IT systems, Mixer allows workers to leverage their expertise in information retrieval to train agents to undertake tedious information tasks for them.

Acknowledgements

This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA), through the Department of the Interior, NBC, Acquisition Services Division, under Contract No. NBCHD030010, Delivery Order No. D0300100003.

REFERENCES

1. Allen, J., Chambers, N., Ferguson, G., Galescu, L., Jung, H., Swift, M., Taysom, W. PLOW: A Collaborative Task Learning Agent. *Proc. of AAAI* (2007).
2. Alterman, R., Feinman, A., Introne, J. and Landsman, S. Coordinating Representations in Computer-Mediated Joint Activities. In *Proc. of the Conference of the Cognitive Science Society* (2001).
3. Bly, S. and Churchill, E. F. Design through Matchmaking: Technology in Search of Users. *Interactions*, 6, 2 (March/April 1999), 23-31.
4. Bresina, J. L. and Morris, P. H. Mixed-Initiative Planning in Space Mission Operations. *AI Magazine*, 28, 2 (2007), 75-89.
5. Burstein, M. H. and Diller, D. E. Cooperative Information Sharing Among Mixed-Initiative Human/Agent Teams. In *Proc. of the Conference on Artificial Intelligence*, (2003).
6. Burstein, M., Laddaga, R., McDonald, D., Cox, M., Benyo, B., Robertson, P., Hussain, T., Brinn, M. and McDermott, D. POIROT: Integrated Learning of Web Service Procedures. *Proc. of AAAI* (2008).
7. Cypher, A. Eager: Programming Repetitive Tasks by Demonstration. In *Watch What I Do: Programming by Demonstration*. A. Cypher (eds). MIT Press, (1993), 205-218.
8. Davis, F. D. Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology. *MIS Quarterly*, 13, 3 (Sept 1989), 319-340.
9. Dix, A., Mancini, R. and Levialdi, S. Communication, Action and History. In *Proc. of CHI*, (1997) ACM Press, 542-543.
10. Fischer, G. and Giaccardi, E. Meta-Design: A Framework for the Future of End User Development. In *End User Development: Empowering People to Flexibly Employ Advanced Information and Communication Technology*. Kluwer Academic Publishers, 2006, 427-457.
11. Fogarty, J., Hudson, S. E., Atkeson, C. G., Avrahami, D., Forlizzi, J., Kiesler, S., Lee, J. C. and Yang, J. Predicting human interruptibility with sensors. *ACM Transactions on Computer-Human Interaction*, 12, 1 (2005), 119-146.
12. Freed, M., Carbonell, J., Gordon, G., Hayes, J., Myers, B., Siewiorek, D., Smith, S., Steinfeld, A., Tomasic, A. RADAR: A Personal Assistant that Learns to Reduce Email Overload. In *Proc. of AAAI Conference on Artificial Intelligence*, (2008)
13. Hearst, M. A. Mixed Initiative Interaction. *IEEE Intelligent Systems*, 14, 5 (1999), 14-23.
14. Horvitz, E. Principles of Mixed-initiative User Interfaces. In *Proc. of CHI*, (1999) ACM Press, 159-166.
15. Horvitz, E. Reflections on Challenges and Promises of Mixed-Initiative Interaction. *AAAI Magazine*, 28(2007).
16. Kosbie, D. S. and Myers, B. A. A System-wide Macro Facility Based on Aggregate Events: A Proposal. In *Watch What I Do: Programming by Demonstration*. A. Cypher (eds). MIT Press, 1993, 433-444.
17. Leshed, G., Haber, E. M., Matthews, T. and Lau, T. CoScripter: automating & sharing how-to knowledge in the enterprise. In *Proc. of CHI*, (2008) ACM Press, 1719-1728.
18. Malone, T. W., Grant, K. R., Lai, K., Rao, R. and Rosenblitt, D. Semistructured Messages Are Surprisingly Useful for Computer-Supported Coordination. *ACM Transactions on Office Information Systems*, 5, 2 (1987), 115-131.
19. Myers, B., Hudson, S. E. and Pausch, R. Past, Present, and Future of User Interface Software Tools. *ACM Transactions on Computer-Human Interaction*, 7, 1 (2000), 3-28.
20. Nardi, B. A. *A Small Matter of Programming: Perspectives on End User Computing*. MIT Press, Cambridge, MA, 1993.
21. Tecuci, G., Boicu, M. and Cox, M. T. Seven Aspects of Mixed-Initiative Reasoning: An Introduction to the Special Issue on Mixed-Initiative Assistants. *AI Magazine*, 28, 2 (2007), 11.
22. Tomasic, A., Zimmerman, J., Hargraves, I., McMullen, R. User Constructed Data Integration via Mixed Initiative Design. In *Proc. of AAAI, Spring Symposium*, (2007).
23. Tuchinda, R., Szekely, P. and Knoblock, C. A. Building Data Integration Queries by Demonstration. In *Proc. of IUI*, (2007) ACM Press, 170-179.
24. Zimmerman, J., Forlizzi, J. and Evenson, S. Research through Design as a Method for Interaction Design Research in HCI. In *Proc. of CHI*, (2007) ACM Press, 493-502.
25. Zimmerman, J., Tomasic, A., Simmons, I., Hargraves, I., Mohnkern, K., Cornwell, J., McGuire, R. VIO: a Mixed-Initiative Approach to Learning and Automating Procedural Update Tasks. In *Proc. of CHI*, (2007), 1445-1454.
26. Zloof, M. M. QBE/OBE: A Language for Office and Business Automation. *IEEE Computer*, 14, 5 (1981), 13-22.