Utility of Human-Computer Interactions: Toward a Science of Preference Measurement

Michael Toomim[1], Travis Kriplean[1], Claus Pörtner[2] and James A. Landay[1]
University of Washington, dub Group
Computer Science & Engineering[1], Economics[2]
{toomim,travis,landay}@cs.washington.edu[1], cportner@u.washington.edu[2]

ABSTRACT
The success of a computer system depends upon a user *choosing* it, but the field of Human-Computer Interaction has little ability to predict this user choice. We present a new method that measures user choice, and quantifies it as a measure of *utility*. Our method has two core features. First, it introduces an economic definition of utility, one that we can operationalize through economic experiments. Second, we employ a novel method of crowdsourcing that enables the collection of thousands of economic judgments from real users.

ACM Classification: H5.m. Information interfaces and presentation: User Interfaces.

General Terms: Design, Economics, Experimentation

INTRODUCTION

The CHI research community grew from the need to understand *discretionary use* of computer interfaces [5]. Discretionary use—the free choices that people make about which interfaces to use to accomplish their tasks—has always been central to our field. And today, people have increasingly more choices in computing.

Unfortunately, the measures used by the CHI research community—time-on-task, the number of errors, and subjective interpretations of think-aloud and survey reports—only indirectly predict whether an interface will be preferred over other alternatives. We usually do not directly measure user *choice* itself. This means that our formative and summative research methods are somewhat misaligned with our values of supporting discretionary use. This limits the CHI community's ability to effect meaningful change in the world.

Industry, on the other hand, has developed a valuable technique to measure user choice: A/B testing [15]. Unfortunately, A/B testing is out of reach for most researchers and small developers because it requires a large up-front investment and an existing userbase to deploy.

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 2011, May 7–12, 2011, Vancouver, BC, Canada.
Copyright 2011 ACM 978-1-4503-0267-8/11/05....$10.00.

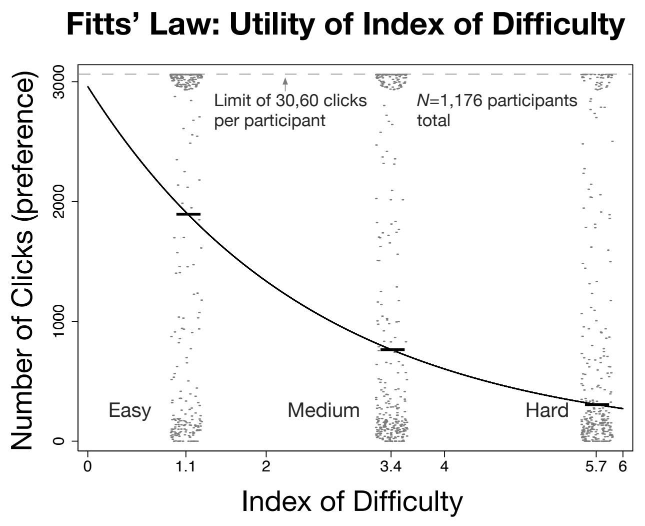


Figure 1. Fitts' law models the *time* required to click a widget of a size and width—our technique can model how much people *prefer to use* a widget. Participants were assigned one of three index of difficulty conditions. Each point is the number of clicks a participant completed before quitting (points jittered to show spread). Participants *preferred* big buttons to small buttons ($p < 0.10$). Participants were allowed a maximum of 3,060 clicks each. The regression line accounts for this maximum using a Tobit analysis.

Our goal is to help researchers and practitioners measure choice itself. We want to make the evaluation of choice and preference accessible and inexpensive. Grudin [6] and Malone [17] demonstrated early on that preference measurements can yield useful design insights. Few others have extended this work ([2]). We want to take this concept to the next level. We propose (1) a science—language, methods, and tools—for discovering, describing, and sharing knowledge about user preference and (2) evaluation techniques that practitioners can use to guide and justify an evidence-driven design process.

In this paper, we take some first steps towards establishing the language, methods, and analytical tools for evaluating choice and preference of different tasks and interfaces. The core technique we introduce is a semi-automated method for posting different interfaces and tasks to a crowdsourced labor market, such as Amazon's Mechanical Turk. These labor markets are websites where anyone can post a small task for someone else to accomplish for a small price. Our method is to create thousands of such tasks that systematically vary the interface, the price, and instructions. We then observe how many workers choose to complete the tasks, and how many times they do so. With this data, we can apply various analytical techniques to characterize user preferences for the given interfaces and tasks.

We can gain a quantitative handle on user preferences by analyzing the dynamics of the discretionary completion of tasks. Consider the classic Fitts' Law experiment [16]. Fitts' Law models the difficulty of target acquisition as a function of target size and distance. To test our methodology, we ran a study online to determine how target size and distance affect *preference*. We posted the standard Fitts' law task to Mechanical Turk, asking workers to click back and forth between a rectangle that switched sides on the screen. Our experiment manipulated the task's index of difficulty, by changing the size of rectangle and distance from cursor. We expected users to *prefer* easy tasks to difficult tasks, and in fact the data displays this trend (see Figure 1). That is, the degree a Mechanical Turk user prefers a Fitts' law task is inversely proportional to the time it takes to use it. With further experiments, one could learn if this preference result generalizes, and perhaps produce a general law of user preference with respect to task-completion time. Such a law would be useful knowledge for designers.

To be more precise, we are proposing that our method enables researchers to measure differences in *net utility* of some classes of interfaces and tasks. HCI often defines usability and utility as separate quantities: usability is the ease and efficiency of the interface apart from its function, and utility is the "usefulness" of its function [18]. However, both values influence user choice, and in fact they can be difficult to separate in practice.

In this paper, we bring the concepts of usefulness and usability together under the *Economic* definition of utility. **In Economics, utility is the degree to which a person prefers a particular choice amongst options [23].** We can infer it from user behavior: when a user chooses to use system A instead of B, it is said that *Utility*(A) > *Utility*(B). Utility encompasses all factors of function and usability that affect preference and use, and measures their net impact on user behavior (Figure 2). Economic utility quantifies aggregate user preference.

To quantify the utility difference of two interfaces, we vary the wage we offer, and find the amount that compensates for a difference in use between the two interfaces. If a user has no preference between being paid 25¢ for using system A over being paid 50¢ to use system B, then we can describe the difference *Utility*(A)–*Utility*(B) = *Utility*(25¢). That is, system A provides a measure of 25¢ more utility than system B. This is a *money-metric of utility*, a number representing the value of an interface change that can be compared across interfaces and situations, providing a lingua franca for communicating results.

This paper contributes the foundational first steps toward a net utility measurement. We define utility, an economic

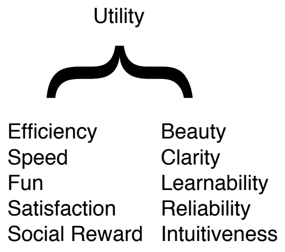


Figure 2. Utility is a summative descriptor of users' decisions of use. Many lower-level factors affect a user's choices of use. Utility aggregates the effects of all lower-level factors into a single number.

concept, in terms applicable to HCI. We show how to measure the difference in utility between two interfaces: by controlling for variables on Mechanical Turk, analyzing the data, and producing a money metric of utility. We present two utility measurement case studies conducted with our method that measure the utility of interface variations in (1) aesthetics & feedback and (2) efficiency. We believe that the further development of utility measurement methods will be able to help align the research community's focus on discretionary use with its toolset, and enable a more scientific understanding of the degree to which our interface improvements are *valued* and motivate *use*.

Using Mechanical Turk facilitates these first steps, but also has acute limitations. First, it is difficult to generalize results from Mechanical Turk to other contexts. We discuss this issue in more depth later. Second, a worker on Mechanical Turk does not complete a task for the same reason a normal user would. For instance, a father gets a particular utility sharing photos with his daughter on Facebook that is lost when we ask a Mechanical Turk worker to complete the same task. Thus, we cannot measure the value of a goal; what we *can* measure is the "cost" of an interface for completing the task, and in particular the difference in utility between two interfaces. Finally, we have yet to calibrate or validate the quantities of our measures, such as the money-metric. In this paper, we only show that their values follow our intuitions on what they should look like.

The rest of the paper is organized as follows. First, we delve deeper into the history of HCI to further motivate the importance of establishing a science for measuring and applying knowledge of interface utility. Second, we give more detail on our method for measuring differences between interfaces and how it can be achieved on Mechanical Turk. This involves (1) defining in greater detail net utility and the money-metric, (2) identifying strategies for running an interface auction and its implication for study design, and (3) detailing factors that can be controlled for on Mechanical turk in order to obtain a more accurate measurement. Third, we describe our two case studies, developing a repertoire of techniques for analyzing data as we present the cases. We conclude with a discussion of the scope of our technique's abilities and limitations as compared to other techniques, and how to develop these techniques further in the future.

HCI IN HISTORY: FROM EFFICIENCY TO UTILITY

Users have not always been able to choose their interfaces, as Grudin describes in his histories of HCI [5]. Most interfaces through the 1970s were designed for paid operators of

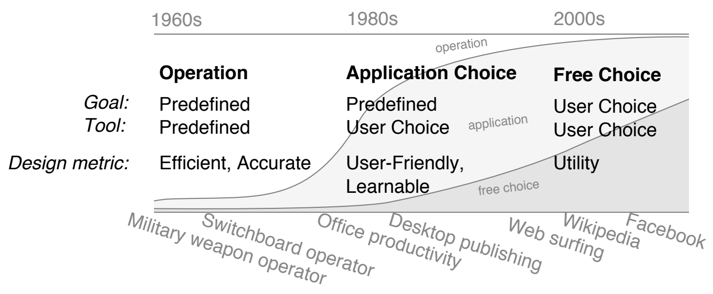


Figure 3. Three phases of choice in HCI. (1) The first user interfaces were designed for fixed tasks, in fixed industrial processes. Users had little choice in their use. (2) In the 1980s, we developed commodity software that users could choose to apply (or not) to tasks of office work. (3) Today's Internet, mobile, entertainment, and persuasive software provides users with entirely new tasks to choose from, not just tools. These interactions must have high utility to motivate use.

large, expensive machinery, such as pilots in an airplane cockpit or nuclear power plant operators in a control room. They obtained mandatory, paid training to learn an interface. A successful system design needed to be performant, efficient, reliable, and little else. It was not until the 1980s that *discretionary use* became common: a technical writer that needed to type a document could choose whether to learn a document typesetting system or use a simpler typewriter. For the first time, user interfaces needed to be easy to learn, or "user-friendly," or users would reject them in favor of alternatives.

The trend has continued, and in fact has sped up rapidly (Figure 3). The Internet makes it easier for users to find and try new software to apply to their tasks. There are more options available to choose amongst, and the ubiquity of computers has given us the option to apply computing to more parts of our lives. Moreover, a new class of computing involves choosing not only a tool to apply to a task, but often the *task itself*. For instance, web surfers constantly choose new interesting pages to surf. Users today routinely choose optional and voluntary tasks, such as maintaining blog entries, sending and accepting friend requests, creating LOLcats and editing Wikipedia articles—tasks which did not even exist two decades ago. And some software is designed specifically to give new goals to users: video gamers *pay money* to find compelling tasks to spend their time on; consumers purchase persuasive products, such as Nike+ iPod, hoping to influence their choices in life; Human Computation and Crowdsourcing applications find ways to wrap tasks like image labeling [22] and protein folding [3] within fun, high-utility games. Simply put, there is a growing, market of interfaces being chosen by (and created) to accomplish a myriad of tasks, each vying with competitors for the real scarcity of the information age: attention [20].

In acknowledging the importance of user choice, industry is increasingly prioritizing metrics of *use* in design. Industry wants their systems to be used, to be desired, popular, to get page views, sales, have millions of active users, reach critical mass, and achieve network effects. It does not make sense for Facebook to optimize the time-on-task for the task of browsing a friend's News Feed. Indeed, when a designer makes an interface fun or valuable, time-on-task should *increase*. Rather, Facebook evaluates features by deploying them and seeing if they obtain and sustain *use*. Amazon, Google, and Yahoo test all important changes with A/B testing, which tells them how their interfaces will affect real user *choices*. Websites measure success in hits, page views, ad clicks, user posts, and monthly active users: all measures of *use*, and the result of *utility*.

The nature of computer use has transformed with the rise of free choice computing, but HCI's methods and metrics have remained relatively static. Our community is not engaged with *choice* in computing. Without studying choice, preference, utility, and real-world use, the field of HCI will become increasingly irrelevant to practice.

METHOD FOR UTILITY MEASUREMENT

In this section, we give more detail on our method for measuring net utility and how it can be achieved on Mechanical Turk. We (1) define in greater detail net utility and the money-metric, (2) identify strategies for running an interface auction and its implication for study design, and (3) detail factors that can be controlled for on Mechanical turk in order to obtain a more accurate utility measurement.

Our approach to measuring the utility of an interface is to determine the *compensating wage differential* between different tasks and computer interfaces. This economic theory dates back to Adam Smith, who defined it as the additional amount a worker must by paid to convince him to do a job that is unpleasant, risky, or otherwise undesirable [21]. Our method determines the desirability or undesirability of an interface to achieve a task by observing how many workers choose to use it at different amounts of pay.

As is standard in contemporary Economics, we generally define utility as an *ordinal* measure: it describes an ordering of preference, but not the strength of preferences. This assumes very little about a user's preferences, only that he has them, and that they are not *e.g.* circular, preferring A to B, B to C, and C to A. (A complete list of assumptions can be found in [23]).

We model user choice as follows. A user finds value in completing a *task*, but to do so he must take actions on a computer *interface*, a process with its own cost or value. These values also depend upon the user's demographic, psychological, social, moral and physical *context*: the variables that determine who he is, what he knows, what he wants, and what situation he is in. All together, we say that utility is a function of three parameters:

$$Utility = f(task, interface, context)$$

Utility experiments can vary any one of these parameters to measure its effect on user choice. For instance, the *context* could theoretically be manipulated by switching to a different labor market, by augmenting or manipulating the labor market with particular community or social dynamics, by surveying workers and controlling for their demographics, or by introducing labor market mechanisms into existing user populations. Consider the task of modifying a wiki page. The value of editing it will change depending on whether that wiki page is part of Wikipedia, Conservapedia, or your research group's homepage; editing each has different meaning to you, has a different community that maintains it, has different visibility through PageRank, etc. Similarly, we might hold the interface and context constant, *e.g.* editing the HCI page on Wikipedia, while the *task* is manipulated (adding a category tag vs. reverting vandalism vs. correcting a spelling error). In this paper, we primarily show how to vary and study the *interface*, which is the easiest of the three variables to study on Mechanical Turk.

Auction Techniques

Our preference measurement technique begins with determining how much you must pay people to convince them to use an interface for a task. This is essentially an auction.

There are many auction techniques one might use. Horton and Chilton [8] measured a worker's *reservation wage*—the wage below which she will not do a task—by starting with a high wage and incrementally reducing it until the worker quit. This tested a rational model of worker behavior, elucidating many interesting aspects of Mechanical Turk as a labor market. Their style of auction could theoretically be used for interface preference measurement, but it is unfortunately biased by strong observer effects: the sequence of declining prices workers are paid has a significant effect on when they decide to stop, and thus changes the "reservation wage" the method calculates. Horton and Chilton's initial experiments found a significant effect of this pricing style on the results, but did not find a significant effect of the Fitts' law index of difficulty condition that they studied—something our technique detects.

Ben-Bassat *et al.* [2] used a traditional second-price auction to determine user preference for interfaces, implemented in a traditional face-to-face laboratory study. Participants provided an explicit money bid for each interface. However, asking a person to explicitly reason about an interface's price inherently alters their process of choice. For instance, the subjects in their study bid extra for *efficient* interfaces, but not for *highly-aesthetic* interfaces. As we will show, our technique detects strong effects for aesthetics. We hypothesize that even if subjects preferentially *choose* pretty interfaces in practice, they may not reason that this is a valid quantity to pay for when asked to reason about it, and bid only for efficiency.

These two auction techniques assume a significant degree of *rationality* of the participants. Our technique's key difference is that it does not ask workers to choose a price. We simply present a worker with a job at a price and observe their behavior. A user chooses to complete a task at a given price, or not. If they choose to complete the task, we observe how much work they do before boredom or other factors reduce the net utility of the task below their other options, and they stop. We then aggregate this data. Holding task and context constant, a data point in utility space is (interface_id, worker_id, wage_compensation_per_completion, number_of_completions).

Our approach is a between-subjects design, and minimizes the explicitness of workers reasoning about their choices. We call jobs "Mystery Tasks," presenting them as a surprise or a game rather than an explicit auction (detailed later). Workers do not know that their activities are being aggregated to infer net utility. This technique is simple, direct, and requires few assumptions. The downside is that it requires a large amount of data, because every completed job provides only one bit of information: whether the user accepted the job, or not. Luckily, obtaining this amount of data is feasible with Mechanical Turk.

MAKING IT WORK IN MECHANICAL TURK

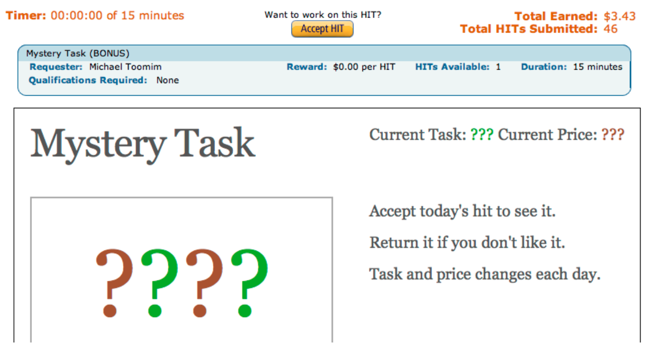
To make our auction work, we had to overcome a number of challenges on Mechanical Turk that would bias the results and compromise their integrity as a measure of net utility. In this section, we first overview the software framework implementing the utility methodology, then describe a couple of the specific problems it helps solve, and finally detail a few outstanding problems that need to be addressed in future work.

Overview of Software Framework

We have automated most of the experimental method. An experimenter first implements a user interface for a task using HTML, JavaScript, Flash, Java, or any other technology embeddable into webpages. She can define a set of experimental *conditions* (a pairing of interface and task), and parameterize the interface for each condition, for instance implementing multiple button widths or interaction styles that she would like to test. She then tells our software to run a study with the interface and task, declaring which conditions she would like to run, and how many workers she would like to receive utility judgements from (this determines how much she will spend on Mechanical Turk). Our software automatically creates hundreds to thousands of jobs on Mechanical Turk and randomly assigns workers to the prescribed conditions. Workers complete jobs interacting directly with the experimenter's web application embedded in an IFrame on the Mechanical Turk website. Our software logs to a database how many workers look at the task, and how many jobs each completes, along with the workers' interactions, worker IDs, geographic locations, and their randomly-assigned pay and interface conditions. The software then applies micro-econometric analyses to the data, computes a money-metric, and produces graphs to explain the results. It also geolocates IP addresses and tracks local time of day for workers automatically, which make it possible to observe differences in utility across regions and time of day. We implement our system in the fantastic web2py web programming framework.

Specific Problems Addressed by Framework

The framework addresses a number of issues that anyone who wants to use our method for running utility auctions on Mechanical Turk need to address. We describe four such problems and solutions here.


Timer: 00:00:00 of 15 minutes

Want to work on this HIT? Accept HIT

Total Earned: $3.43
Total HITs Submitted: 46

Mystery Task (BONUS)
Requester: Michael Toomim
Reward: $0.00 per HIT
HITs Available: 1
Duration: 15 minutes
Qualifications Required: None

Mystery Task

Current Task: ??? Current Price: ???

????

Accept today's hit to see it.

Return it if you don't like it.

Task and price changes each day.

Figure 4. Until a user accepts a Mystery Task, they do not know the type of task or how much it pays. This allows us to log the percentage of users who *see* the task vs. *accept* it, and dispatch a different experimental condition and price to each worker.

Implementing multiple conditions.
We want to measure the varying work done per condition, so we need to post multiple conditions of jobs. However, if we post them all at the same time, workers will only consider completing the best, highest-priced job. If, on the other hand, we change the price over time, we have to control for the other factors that change over time, such as worker populations and market conditions, and have to control for the increases in experience and boredom that occur with workers who happen to remain on Mechanical Turk during multiple phases of the conditions. Our solution is to post all conditions to Turk simultaneously, but allow each worker to see only one of the many conditions. Mechanical Turk itself does not support this, but we were able to implement a novel workaround using Mechanical Turk's *bonus* and *IFrame* APIs. We will describe this below.

Selection bias.
The second problem is that the rate of work depends on how many workers *find* a job in the first place. We want to measure a job's inherent labor completion rate, but other factors confound the rate. For instance, we were able to get a 12x increase in work rate by modifying a job so that it appeared on the Mechanical Turk front page for workers who do not change Amazon's default search settings. Our solution is to measure the number of people who look at our jobs, separately from the number who complete them, so that we can measure instead the *proportion* of those that complete tasks out of those who consider tasks. Unfortunately, Mechanical Turk does not tell us how many people look at our task listings, since they appear in a large list of search results on the Mechanical Turk server.

Luckily, we can solve the multiple conditions and selection bias problems with one technique. Our solution is to post each job as a "Mystery Task," with a listed pay of $0.00, and a description that tells the worker she must preview the job to see the task and how much it actually pays (Figure 4). When the worker opens a Mystery Task, Mechanical Turk opens an IFrame (using the externalQuestion API) to our web server, which tracks that the worker has *seen* our task. Our framework is then able to initialize an appropriate interface, task, and price that instantiates an experimental condition. If the framework has not yet seen the worker's ID, the price is randomly assigned, and sticks with them for subsequent jobs. Our jobs pay entirely in *bonus*, a Mechanical Turk feature that allows employers to pay a worker a discretionary amount beyond the initial payment. The randomization of condition and pay is hidden from the worker, who is not told that there are multiple conditions available behind the scenes. Unfortunately, Mechanical Turk does not provide external webservers with the worker's ID when she previews the job, only when she accepts it, so we display the page in Figure 4 for a previewing worker until she accepts the job. This set of techniques removes all condition-specific information from the job description listings, which we cannot control on a per-user basis, and gives us control over every step in a participant's process of choice.

Lopping off the long tail of job completions.
Some small proportion of workers will continue to complete the same job *ad infinitum.* However, this does not necessarily yield more interesting data for informing utility analysis (see e.g. later section on survival analysis). Our framework allows the experimenter to set a parameter for the maximum number of jobs (*e.g.,* 50) any single worker can complete. This reduces the cost of the study if the experimenter is not interested in the utility of doing a job more than a certain number of times in a row. It also makes it less worthwhile for a worker to try to cheat by programming a script to parse the webpages and perform tasks automatically.

Temporal Market Price Fluctuations.
The market clearing price on Mechanical Turk can change over time because of an increase in jobs or workers. Labor supply is sensitive to short term fluctuations like time of day and day of week, as well as longer term boom and bust of the Turk economy. This can create problems for analyzing the results of utility studies conducted over a longer period of time, as well as problems comparing studies conducted at different times. To control for this, we can post a baseline control condition (answering CAPTCHAs for 1¢) along with the experimental conditions. Note that the data we present in this paper has not been controlled for in this way.

CASE STUDIES
We applied our method in two case studies, to verify that it detects significant differences in utility that we would expect. The first study is the Fitts' Law study, which we summarized in the introduction. Its purpose is to measure the utility of *time-on-task*, a factor that HCI routinely measures. The second study, on the other hand, measures the utility of two user interface factors that HCI has had a difficult time quantifying because they do not affect efficiency: aesthetics and feedback. As we describe the studies, we will also present a repertoire of techniques that are useful for analyzing and communicating about the data produced by utility measurement.

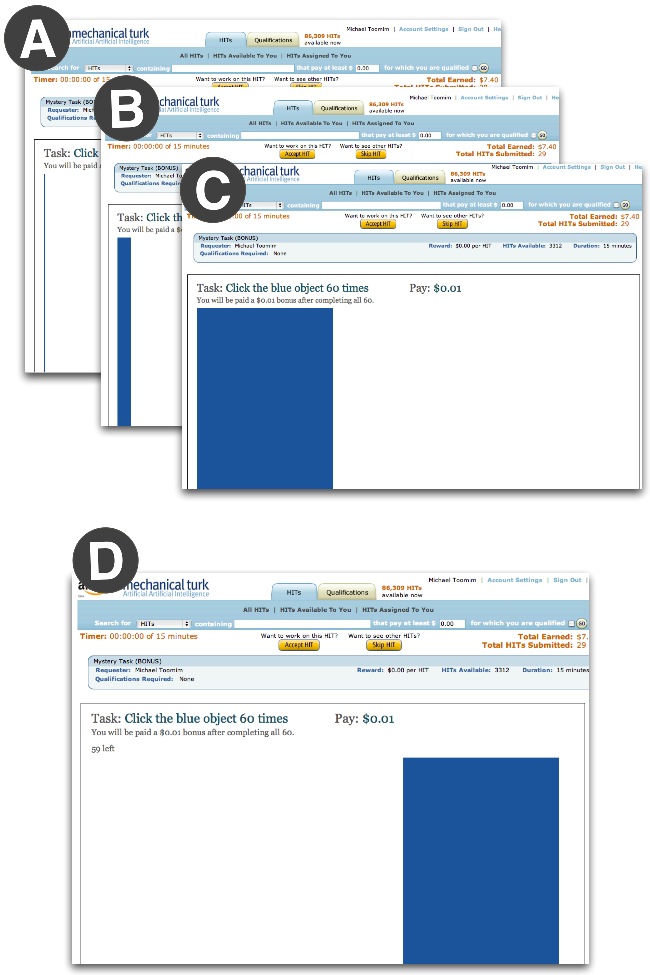

A
B
C
D
mechanical turk
Artificial Artificial Intelligence
HITs
Qualifications
86,309 HITs available now
Michael Toomim | Account Settings | Sign Out
All HITs | HITs Available To You | HITs Assigned To You
Search for HITs containing that pay at least $ 0.00 for which you are qualified GO
Timer: 00:00:00 of 15 minutes
Want to work on this HIT? Accept HIT
Want to see other HITs? Skip HIT
Total Earned: $7.40
Total HITs Submitted: 29
Mystery Task (BONUS)
Requester: Michael Toomim
Reward: $0.00 per HIT
HITs Available: 3312
Duration: 15 minutes
Qualifications Required: None
Task: Click the blue object 60 times
Pay: $0.01
You will be paid a $0.01 bonus after completing all 60.
59 left

Figure 5. Screenshots of the Fitts' law task. Subjects clicked on a blue rectangle 60 times. We created three variations of bar width and the distance it moved: hard (a), medium (b), and easy (c). Each time they clicked on the bar, it moved to the opposite side of the screen (d).

These case studies demonstrate that utility measurement can apply to a range of factors—both those that we measure today, and those that we do not—and that it replicates our existing design knowledge and achieves statistically significant results.

Utility of Efficiency: Testing Fitts' Law

To get a baseline understanding of the relationship between utility and HCI's existing metrics, we studied the utility of *efficiency* in a Fitts' law task.

Study design and execution.

We implemented a traditional Fitts' law task in JavaScript, where the user must click back and forth between rectangles on the left and right sides of the screen (see Figure 5). We parameterized the task with three conditions of bar width and distance, or *indices of difficulty*. The conditions as `[width, distance]` were `[300px, 700px]` for the easy task, `[30px, 870px]` for the medium, and `[3px, 897px]` for the hard. We posted 22,190 jobs to Mechanical

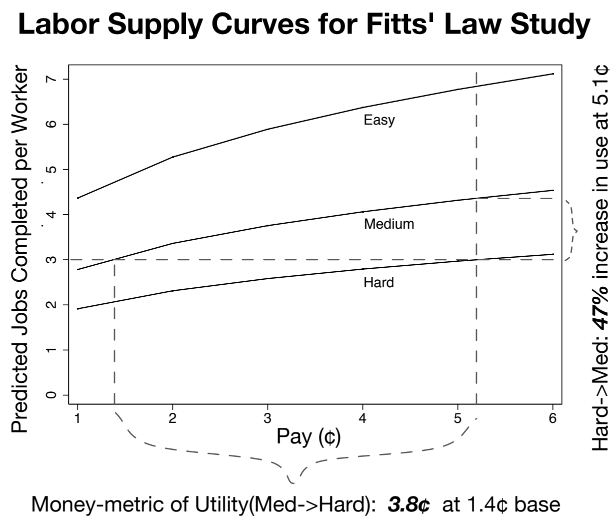


Figure 6. We paid six different prices (1-6¢) for each of the three experimental conditions: a total of eighteen conditions. By regressing on the number of jobs completed in each condition, we estimate these labor supply curves. Holding pay constant, we can quantify the effect of an interface on *use*. Holding the number of jobs constant, we can compute a *money-metric*: the amount of pay required to obtain the same amount of work between two interfaces. For 3 jobs, the utility of Medium over Hard is equivalent to 3.8¢ per 60-click job. (This data is filtered to U.S. workers.)

Turk, recruiting 1,176 distinct workers, at six prices: 1¢ through 6¢. Our software automatically crossed the six prices and three experimental conditions to create eighteen conditions total. Each job required 60 clicks to complete, giving us time-to-click data on 1,331,400 clicks. We set an upper limit of 51 jobs, or 3,060 clicks, per worker. The study took 5 hours 15 minutes to complete, and cost $970.

Analyzing the data.

Now that we have a database containing the choices of hundreds of workers completing the Fitts' law task at different prices, we would like to answer a set of questions. How much use occurs in the different conditions? How much money is this difference in use worth to users? How does utility vary over time; for instance, what is the difference between novice utility and expert utility? On the converse, how quickly do users bore of the task? And how does use vary with user context, such as a user's geographic location or local time of day? Here we define two analytical techniques that can help make sense of choice data.

Creating a Labor Supply Curve. The labor supply curve predicts the amount of use that a particular interface and degree of incentive (pay) will produce. It is a plot of the number of jobs completed at each price for each condition. We estimate the labor supply curve using a Tobit regression model [4]. The Tobit model takes into account the upper limit of 51 jobs per worker, known as "censored" data.

Figure 6 shows the predicted labor supply curves for the Fitts' Law study. In this example, we filtered to workers

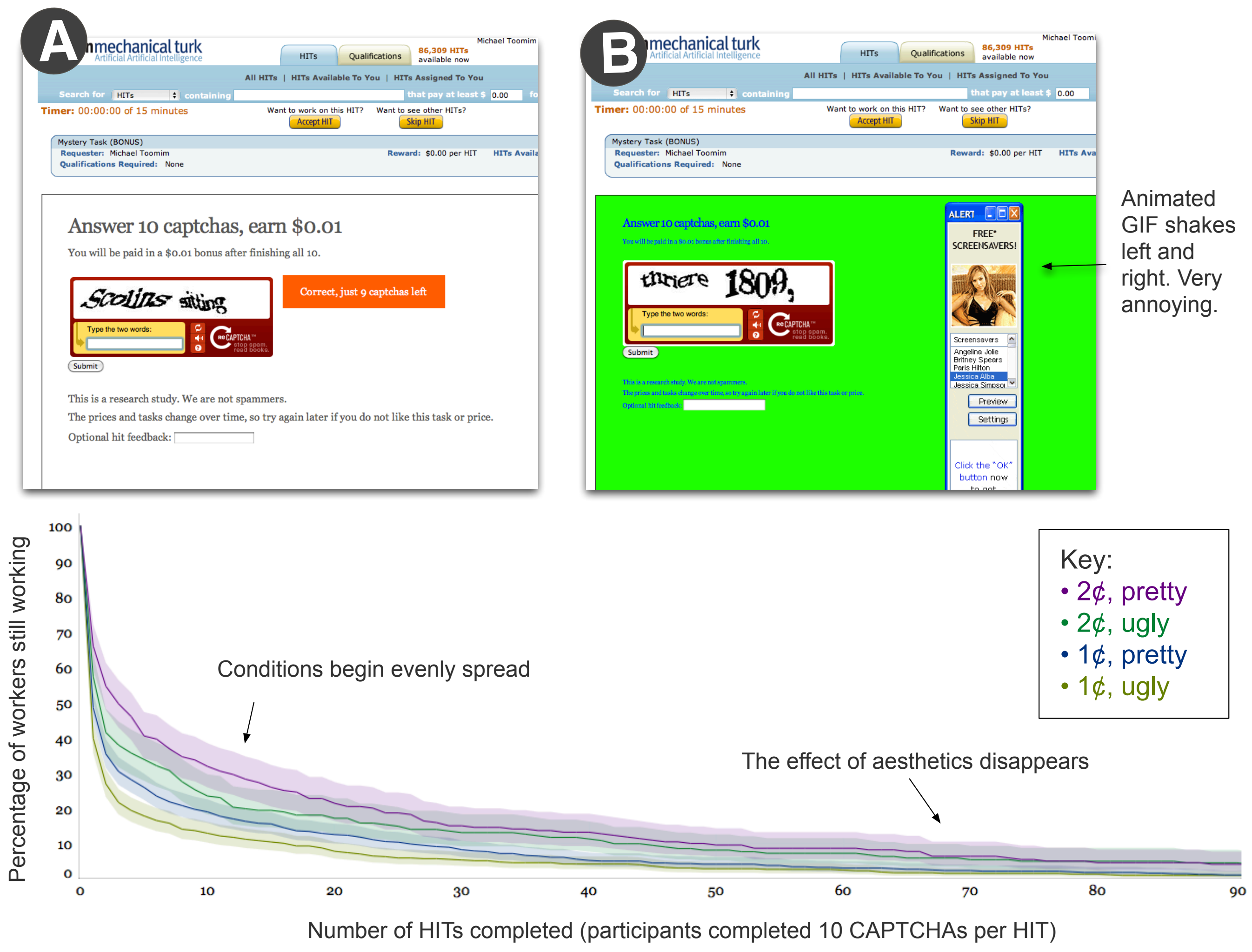


Figure 7. *Survival graph for the Aesthetics & Feedback study.* We made two interfaces for answering CAPTCHAs: one "pretty" (a), one "ugly" (b), but identical in behavior. The *survival graph* shows how many workers made it through how many tasks, for each of our four experimental conditions. The shaded regions are 95% confidence intervals. At the far left, 100% of these workers looked at the task, but only 10% to 40% completed 10 tasks (100 CAPTCHAs). Note that the *pretty* and *ugly* lines are separated at the left, but converge toward the right. This suggests either that the utility effect of aesthetics fades over time, or that the types of users who complete many CAPTCHAs are more concerned with pay than aesthetics.

from the United States (31% of the total data). As expected, workers do more work if they are paid more money. Furthermore, the curve shows they prefer the conditions in the order easy > medium > hard. These differences are significant at $p = 0.03$ from easy to medium, and 0.07 from medium to hard. In Figure 1, we can see this preference ordering even clearer. There is an inverse relationship between task difficulty and utility for Fitts' law tasks. These results are qualitatively the same for Indian workers (44% of the data), but the data show they respond much more strongly to both pay and index of difficulty than Americans. Our data can also be used to quantitatively predict the amount of work, within error bounds, that will be produced given the variables of interface, context, and incentive (pay).

Computing a Money Metric: the Compensating Wage Differential. We can also infer the amount of money an interface variation is worth: the money metric of utility. To derive the money metric, we measure the horizontal distance between two curves—the change in pay that makes them equivalent. Note that this *compensating wage differential* will generally increase with the base pay rate, as the curves spread apart. We demonstrate calculating the money metric graphically in Figure 6. With a base pay of 1.4¢ (producing 3 jobs, or 180 clicks per worker for the medium difficulty condition), we would need to pay 3.8¢ more to achieve the equivalent work rate in the hard condition.

In conclusion, our efficiency case study showed that utility can capture an existing metric: *efficiency*. Users prefer clicking efficiently-designed targets. Labor supply curves predict the amount of use an interface will get. By calculating a money-metric, we can quantify the magnitude of the utility differences across conditions, and reason and hypothesize about them in terms of the *lingua franca* of utility: dollars and cents.

Utility of Aesthetics & Feedback: CAPTCHAs

This second study demonstrates that we can measure the utility of two particularly elusive quantities in HCI: *aesthetics* and *feedback*. These quantities are elusive because they do not make an interface slower to use, or otherwise affect the user's actual behavior. They only affect his perception of the interface and his understanding of its internal process.

Study design and execution.

We experimented with two interface variations for the task of answering CAPTCHAs. One interface had a clear, minimalist design, and the other had gaudy colors, small fonts, and a distracting animated GIF advertisement (Figure 7A & B). Both tasks had the same instructions and wording, required 10 CAPTCHAs to be completed per job, and took the same amount of time to complete. The *pretty* condition implemented an elegant animated countdown reminding the user how many CAPTCHAs they had left, and the *ugly* condition only told them when they had completed all 10. We posted 15,000 jobs to Mechanical Turk, with one 10-CAPTCHA task per job. Workers were paid either 1 or 2 cents, for a total of 4 conditions. 1,270 workers completed our jobs, and the entire study cost us $388. In this study we did not limit the number of jobs a worker could complete.

Survival Analysis.

Our between-subjects auction method collects a binary choice for a user, over time until he quits. One intuitive way to represent this data is with a *survival function* [14]: a function $S(t)$ that represents the probability of a user "surviving" t tasks before quitting. Analysis starts by preprocessing the data to identify how many tasks each worker completed. Then we group data by condition and price, and plot a graph of the percentage of users who continued to use the interface after N jobs. If a line is higher in the survival graph, it means more workers completed more tasks. We compute 95% confidence intervals using Wilson's estimate [24], since survival data is binomial. When the study ends, it artificially stops, or *censors* the work of some users who might otherwise have completed more tasks. We label those users as *censored* and account for them statistically using standard survival analysis techniques.

The survival graph for the CAPTCHA experiment is shown in Figure 7. The confidence intervals for each line are shaded. The survival analysis shows how use changes over time. We can see that all four conditions are spaced apart roughly equivalent for the first 20 tasks, but for work done at 80 tasks, the top two lines (2¢) and bottom two lines (1¢) converge. This means that price dominates the utility for workers who acquire more experience with the task, and aesthetics is primarily important for those who are inexperienced. Or, those who stick with the task are more resilient to aesthetic quality.

We also estimated the effect of aesthetics on labor supply, as we did with the Fitts' law study. The results show that the effect of aesthetics and feedback is substantial: all else equal, the *pretty* style of the interface produces 58% more use. This is statistically significant at $p = 0.02$.

SCOPE, LIMITATIONS, & FUTURE WORK

We view our technique as just one point in the space of *preference measurement*, along with A/B testing and techniques that have not yet been invented. Our technique has two key features. (1) We measure behavior for controlled tasks and interfaces in a labor market, instead of experimenting with real users on a production website like A/B testing. (2) We vary pay, artificially manipulating a user's motivation, and from it infer a money metric of utility.

These unique aspects separate it from A/B testing. Interfaces to be A/B tested must be developed and tested to production quality and deployed to a real userbase. This constrains applicability in most companies, and indeed A/B testing is often limited to optimizing small variations in an existing UI—*i.e.*, for late-stage designs. Our techniques can be used to obtain quick feedback for early-stage design decisions. Furthermore, A/B testing is adept at answering "what" users will do for a particular website, but not "why." [15] By paying users, we have more discretion to ask them survey questions to elicit "why," and we also gain the ability to test abstract interfaces and tasks—such as our Fitts' law interface—to isolate factors and remove confounds of real-world websites. Future researchers can replicate and extend the results, in the same labor market, by copying and re-running the study's source code. This lets us develop and validate scientific models that generalize across concrete website instances. And by calculating utility in terms of money, we create a language to compare interfaces, tasks, and contexts and create generalizable knowledge. In summary, our techniques fill a niche in early-stage and generalizable interaction studies.

Mechanical Turk & Other Labor Markets

Although it is common to expect that Mechanical Turk workers are somehow "different" from normal computer users, the demographic data actually show workers to be remarkably representative of the Internet population. A substantial portion of workers have bachelors, masters, and PhD degrees, for instance [9,19]. Workers use Turk not just to make money, but also to have fun and spend free time, and avoid boring or distasteful tasks similarly to other Internet users [12]. We encourage the skeptical reader to investigate this data. In fact, Mechanical Turk's population is much more diverse and ecologically valid than the small-sample college populations commonly employed in HCI and Psychology research. Our Fitts' law study recruited workers from 32 countries in five hours. Moreover, the setting of use for Turk workers—*e.g.* at home, at work, at a cafe, watching TV, on one's own hardware—is often more naturalistic than a laboratory. Furthermore, we can survey workers for demographic or other information, and store it in a database with their worker-id. This allows us to examine the effect of *context* in a study, for a variety of personal characteristics, without additional experimental effort.

Yet at a higher level, our techniques are by no means limited to Mechanical Turk. In fact, there are more than ten alternative crowdsourced labor markets in current deployment [10], and we expect more to develop in the near future. Each market has different characteristics. Some markets even "pay" users with non-monetary incentives. For instance, the company CrowdFlower deploys micro-tasks through gaming company Zynga, which rewards game players with upgraded "cows" in the game FarmVille in exchange for doing small pieces of work.

Finally, we expect that our techniques could be used without a labor market at all, by finding other ways to recruit users. Facebook or Google could run utility experiments by recruiting their own users through advertisements, and paying them small amounts on PayPal. Researchers could run a custom ad campaign in this way, targeting a subpopulation. Going one step further, our economic methods could in theory be applied within A/B tests themselves, creating *utility-augmented A/B tests*. For instance, Amazon might offer randomly-selected users the opportunity to discuss a product within an experimental social system in exchange for a few cents of store credit, and thus combine many of the benefits of a traditional A/B test with many of the benefits of economic utility analysis. Indeed, it is important to remember that our existing research techniques are all biased and limited, but with time we have learned how and when to trust them. We believe the crowd enables the future of HCI evaluation.

However, when we assume we can replace a user's existing goals with money, we run into a number of potential hurdles. First, we cannot measure the value of their existing goals—only the cost (or value) inherent to the process of using the interface itself. This is a significant limitation, and difficult to get around in theory without data from actual use of the real system (*e.g.* a utility-augmented A/B test). Second, the researcher must be careful to avoid situations where the use of a money incentive adversely effects one's decision process, as has been recorded in Behavioral Economics [1]. Third, the researcher's freedom in defining the labor market worker's goal with an interface comes with the difficulty of *enforcing* it. To do so, the researcher might employ quality measurement, which we will describe next.

Quality measurement. Many computer tasks, such as writing articles, blog posts, and authoring presentations, are *open-ended.* The quality of results is difficult to verify with a computer. We have not yet studied such tasks, because we need to know which tasks were completed successfully so we can determine who to pay. The standard technique on Mechanical Turk is to post the result of tasks back to Mechanical Turk as new "reviewing" tasks, having workers review the work of other workers. This is the basis of a growing body of quality-assurance techniques used on Mechanical Turk [11]. We hope to implement this technique in our software framework. Moreover, this will enable us to study the relationship between quality and utility. High quality articles are often more difficult to write, and likely to cost more. But we can measure, for instance, whether writing on a topic of personal interest to the worker results in both higher quality *and* lower cost utility.

Cheating. Related to quality measurement is preventing cheaters and spammers on Mechanical Turk from abusing our experiments for money. Quality measurement will be critical for open-ended tasks, to prevent workers from submitting garbage results. It is also possible for someone to write a browser script to automate the submission of tasks without doing them himself. Our aesthetics task used CAPTCHAs to guard against automation, and our Fitts' law task recorded the time of each click, of which we ran simple data analyses to validate they looked human. Furthermore, we generally set a limit, such as 50 tasks, on the amount of work a worker can do. This way, even a script would make at most a dollar for its author, reducing the incentive to writing scripts.

Macroeconomic analysis. Analogous to how we are building a microeconomic analysis of user interfaces, we also imagine that future work might be able to build a *macroeconomic* analysis of entire systems, such as Wikipedia. This could attempt to answer questions such as "What is the net utility of the billions of Wikipedia edits?" analogous to "What is the GDP, or net economic output of the billions of products produced in the United States?" The path to studying this would involve first measuring the labor supply curves for users editing Wikipedia articles given different levels of pay, and exploring the different conditions—article subject, user background, education, and personality—that affect these curves. Then, by comparing these "editing costs" in to the actual amount of article editing taking place on Wikipedia, we could infer the net motivation that must be present on Wikipedia to produce the edits. We are interested in investigating where this general chain of reasoning—connecting studies relating the cost of using an interface to its use, and comparing this to actual use on a website—will allow utility studies to infer the intrinsic value of Internet user behaviors, and not just the costs of their actions.

Uniting game design and HCI. Game design and HCI often seem to pursue different metrics. HCI strives to make interactions usable, and game design strives to make them fun. HCI wants tasks to be fast, and game designers want users to spend more time on tasks. However, both fields strive for their interfaces to have high utility, and we may be able to bring the fields closer together, in common pursuit of a single metric. We note that utility is a combination of *value* and *cost*. A user often wants to achieve a goal that has some value, but in order to do so, must complete a process with a computer interface, a task with some *cost*. HCI has focused on reducing this cost, and game design has focused on increasing value. We are interested in learning how to design value into existing interfaces, rather than just reducing cost. For instance, we are currently running a utility study where we turned the traditional Fitts' law task into a video game by adding a scoreboard and graphics. By measuring the effect of these game conditions on utility, we can analyze the interaction between game mechanics and user interface costs within the same evaluative framework.

CONCLUSION

This paper presents a way to crowdsource the evaluation of utility, or the study of preference. We argue that utility measurement is necessary to make HCI relevant to free choice in computing.

In the future, we hope this class of methods will enable HCI to build a science of the human motive in computer interactions. By using the same labor marketplaces, scientists will have the equivalent of the same laboratories for interface utility studies. This would make their studies reproducible. A researcher could share her code with another researcher, who could then re-run the study and reproduce prior results, or alter the study and build on previous insights. Running studies on Mechanical Turk is cheaper and faster than traditional lab studies, as has been explored by Kittur *et al*. [13], Heer and Bostock [7], and a growing number of others.

We believe utility studies will also benefit design *practice*. Designers will be able to experiment with new interfaces and evaluate their use without building infrastructure and critical mass in a userbase. They could break up a complicated interaction into pieces, and estimate the motivation required of users for each subcomponent. This may allow practitioners to develop and test large social systems in modules, evaluating each part incrementally, before building critical mass, with greater certainty of success.

In this paper, we present just one possible method for preference measurement, using one particular labor market, and one set of economic analyses. However, preference measurement is a much larger space. In it, we can vary the user population (*e.g.* labor market, real website users), the incentive and goal structure (*e.g.* pay or no pay, real website user goals or artificial paid goals), and the experimental methods for determining value (*e.g.* auction techniques). We believe this space of investigation—understanding and designing for user preference and choice—is the future of HCI research and practice. By shifting our evaluation techniques to questions of *what people will choose to use*, we can better align our methods with our community's values.

ACKNOWLEDGEMENTS

We thank Google for a research grant that funded this work, and the anonymous reviewers for their excellent feedback. Hal Varian provided support and guidance. Lydia Chilton and Rob Miller generated early economic ideas. Jon Froehlich helped edit. Neema Moraveji and Jeffrey Heer were supportive kindred spirits. Brought to you by the letter G.

REFERENCES

1. Ariely, D. *Predictably Irrational: The Hidden Forces That Shape Our Decisions*. HarperCollins. 2008.
2. Ben-Bassat, T., Meyer, J., and Tractinsky, N. Economic and Subjective Measures of the Perceived Value of Aesthetics and Usability. *in ACM Transactions of Computer-Human Interaction*, 13 (2). 2006. 210–234.
3. Cooper, S., Khatib, F., Treuille, A., Barbero, J., Lee, J-H., Beenen, M., Leaver-Fay, A., Baker, D., Popović, Z. & Foldit players. Predicting Protein Structures with a Multi-player Online Game. *Nature* 466, 756-760.
4. Greene, W.H. *Econometric Analysis*. Macmillan. 1990.
5. Grudin, J. Utility and Usability: Research Issues and Development Contexts. *Interacting with Computers*, 4, (1992) 209–217.
6. Grudin J. and Wang Laboratories. Adapting a Psychophysical Method to Measure Performance and Preference Tradeoffs in Human-Computer Interaction. *in Proc. INTERACT (1984)*. 737–741.
7. Heer, J. and Bostock, M. Crowdsourcing Graphical Perception: Using Mechanical Turk to Assess Visualization Design. *in Proc. CHI 2010*. (2000)
8. Horton, J.J. and Chilton, L. The Labor Economics of Paid Crowdsourcing. *in Proc. ACM Electronic Commerce 2010.* (2010)
9. Ipeirotis, P.G. Analyzing the Amazon Mechanical Turk Marketplace. *New York University Report no. CeDER-10-04*. 2010.
10. Ipeirotis, P.G. The Explosion of Micro-Crowdsourcing Services. http://behind-the-enemy-lines.blogspot.com /2010/10/explosion-of-micro-crowdsourcing.html.
11. Ipeirotis, P.G. Provost, F. and Wang, J. Quality Management on Amazon Mechanical Turk. *In Proceedings of KDD-HCOMP* (2010).
12. Ipeirotis, P.G. Why People Participate on Mechanical Turk, Now Tabulated. http://behind-the-enemy-lines.blogspot.com/2008/09/why-people-participate-on-mechanical.html
13. Kittur, A.; Chi, E. H. ; Suh, B. Crowdsourcing user studies with Mechanical Turk. In Proc. CHI 2008, ACM Press (2008). 453–456.
14. Klein, J.P. *Survival Analysis*. Springer, 2003.
15. Kohavi, R.. Practical Guide to Controlled Experiments on the Web: Listen to Your Customers not to the HiPPO. *in Proc KDD 2007*. (2007)
16. MacKenzie, S. Fitts' Law as a Research and Design Tool. *in Human-Computer Interaction (HCI)*, 1992 7(1), 91–139.
17. Malone, T.W. Heuristics for Designing Enjoyable User Interfaces: Lessons from Computer Games. *in CHI 1982*. ACM Press (1982)
18. Nielsen, J. *Usability Engineering*. Morgan Kaufmann, 2001.
19. Ross, J., Irani, L., Silberman, M.S., Zaldivar, A. and Tomlinson, B. Who are the Crowdworkers? Shifting Demographics in Mechanical Turk. *alt.chi '10*, ACM Press.
20. Simon, H.A. Designing Organizations for an Information-Rich World. in Martin Greenberger, Computers, Communication, and the Public Interest, Baltimore, MD: The Johns Hopkins Press (1971).
21. Smith, A. An Inquiry into the Nature and Causes of the Wealth of Nations. *London: Methuen and Co., Ltd*., ed. Edwin Cannan, 1904.
22. Van Ahn, L. and Dabbish, L. Labeling Images with a Computer Game. *in Proc. CHI 2004.* ACM Press (2004)
23. Varian, H.R. *Microeconomic Analysis*. Norton, 1984.
24. Wilson, E. Probable Inference, the Law of Succession, and Statistical Inference. Journal of the American Statistical Association. 1927. 22: 209–212.