

Human Computation Tasks with Global Constraints

Haoqi Zhang¹, Edith Law², Robert C. Miller³, Krzysztof Z. Gajos¹, David C. Parkes¹, Eric Horvitz⁴
¹Harvard SEAS Cambridge, MA {hq, kgajos, parkes}@eecs.harvard.edu
²Carnegie Mellon University Pittsburgh, PA edith@cmu.edu
³MIT CSAIL Cambridge, MA rcm@mit.edu
⁴Microsoft Research Redmond, WA horvitz@microsoft.edu

ABSTRACT

An important class of tasks that are underexplored in current human computation systems are complex tasks with global constraints. One example of such a task is itinerary planning, where solutions consist of a sequence of activities that meet requirements specified by the requester. In this paper, we focus on the crowdsourcing of such plans as a case study of constraint-based human computation tasks and introduce a collaborative planning system called *Mobi* that illustrates a novel *crowdware* paradigm. *Mobi* presents a single interface that enables crowd participants to view the current solution context and make appropriate contributions based on current needs. We conduct experiments that explain how *Mobi* enables a crowd to effectively and collaboratively resolve global constraints, and discuss how the design principles behind *Mobi* can more generally facilitate a crowd to tackle problems involving global constraints.

Author Keywords

Human Computation, Crowdware, Mixed-Initiative Interaction, Groupware, Collaborative Planning

ACM Classification Keywords

H.5.3 [Information Interfaces and Presentation]: Group and Organization Interfaces

General Terms

Design, Human Factors

INTRODUCTION

Human computation [16, 9] is an evolving paradigm with numerous studies and applications over the last decade. Most human computation tasks explored to date are simple and easy to parallelize. However, several recent studies (e.g., Soylen [1], CrowdForge [7], PlateMate [13]) have tackled more complex tasks by using workflows that link together independent modules that harness different types of human effort. These workflows serve as algorithms that coordinate among the inputs and outputs of different task modules, allowing complex tasks to be decomposed into manageable subtasks

*The first two authors have equal contribution in this work.

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, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CHI'12, May 5–10, 2012, Austin, Texas, USA.

Copyright 2012 ACM 978-1-4503-1015-4/12/05...\$10.00.

I am traveling to San Francisco for a conference and will have a day to explore the city before the conference starts. I have heard great things about San Francisco having so many restaurants serving fresh, local food, and it'd be awesome to get some concrete suggestions on places to go. I'd love to just check out cool things in the city, especially artsy things, and to stop by here and there, either to people watch, get some amazing coffee at a cafe, and just to relax and read a little bit. It is not important that I get to all the touristy places – I don't need to go to Fisherman Wharf or check out the Golden Gate Bridge (I am sure I will see it from somewhere).

In planning our trip, please pay attention to our wishes below:

- have exactly 2 fresh local food restaurants activities.
- have at least 2 cool artsy things activities.
- have at least 1 people watch activity.
- have at least 1 amazing coffee/cafes activity.
- have at least 1 somewhere to read activity.
- have at most 2 amazing coffee/cafes activities.
- spend at least 1 hour on amazing coffee/cafes.
- spend at least 3 hours on fresh local food restaurants.

- arrive at Hyatt Regency
- 1 Cool cafe + People watching
- 2 place to read
- 3 San Fran Museum of Modern Art
- 4 Visit Japantown
- 5 Ferry Plaza Farmers Market
- 6 Art at Grace Cathedral
- 7 Stop at Philz Coffee
- 8 balmy alley murals
- 9 Local food at unique Localvore restaurant
- 10 Westin St. Francis glass elevators
- arrive at Hyatt Regency

Figure 1. Planning mission (left) and itinerary (right)

that the crowd can contribute to independently without having to reason about other modules or the task at large.

Within studies of human computation, an important class of underexplored tasks are those in which the solution must satisfy a set of global requirements. For example, in leveraging the crowd to write an essay, a requester may want to specify requirements on the desired tone, tense, length, structure of arguments, and style of exposition that must hold consistently throughout a piece of writing. Some requirements, e.g., presenting a balanced perspective on a situation, touch upon different components of the essay and depend on its composition. Similar considerations arise in creative tasks such as graphic design, and more mundane tasks such as meeting scheduling. As good solutions rely on the composition as a whole and are marked by interdependence among solution components, such tasks are not amenable to the divide and conquer approach used in most crowdsourcing systems.

As a focal example, consider the problem of crowdsourcing itinerary planning. Planning events such as vacations, outings, and dates often involve an *itinerary* (Figure 1), which contains an ordered list of activities that are meant to be executed in sequence over the course of an event. People going on a trip have preferences and constraints over the types of activities of interest (e.g., “I want a coffee break right after lunch”), how long to spend on different activities (e.g., “I want to spend at least 2 hours in parks”), the composition of activities (e.g., “I want to focus on art galleries and museums for the day”), the budget, and the time available, which define a set of global requirements that an itinerary should satisfy.

Decisions on any particular activity in the itinerary may naturally influence other decisions. As simple examples, spending

time on one activity leaves less time for another, and moving to one location introduces distances to other locations.

To handle tasks with global requirements, we introduce a novel *crowdware* paradigm that provides a single workspace in which a crowd of individuals contribute opportunistically based on their knowledge and expertise and the current solution context, and in which the system (indirectly) coordinates the crowd problem solving effort to resolve global constraints. Crowdware takes inspiration from *groupware* [3], which suggest principles and ideas on communication and collaboration within a shared context, that help a group to accomplish a joint task. We consider how to apply such principles and ideas to crowd workers, who differ from groups in that individuals are only briefly involved, are less willing to spend time grasping the solution context or take meta-level actions, and may not consider the desires of other crowd workers when making decisions.

We focus on itinerary planning as a case study of coordinating a crowd to tackle tasks with global constraints. We introduce a collaborative itinerary planning system called *Mobi*, which takes as input a planning mission containing a set of qualitative and quantitative constraints as articulated by the user and produces as output an itinerary that satisfies the mission. The crowd participates via a single interface—displaying the current itinerary and a stream of ideas generated thus far—that allows individuals to contribute opportunistically given the current context and to see their contributions incorporated into the solution in real-time. *Mobi* focuses the crowd’s attention on aspects of the evolving plan that needs work by prominently displaying a list of automatically generated *todo* items, which point out violated constraints, provide suggestions on how to address them, and promote activities directed at the refinement of the itinerary.

Mobi allows users to specify their desires and needs in *natural language*, thereby enabling complex constraints and preferences to be expressed and used in the planning process. We present two studies, which show that *Mobi*’s design promotes a collaborative planning environment in which the crowd can effectively produce custom-tailored itineraries that satisfy the stated global constraints in user missions. In the first study, we test the effect of displaying the automatically generated *todo* items on the rate at which quantitative constraints are resolved by the crowd, and measure the contribution patterns of the crowd workers. We find that the use of *todo* items promotes satisfaction of constraints at a significantly faster rate than when the items are not displayed, and that the crowd’s editing patterns show evidence of both collaboration and opportunistic planning.

The second study seeks to understand whether the end users believe that crowd-generated itineraries satisfy their stated requirements. We find that users feel that the itineraries contain many activities of interest, mostly or fully satisfy their mission requirements, and are useful for their actual trips. After presenting these results, we revisit and summarize the design principles behind *Mobi* and discuss how they can in general facilitate a crowd to tackle problems involving global constraints.

RELATED WORK

Planning can be viewed as an *iterative task* in which workers make successive edits to improve the solution. There has been some attention on iterative tasks in human computation [11], and an interesting recent example is work by Kittur [6] that recruits workers to collaborate in Etherpad to translate a poem. Workers were able to see their edits reflected in real time (as in *Mobi*), and could communicate to explain their edits via chat. One difference in *Mobi* is that the system uses its sense of the progress made so far, e.g., how full the itinerary is, which constraints are violated, etc., to prompt users on what needs work so as to guide the problem-solving process. Other recent systems recruit a crowd to plan (e.g., Turkomatic [8] and CrowdPlan [10]); our work differs in that we consider planning tasks with global requirements.

Wikipedia can be viewed as an example of a system in which (mostly expert and highly dedicated) contributors write and edit articles to resolve a set of global constraints as defined by Wikipedia’s standards. Much like the way *todo* items are used in *Mobi* to drive progress, template messages and cleanup tags are used in Wikipedia [17] to alert editors of changes that need to be made to improve an article. Such messages are typically managed by human contributors, whereas in *Mobi* *todo* items are managed in an automated manner whenever possible.

Several models have been proposed to describe how people generate plans to achieve goals. The *successive refinement* model advocates a top-down approach, where a high-level goal is decomposed into subgoals iteratively, down to a sequence of elementary actions [14]. In contrast, the planning of many everyday activities (e.g., errands) is often *opportunistic*—i.e., planning decisions happen whenever opportunities arise [4, 5], so that a decision or observation in one part of the plan may suggest new ideas or illuminate problems in a different part of the plan, causing the planner to refocus his attention. Opportunistic planning may involve both top-down and bottom-up processing. For example, in an errand planning experiment [4], it was found that subjects would start making detailed plans (e.g., sequencing individual errands), and then switch to planning on a more abstract level (e.g., by discovering clusters of errands), and back and forth as they refined the plan. *Mobi* is designed with the opportunistic planning model in mind, where individuals in the crowd are allowed to contribute freely as they see fit based on their observations of what needs work given the current solution context.

Real-life planning is a difficult problem for computers; despite advances in automated planning [12], a major challenge is in making sense of people’s goals, preferences and other ‘soft’ considerations [2]. Currently, the automated planner in *Mobi* supports workers by automatically checking constraints and computing trip times and routes. In the future, automation may play a more active role in the planning process by learning about different requirements, suggesting activities and their composition in the itinerary, or even detecting and adding important constraints that may have been missed by the requester.

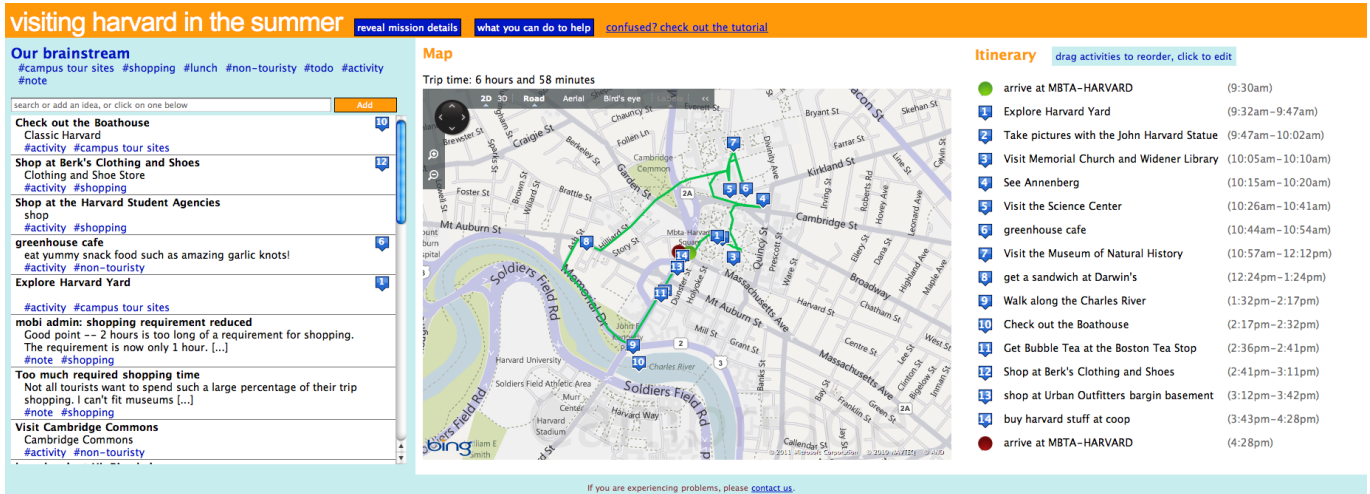


Figure 2. The Moby planning interface consists of the information panel (top), the brainstream (left), and the itinerary viewer (right).

There are several existing commercial systems that either allow groups to plan trips for themselves or to ask friends and other members for suggestions. Examples include Gogobot, triporama, Kuku, and FriendTripper. Moby differs from these systems in that it seeks to produce not only suggestions for activities but an itinerary satisfying a set of global requirements, and can focus the crowd on making contributions where they are most needed.

MOBI: A SYSTEM FOR CROWD ITINERARY PLANNING

Moby takes as input a planning mission consisting of preferences and constraints, and generates an itinerary by having a crowd plan asynchronously using a shared interface. Workers invited to contribute can view the current plan and all ideas proposed thus far, and make contributions as they see fit. Edits can be made at any time and without restrictions, and the itinerary is automatically saved after each change. We now describe Moby's interfaces for specifying the planning mission and for assembling the itinerary, and discuss how these two interfaces support the process of generating itineraries and resolving constraints. Moby's planning interface is shown in Figure 2.

Specifying the Planning Mission

Our target users, also referred to as *requesters*, are people who are interested in planning a trip. To start planning, the requester enters a planning mission using a simple web interface, which specifies the title and description of the trip, start/end locations and times, and whether they will use public transit or drive between locations in addition to walking.

Requesters can express two kinds of constraints: *qualitative* and *quantitative*. Figure 1 shows an example of a planning mission that includes both types of constraints. Qualitative constraints are specified in natural language (e.g., in a paragraph), and describe the nature of the trip, what the user hopes to accomplish, who they are traveling with, etc. Quantitative constraints are specified by creating *categories* using arbitrary natural language phrases (e.g., “cool artsy things,” “by the ocean”), and assigning preferences and limitations over

those categories. One can specify constraints on the number of activities in each category (e.g., “I want to have up to two *museum visits*”), as well as on the amount of time to spend on activities in each category (e.g., “I want to spend at least two hours on *cool artsy things*”). Such constraints can also be used to express the preferred combination of activities in the plan (e.g., “I want to spend half of my time on activities *by the ocean*, and the other half on activities *in the city*”). In our prototype, the domain-specific language for quantitative constraints allows for constraints encoded in the form of “I want {*at most, at least, exactly*} [number] {*activities, hours*} of {*cat₁, cat₂, . . . , cat_n*},” where *cat_i* refers to the *i*-th category.

Both qualitative and quantitative constraints contain natural language, and can express ‘soft’ considerations that the computer cannot tackle alone. In addition to these constraints, the system maintains a pair of time constraints, which state that the cumulative duration of the activities in the itinerary should not be significantly less than, or greater than, the duration of the trip specified by the user.

Assembling the Itinerary

Once a requester specifies a planning mission, workers can use Moby's planning interface to view the mission by clicking on the *reveal mission details* button in the *information panel* (see Figure 2, on top). The planning interface consists of two key components – the brainstream and the itinerary viewer.

Brainstream

The *brainstream* (see Figure 2, on left) is a collection of everyone's ideas. An idea can be an activity (“something to do or see”) or a note (“a thought about the plan”).

To view ideas in the brainstream, one can either scroll down the list, click on a hashtag to display ideas belonging to a particular category, or use the auto-completing search box. Clicking on an idea reveals additional details via a dialog box, and presents the option to edit the idea, or in the case of an activity, an option to add it to or remove it from the current

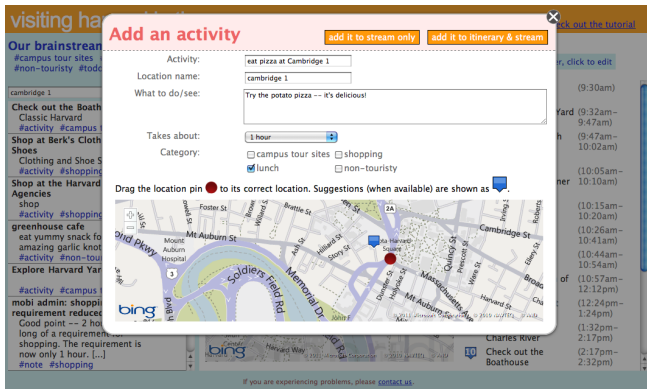


Figure 3. Adding a new activity to the brainstorm via a dialog box

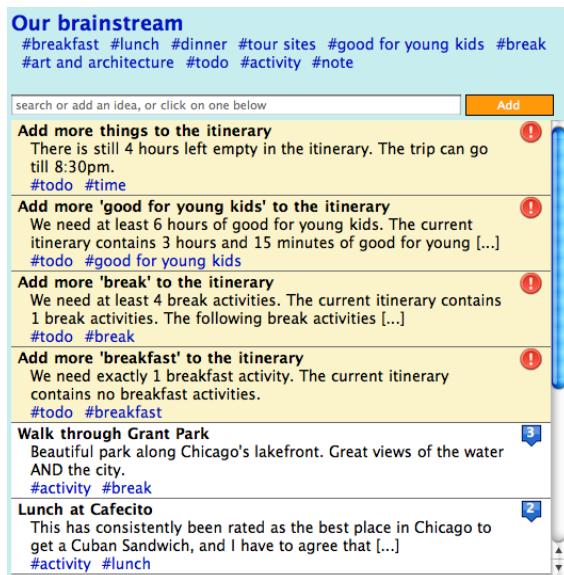


Figure 4. The brainstorm displays system-generated todo items, alerting workers about what needs work.

itinerary. A blue badge next to an activity indicates that it is already in the current itinerary.

To add a new idea (an activity or a note), one can type a title into the search box and click 'add'. If similar ideas already exist, a drop down list will appear, which helps to prevent duplicates and promote editing. For notes, workers can fill in a description if desired. For activities, the activity editor (see Figure 3) asks workers to provide the name of the location, what to do or see, the activity's duration, and the (requester-defined) categories that the activity belongs in. In the same editor, workers can view a map, which allows them to mark the location of the point of interest. Workers can decide to add the activity to both the itinerary and the brainstorm, or only to the brainstorm for the time being.

The brainstorm allows people to brainstorm together and build upon each other's ideas. It keeps around all alternative activities, and allows workers to quickly access them through the hashtags and the search box. By adding notes, workers can identify areas that need work or raise questions about the

plan's feasibility, that other workers or the requester can help to address or provide further comments on. The brainstorm's design draws inspirations from social technologies such as Twitter and Piazza, which aggregate information into a feed or stream that one can then easily process.

If the current itinerary does not satisfy a stated quantitative constraint or is over time or under time, the violated constraints are automatically turned into *todo items* that are displayed at the top of the brainstorm in a different color, alerting workers to what needs work (e.g., see Figure 4). The todo items suggest specific actions, e.g., "Add more lunch activities," or, "The itinerary is over time. Try reordering itinerary items. You can also edit or remove items." Todo items also provide natural language explanations of how the current itinerary violates particular constraints, e.g., "You need exactly one lunch activity but there is currently none in the itinerary," or "The itinerary is over time because the trip must end by 9pm."

We note that the system is able to check arbitrary quantitative constraints and generate todo items without understanding the meaning of the natural language categories; this is because workers associate activities with the categories they belong in when they are suggested. As we will show in the next section, todo items are an important design element that helps to accelerate the speed with which quantitative constraints are resolved.

Itinerary Viewer

The *itinerary viewer* (see Figure 2, on right) consists of an itinerary and a map. The itinerary displays the activities in order, alongside the times during which they are scheduled to take place, with travel times between locations automatically computed and accounted for. It is accompanied by a map showing the activities' locations and routes between locations. The map and itinerary allow crowd workers to see at a glance whether the plan is coherent, that is, if any activities are out of place (e.g., lunch happening too early, activities whose order can be swapped to avoid unnecessary back-and-forth travel), or if too much or too little time is spent on an activity. The itinerary doubles as an editor—workers can drag and drop activities to rearrange their order, and click an activity to see its details, edit it, or remove it from the itinerary. On any itinerary change (i.e. via adding, removing, editing, or reordering of activities), the itinerary, activity times, map display, trip time, and todo items automatically update, which provides direct feedback to the workers as they refine the itinerary.

Mobi promotes collaboration by making the plan always visible and editable by everyone. This follows the WYSIWIS ('What You See Is What I See') principle [15], which ensures that all participants have equal access to shared information. Mobi also supports *opportunistic planning*, by providing support for both top-down and bottom-up planning, and a fluid way to move back and forth between the two. For example, as workers plan at a detailed level (e.g., suggesting activities in the brainstorm), they may become aware of shortcomings of the current itinerary, which in turn prompts them to start considering the itinerary as a whole; likewise, when workers

I am taking my teenage daughter to NYC in a few weeks for a mother and daughter getaway. We are looking for places to shop and visit -- my daughter loves fashion and design, so suggestions along those lines are most helpful. We are not big on the touristy places, but instead, finding amazing dessert is a must! :) I have also always wanted to see a Broadway show, so suggestions for shows to see would be helpful. Thanks so much!

In planning our trip, please pay attention to our wishes below:

- have exactly 1 simple lunch (not fast-food) activity.
- have exactly 1 italian for dinner activity.
- have exactly 1 broadway shows activity.
- have at least 2 amazing desserts activities.
- spend at least 3 hours on fashion or design.

- arrive at Eventi Hotel
- 1 Visit The Museum at FIT
- 2 Visit to Mood
- 3 Madison Square Garden
- 4 Chocolate to the Max
- 5 Sara D Roosevelt Park
- 6 Lunch at Soho Park
- 7 Shop the boutiques of Soho
- 8 Cheesecake at Eileen's
- 9 Tour the fashion flagship stores of 5th Ave.
- 10 Carmine's Dinner
- 11 Wonderland
- arrive at Eventi Hotel

(a) Mother/Daughter NYC

We are a family visiting Chicago for the first time. As we are traveling with our children (who are 5 and 8), we are looking for an itinerary for touring the city that would be suitable for young kids. We love anything from seeing the sites, checking out art and architecture, to having delicious food, but want to make sure we take things easy and that our children have a great day as well!

In planning our trip, please pay attention to our wishes below:

- have exactly 1 breakfast activity.
- have exactly 1 lunch activity.
- have exactly 1 dinner activity.
- have at least 4 break activities.
- have at least 1 art and architecture activity.
- spend at least 6 hours on good for young kids.

- arrive at Hampton Inn & Suites Chicago-Downtown
- 1 Breakfast at Melicafé
- 2 Adler Planetarium
- 3 Art and Architecture
- 4 Coffee & Tea break
- 5 Grant Park
- 6 Lunch at Cafecito
- 7 Navy Pier
- 8 Architectural Boat Tour
- 9 Shedd Aquarium
- 10 Dinner at Gino's East
- arrive at Hampton Inn & Suites Chicago-Downtown

(b) Chicago with young children

Me and three of my best friends just graduated college, and we are looking to have a wild time in Vegas! We don't really care for the shows, and would rather spend our time playing poker in the casinos, hanging out at cocktail bars, and partying till late in the clubs. We are on a bit of a budget though, so low stakes tables suggestions and affordable lunch and dinner options are particularly helpful.

In planning our trip, please pay attention to our wishes below:

- have exactly 1 buffet lunch (\$20 or less) activity.
- have exactly 1 dinner at a steakhouse activity.
- have at least 1 cocktail bars activity.
- have at least 1 clubs activity.
- spend at least 3 hours on casino poker rooms (low stakes).

- arrive at MGM Grand Hotel and Casino
- 1 Breakfast
- 2 Forum Shops
- 3 Poker at Bill's Gamblin Hall
- 4 Lunch Buffet at Aladin
- 5 Check out the fountains at the Bellagio
- 6 low stakes table games
- 7 House of Blues
- 8 Dinner at Delmonico's
- 9 Head up to Mix Lounge Mandalay Bay
- 10 Clubbing at Tao
- 11 strip club
- arrive at MGM Grand Hotel and Casino

(c) Vegas with buddies

We are going on a family vacation to DC together with our 10 year old boy and 12 year old daughter. I want our kids to get a sense of our nation's history and to have some understanding of how our government works, so sites along these lines would be most helpful. We want it to be a relaxing vacation (no need to see everything), but at the same time we definitely want the kids to learn a lot and be engaged.

In planning our trip, please pay attention to our wishes below:

- have exactly 1 quick healthy lunch activity.
- have exactly 1 ethnic restaurant for dinner activity.

- arrive at Embassy Suites Washington D.C. - Convention Center
- 1 Visit the Spy Museum
- 2 Declaration of Independence
- 3 The Newseum. Great exhibits about history and current events
- 4 United States Botanic Garden
- 5 National Museum of Natural History
- 6 Lunch
- 7 United States Holocaust Memorial Museum
- 8 Visit the WWII Memorial
- 9 Visit the Korean War Memorial
- 10 Visit the Lincoln Memorial
- 11 Visit the Vietnam War Memorial
- 12 The White House
- 13 dinner at "The Source"
- arrive at Embassy Suites Washington D.C. - Convention Center

(d) Family in DC

Figure 5. Experiment: planning missions and the corresponding itineraries generated by the crowd in the todo items condition

refine the itinerary, they may think of new activities to add to the mainstream, or ways to elaborate on the details of a particular activity in the current itinerary.

EXPERIMENT: TODO OR NOT TODO

We hypothesize that elements of Mobi's design, namely todo items and having a shared interface in which the crowd can work off the current solution context and existing ideas, promotes the crowd to effectively and collaboratively resolve the users' stated constraints so as to produce itineraries that satisfy planning missions. Focusing first on quantitative constraints, we conducted an experiment using two versions of Mobi—one that displays todo items and one that does not—to evaluate the effect of todo items on how quickly the crowd can reach feasible solutions that satisfy the stated constraints.

Method

We created custom day-trip planning missions for each of eight major U.S. cities: New York, Chicago, DC, Las Vegas, Los Angeles, San Francisco, Seattle, and San Diego. We recruited Mechanical Turk workers in the U.S. who have a 95% or higher approval rating to contribute to the planning missions, by working on human intelligence tasks (HITs) in which the Mobi interface was fully embedded. The interface

is nearly identical to that shown in Figure 2, with the addition of a submit button on the bottom, a 'HIT instructions' button replacing the 'what you can do to help' button in the information panel, and the addition of a 'continue to improve the itinerary' todo item that displays only when there are no other todo items (i.e. all quantitative constraints are satisfied). Turkers were asked to make 'micro-contributions' as they plan the trip with other Turkers, and were told that they can submit a HIT as soon as they have made *any* contribution. Turkers were paid 15 cents per HIT, and no verification was used other than requiring Turkers to have made some edit (however small) to the mainstream or itinerary before submitting the task. For half of the cities, the version with todo items was posted prior to that with no todo items, and the order of posting was reversed for the other cities. Missions were posted for up to four days. Other than the display of todo items, the interface, job description, and instructions are identical in the two conditions.

Results I: The Generated Itineraries

Figure 5 shows a sample of the planning missions and the corresponding itineraries generated by Turkers in the todo condition. All eight itineraries satisfy the stated quantitative con-

	NYC	Chicago	Las Vegas	DC
# unique workers	17	15	16	21
# workers with winning ideas	6	5	7	8
# activities in brainstorm	35	16	18	28
# activities in itinerary	11	10	11	13
# edits in brainstorm	50	54	43	57
# edits in itinerary	193	140	75	154
# notes in brainstorm	1	0	9	1
# of HITs	64	31	47	50
Total cost	\$9.60	\$4.65	\$7.05	\$7.50

Table 1. Summary statistics about the final itineraries of the examples shown in Figure 5, including contributions by and payments to Turkers. Winning ideas are activity suggestions that are in the final itinerary.

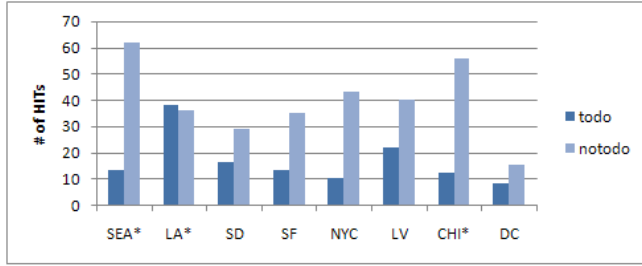


Figure 6. Number of HITs required to satisfy all quantitative and system generated time constraints for each city in the todo and no todo conditions. For cities marked by an asterisk, itineraries in the no todo condition still have violated constraints; in such cases we reported the number of HITs thus far.

straints. From the final itineraries, it appears that Turkers not only pay attention to the quantitative constraints, but also the mission description, e.g., by filling the itinerary with government/history related activities in DC, and by offering low-cost options in Vegas.

Table 1 summarizes, for each of the examples shown in Figure 5, statistics about the final itineraries, different types of edits Turkers made, and the amount of money paid to workers. We see that the final itineraries contain original ideas from multiple workers. Specifically, Turkers generated just over twice as many ideas for activities as are in the final itineraries, and generally used notes sparingly. When notes were added, they provided commentary on alternative suggestions (“They are a better place than Pasty’s by far, and have better service, plus that perfect dessert.”), noted errors in activities (“Barbary Coast isn’t called ‘Barbary Coast’ anymore”), presented general advice (“You can buy a MealTicket which will allow you to eat free at many places.”), or pointed out problems with the plan (“why are we eating so much dinner?”).

Results II: Influence of Todo Items

Results show that when prompting users with todo items, quantitative constraints are satisfied significantly more quickly than when todo items are not displayed. We measure the speed of the process in number of HITs performed. One spammer in the no-todo condition submitted multiple HITs for a single piece of work (e.g. adding an activity, filling in its details, and placing it into the itinerary) as three separate HITs. For this worker, only the itinerary-changing HITs were counted, but for all other workers, all HITs were counted.

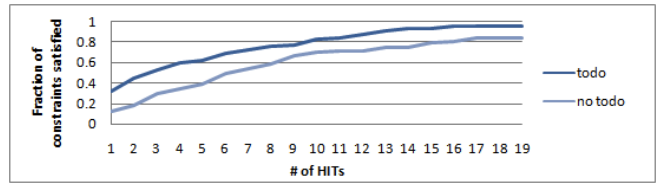


Figure 7. Cumulative distribution of the violation duration of constraints in the todo versus no todo conditions, showing the fraction of constraints satisfied after at most k HITs since the time it was last violated.

We make three observations. First, we found a significant difference ($t(7) = 3.65, p = 0.0082$) in the number of HITs it took to satisfy (for the first time) all of the stated quantitative constraints between the todo condition ($\mu = 16.5, \sigma = 9.65$) and the no todo condition ($\mu = 39.5, \sigma = 14.8$);¹ see Figure 6 for a city-by-city breakdown.

Second, there is also a significant difference ($t(7) = 4.247, p = 0.0038$) in the number of HITs it took to satisfy *all* constraints for the first time (this includes system generated time constraints) between the todo condition ($\mu = 22.5, \sigma = 8.5$) and the no todo condition ($\mu = 45.38, \sigma = 13.9$).

Finally, as constraints can be violated and satisfied repeatedly throughout the planning process, we sought to understand how quickly constraints are satisfied on average. We introduce the notion of the *violation duration* of a constraint, which is the number of HITs it takes for a constraint to be satisfied by the itinerary since it was last violated (which could be when it is first introduced). The average violation duration of quantitative constraints is shorter for the todo condition ($\mu = 5.64, \sigma = 6.34$) compared to the no todo condition ($\mu = 10.5, \sigma = 10.97$); the result is statistically significant ($t(134) = 3.206, p = 0.0017$).

Figure 7 shows the cumulative distribution of the violation durations of constraints in the todo versus no todo conditions. We observe that for any violation duration (in number of HITs), a larger fraction of the constraints are satisfied within that duration in the todo condition than the no todo condition. We also see that more than half of all violated constraints become satisfied after three or fewer HITs in the todo condition.

Figure 8 shows, for the todo versus no todo conditions, the rate at which *each* constraint gets satisfied as workers contribute to the planning effort for the Seattle and Chicago planning missions. We observe that constraints are satisfied much more quickly in the todo condition. The Chicago case is particularly interesting; whereas in the todo condition a worker violated a previously satisfied constraint while editing and proceeded to make successive edits that led to satisfying all of the constraints, in the no todo condition a satisfied constraint was violated and then left unaddressed for a significant amount of time. This example illustrates the power of immediate feedback—when an edit to the itinerary violates

¹ In some cases for the no todo condition, no itinerary satisfied all the stated requirements in the course of the experiment. In such cases the number of HITs completed thus far was used as a lower bound for comparison.

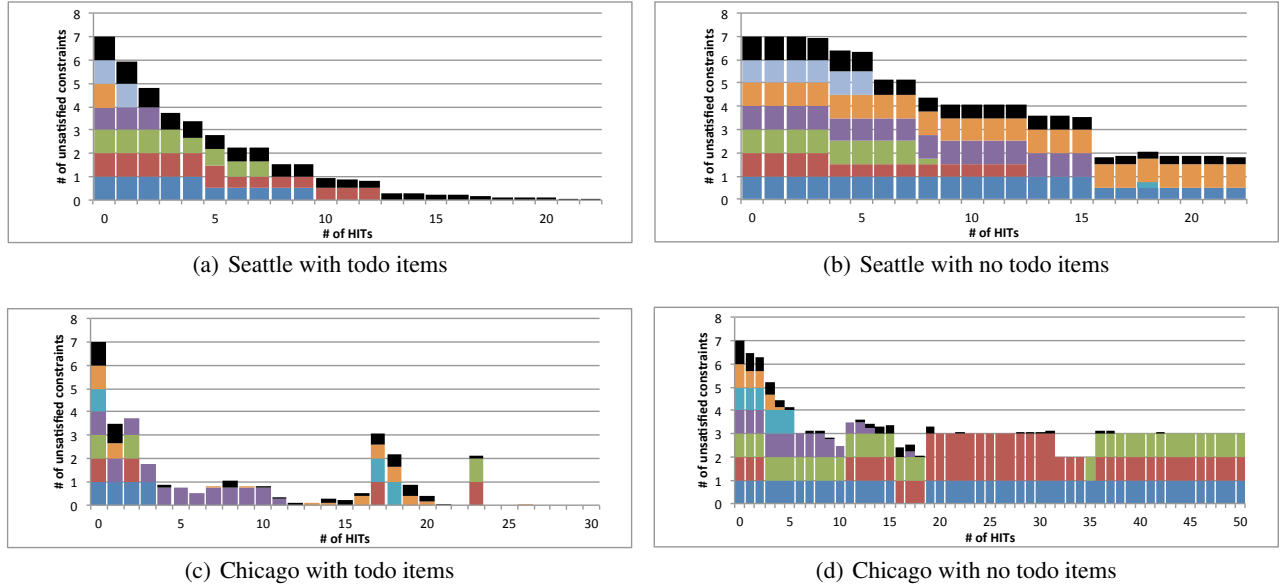


Figure 8. The unsatisfied constraints during the planning process for the Seattle and Chicago missions. The height of each bar indicates the number of constraints unsatisfied after k HITs. Each colored segment represents a particular quantitative constraint, and its height indicates the extent to which it is violated. The black segment represents the percent by which the itinerary is over time or under time (when it is greater or less than 5%).

some constraints, the automatically generated todo items are able to not only alert workers as to what needs fixing, but also make them aware that their edits have direct effects on the constraints associated with the planning mission.

Results III: Editing Patterns

Having shown that todo items play an important role in focusing the crowd’s effort towards satisfying the quantitative constraints, we turn to investigate the crowd’s work process while using Mobi in the todo condition. In particular, we look for evidence of collaborative behavior from the crowd, and at the way that they plan using the current context of the plan.

We focus first on the process of generating ideas for activities. We observe that roughly half the edits to the brainstorm contain new suggestions (52%) while the other half (48%) are edits of existing ideas in the itinerary. Of the edits, 72% are edits on ideas that originated from someone else (i.e., other than the person editing), which suggests that workers are working off others’ contributions as they refine ideas and the itinerary. When editing an activity, we see that edits are predominantly on an activity’s duration (80%), but there are also edits to change title/descriptions (7%) and to correct an item’s location coordinates (12%). Although duration edits are often in the context of resolving some constraints, edits to the title, description, and location are particularly encouraging to see as they suggest that the interface is providing (via the brainstorm, map, and itinerary) means for users to discover and improve on existing ideas.

Turning to the patterns of itinerary edits, we observe that most of the contributions come from adding (31%) and re-ordering activities (32%), but that workers also edit existing ideas (22%) and remove activities (14%). This is encouraging to see because workers are using the different actions

available to them to plan as they see fit as they improve the itinerary. When tasks are left to run after the quantitative constraints are all satisfied, we observe that itineraries continue to evolve; workers replace activities in the itinerary with other activities, reorder the itinerary, edit existing items, and so on. While constraints may be violated during such edits, workers were reminded of such violations by the todo items and violated constraints were quickly satisfied again (e.g., see Figure 8(c)). This suggests that new ideas can continue to be generated and incorporated into the itinerary; in fact, workers are encouraged to do both because they are paid for such contributions and because we display a todo item that asks workers to continue to improve the itinerary whenever all quantitative constraints are met.

Throughout the experiment, we saw very few Turkers who blatantly tried to game the system. The kinds of gaming behavior we did observe generally fell into two categories. In one, a Turker under-specifies an activity, either by creating an activity without filling in its description and location, or by adding a note containing a suggestion for an activity instead of just adding the suggested activity. In the other, a Turker would fully specify an activity, but use two or even three HITs to do so, by spending a HIT on creating the activity, another to edit its details, and another to add it to the itinerary – when all this can be accomplished with a single ‘add activity’ action.

While it is certainly useful to consider refinements that would curb such behaviors (e.g., by requiring activities to contain descriptions, by not allowing workers to submit HITs in which they have only edited their own ideas, etc.), we note that such gaming behaviors from a small set of Turkers did not seem to have a negative influence on the planning processes nor the resulting solutions. In particular, we saw that

I am only here for 4 days, arriving the morning of October 27th and leaving the 30th, so I would like to see as much of the touristy attractions as possible. I will be with 3 others, my father, my older brother, and my uncle, and we are celebrating my 21st birthday. I have a nice camera and enjoy taking pictures. Restaurants are very important to my father, and we would like to go to some of the nicer restaurants. We all eat everything and go out to eat very often, so don't worry about the types of restaurants. We obviously will be wanting to gamble and see some of the casinos, but I would also like to go out to a club or see what other nightlife activities they have. I would like to do more active than less active things. Sitting by the pool for hours isn't my idea of fun on vacation because I can do that at home.

In planning our trip, please pay attention to our wishes below:
 - spend at least 1 hour on Nice dinner.
 - spend at most 3 hours on Gambling.
 - spend exactly 2 hours on Nap/Relaxing.

- arrive at Harrah's Las Vegas Hotel
- 1 Breakfast at Bouchon
- 2 Forum Shops at Caesars Palace
- 3 Madame Tussauds Las Vegas
- 4 Joe's Seafood, Prime Steak & Stone Crab
- 5 Cosmopolitan Hotel
- 6 Spa Mandalay
- 7 Secret Pizza
- 8 Nap/Relaxing
- 9 Circus Circus
- 10 Bellagio Fountain Show
- 11 Hennessy's Tavern
- 12 Marquee Nightclub
- 13 Fremont Street Experience
- arrive at Harrah's Las Vegas Hotel

(a) Subject 1: Las Vegas

- arrive at 8442 Palm Pkwy, Orlando, FL
 - 1 Travel to Sea World
 - 2 Pets Ahoy
 - 3 Sharks Underwater Grill
 - 4 Clyde and Seamore Take Pirate Island
 - 5 Blue Horizons
 - 6 One Ocean
 - 7 A'Lure the Call of the Ocean
 - 8 Eat at The Ravenous Pig
 - arrive at 8442 Palm Pkwy, Orlando, FL
- Visit SeaWorld. My plane arrives at Sep. 2nd, 11:30AM and I'll arrive at SeaWorld around 1:00PM. However, I don't want to miss any show. Also, I would be appreciated if there is any money tips or coupons I can get with regard to theme parks.
- I'm traveling with my wife pregnant, so it would be very helpful if I could get some information related to my pregnant wife.
- We'd like to find some good but not necessarily too cheap restaurants for dinner for two people, around \$30~50 total.
- In planning our trip, please pay attention to our wishes below:
 - Plan a fun outing that you'd personally love to go on!

(b) Subject 2: Orlando

Figure 9. User study: planning missions and corresponding itineraries generated by the crowd

poorly formed ideas were simply ignored, removed from the itinerary, or edited by another worker who discovered them via the auto-complete search box in the brainstream, which occurred as a natural part of the iterative process through which workers improved the itinerary.

END-TO-END USER STUDY

Having seen that workers can resolve quantitative constraints effectively using Mobi, we conducted a user study through which we seek to evaluate how well the generated itineraries satisfy not only *quantitative* constraints, but also the stated *qualitative* constraints, from the perspective of requesters.

Method

We recruited 11 subjects to participate in the study via the use of several university mailing lists. We specified requirements that participants be actually planning a forthcoming trip to a major U.S. city. Recruited subjects are a mix of undergrads, graduate students, and research scientists. We report on the responses of 10 of the users, as one of the users specified a trip destination that is not a major U.S. city. Subjects were instructed to describe their planning mission, which includes qualitative and quantitative preferences and constraints. Participants were given unlimited access to Mobi for over a week, during which they were free to modify their planning mission or to participate in the planning process. Missions were crowdsourced on Mechanical Turk as was done in the todo versus no todo experiment. At the end of the study, subjects were given a questionnaire, which asked them to evaluate the final itinerary and to describe their experiences using Mobi. Subjects were each paid \$30 for their participation.

The trip destinations specified by the users include Boston, New York City, San Francisco, Las Vegas, Orlando, and Washington DC. The planning missions vary in length and specificity. Figure 9 provides two examples of user missions and the generated itineraries.

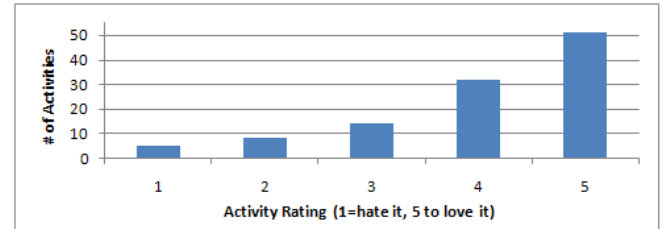


Figure 10. Histogram of activity ratings

Results

To see how well the generated itineraries satisfy the end users' requirements, we consider three measures of the *quality* of an itinerary, namely the extent to which it (1) contains activities that the requester likes, (2) satisfies the qualitative and quantitative constraints specified in the planning mission, and (3) serves its purpose as a plan that is feasible, useful, and executable in real life.

1. Do itineraries contain activities that the requesters like?

Users were shown each of the itinerary activities (title, description, start time, end time, duration) and asked to rate how much they think they would enjoy each activity on a 5-point scale (1="hate it", 5="love it").

Figure 10 shows a histogram of the activity ratings across all 10 participants. The mean rating was 4.03 ($\sigma = 0.44$). Users also mentioned that the activities are diverse, interesting, and often unknown to them prior to using Mobi.

2. Do itineraries satisfy the qualitative and quantitative constraints specified in the planning mission?

All of the users answered that their itinerary fulfilled most or all of the requirements they had specified. Some users noted specific activities that they do not like (e.g., "I just happen to be afraid of bungee jumping because it seems so unsafe, but

a similar activity would be fun,” “I am under age so the wine thing would not be great for me but everything else sounds great.”) or complained about too many activities (e.g., “There was far far far too much packed into a single day, but the ideas were all totally interesting.”) or visiting too many parts of a city in one day (e.g., “For the most part, it was a good mix of things to do. I was not expecting to travel so much up-town/downtown in one day though.”) These problems can be partially explained by the fact that many constraints (e.g., the notion that an itinerary shouldn’t be too packed) are assumed or missed and therefore not explicitly stated by the users. One potential solution is to have the requesters evaluate the itineraries as they are being created and add the missing constraints to the planning mission. In fact, as a preliminary test, we took two of the users’ feedback and entered them as todo items (i.e., “Let’s just stay midtown and remove downtown activities,” “The harbor island suggestions are great but one island would be enough. Please adjust time durations accordingly so the day is not so packed.”). After resuming the task, we observe that after just a few HITs workers have already addressed the issue by removing offending activities, reordering activities (so that meals occur at reasonably hours), and adding additional activities (replacing a concert downtown with a Broadway show in midtown).

3. Are the itineraries feasible, useful, and executable in real-life settings?

We asked users if they *would* or *did* use the itinerary in real-life. All users expressed that they would use the itinerary as is, some version of the itinerary, or selective ideas in the itinerary. When asked “*If Mobi were made available for general use, how likely would you want to use such a tool again for recruiting the crowd to help you plan a trip?*”, 7 out of 10 users answered *likely* or *very likely*, 2 answered *neutral* and only 1 answered *unlikely*, showing the usefulness of Mobi in practice.

Three users actually followed the itinerary or used the ideas in the itinerary in their real-life trips. One subject reported that “having other people involved in the idea-creation process was extremely helpful. It sparked all sorts of ideas that I kept in the back of my head throughout the weekend.” Another subject remarked that his “trip was mostly in the plan, but all restaurant plans changed due to necessity during the trip.”

We found a dichotomy of users: those who are interested in obtaining a *fully-specified itinerary* and those who are interested in a *loose itinerary* that contains an unordered set of suggested activities that leave room for exploration. An interesting solution is to allow requesters to choose between a fully specified or loose itinerary, which in turn translate into constraints that specify the maximum number of activities in the itinerary, the amount of buffer time between activities, and whether activities need to be ordered.

One of the most frequently mentioned benefits of Mobi is that both the *idea generation* and the *planning* are fully automated, thereby “integrating all the factors one would consider in planning an itinerary,” yet making “the time spent creat-

ing the plan minimal.” Most users (7 out of 10) said that they are comfortable with an anonymous crowd planning their trip. Furthermore, results show that requesters mostly left the planning up to the crowd. In particular, 3 out of 10 users said that they never or rarely checked on the progress of the itinerary, 5 did so occasionally, and only 2 frequently. Likewise, 7 out of 10 users said that they never went back to modify the mission details or add notes. As one user puts this succinctly: “the process seemed to work smoothly without my intervention.”

DISCUSSION

Having demonstrated the effectiveness of Mobi for helping the crowd to resolve qualitative and quantitative constraints in the itinerary planning setting, we now revisit the elements of Mobi’s design and discuss how these elements may in general inform the design of systems that facilitate a crowd to tackle problems involving global constraints.

Keeping the crowd, the solution, and the context together

Compared to the design of most other crowdsourcing systems for tackling complex tasks, Mobi is distinguished in its use of a single structured interface through which the crowd is exposed to the current solution and the global problem-solving context. This unified view provides a shared context that allows contributors to coordinate and communicate more effectively with one another than approaches where participants are forced to work on different subtasks in separate contexts.

Interactions are less controlled, but still structured

Mobi allows workers to choose how they want to contribute to the task. In our studies, we found that workers generate a diverse set of ideas, and make various types of contributions while problem solving. This freedom is particularly important for resolving global constraints as we do not know *a priori* the specific contributions that are needed. Rather, contributions are context dependent. While interactions are less controlled this way, note that the interaction is still highly structured; the crowd selects from a well-specified set of actions, todo items guide the crowd towards useful actions, and real-time feedback is provided on the effects of actions (e.g., via map, trip times, and todo items).

A language for human-computer communication

In the background, Mobi’s automation computes routes and times, checks for violated constraints, and generates todo items. Mobi understands, for example, when all of the quantitative constraints are satisfied; this ability enables Mobi to take such actions as prompting the crowd for future revisions and asking the crowd or requester to check for potential problems. Mobi can do these things without knowing what the constraints mean, because the inputs that it seeks from the crowd include the categories of suggested activities. This information is sufficient for the system to check for violated constraints, and therefore assist in the planning process.

A fluid way to refine goals

With complex problems, requirements can change over time as ideas and partial solutions stream in. In Mobi, a requester can add or revise requirements, write notes, or even directly alter the plan throughout the planning process. The crowd can react to such changes just as they react to the current solution

at any other point in the planning process. The iterative nature of the task and the ease with which workers can grasp the current solution and access alternative suggestions make it easy for the crowd to see and respond to such refinements.

CONCLUSION

To date, many human computation systems have relied on the assumption that problems can be solved in an algorithmic manner, using explicit procedures that outline the operations that need to be done and how they are ordered. We argue for an alternative *crowdware* paradigm, where workers contribute to solving complex tasks in a less controlled environment, allowing them to view and build upon each other's ideas and to contribute as they wish, while being steered towards a solution by system-generated advice and alerts.

Using itinerary planning as a case study, we introduce a prototype named *Mobi*, which draws on groupware ideas and uses explicit processes (e.g., automatic generation of todo items) to generate itineraries that satisfy complex, interdependent constraints. Our results show that constraints are resolved efficiently using this design, and that end users find that the generated itineraries satisfy their stated quantitative and qualitative constraints. The design principles explored in research on *Mobi* bring into focus several research opportunities, including the formulation of novel combinations of crowdware and workflow approaches to enhance the ability of participants to effectively contribute to solving complex problems that are hard to decompose.

On potential extensions of *Mobi*, we are interested in studying ways to handle the implied constraints that are assumed or missed, such as the common sense knowledge that people may desire a bathroom break every few hours. The challenge is to make implied constraints visible so they can be tackled like other constraints; possible approaches include having the crowd identify them, using automated procedures to detect and learn about such constraints, and asking requesters to provide feedback. A related direction is to encapsulate *qualitative* constraints in todo items, which would allow workers to see everything that needs work in one place.

While we are using Mechanical Turk as an experimentation platform, we are interested in exploring *Mobi* as a social system, with friends and community members coming to help. This introduces opportunities for identifying individuals who can best contribute to (particular aspects of) a mission and routing tasks appropriately. Finally, we envision rich opportunities to integrate different types of automation into *Mobi*—to detect failures, handle uncertainties, incorporate richer forms of user preferences, and combine automated and human planners in a synergistic way.

ACKNOWLEDGMENTS

Mobi was initially formulated and prototyped at Microsoft Research during summer internships there by Haoqi Zhang and Edith Law, who thank Anne Loomis Thompson and Paul Koch for technical assistance on an earlier prototype. Haoqi Zhang and Edith Law are generously funded by a NSF Graduate Research Fellowship and a Microsoft Graduate Research Fellowship respectively.

REFERENCES

1. Bernstein, M. S., Little, G., Miller, R. C., Hartmann, B., Ackerman, M. S., Karger, D. R., Crowell, D., and Panovich, K. Soylent: a word processor with a crowd inside. In *UIST* (2010), 313–322.
2. Chen, L., and Pu, P. Survey of preference elicitation methods. Tech. rep., Swiss Federal Institute Of Technology In Lausanne (EPFL), 2004.
3. Ellis, C. A., Gibbs, S. J., and Rein, G. Groupware: some issues and experiences. *Communications of the ACM* 34, 1 (1991), 38–58.
4. Hayes-Roth, B., and Hayes-Roth, F. A cognitive model of planning. *Cognitive Science* 3 (1979), 275–310.
5. Kamar, E., Horvitz, E., and Meek, C. Mobile opportunistic commerce: Mechanisms, architecture, and application. In *AAMAS* (2008).
6. Kittur, A. Crowdsourcing, collaboration and creativity. *XRDS* 17 (December 2010), 22–26.
7. Kittur, A., Smus, B., Kraut, R., and Khamkar, S. Crowdforge: Crowdsourcing complex work. In *UIST* (2011).
8. Kulkarni, A. P., Can, M., and Hartmann, B. Turkomatic: automatic recursive task and workflow design for mechanical turk. In *CSCW* (2011).
9. Law, E., and von Ahn, L. *Human Computation*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan and Claypool, June 2011.
10. Law, E., and Zhang, H. Towards large-scale collaborative planning: Answering high-level search queries using human computation. In *AAAI* (2011).
11. Little, G., Chilton, L. B., Goldman, M., and Miller, R. C. Turkit: human computation algorithms on mechanical turk. In *UIST* (2010), 57–66.
12. Nau, D., Ghallab, M., and Traverso, P. *Automated Planning: Theory & Practice*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2004.
13. Noronha, J., Hysen, E., Zhang, H., and Gajos, K. Z. Platemate: Crowdsourcing nutrition analysis from food photographs. In *UIST* (2011).
14. Sacerdoti, E. D. *A structure for plans and behavior*. Elsevier North-Holland, 1977.
15. Stefik, M., Bobrow, D. G., Foster, G., Lanning, S., and Tatar, D. Wysiwiw revised: early experiences with multiuser interfaces. *ACM Trans. Inf. Syst.* 5 (April 1987), 147–167.
16. von Ahn, L. *Human Computation*. PhD thesis, Carnegie Mellon University, 2005.
17. Wikipedia:template messages/cleanup, 2011. Available at http://en.wikipedia.org/wiki/Wikipedia:Template_messages/Cleanup. Accessed September 22, 2011.