A Computational Theory of Awareness and Decision Making

Nikhil R Devanur*
Microsoft Research
One Microsoft way
Redmond, WA

Lance Fortnow†
Department of EECS
Northwestern University
Evanston, IL

Abstract

We exhibit a new computational-based definition of awareness, informally that our level of unawareness of an object is the amount of time needed to generate that object within a certain environment. We give several examples to show this notion matches our intuition in scenarios where one organizes, accesses and transfers information. We also give a formal process-independent definition of awareness based on Levin's universal enumeration.

We show the usefulness of computational awareness by showing how it relates to decision making, and how others can manipulate our decision making with appropriate advertising, in particular, we show connections to sponsored search and brand awareness. Understanding awareness can also help rate the effectiveness of various user interfaces designed to access information.

1 Introduction

Beer companies advertise to raise their brand awareness, to make sure someone walking into a bar asks for Bud instead of Miller. Search engine companies do not only want to give the world access to information, but to let people be highly aware of the information when they need it. One typically chooses a restaurant to eat at among those with which one has a high awareness of. We keep an address book so we can be aware of a phone number when we need to call that person. A judge tries to determine what circumstances that a legislature was aware of when they passed a certain law. We used to be highly aware of John Edwards when he was an active candidate but our awareness of him has since faded.

We shine the "computational lens" on awareness to develop a new definition that captures our intuition in a number of scenarios such as the ones above. Awareness does not occur in a vacuum so we consider two types of inputs. The first is the environment which encodes all information sources such as a person's memory, possible interactions with other people and nature, books, the entire Internet and everything else one might have access to. The other is a context such as "Restaurants in Chicago." Informally we define the unawareness level of an object as *the amount of time needed to enumerate that object* where the enumerator gets the context as input and has random access to the environment. The intuition is that objects that you are more aware of you will generate first. We give a formal definition in Section 4. Using a universal enumeration procedure due to Levin [Lev73], our formal definition is independent of the particular enumeration procedure of the agent. There have been some previous notions of awareness in the literature (see Section 3 for a more detailed discussion), particularly a knowledge-based concept due to Fagin and Halpern [FH87]. Our model has the advantage of talking about abstract objects instead of formulas with truth values and allowing for a gradation of awareness that can possible decrease over time.

We then show how one can use our definition to analyze decision making processes. How do you decide on what restaurant to eat at? A large city typically has thousands of restaurants. You would consider restaurants where you have eaten before, recommendations from friends,

*Part of the work done when the author was affiliated with Toyota Technological Institute–Chicago

†Supported in part by NSF grants CCF-0829754 and DMS-0652521.

reviews from newspapers, restaurant guides and search engines, some restaurant you might have walked by or seen an advertisement for. What you don't do is examine every restaurant in the city and making the choice that optimizes the various criteria, such as type and quality of food, cost, atmosphere, location, etc. You simply lack awareness of all the restaurants. That's not quite precisely true, since you can find nearly every restaurant with appropriate web surfing. In this paper we measure a certain cost of awareness, precisely a *computational* cost of awareness. In the restaurant example, you would likely choose restaurants with high awareness, balanced against your particular desires in food type, etc. at the time.

A typical decision making process goes like this:
A process starts with an agent who needs to make a decision based on a certain *criteria*. The agent interacts with the environment, and outputs a *decision* that satisfies the given criteria. For instance, a criteria could be, "find a used car that has less than 50,000 miles, costs less than $8,000 and within 50 miles of Chicago". The interaction with the environment could involve going through craigslist to look at used cars for sale, placing an message asking for used cars which meet the criteria, and so on. At some point, the agent decides to stop the interaction and outputs a decision, which could also be "there are no such cars". Note that cars with low awareness will be enumerated earlier and thus be more likely bought by the agent.

We can use awareness to capture how the decision making process can be manipulated by someone who benefits from the decision made. For example, a used car dealer might want to make sure that the decision turns out to be one of her cars. Awareness also help us understand the cost to computation which factors into the decision making process.

These definitions of awareness and decision making turn out to be quite useful: we show how they can be applied in a number of scenarios including sponsored search, competition between brands (focusing on awareness of the brands themselves as well as the awareness of particular attributes that would make one brand look better than the other) and in analyzing user interfaces.

We end with a discussion of future applications and questions about awareness and decision making.

2 Examples of Computational Awareness

In this section we motivate our definition by giving examples where our definition coincides with the intuitive notion of awareness.

2.1 Decreasing Awareness

Consider a stack of research papers that you want to read. You download and print a new paper by Karp and put it on top of the stack. The Karp paper has high awareness as it is very easy to enumerate that paper, picking it off the top of the stack. But over time you will put more papers on the stack and the Karp paper now falls towards the middle of the stack. Your awareness of that paper goes down as enumerating papers would take longer before the Karp paper is enumerated. As the environment changes over time, you can become less aware of a given object.

Awareness can be recovered. You might see a talk mentioning the Karp paper. In the context of the talk, your awareness of the paper becomes high again. This may cause you to put the Karp paper once again at the top of the stack, in effect changing your environment to increase the awareness of the paper.

2.2 Awareness of Unawareness

In the context of US Presidents, you are likely aware of recent presidents like Barack Obama and George W. Bush, early presidents like George Washington and Thomas Jefferson and famous presidents like Abraham Lincoln and Franklin D. Roosevelt. But you likely have far less awareness of say James Garfield. Until you have read the last sentence. As a basic principle you cannot be aware of what you are unaware of.

2.3 Awareness and the Legal System

Consider a civil court case where the judge orders the defendant to hand over a specific document to the plaintiffs. A common tactic is to hand over a large stack of documents with the requested document placed in a random location. The defendants have followed the letter of the law by physically giving the document to the plaintiffs. But even though the plaintiffs have the document in this stack, they are not aware of the document because it would take a very long time to find the document given the environment of the large stack produced by the defense.

With a proper definition of awareness, the judge could order the defense to not only make the document available but produce an environment where the plaintiffs have high awareness of the document in the context of the court case. If the defense tried the above tactic they would be found in contempt of court unless they could exhibit a simple enumeration procedure that quickly produces the document.

In a different legal context, Chung and Fortnow [CF08], examined why loopholes occur in laws and contracts. A legislature creates a law and at some point in the future the judge interprets the law as it applies to a particular circumstance. Think of the judge as a function mapping a law and circumstance into a penalty. The goal of the judge is to apply the legislature's intent of the law.

This may not always be possible. A legislature, when they create a law, may not be aware of all future circumstances. Furthermore, a judge may not be aware of what the legislature is or is not aware of. Chung and Fortnow created a simple model and showed that legislatures, sometimes knowingly, create loopholes to keep laws interpreted as best as possible in the future. The model of awareness used by Chung and Fortnow was a very simple cost function. A better notion of awareness was needed to create better models and find other applications of awareness in the legal community. This goal of a new awareness model is one of the drivers for this current paper.

2.4 Future Awareness

Why do we keep calendars? After all we only add entries to a calendar that we already know. The key is awareness. When we add the TARK conference to our calendar we do so that in the context of July 6, 7 or 8, we will quickly enumerate the event. This is useful not only during the event themselves but if we later schedule other events on those days, our context switches to those days and we have a high awareness of TARK, allowing us to avoid conflicts. In short, keeping a calendar creates an environment to help us to have high awareness in the contexts where we need high awareness.

We can tell similar stories about address books, email programs, to do lists, file directories, desktop search, file cabinets, etc. Most organizational tools try to organize our information to maximize the awareness of information in the future contexts where we need it. One must apply these organizational tools carefully. Putting too many events on one day (say including sports teams schedules, theater events, colloquium talks all over campus, etc.) causes unawareness of some of the possibly important events schedule that day as it could take a long time to enumerate those events even in the context of that day. Organizational tools need the right balance to maximize awareness in the right context and using a notion of computational awareness can help us use those tools to maximize our efficiency.

3 Relationship to Other Notions of Awareness

Awareness as a concept has had extensive study in many disciplines including psychology, philosophy and economics. We cannot, in the limited space of this paper, even begin to cover these approaches. Instead we focus on a computer science approach based on knowledge representation.

Fagin and Halpern [FH87] build awareness on top of the Kripke model of knowledge (see Halpern and Moses [HM84]. In the Kripke model, the world is in one of many states, for each person i there is a partition of the states into information sets. Person i knows what set his states lies in but not the state itself. Person i knows a formula ϕ ($K_i\phi$) if the formula ϕ is true in every state in the partition. Fagin and Halpern add to this two new modal operators $A_i\phi$ and $X_i\phi$. $A_i\phi$ represents some intuitive notion of awareness and define explicit knowledge, $X_i\phi$, as having implicit knowledge of ϕ, ($K_i\phi$) and being aware of ϕ. Fagin and Halpern do not put any restrictions on $A_i\phi$.

Two economists Mudica and Rusticini [MR94] suggest defining knowledge in terms of awareness: You are aware of ϕ if you know ϕ or you know you don't know ϕ. In the Kripke model, this definition is too broad, as for every formula ϕ you either know ϕ or you know you don't know ϕ. In a follow-up paper Mudica and Rusticini [MR99] give a more nuanced definition of awareness which Halpern [Hal01] later showed, under reasonable assumptions, is equivalent to saying you are aware of ϕ if you explicitly know ϕ or you explicitly know you don't explicitly know ϕ.

In that last paper [Hal01], Halpern suggests looking at a computational version of awareness:

> It is possible to consider more computationally oriented notions of awareness. The problem is then to come up with interesting notions of awareness that have enough structure to allow for interesting mathematical analysis. I believe it should also be possible to use awareness structures to allow for natural rea-

soning about awareness and lack of it (so that an agent can reason, for example, about the possibility that she is unaware of certain features that another may be aware of). I am currently working on modeling such reasoning.

Halpern [Hal] has not yet fully pursued this direction and in any case has focused on the complexity of determining whether a formula is true or false.

While the model of awareness developed above has some appeal, we don't believe it properly models the intuitive ideas of awareness we express in this paper. Our model has many features that differ, including

- The Halpern-Fagin model focuses on formulas that have some truth value. We focus on awareness of strings, like "McDonalds" without any underlying truth value.

- Instead of a binary notion of awareness, we give a gradation of awareness, so one can say that they are more aware of a restaurant they ate at yesterday than one they ate at a few years ago. We also allow awareness to possibly decrease over time.

- We give a full formal definition of awareness based on a computational basis, as opposed to just an axiomatic definition that allows for many different awareness operators.

4 Formal Model

In this section we develop a formal model of awareness.

We fix an alphabet $\Sigma = \{0, 1\}$ though the theory easily generalizes to larger alphabets. We define the environment formally as a function $E : \Sigma^* \rightarrow \Sigma^*$. This is general enough to encode any of the intuitive notions of the environment we describe above. The context y is just a string in Σ^*. We define awareness (actually unawareness) on strings as well.

An enumeration process is an oracle Turing machine M such that given an oracle A and input w, $M^A(w)$ will output a potentially infinite series of strings $z_1, z_2, \ldots$. The oracle A can be a function whose output will appear on a special oracle output tape. We will count each oracle query as a single time step, though it will take more time to write each bit of the query and read each bit of the response.

We define the unawareness of a string x in environment E and context y using enumeration process M, $U^E_M(x|y)$ as the amount of time until $M^E(y)$ enumerates the string x. Note we are counting time, not the index of the string being enumerated. Also we do not require that $M^E(y)$ halt after producing string x on its list of outputs. If $M^E(y)$ never enumerates x we say the unawareness is infinite.

We can eliminate the dependence on the enumeration process by applying a universal enumeration procedure due to Levin [Lev73]. Levin shows that there exists a single enumeration procedure N such that for every enumeration procedure M, constant c_M such that for all x, context y and environment E,

$$U^E_N(x|y) \leq c_M U^E_M(x|y).$$

The value c_M does not depend on x, y or E. As a shortcut we use $U^E(x|y)$ to represent $U^E_N(x|y)$, the enumeration-independent unawareness measure of object x in context y and environment E.

Levin [Lev73] also gives a Kolmogorov-complexity definition equivalent up to constant factors: Fix a universal oracle Turing machine M_U that takes inputs (p, y), where p is from a set of prefix-free programs, and simulates program p on input y. $U^E(x|y)$ is the minimum over all programs p of the quantity $t2^{|p|}$ where $M^E_U(p, y)$ outputs x within t steps.

The enumeration-independent definition has nice mathematical properties but can be difficult to properly analyze. Often in this paper we will often focus on limited though natural enumeration processes in a specifically structured environment.

This definition does not involve any actual actions or decisions made by an agent. One approach involves combining awareness with a fitness function. Some candidate functions include

- The input is a function $f : \Sigma^* \rightarrow \{0, 1\}$, measuring the "fitness" of a string. This models binary decision processes, where a string is either fit or not, and the goal is to find some string that is.

- The input is a function $F : \Sigma^* \times \Omega \rightarrow \{0, 1\}$, where Ω is a probability space. In this case, F is interpreted as a random variable that measures fitness. A further special case of this is that the fitnesses of the strings are independent of each other. In this case, we could simply have $f : \Sigma^* \rightarrow [0, 1]$, which measures the probability that a string satisfies the criteria. Such random functions allow us to model multiple agents, with a single process.

- The input is a function $f : \Sigma^* \to \mathbb{R}$. In this case fitness is a continuous measure, and the agent seeks to find a string that maximizes the fitness.

Each of the above cases can be extended to include auxiliary information, so that the domain of f is $\Sigma^* \times \Sigma^*$. The auxiliary information could involve such things as product reviews.

One can then get a decision making process by combining the fitness function with an enumeration procedure, for example, in the first case above, choosing an object x if it is the first object enumerated by $N^E(y)$ such that $f(x) = 1$. We can also explicitly incorporate the cost of computation into the decision making process, such an approach is shown in Appendix A.

One can manipulate the decision making process by manipulating the environment E, by changing the output of the function E on some inputs (for example manipulating search engine results). Let A be some set of actions (a subset of all possible actions), and $c(A)$ be the cost of changing all the entries of E corresponding to A. In other words, the cost of advertising on the actions in A is $c(A)$. A special case is when $c(A) = \sum_{a \in A} c(a)$. Also an advertiser w makes a profit $p_w(x)$ if x is the decision made. Hence it is profitable for an advertiser to advertise on actions in A if the resulting change in the decision made gets him an increase in profit more than $c(A)$.

5 Applications

Having defined a general model, we now describe a few situations in which the notion of computational awareness arises naturally. We show that specialising our model to the particular situation provides a useful tool for formal reasoning and analysis. Moreover, we give a uniform way to do it, by providing a set of questions that we are repeatedly led to answer in all such situations.

- Awareness of what? Although our model allows awareness to be defined for arbitrary strings, it is often useful to define a particular set of strings whose awareness we are interested in, given the context.

- What is the environment? Again, as before, a contextual definition of environment may be restricted to a particular set of strings that are of interest. Also defining the rules on how the environment might be modified is often a useful exercise.

- What is the enumeration process? Defining an enumeration process amounts to modeling the behavior of a person; hence it is bound to be an inexact choice. Traditionally such modeling is done statistically and no consideration is given to the algorithmic "resonableness" of a process, which we consider to be important. Further, an algorithmic model as defined by an enumeration process might be quite different from the traditional models in statistics, and have interesting new properties.

- What is the decision making process? This involves picking a way to make decisions, and defining the fitnesses of the strings, and might involve defining a cost of computation as well.

We make several simplifying assumptions to keep the analysis tractable. Analyzing more general models might lead to deeper insights.

5.1 Sponsored Search

In sponsored search, a user queries a search engine for a keyword, and is served a list of "organic" results along with the advertisements (ads) for the query. The list of ads is referred to as sponsored search listings. The advertiser pays the search engine a certain amount if the user clicks on his ad. We now give a model for sponsored search based on our model of awareness.

Awareness of what? The goal of an ad is to increase awareness of say, some product and influence the decision of users who might be interested in that product. For the sake of this discussion, we will simply talk of awareness of an ad and not make a distinction between the ad and the product it is peddling. An advertiser picks a list of keywords for which his ad is to be considered. So given the keyword the user searched for, we are interested in his awareness of all the ads that are to be considered for that keyword.

What is the environment? The environment is essentially the page returned by the search engine. The queries to the environment are the various actions the user does, such as looking at an ad in a particular position, and the string returned by the environment is simply the ad.

What is the decision making process? The user decides on which ads to click on, if any. Define the fitness f_i of an ad i to be the probability[1] that a user finds the ad

[1] We think of an user as randomly drawn from a large population. Hence, the probability is over the choice of the user.

relevant to his query, and would click on it if presented by itself.

What is the enumeration process? First, we reverse-engineer an enumeration process from one of the standard assumptions in the sponsored search literature. A quantity of great interest is the click-through rate (CTR) of an ad, which is the ratio of the number of clicks on an ad to the number of times the ad is displayed. Clearly the CTR of an ad depends on a lot of contextual factors, like the relevance of the ad to the keyword queried, the text displayed by the ad, and position of all the ads. A common assumption is that the CTR is a product of the fitness f_i of the ad and a slot dependent factor, say θ_j for slot j (see Varian [Var07], for example). One could reverse-engineer this assumption to see what processes generate such clicks. One such process is, the user enumerates slot j with probability θ_j, and clicks on the ad in that slot if relevant, which happens with probability f_i. Otherwise he does not click anything. This is really a binary notion of awareness, where the order of enumeration is not important.

An alternate and simpler enumeration process is the following: The user goes down the list of ads starting from the top to bottom. Further, let's say that the user only clicks on the first ad that he finds relevant. This gives an alternate formula for the CTRs. If ad i is shown in slot $\pi(i)$, then the CTR of the ad is

$$f_i.\Pi_{j:\pi(j)<\pi(i)}(1-f_j).$$

Note that we have made a simplifying assumption that the fitnesses of different ads are independent of each other.

A similar model has been considered by Crasswell et. al.[CZTR08] for CTRs of organic seach results (not ads), under the name, "cascade model". Their experiments indicate that the cascade model gives a better fit to the data than the standard one. Such models have also been proposed for sponsored search, independently, by Agarwal et. al. [AFMP08] and Kempe and Mahdian [KM08]. Athey and Ellison [AE08] have developed and analyzed a different model based on a similar enumeration principle.

Given such a model, we can analyze various aspects of sponsored search. Typically the slots are priced through an auction. Two of the standard auctions are the Vickery-Clarke-Groves (VCG) auction and the Generalized Second Price (GSP) auction. Various questions related to these auctions are:

1. The VCG auction allocates the slots to maximize social welfare. What is the order of the ads that achieves this maximum?
2. What are VCG prices?
3. What is the equilibrium of the GSP auction?

The answers to some of these questions provide surprising insight. For instance, for the first question, suppose advertiser i derives a value v_i from one click on his ad. The social welfare of a particular order of listings is the expected value of a click from one user. The answer is that the ads are to be ranked by their valus v_i regardless of their fitnesses f_i (Theorem 1). This is in contrast to the standard model in which ranking by $f_i v_i$ is optimal.

Remark: The conventional wisdom is that ranking by expected revenue ($f_i v_i$) is better. However, in the analysis, we have assumed that we can display all the ads. When the number of ads that can be displayed is restricted, the set of ads should be chosen in order to maximize welfare, and this set of ads should be ranked by v_i's. This makes sure that ads with very small f_i's are not displayed, which was the original motivation behind ranking by revenue.

Theorem 1. *Given a set of ads, their values and fitnesses, ordering by decreasing values of v_i maximizes*

$$W(\pi) := \sum_i v_i f_i \prod_{j:\pi(j)<\pi(i)} (1-f_j)$$

among all permutations π.

Proof. Consider any permutation π. We will show that if there exist i and j such that $\pi(i) < \pi(j)$ and $v_i < v_j$, then switching the positions of i and j increases W. Without loss of generality, we may assume that π is the identity permutation and $j = i+1$. With these assumptions, the only terms in W that change when i and $i+1$ are switched are the ones with v_i and v_{i+1}. They both have as a common factor $\prod_{j:\pi(j)<\pi(i)}(1-f_j)$ which we can ignore for the sake of comparison. Hence the terms of interest are initially

$$v_i f_i + v_{i+1} f_{i+1}(1-f_i) = v_i f_i + v_{i+1} f_{i+1} - v_{i+1} f_i f_{i+1},$$

which is easily seen to be smaller than the terms after switching,

$$v_{i+1} f_{i+1} + v_i f_i (1-f_{i+1}) = v_i f_i + v_{i+1} f_{i+1} - v_i f_i f_{i+1}.$$

□

5.2 Two competing brands

Consider a person who has to decide between two presidential candidates. His decision would depend on their stands on various issues of importance to the person. Or consider someone choosing between two competing brands, say Corn Flakes and Cheerios. The decision to choose one over the other depends on how the two brands score on different features the user considers. We now formalize such situations.

Awareness of what? In this situation, the agent is aware of both the brands, so it does not really help us to talk about their awareness. Instead we consider his awareness of the various features of either brand. For simplicity, assume that there is a given set of features, and for each feature l, there are weights w_l and v_l. The weights indicate how well the two brands score on the feature. Awareness of a feature l means the awareness of the weights w_l and v_l for that feature. Formally, we consider the awareness of the agent for triples of the form (l, w_l, v_l).

What is the environment? The environment includes the various advertisement channels that the agent interacts with. It could be different media, such as print, TV, Internet, billboards, etc. The queries to the environment correspond to actions accessing the media, such as the location of a billboard, the TV channel tuned into at a given time, or the url of the web site accessed. The environment returns a triple of the form (l, w_l, v_l). In this situation, we are interested in how the two brands can manipulate the environment to their advantage. (Which is what advertisement is all about.) This is essentially the marketing strategy for the advertisers: choosing the channels to place their ads on, deciding the features to highlight, and so on. We consider a simplification and assume that there are n possible queries to the environment and the two brands control the answer for n_1 and n_2 of these queries with $n_1+n_2 = n$. We'd like to study their optimal strategies: which features should they return for the queries they control.

Apart from the advertising channels, we also consider the agent's memory as part of the environment. We assume that the interaction with the advertisement channels is like pre-processing, and affects the agent's memory in a certain way. The enumeration process at the time of decision making simply accesses the memory that has been formed as a result.

What is the enumeration process? First, we have to specify how the interaction with the advertisements affects the memory. Again, for simplicity, assume that the agent makes each of the n possible queries exactly once. He retains in memory the k most repeated features, in the order of the frequency of repetition, for some k. For now, we think of k as a fixed[2] number. The idea is that an agent has a higher awareness of features that have been repeated more often in the recent past.

During the decision making process, the agent simply enumerates over all k features he has retained in the order of frequency.

What is the decision making process? Let's say that the k most often repeated features are $1, 2, \ldots, k$. Then the agent picks brand 1 if

$$w_0 + \sum_{l=1}^{k} w_l > \sum_{l=1}^{k} v_l,$$

and otherwise picks brand 2. w_0 is an initial bias in favor of brand 1 (if it is negative, then it actually favors brand 2). The initial bias can be used to include all the factors that are not susceptible to be changed by advertising.

Given the model of decision making as above, one can ask what are the optimal strategies for the two brands? This is a min-max game between the 2 brands where their strategies are to decide which features to return on the queries they control. The strategy set of brand 1 is the set of all partitions of n_1 into $\leq k$ parts (and respectively for 2). Note that clearly, the features that brand 1 returns on the (n_1) queries it controls are the ones that maximize the difference in scores, $w_l - v_l$. Similary brand 2 returns those features that maximize $v_l - w_l$. Rank the features by decreasing order of $w_l - v_l$, and let $i_1, i_2, \ldots, i_k$ be the top k features. Similary let $j_1, j_2, \ldots, j_k$ be the top k features when ordered by decreasing order of $v_l - w_l$. The i's are the best features of brand 1 and the j's are those of 2.

Let us consider simple cases. If $k = 1$, then the brand who controls the larger number of queries wins. If $k = 2$, then 1 wins if $w_0 + w_{i_1} \geq v_{j_1}$ and $n_1 \geq n_2/2$, or if $n_1 \geq 2n_2$. In general, let k_1 and k_2 be such that

$$w_0 + \sum_{l=i_1}^{i_{k_1}} w_l > \sum_{l=j_1}^{j_{k_2-1}} v_l,$$

$$w_0 + \sum_{l=i_1}^{i_{k_1-1}} w_l < \sum_{l=j_1}^{j_{k_2}} v_l$$

[2] One could consider cases where k is drawn from a probability distribution.

and $k_1 + k_2 = k + 1$. That is, entity 1 is guaranteed to win if he can control at least k_1 of the features picked (and respectively for entity 2). So the best strategy for 1 is to spread his top k_1 features among the n_1 queries that he controls. Thus entity 1 wins if $\lfloor \frac{n_1}{k_1} \rfloor > \lfloor \frac{n_2}{k_2} \rfloor$ and vice versa. If they are the same, then let's say there is a tie.

There are several ways to refine the model to make it more realistic, for instance, k could be a random variable with some known distribution, or the costs of advertising could be different for different queries. We do not pursue it further, as our goal is to simply show how interesting analysis emerges from our model.

5.3 User Interface Design

In this section we show that the notion of awareness is useful in the design and evaluation of user interfaces, such as designing the interface of a particular application, file system design, or more generally, the interface of an operating system. The framework outlined in the beginning of this section gives a systematic and formal way to do the same.

Awareness of what? For a calendar application, one wants to keep the user aware of the events at the appropriate time. For a UI of an operating system, the user needs to be aware of the various applications that are running, their status, potential applications that can be launched, potential actions that can be performed in an application, the locations of various files in the directory structure and so on and so forth. For a formal analysis, one would list in detail all such objects whose awareness are of interest to the particular interface design.

What is the environment? The environment is the user interface. The queries to the environment are various actions the user can take, such as pointing or clicking a mouse, keyboard shortcuts, etc. What the environment returns is the feedback it gives the user on performing the said actions. For example, clicking a menu might return a list[3] of applications that could be launched, in the order of frequency of use. Again, a formal analysis would include all possible queries (actions) to the environment and the corresponding answers.

What is the enumeration process? There may be multiple enumeration processes, since typically an interface is intended for multiple purposes. The design of the interface would specify an intended enumeration process for each purpose. One could then examine if those enumeration processes are natural, easy to remember, and in general, if a user might be reasonably expected to follow it. For instance, a user looking to switch between two applications might be expected to go through a list of icons depicting all running applications to find the one he wants to switch to.

What is the decision making process? In this situation, the goal of the user is more often to perform a specific task, than to make a decision, or choice. Hence, it makes sense to specify the process by which he performs the task, which is often similar to the enumeration process.

What is the cost? Although the enumeration time gives a good measure of evaluating the complexity of a design, a cost function that is different from the enumeration time might be more useful. This allows one to assign different costs to different types of actions, a gesture using a multitouch trackpad might cost less than using the mouse to drag a button. An interface design may be evaluated by considering the cost of performing various tasks.

6 Conclusions

In this paper, we have given a new computational-based definition of awareness that appears to fit well with many intuitive uses of the concept. We apply our notion to give a new way to think about decision making and show applications to sponsored search, brand awareness and user interfaces.

We have only scratched the surface with our uses of awareness. Deeper analysis of the models described in Section 5 should lead to a greater understanding of those topics. We also hope to see several new and unexpected applications of awareness as well as other ways we can use computational thinking to understand the various ways we process information.

Acknowledgments

The second author would like to thank Kim-Sau Chung. Their paper [CF08] and subsequent discussions exhibited the need for a new computational-based definition of awareness that eventually led to the model described here. The second author also would like to acknowledge useful discussions with Rakesh Vohra and Nabil Al-Najjar. Jason Hartline pointed us to the sponsored search work of Athey and Ellison [AE08] and Abie Flaxman pointed us to the experiments of Craswell, et. al. [CZTR08].

[3] We could be more detailed and say that clicking the menu is one action, and looking at an element in the list that pops up is another and so on.

References

[AE08] S. Athey and G. Ellison. Position auctions with consumer search. Working paper, 2008.

[AFMP08] Gagan Aggarwal, Jon Feldman, S. Muthukrishnan, and Martin Pál. Sponsored search auctions with markovian users. In *Proceedings of The 4th International Workshop On Internet And Network Economics*, pages 621–628, 2008.

[CF08] K. Chung and L. Fortnow. Loopholes. Submitted, 2008.

[CZTR08] Nick Craswell, Onno Zoeter, Michael Taylor, and Bill Ramsey. An experimental comparison of click position-bias models. In *Proceedings of the First ACM International Conference on Web Search and Data Mining*, 2008.

[FH87] R. Fagin and J. Halpern. Belief, awareness, and limited reasoning. *Artificial Intelligence*, 34(1):39–76, December 1987.

[Hal] J. Halpern. Personal Communication.

[Hal01] J. Halpern. Alternative semantics for unawareness. *Games and Economic Behavior*, 37(2):321–339, November 2001.

[HM84] J. Halpern and Y. Moses. *Knowledge and common knowledge in a distributed environment*. ACM Press, New York, 1984.

[KM08] David Kempe and Mohammad Mahdian. A cascade model for externalities in sponsored search. In *Proceedings of The 4th International Workshop On Internet And Network Economics*, pages 585–596, 2008.

[Lev73] L. Levin. Universal'nyĭe perebornyĭe zadachi (Universal search problems: in Russian). *Problemy Peredachi Informatsii*, 9(3):265–266, 1973. Corrected English translation in [**?**].

[MR94] S. Modica and A. Rustichini. Awareness and partitional information structures. *Theory and Decision*, 37(1):107–124, July 1994.

[MR99] S. Modica and A. Rustichini. Unawareness and partitional information structures. *Games and Economic Behavior*, 27(2):265–298, May 1999.

[Var07] H. R. Varian. Position auctions. *International Journal of Industrial Organization*, 25(6):1163–1178, 2007.

A Cost of Awareness

In this section we show how to explicitly incorporate the computational cost of awareness. We consider a continuous measure of fitness, $f : \Sigma^* \to \mathbb{R}$. We introduce a cost function $g : \mathcal{N} \to \mathbb{R}$ that measures the cost incurred by t time steps. This captures the fact that the cost may scale with time. Finally, suppose that at every time step, the process has a *belief* about the outcome of the process if it was continued further. The belief is a probability distribution over all possible outcomes of the process. What we need about this belief is that at time t, the process can calculate

$$B(t) := \max_{t'>t} \mathbf{E}[f(x^*(t'))] - g(t'), \text{ where}$$

$$x^*(t) = \arg\max\{f(x) : U^E(x|y) \leq t\}.$$

Given this, we can give an explicit stopping criteria for the process: stop when

$$B(t) \leq f(x^*(t)) - g(t)$$

and output $x^*(t)$. This gives us a clean way of balancing the potential increased benefit from continuing the search with the expected cost.